Limiting the Impact of Stealthy Attacks on Industrial Control Systems

David I. Urbina[1], Jairo Giraldo[1], Alvaro A. Cardenas[1], Nils Ole Tippenhauer[2], Junia Valente[1], Mustafa Faisal[1], Justin Ruths[1], Richard Candell[3], and Henrik Sandberg[4]

[1]University of Texas at Dallas, [2]Singapore University of Technology and Design, [3]National Institute of Standards and Technology, and [4]KTH Royal Institute of Technology

{david.urbina, jairo.giraldo, alvaro.cardenas, juniavalente, mustafa.faisal, jruths}@utdallas.edu, nils_tippenhauer@sutd.edu.sg, richard.candell@nist.gov, and hsan@kth.se

ABSTRACT

While attacks on information systems have for most practical purposes binary outcomes (information was manipulated/eavesdropped, or not), attacks manipulating the sensor or control signals of Industrial Control Systems (ICS) can be tuned by the attacker to cause a continuous spectrum in damages. Attackers that want to remain undetected can attempt to hide their manipulation of the system by following closely the expected behavior of the system, while injecting just enough false information at each time step to achieve their goals.

In this work, we study if physics-based attack detection can limit the impact of such stealthy attacks. We start with a comprehensive review of related work on attack detection schemes in the security and control systems community. We then show that many of these works use detection schemes that are not limiting the impact of stealthy attacks. We propose a new metric to measure the impact of stealthy attacks and how they relate to our selection on an upper bound on false alarms. We finally show that the impact of such attacks can be mitigated in several cases by the proper combination and configuration of detection schemes. We demonstrate the effectiveness of our algorithms through simulations and experiments using real ICS testbeds and real ICS systems.

Keywords

Industrial Control Systems; Intrusion Detection; Security Metrics; Stealthy Attacks; Physics-Based Detection; Cyber-Physical Systems

1. INTRODUCTION

One of the fundamentally unique and intrinsic properties of Industrial Control Systems (ICS)—when compared to general Information Technology (IT) systems— is that changes in the system's state must follow immutable laws of physics. For example, the physical properties of water systems (fluid dynamics) or the power grid (electromagnetics) can be used to create prediction models that we can 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: if we opened an intake valve, we would expect the water tank level to rise, otherwise we may have a problem with the control, actuator, or the sensor.

The idea of using physics-based models of the normal operation of control systems to detect attacks has been used in an increasing number of publications in security conferences in the last couple of years. Applications include water control systems [21], state estimation in the power grid [35,36], boilers in power plants [67], chemical process control [10], electricity consumption data from smart meters [40], and a variety of industrial control systems [42].

The growing number of publications shows the importance of leveraging the physical properties of control systems for security; however, a missing element in this growing body of work is a unified adversary model and security metric to help us compare the effectiveness of previous proposals. In particular, the problem we consider is one where the attacker knows the attack-detection system is in place and bypasses it by launching attacks imitating our expected behavior of the system, but different enough that over long periods of time it can drive the system to an unsafe operating state. This attacker is quite powerful and can provide an upper bound on the worst performance of our attack-detection tools.

Contributions. (i) We propose a strong adversary model that will always be able to bypass attack-detection mechanisms and propose a new evaluation metric for attack-detection algorithms that quantifies the negative impact of these stealthy attacks and the inherent trade-off with false alarms. Our new metric helps us compare in a fair way previously proposed attack-detection mechanisms.

(ii) We compare previous attack-detection proposals across three different experimental settings: a) a testbed operating real-world systems, b) network data we collected from an operational large-scale Supervisory Control and Data Acquisition (SCADA) system that manages more than 100 Programmable Logic Controllers (PLCs), and c) simulations.

(iii) Using these three scenarios we find the following results: (a) while the vast majority of previous work uses stateless tests on residuals, stateful tests are better in limiting the impact of stealthy attackers (for the same levels of false alarms), (b) limiting the impact of a stealthy attacker can also depend on the specific control algorithm used and not only on the attack-detection algorithm, (c) linear state-space

CCS'16, October 24 - 28, 2016, Vienna, Austria

ACM ISBN 978-1-4503-4139-4/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2976749.2978388

models outperform output-only autoregressive models, (d) time and space correlated models outperform models that do not exploit these correlations, and (e) from the point of view of an attacker, launching undetected actuator attacks is more difficult than launching undetected false-data injection for sensor values.

The remainder of this paper is organized as follows: In § 2, we provide the scope of the paper, and provide the background to analyze previous proposals. We introduce our attacker model and the need for new metrics in § 3. We introduce a way to evaluate the impact of undetected attacks and attack-detection systems in § 4, and then we use this adversary model and metric to evaluate the performance of these systems in physical testbeds, real-world systems, and simulations in § 5.

2. BACKGROUND AND TAXONOMY

Scope of Our Study. We focus on using real-time measurements of the physical world to build indicators of attacks. In particular, we look at the physics of the process under control but our approach can be extended to the physics of devices as well [18]. Our work is motivated by false sensor measurements [35, 58] or false control signals like manipulating vehicle platoons [19], manipulating demand-response systems [58], and the sabotage Stuxnet created by manipulating the rotation frequency of centrifuges [17, 32]. The question we are trying to address is how to detect these false sensor or false control attacks in real-time.

2.1 Background

A general feedback control system has five components: (1) the physical phenomena of interest (sometimes called the process or plant), (2) sensors that send a time series y_k denoting the value of the physical measurement z_k at time k (e.g., the voltage at 3am is 120kV) to a controller, (3) based on the sensor measurements received y_k, the controller $\mathcal{K}(y_k)$ sends control commands u_k (e.g., open a valve by 10 %) to actuators, and (4) actuators that produce a physical change v_k in response to the control command (the actuator is 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 Fig. 1.

2.2 Taxonomy

Anomaly detection is usually performed in two steps. First we need a model of the physical system that predicts the output of the system $\hat{y}_k$. The second step compares that prediction $\hat{y}_k$ to the observations y_k and then performs a statistical test on the difference. The difference between prediction and observation is usually called the **residual** r_k. We now present our new taxonomy for related work, based on four aspects: (1) physical model, (2) detection statistic, (3) metrics, and (4) validation.

Physical Model. 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* [4, 38]. In system identification one often has to use either Auto-Regressive Moving Average with eXogenous inputs (ARMAX) or linear state-space models. Two

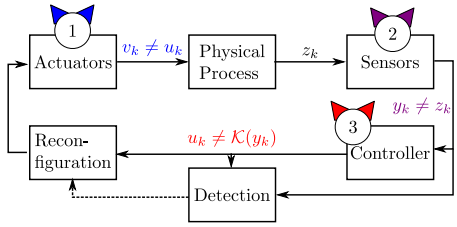

Figure 1: Different attack points in a control system: (1) Attack on the actuators (blue): $v_k \neq u_k$, (2) Attack on the sensors (purple): $y_k \neq z_k$, (3) Attack on the controller (red): $u_k \neq \mathcal{K}(y_k)$

popular models used by the papers we survey are **Auto-Regressive (AR)** models and **Linear Dynamical State-space (LDS)** models.

An AR model for a time series y_k is given by

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

where α_i are obtained through system identification and y_i the last N sensor measurements. The coefficients α_i can be obtained by solving an optimization problem that minimizes the residual error (e.g., least squares) [37].

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

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

where A, B, C, and D are matrices modeling the dynamics of the physical system. Most physical systems are strictly causal and therefore $D = 0$ in general. The control commands $u_k \in \mathbb{R}^p$ affect the next time step of the state of the system $x_k \in \mathbb{R}^n$ and sensor measurements $y_k \in \mathbb{R}^q$ 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. To make a prediction, we i) first need y_k and u_k to obtain a *state estimate* $\hat{x}_{k+1}$ and ii) use the estimate to predict $\hat{y}_{k+1} = C\hat{x}_{k+1}$. A large body of work on power systems employs the second equation from Eq. (2) without the dynamic state equation. We refer to this special case of LDS used in power systems as **Static Linear State-space (SLS)** models.

Detection Statistic. If the observations we get from sensors y_k are significantly different from the ones we expect (i.e., if the residual is large) we generate an alert. A **Stateless** test, raises an alarm for every 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 nonparametric 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 $S_{k+1} = 0$. The summary of our taxonomy for modeling the system and to detect an anomaly in the residuals is given in Fig. 2

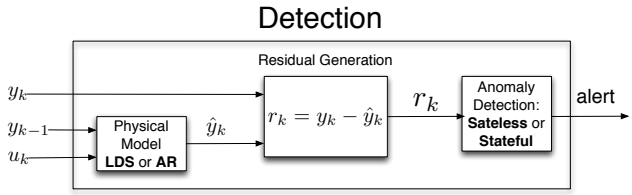

Figure 2: The detection block from Fig. 1 focusing on our taxonomy.

Metrics. An evaluation metric is used to determine the effectiveness of the physics-based attack detection algorithm. Popular evaluation metrics are the True Positive Rate (TPR) and the False Positive Rate (FPR)—the trade-off between these two numbers is called the Receiver Operating Characteristic (ROC) curve. Some papers just plot the residuals (without quantifying the TPR or FPR values), and other papers just measure the impact of attacks.

Validation. The experimental setting to validate proposals can use simulations, data from real-world operating systems, and testbeds. Testbeds can be classified as testbeds controlling a real-system or a testbed with Hardware-in-the-Loop (HIL) where part of the physical system is simulated in a computer. For our purposes a HIL testbed is similar to having pure simulations, because the model of the physical system is given by the algorithm running on a computer.

2.3 Limitations of Previous Work

There is a large variety of previous work but because of the diversity of domains (e.g., power systems, industrial control, and theoretical studies) and academic venues (e.g., security, control theory, and power systems conferences), the field has not been presented in a unified way with a common language that can be used to identify trends, alternatives, and limitations. Using our previously defined taxonomy, in this section we discuss previous work and summarize our results in Table 1.

The columns in Table 1 are arranged by conference venue (we assigned workshops to the venue that the main conference is associated with), we also assigned conferences associated with CPSWeek to control conferences because of the overlap of attendees to both venues. We make the following observations: (1) the vast majority of prior work use 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 or adversary model used to evaluate proposed attack-detection algorithms; and (4) no previous work has validated their work with all three options: simulations, testbeds and real-world data.

The first three observations (1-3) are related: while previous work has used different statistical tests (stateless vs. stateful) and models of the physical system to predict its expected behavior, so far they have not been compared against each other, and this makes it difficult to build upon previous work (it is impossible to identify best practices without a way to compare different proposals). To address this problem we propose a general-purpose evaluation metric in § 4 that leverages our stealthy adversary model, and then compare previously proposed methods. Our results show that while stateless tests are more popular in the literature, stateful tests are better to limit the impact of stealthy attackers. In addition, we show that LDS models are better than AR models, that AR models proposed in previous work can be improved by leveraging correlation among different signals, and that having an integral controller can limit the impact of stealthy actuation attacks.

To address point (4) we conduct experiments using all three options: a testbed with a real physical process under control § 5.1, real-world data § 5.2, and simulations § 5.3. We show the advantages and disadvantages of each experimental setup, and the insights each of these experiments provide.

3. MOTIVATING EXAMPLE

The testbed we use for our experiments is a room-size, water treatment plant consisting of 6 stages to purify raw water. The testbed has a total of 12 PLCs (6 main PLCs and 6 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 acts as the main water buffer supplying 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 *Pretreatment* 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. This stage is illustrated in Fig. 3 and will be used in our motivating example. *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. After the small residuals are removed by the UF system, the remaining chlorines are destroyed in the *Dechlorinization* stage, using ultraviolet chlorine destruction unit and by dosing a solution of sodium bisulphite. *Reverse Osmosis* (RO) system is designed to reduce inorganic impurities by pumping the filtrated and dechlorinated water with a high pressure. Finally, in *RO final product* stage stores the RO product (clean water).

Figure 3: Stage controlling the pH level.

Attacking the *pH* level. In this process, the water's *pH* level is controlled by dosing the water with Hydrochloric Acid (HCl). Fig. 4 illustrates the normal operation of the plant: if the *pH* sensor reports a level above 7.05, the PLC sends a signal to turn On the HCl pump, and if the sensor reports a level below 6.95, it sends a signal to turn it Off.

Table 1: Taxonomy of related work. Columns are organized by publication venue.

	[8] Bobba *et al.*	[54] Sandberg *et al.*	[59] Teixeira *et al.*	[6] Bai, Gupta	[46] Mo *et al.*	[44] Mo, Sinopoli	[7] Bai *et al.*	[43] Miao *et al.*	[23] Hou *et al.*	[16] Eyisi *et al.*	[45] Mo *et al.*	[50] Pasqualetti *et al.*	[61] Teixeira *et al.*	[31] Kwon *et al.*	[60] Teixeira *et al.*	[15] Do *et al.*	[1, 2] Amin *et al.*	[56] Smith	[25] Kerns *et al.*	[33] Liang *et al.*	[20] Giani *et al.*	[13] Dan, Sandberg	[28] Kosut *et al.*	[26] Kim, Poor	[14] Davis *et al.*	[57] Sridhar, Govindarasu	[29] Koutsandria *et al.*	[40] Mashima *et al.*	[35, 36] Liu *et al.*	[49] Parvania *et al.*	[34] Lin *et al.*	[55] Shoukry *et al.*	[21] Hadziosmanovic *et al.*	[10] Cardenas *et al.*	[67] Wang *et al.*	[42] McLaughlin	[53] Sajjad *et al.*	[65] Valente, Cardenas	[30] Krotofil *et al.*	[66] Vukovic, Dan	[47] Morrow *et al.*	[12] Cui *et al.*	[9] Carcano *et al.*	[22] Hei *et al.*	[27] Kiss *et al.*
Venue	Control																		Smart/Power Grid									Security											Misc.							
Detection Statistic																																												
stateless	●	●	●	-	-	-	●	●	●	◐	●	●	●	●	●	-	●	◐	●	-	●	●	●	●	●	-	●	-	●	◐	●	-	●	-	◐	◐	-	-	●	●	●	●	-	●
stateful	-	-	-	◐	⊛	⊛	-	-	-	-	-	-	-	-	-	●	-	-	-	⊛	-	-	-	-	-	⊛	-	●	-	-	-	⊛	-	●	-	-	●	●	●	-	-	●	-	⊛	-
Physical Model																																											
AR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-
SLS	●	●	◐	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	●	●	●	●	◐	-	-	-	●	-	-	-	-	-	-	-	-	-	-	◐	◐	●	-	-	-
LDS	-	-	-	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	●	-	●	-	-	●	-	-	-	-	-	-	-	-
other	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	●	-	-	◐	◐	-	-	-	●	●	-	●	●	-	-	●	◐	●
Metrics*																																											
impact	-	●	-	●	-	-	●	-	●	●	-	●	●	●	●	-	-	-	●	●	-	●	-	●	-	-	●	●	●	-	●	-	-	-	-	-	-	●	-	●	-	●	-
statistic	-	-	●	-	●	-	-	●	●	-	-	●	●	●	●	-	●	-	-	●	-	-	●	-	-	-	●	-	●	-	-	-	-	-	-	-	●	●	●	-	●	-	-	●
TPR	-	-	-	-	●	●	-	-	-	-	●	-	-	-	-	●	-	-	-	-	●	-	●	-	-	●	-	-	-	-	-	●	●	-	●	-	-	-	-	-	-	●	-	-
FPR	-	-	-	-	●	-	-	-	-	-	●	-	-	-	-	●	-	-	-	-	-	-	●	-	-	●	-	●	-	-	-	●	-	-	●	-	-	-	-	-	-	●	-	-
Validation																																											
simulation	-	●	●	●	●	●	●	●	●	●	●	●	●	●	-	●	-	●	●	●	-	●	●	●	●	◐	-	●	-	●	-	-	●	●	-	●	-	●	●	●	●	-	-	●
real data	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	◐	●	-	-	-	-	-	●	-	-	-	-	●	-	-	-	-	-	●	-
testbed	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-	-	-	-	-	◐	-	●	●	-	-	●	-	-	-	-	-	●	-	-

Legend: ●: feature considered by authors, ◐: feature assumed implicitly but exhibits ambiguity, ⊛: a windowed stateful detection method is used, *Evaluation options have been abbreviated in the table: Attack Impact, Statistic Visualization, True Positive Rate, False Positive Rate.

The wide oscillations of the pH levels occur because there is a delay between the control actions of the HCl pump, and the water pH responding to it.

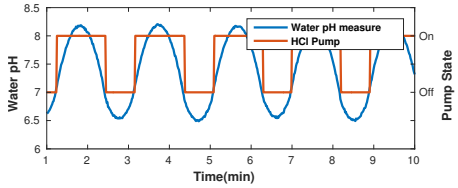

Figure 4: During normal operation, the water pH is kept in safe levels.

To detect attacks on the PLC, the pump or the sensor, we need to create a model of the physical system. While the system is nonlinear, let us first attempt it to model it as time-delayed LDS of order 2. The model is described by $pH_{k+1} = pH_k + u_{k-T_{delay}}$, where we estimate (by observing the process behavior) $u_{k-T_{delay}} = -0.1$ after a delay of 35 time steps after the pump is turned On, and 0.1 after a delay of 20 time steps after it is turned Off. We then compare the predicted and observed behavior, compute the residual, and apply a stateless, and a stateful test to the residual. If either of these statistics goes above a defined threshold, we raise an alarm.

We note that high or low pH levels can be dangerous. In particular, if the attacker can drive the pH below 5, the acidity of the water will damage the membranes of the *Ultra Filtration* and *Reverse Osmosis* stages, the pipes, and even sensor probes.

We launch a wired Man-In-The-Middle (MitM) attack between the field devices (sensors and actuators) and the PLC by injecting a malicious device in the EtherNet/IP ring of the testbed, given that the implementation of this protocol is unauthenticated. A detailed implementation of our attack is given in our previous work [64]. In particular, our MitM intercepts sensor values coming from the HCL pump and the pH sensor, and intercept actuator commands going to the HCl pump, to inject false sensor readings and commands sent to the PLC and HCl pump.

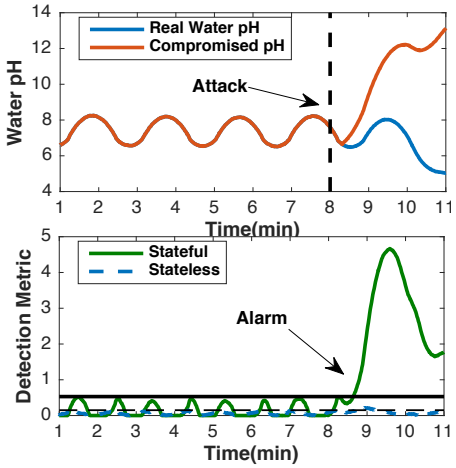

Figure 5: Attack to the pH sensor.

Our attack sends false sensor data to the PLC, faking a high pH level so the pump keeps running, and thus driving the acidity of the water to unsafe levels, as illustrated in Fig. 5. Notice that both, stateless and stateful tests detect this attack (each test has a different threshold set to main-

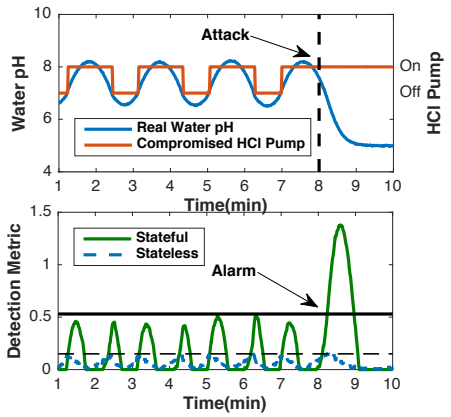

Figure 6: Attack to the pump *actuator*.

tain a probability of false alarm of 0.01). We also launched an attack on the pump (actuator). Here the pump ignores Off control commands from the PLC, and sends back messages stating that it is indeed Off, while in reality it is On. As illustrated in Fig. 6, only the stateful test detects this attack. We also launched several random attacks that were easily detected by the stateful statistic, and if we were to plot the ROC curve of these attacks, we would get 100% detection rate.

Observations. As we can see, it is very easy to create attacks that can be detected. Under these simulations we could initially conclude that our LDS model combined with the stateful anomaly detection are good enough; after all, they detected all attacks we launched. However, are these attacks enough to conclude that our LDS model is good enough? And if these attacks are not enough, then which types of attacks should we launch?

Notice that for any physical system, a sophisticated attacker can spoof deviations that follow relatively close the "physics" of the system while still driving the system to a different state. How can we measure the performance of our anomaly detection algorithm against these attacks? How can we measure the effectiveness of our anomaly detection tool if we assume that the attacker will always **adapt** to our algorithms and launch an undetected attack? And if our algorithms are not good enough, how can we design better algorithms? If by definition the attack is undetected, then we will always have a 0% true positive rate, therefore we need to devise new metrics to evaluate our systems.

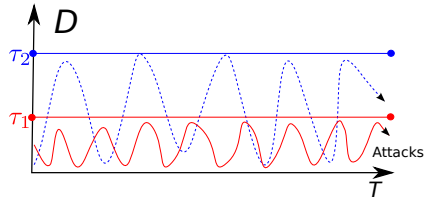

Figure 7: Our attacker adapts to different detection thresholds: If we select τ_2 the adversary launches an attack such that the detection statistic (dotted blue) remains below τ_2. If we lower our threshold to τ_1, the adversary selects a new attack such that the detection statistic (solid red) remains below τ_1.

4. A STRONGER ADVERSARY MODEL

We assume an attacker that has compromised a sensor (e.g. pH level in our motivating example) or an actuator (e.g. pump in our motivating example) in our system. We also assume that the adversary has complete knowledge of our system, i.e. she knows the physical model we use, the statistical test we use, and the thresholds we select to raise alerts. Given this knowledge, she generates a stealthy attack, where the detection statistic will always remain below the selected threshold.

While similar stealthy attacks have been previously proposed [13, 35, 36], in this paper we extend them for generic control systems including process perturbations and measurement noise, we force the attacks to remain stealthy against stateful tests, and also force the adversary to optimize the negative impact of the attack. In addition, we assume our adversary is **adaptive**, so if we lower the threshold to fire an alert, the attacker will also change the attack so that the anomaly detection statistic remains below the threshold. This last property is illustrated in Fig. 7.

Notice that this type of adaptive behavior is different from how traditional metrics such as ROC curves work, because they use the same attacks for different thresholds of the anomaly detector. On the other hand, our adversary model requires a new and unique (undetected) attack specifically tailored for every anomaly detection threshold. If we try to compute an ROC curve under our adversary model we would get a 0% detection rate because the attacker would generate a new undetected attack for every anomaly detection threshold.

This problem is not unique to ROC curves: most popular metrics for evaluating the classification accuracy of intrusion detection systems (like the intrusion detection capability, the Bayesian detection rate, accuracy, expected cost, etc.) are known to be a multi-criteria optimization problem between two fundamental trade-off properties: the false alarm rate, and the true positive rate [11], and as we have argued, using any metric that requires a true positive rate will be ineffective against our adversary model launching undetected attacks.

Observation. Most intrusion detection metrics are variations of the fundamental trade-off between false alarms and true positive rates [11], however, our adversary by definition will never be detected so we cannot use true positive rates (or variations thereof). Notice however that by forcing our adversary to remain undetected, we are effectively forcing her to launch attacks that follow closely the physical behavior of the system (more precisely, we are forcing our attacker to follow more closely our *Physical Model*), and by following closer the behavior of the system, then the attack impact is reduced: the attack needs to appear to be a plausible physical system behavior. So the trade-off we are looking for with this new adversary model is not one of *false positives* vs. *true positives*, but one between *false positives* and *the impact of undetected attacks.*

New Metric. To define precisely what we mean by *impact of undetected attack* we select one (or more) variables of interest (usually a variable whose compromise can affect the safety of the system) in the process we want to control– e.g., the pH level in our motivating example. The impact of the undetected attack will then be, how much can the attacker drive that value towards its intended goal (e.g., how much can the attacker lower the pH level while remaining undetected) per unit of time.

Therefore we propose a new metric consisting of the trade-

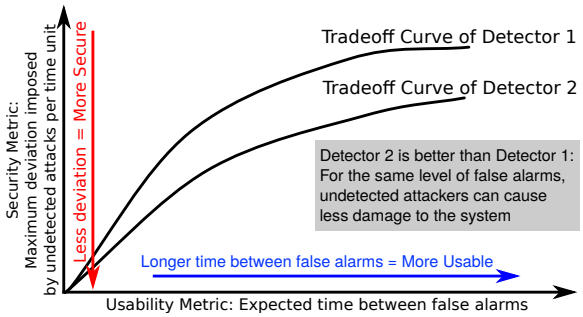

Figure 8: Illustration of our proposed tradeoff metric. The y-axis is a measure of the maximum deviation imposed by undetected attacks per time unit Δ_X/TU, and the x-axis represents the expected time between false alarms $\mathbb{E}[T_{fa}]$. Anomaly detection algorithms are then evaluated for different points in this space.

off between the maximum deviation per time unit imposed by undetected attacks (y-axis) and the expected time between false alarms (x-axis). Our proposed trade-off metric is illustrated in Fig. 8, and its comparison to the performance of Receiver Operating Characteristic (ROC) curves against our proposed adversary model is illustrated in Fig. 9.

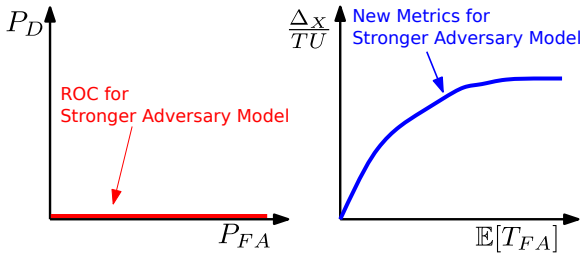

Figure 9: Comparison of ROC curves with our proposed metric: ROC curves are not a useful metric against a stealthy and adaptive adversary.

Notice that while the y-axis of our proposed metric is completely different to ROC curves, the x-axis is similar, but with a key difference: instead of using the probability of false alarms, we use instead the expected time between false alarms $\mathbb{E}[T_{fa}]$. This quantity has a couple of advantages over the false alarm rate: (1) it addresses the deceptive nature of low false alarm rates due to the base-rate fallacy [5], and (2) it addresses the problem that several anomaly detection statistics make a decision ("alarm" or "normal behavior") at non-constant time-intervals.

We now describe how to compute the y-axis and the x-axis of our proposed metric.

4.1 Computing the X and Y axis of Fig. 8

Computing Attacks Designed for the Y-axis of our Metric. The adversary wants to maximize the deviation of a variable of interest y_k (per time unit) without being detected. The true value of this variable is $y_k, y_{k+1}, \ldots, y_N$, and the attack starts at time k, resulting in a new observed time series $y^a_k, y^a_{k+1}, \ldots, y^a_N$. The goal of the attacker is to maximize the distance $\max_i ||y_i - y^a_i||$. Recall that in general y_k can be a vector of n sensor measurements, and that the attack y^a_k is a new vector where some (or all) of the sensor measurements are compromised.

An optimal greedy-attack (y^{a*}) at time $k \in [\kappa, \kappa_f]$ (where κ and κ_f are the initial and final attack times, respectively), satisfies the equation: $y^{a*}_{k+1} = \arg\max_{y^a_{k+1}} f(y^a_{k+1})$ (where $f(y^a_{k+1})$ is defined by the designer of the detection method to quantify the attack impact) subject to not raising an alert (instead of max it can be min). For instance, if $f(y^a_{k+1}) = ||y_{k+1} - y^a_{k+1}||$, the greedy attack for a stateless test is: $y^{a*}_{k+1} = \hat{y}_{k+1} \pm \tau$. The greedy optimization problem for an attacker facing a stateful CUSUM test becomes $y^{a*}_{k+1} = \max\{y^a_{k+1} : S_{k+1} \leq \tau\}$. Because $S_{k+1} = (S_k + r_k - \delta)$ the optimal attack is given when $S_k = \tau$, which results in $y^{a*}_{k+1} = \hat{y}_{k+1} \pm (\tau + \delta - S_k)$. For all attack times k greater than the initial time of attack κ, $S_k = \tau$ and $y^{a*}_{k+1} = \hat{y}_{k+1} \pm \delta$.

Generating undetectable **actuator attacks** is more difficult than **sensor attacks** because in several practical cases it is impossible to predict the outcome y_{k+1} with 100% accuracy, given the actuation attack signal v_k in Fig. 1. For our experiments when the control signal is compromised in § 5.3, we use the linear state space model from Eq. (2) to do a reverse prediction from the intended y^{a*}_{k+1} to obtain the control signal v_k that will generate that next sensor observation.

Computing the X-axis of our Metric. Most of the literature that reports false alarms uses the false alarm rate metric. This value obscures the practical interpretation of false alarms: for example a 0.1% false alarm rate depends on the number of times an anomaly *decision* was made, and the time-duration of the experiment: and these are variables that can be selected: for example a *stateful* anomaly detection algorithm that monitors the difference between expected $\hat{y}_k$ and observed y_k behavior has three options with every new observation k: (1) it can declare the behavior as *normal*, (2) it can generate an *alert*, (3) it can decide that the current evidence is inconclusive, and it can decide to take one more measurement y_{k+1}.

Because *the amount of time T that we have to observe the process and then make a decision is not fixed, but rather is a variable that can be selected*, using the false alarm rate is misleading and therefore we have to use ideas from *sequential detection theory* [24]. In particular, we use the average *time between false alarms* T_{FA}, or more precisely, the expected time between false alarms $\mathbb{E}[T_{FA}]$. We argue that telling security analysts that e.g., they should expect a false alarm every hour is a more direct and intuitive metric rather than giving them a probability of false alarm number over a decision period that will be variable if we use *stateful* anomaly detection tests. This way of measuring alarms also deals with the *base rate fallacy*, which is the problem where low false alarm rates such as 0.1% do not have any meaning unless we understand the likelihood of attacks in the dataset (the base rate of attacks). If the likelihood of attack is low, then low false alarm rates can be deceptive [5].

In all the experiments, the usability metric for each evaluated detection mechanism is obtained by counting the number of false alarms nFA for an experiment with a duration T_E under normal operation (without attack), so for each threshold τ we calculate the estimated time for a false alarm by $E[T_{fa}] \approx T_E/nFA$. Computing the average time between false alarms in the CUSUM test is more complicated than with the stateless test. In the CUSUM case, we need to compute the evolution of the statistic S_k for every threshold we test, because once S_k hits the threshold we have to reset it to zero.

Notice that while we have defined a specific impact for

Algorithm 1: Computing Y axis

1: Define $f(y^a_{k+1})$
2: Select $\tau_{set} = \{\tau_1, \tau_2, \ldots\}$, κ, κ_f, and $K_{set} = \{\kappa, \ldots, k_f - 1\}$
3: $\forall(\tau, k) \in \tau_{set} \times K_{set}$, find
4:

$$y^{a*}_{k+1}(\tau) = \arg\max_{y^a_{k+1}} f(y^a_{k+1})$$

$$s.t.$$

$$\text{Detection Statistic} \leq \tau$$

5: $\forall \tau \in \tau_{set}$, calculate

$$y - axis = \max_{k \in K_{set}} f(y^{a*}_{k+1}(\tau))$$

Algorithm 2: Computing X axis

1: Observations Y^{na} with no attacks of time-duration T_E
2: $\forall \tau \in \tau_{set}$, compute

$$\text{Detection Statistic: } D_S(Y^{na})$$
$$\text{Number of false alarms: } nFA(D_S, \tau)$$
$$x - axis = E[T_{fa}(\tau)] = T_E/nFA$$

undetected attacks in our y-axis for clarity, we believe that designers who want to evaluate their system using our metric should define an appropriate *worst case undetected attack* optimization problem specifically for their system. In particular, the y-axis can be a representation of a cost function f of interest to the designer. There are a variety of metrics (optimization objectives) that can be measured such as the product degradation from undetected attacks, or the historical deviation of the system under attack $\sum_i |y_i - \hat{y}^a_i|$ or the deviation at the end of the attack $|y_N - \hat{y}^a_N|$, etc. A summary of how to compute the y-axis and the x-axis of our metric is given in Algorithms 1 and 2.

5. EXPERIMENTAL RESULTS

Table 2: Advantages and disadvantages of different evaluation setups.

Reliability of:	X-Axis	Y-Axis
Real Data	●	○
Testbed	◐	◐
Simulation	○	●

● = well suited, ◐ = partially suitable, ○ = least suitable

We evaluate anomaly detection systems under the light of our *Stronger Adversary Model* (see section § 4), using our new metrics in a range of test environments, with individual strengths and weaknesses (see Table 2). As shown in the table, real-world data allows us to analyze operational large-scale scenarios, and therefore it is the best way to test the x-axis metric $\mathbb{E}[T_{fa}]$. Unfortunately, real-world data does not give researchers the flexibility to launch attacks and measure the impact on all parts of the system. Such interactive testing requires the use of a dedicated physical testbed.

A physical testbed has typically a smaller scale than a real-world operational system, so the fidelity in false alarms might not be as good as with real data, but on the other hand, we can launch attacks. The attacks we can launch are, however, constrained because physical components and devices may suffer damage by attacks that violate the safety requirements and conditions for which they were designed for. Moreover, attacks could also drive the testbed to states that endanger the operator's and environment's safety. Therefore, while a testbed provides more experimental interaction than real data, it introduces safety constraints for launching attacks.

Simulations on the other hand, do not have these constraints and a wide variety of attacks can be launched. So our simulations will focus on attacks to actuators and demonstrate settings that cannot be achieved while operating a real-world system because of safety constraints. Simulations also allow us to easily change the control algorithms and to our surprise, we found that control algorithms have a big impact on the ability of our attacker to achieve good results in the y-axis of our metric. However, while simulations allow us to test a wide variety of attacks, the problem is that the false alarms measured with a simulation are not going to be as representative as those obtained from real data or from a testbed.

5.1 Physical Testbed (EtherNet/IP packets)

In this section, we focus on testbeds that control a real physical process, as opposed to testbeds that use a *Hardware-In-the-Loop* (HIL) simulation of the physical process. A HIL testbed is similar to the experiments we describe in § 5.3.

We developed an attacker who has complete knowledge of the physical behavior of the system and can manipulate EtherNet/IP packets and inject attacks. We now apply our metric to the experiments we started in section § 3.

Attacking pH Level. Because this system is highly nonlinear, apart from the simple physical model (LDS) of order 2 we presented in section § 3, we also applied a system identification to calculate higher order system models: an LDS model of order 20 and two nonlinear models (order 50 and 100) based on wavelet networks [52]. Fig. 10 shows the minimum pH achieved by the attacker after 4-minutes and against three different models. *Notice that the nonlinear models limited the impact of the stealthy attack by not allowing deviations below a pH of 5*, while our linear model (which was successful in detecting attacks in our motivating example) was not able to prevent the attacker from taking the pH below 5.

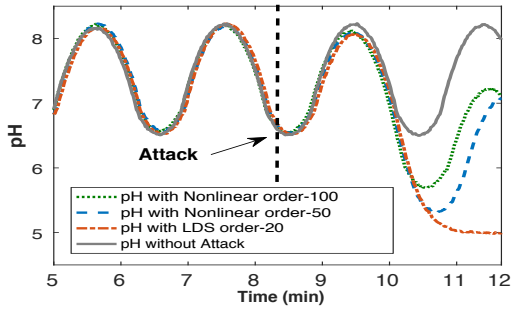

Figure 10: pH deviation imposed by greedy attacks while using stateful detection ($\tau = 0.05$) with both, LDS and nonlinear models.

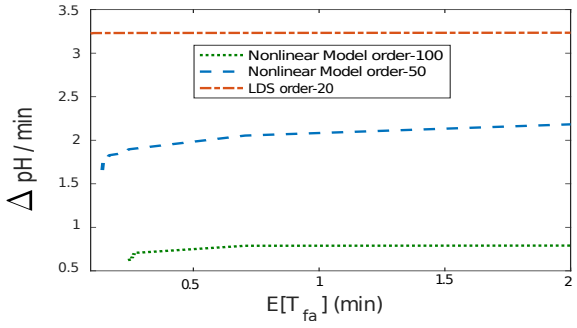

Figure 11: Comparison of LDS and nonlinear models to limit attack impact using our metric. Higher order nonlinear models perform better.

Fig. 11 illustrates the application of our proposed metric over 10 different undetected greedy attacks, each averaging 4 minutes, to evaluate the three system models used for detection. Given enough time, it is not possible to restrict a deviation of pH below 5. Nevertheless, for all $E[T_{fa}](min)$, the nonlinear model of order 100 performs better than the nonlinear model of order 50 and the LDS of order 20, limiting the impact of the attack per minute Δ_{pH}/min. It would take over 5 minutes for the attacker to deviate the pH below 5 without being detected using a nonlinear model of order 100, whereas it would take less than 3 minutes with the nonlinear of order 50 and the LDS of order 20.

5.1.1 Attacking the Water Level

Now we turn to another stage in our testbed. The goal of the attacker this time is to deviate the water level in a tank as much as possible until the tank overflows.

While in the pH example we had to use system identification to learn LDS and nonlinear models, the evolution of the water level in a tank is a well-known LDS system that can be derived from first principles. 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 time-discretization of 1 s, we obtain an LDS 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 $C = 1$.

Recall that the goal of the attacker is to deviate the water level in a tank as much as possible until the tank overflows. In particular, the attacker increases the water level sensor signal at a lower rate than the real level of water (Fig. 12) with the goal of overflowing the tank. A **successful attack** occurs if the PLC receives from the sensor a *High* water-level message (the point when the PLC sends a command to close the inlet), and at that point, the deviation (Δ) between the real level of water and the "fake" level (which just reached the High warning) is $\Delta \geq Overflow - High$. Fig. 12 shows three water level attacks with different increment rates, starting from the *Low* level setting and stopping at the *High* level setting, and their induced maximum Δ over the real level. Only attacks a_1 and a_2 achieve a successful overflow (only a_2 achieves a water spill), while a_3 deviates the water level without overflow. In our experiment, *High* corresponds to a water level of 0.8 m and *Low* to 0.5 m. Overflow occurs at 1.1 m. The testbed has a drainage system to allow attacks that overflow the tank.

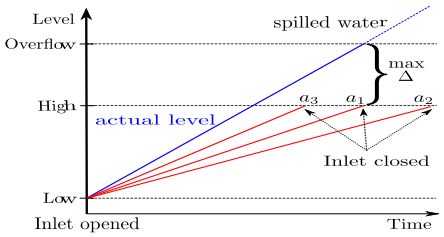

Figure 12: Impact of different increment rates on overflow attack. The attacker has to select the rate of increase with the lowest slope while remaining undetected.

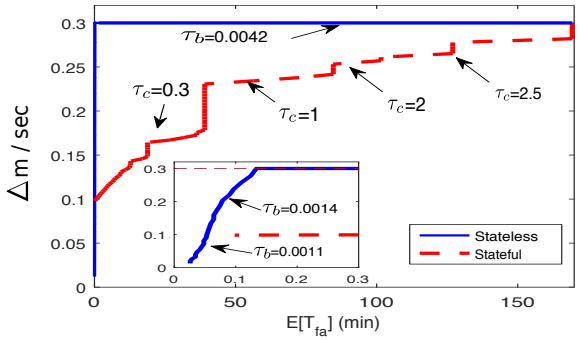

Figure 13: Comparison of stateful and stateless detection. At 0.3m the tank overflows, so stateless tests are not good for this use case. τ_b, τ_c correspond to the threshold associated to some $E[T_{fa}]$.

Because it was derived from "first principles", our LDS model is a highly accurate physical model of the system, so there is no need to test alternative physical models. However, we can combine our LDS model with a stateless test, and with a stateful test and see which of these detection tests can limit the impact of stealthy attacks.

In particular, to compute our metric we need to test stateless and stateful mechanisms and obtain the security metric that quantifies the impact Δ of undetected attacks for several thresholds τ. We selected the parameter $\delta = 0.002$ for the stateful (CUSUM) algorithm, such that the detection metric S_k remains close to zero when there is no attack. The usability metric is calculated for $T_E = 8$ h, which is the time of the experiment without attacks.

Fig. 13 illustrates the maximum impact caused by 20 different undetected attacks, each of them averaging 40 minutes. Even though the attacks remained undetected, the impact using stateless detection is such that a large amount of water can be spilled. Only for very small thresholds is it possible to avoid overflow, but it causes a large number of false alarms. On the other hand, stateful detection limits

the impact of the adversary. Note that to start spilling water (i.e., $\Delta > 0.3\ m$) a large threshold is required. Clearly, selecting a threshold such that $E[T_{fa}] = 170\ min$ can avoid the spilling of water with a considerable tolerable number of false alarms.

In addition to attacking sensor values, we would like to analyze undetected actuation attacks. To launch attacks on the actuators (pumps) of this testbed, we would need to turn them On and Off in rapid succession in order try to maintain the residuals of the system low enough to avoid being detected. We cannot do this on real equipment because the pumps would get damaged. Therefore, we will analyze undetected actuator attacks with simulations (where equipment cannot be damaged) in § 5.3.

5.2 Large-Scale Operational Systems (Modbus packets)

We were allowed to place a network sniffer on a real-world operational large-scale water facility in the U.S. We collected more than 200GB of network packet captures of a system using the Modbus/TCP [63] industrial protocol. Our goal is to extract the sensor and control commands from this trace and evaluate and compare alternatives presented in the survey.

The network has more than 100 controllers, some of them with more than a thousand registers. In particular, 1) 95% of transmissions are Modbus packets and the rest 5% corresponds to general Internet protocols; 2) the trace captured 108 Modbus devices, of which one acts as central master, one as external network gateway, and 106 are slave PLCs; 3) of the commands sent from the master to the PLCs, 74% are *Read/Write Multiple Registers* (0x17) commands, 20% are *Read Coils* (0x01) commands, and 6% are *Read Discrete Inputs* (0x02) commands; and 4) 78% of PLCs count with 200 to 600 registers, 15% between 600 to 1000, and 7% with more than 1000.

We replay the traffic traces in packet capture (pcap) format and use Bro [51] to track the memory map of holding (read/write) registers from PLCs. We then use Pandas [68], a Python Data Analysis Library, to parse the log generated by Bro and to extract per PLC the time series corresponding to each of the registers. Each time series corresponds to a signal (y_k) in our experiments. We classify the signals as 91.5% *constant*, 5.3% *discrete* and 3.2% *continuous* based on the data characterization approach proposed to analyze Modbus traces [21] and uses AR models (as in Eq. (1)). We follow that approach by modeling the continuous time-series in our dataset with AR models. The order of the AR model is selected using the *Best Fit* criteria from the Matlab system identification toolbox [39], which uses unexplained output variance, i.e., the portion of the output not explained by the AR model for various orders [41].

Using the AR model, our first experiment centers on deciding which statistical detection test is better, a stateless test or the stateful CUSUM change detection test. Fig. 14 shows the comparison of stateless vs. stateful tests with our proposed metrics (where the duration of an undetected attack is 10 minutes). As expected, once the CUSUM statistic reaches the threshold $S_k = \tau$, the attack no longer has enough room to continue deviating the signal without being detected, and larger thresholds τ do not make a difference once the attacker reaches the threshold, whereas for the stateless test, the attacker has the ability to change the measurement by τ units at every time step.

Having shown that a CUSUM (stateful) test reduces the impact of a stealthy attack when compared to the stateless test we now show how to improve the AR physical model previously used by Hadziosmanovic et al. [21]. In particular, we notice that Hadziosmanovic et al. use an AR model *per signal*; this misses the opportunity of creating models of how multiple signals are correlated, creating correlated physical models will limit the impact of undetected attacks.

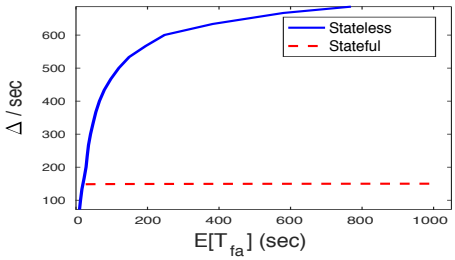

Figure 14: Stateful performs better than stateless detection: The attacker can send larger undetected false measurements for the same expected time to false alarms.

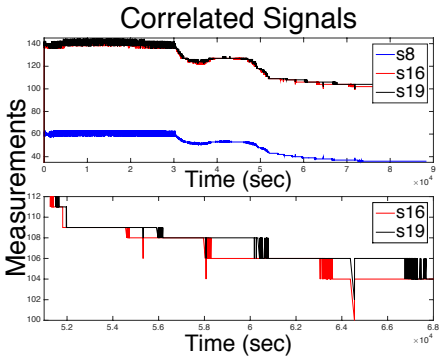

Figure 15: Three example signals with significant correlations. Signal S_{16} is more correlated with S_{19} than it is with S_8.

Spatial and Temporal Correlation. In an ideal situation the water utility operators could help us identify all control loops and spatial correlations of all variables (the water pump that controls the level of water in a tank etc.); however, this process becomes difficult to perform in a large-scale system with thousands of control and sensor signals exchanged every second; therefore we now attempt to find correlations empirically from our data. We correlate signals by computing the correlation coefficients of different signals $s_1, s_2, \cdots, s_N$. The correlation coefficient is a normalized variant of the mathematical covariance function: $\text{corr}(s_i, s_j) = \frac{\text{cov}(s_i,s_j)}{\sqrt{\text{cov}(s_i,s_i)\text{cov}(s_j,s_j)}}$ where $\text{cov}(s_i, s_j)$ denotes the covariance between s_i and s_j and correlation ranges between $-1 \leq \text{corr}(s_i, s_j) \leq 1$. We then calculate the *p-value* of the test to measure the significance of the correlation between signals. The *p-value* is the probability of having a correlation as large (or as negative) as the observed value when the true correlation is zero (i.e., testing the null hypothesis of no correlation, so lower values of p indicate higher evidence of correlation). We were able to find 8,620 correlations to be highly significant with $p = 0$.

Because $\text{corr}(s_i, s_j) = \text{corr}(s_j, s_i)$ there are 4,310 unique significant correlated pairs. We narrow down our attention to $\text{corr}(s_i, s_j) > .96$. Fig. 15 illustrates three of the correlated signals we found. Signals s_{16} and s_{19} are highly correlated with $\text{corr}(s_{16}, s_{19}) = .9924$ while s_8 and s_{19} are correlated but with a lower correlation coefficient of $\text{corr}(s_8, s_{19}) = .9657$. For our study we selected to use signal s_8 and its most correlated signal s_{17} which are among the top most correlated signal pairs we found with $\text{corr}(S_8, S_{17}) = .9996$.

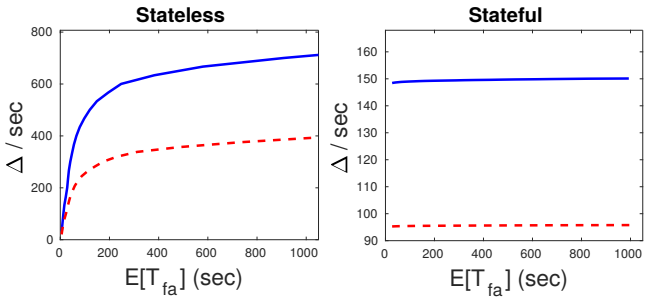

Figure 16: Using the defined metrics, we show how our new correlated AR models perform better (with stateless or stateful tests) than the AR models of independent signals.

Our experiments show that an AR model trained with correlated signals (see Fig. 16) is more effective in limiting the maximum deviation the attacker can achieve (assuming the attacker only compromises one of the signals). For that reason, we encourage future work to use correlated AR models rather than AR models of single signals.

5.3 Simulations of the Physical World

With simulations we can launch actuator attacks without the safety risk of damaging physical equipment. In particular, in this section we launch actuation attacks and show how the control algorithm used can significantly limit the impact of stealthy attackers. In particular we show that the Integrative part of a Proportional Integral Derivative (PID) control algorithm (or a PI or I control algorithm) can correct the deviation injected by the malicious actuator, and force the system to return to the correct operating state.

We use simulations of primary frequency control in the power grid as this is the scenario used by the Aurora attack [69]. Our goal is to maintain the frequency of the power grid as close as possible to 60Hz, subject to perturbations—i.e., changes in the Mega Watt (MW) demand by consumers—and attacks.

We assume that the attacker takes control of the actuators. When we consider attacks on a control signal, we need to be careful to specify whether or not the anomaly detection system can observe the false control signal. In this section, we assume the worst case: our anomaly detection algorithm cannot see the manipulated signal and indirectly observes the attack effects from sensors (e.g., v_k is controlled by the attacker, while the detection algorithm observes the valid u_k control signal, see Fig. 1).

Attacking a sensor is easier for our stealthy adversary because she knows the exact false sensor value $\hat{y}$ that will allow her to remain undetected while causing maximum damage. On the other hand, by attacking the actuator the attacker needs to find the input u_k that deviates the frequency enough, but still remains undetected. This is harder because even if the attacker has a model of the system, the output signal is not under complete control of the attacker: consumers can also affect the frequency of the system (by increasing or decreasing electricity consumption), and therefore they can cause an alarm to be generated if the attacker is not conservative. We assume the worst possible case of an omniscient adversary that knows how much consumption will happen at the next time-step (this is a conservative approach to evaluate the security of our system, in practice we expect the anomaly detection system to perform better because no attacker can predict the future).

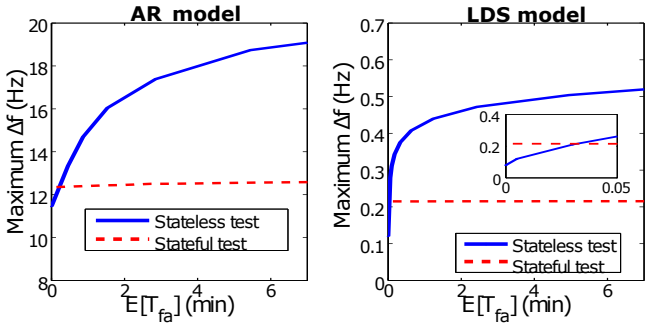

Figure 17: These figures show two things: (1) the stateful (CUSUM) test performs better than stateless tests when using AR (left) or LDS (right) models, and (2) LDS models perform an order of magnitude better than AR models (right vs left). Only for really small values of $\tau < \delta$ (0.04 minutes on average between false alarms), will the stateless test performs better than the stateful test.

We now evaluate all possible combinations of the popular *physical models* and *detection statistics* illustrated in Table 1. In particular we want to test AR models vs. LDS models estimated via system identification (SLS models do not make sense here because our system is dynamic) and stateless detection tests vs. stateful detection tests.

We launch an undetected actuator attack after 50 seconds using stateless and stateful detection tests for both: AR and LDS physical models. Our experiments show that LDS models outperform AR models, and that stateful models (again) outperform stateless models, as illustrated in Fig 17. These wide variations in frequency would not be tolerated in a real system, but we let the simulations continue for large frequency deviations to illustrate the order of magnitude ability from LDS models to limit the impact of stealthy attackers when compared to AR models.

Having settled for LDS physical models with CUSUM as the optimal combination of physical models with detection tests, we now evaluate the performance of different control algorithms, a property that has rarely been explored in our survey of related work. In particular, we show how Integrative control is able to correct undetected actuation attacks.

In particular we compare one of the most popular control algorithms: P control, and then we compare it to PI control. If the system operator has a P control of the form $u_k = Ky_k$, the attacker can affect the system significantly, as illustrated in Fig. 18. However, if the system operator uses a PI control, the effects of the attacker are limited: The actuator attack will tend to deviate the frequency signal, but this deviation will cause the controller to generate a cumulative compensation (due to the integral term) and because the LDS model knows the effect of this cumulative compensation, it is going to expect the corresponding change in the sensor measure-

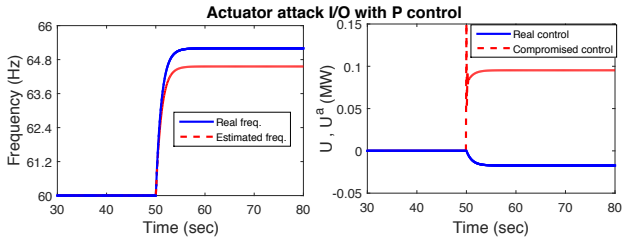

Figure 18: Left: The real (and trusted) frequency signal is increased to a level higher than the one expected (red) by our model of physical system given the control commands. Right: If the defender uses a P control algorithm, the attacker is able to maintain a large deviation of the frequency from its desired 60Hz set point.

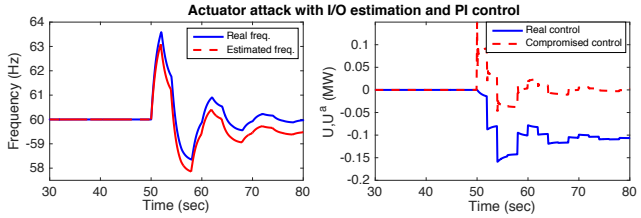

Figure 19: Same setup as in Fig. 18, but this time the defender uses a PI control algorithm: this results in the controller being able to drive the system back to the desired 60Hz operation point.

ment. As a consequence, to maintain the distance between the estimated and the real frequency below the threshold, the attack would have to decrease its action. At the end, the only way to maintain the undetected attack is when the attack is non-existent $u_k^a = 0$, as shown in Fig. 19.

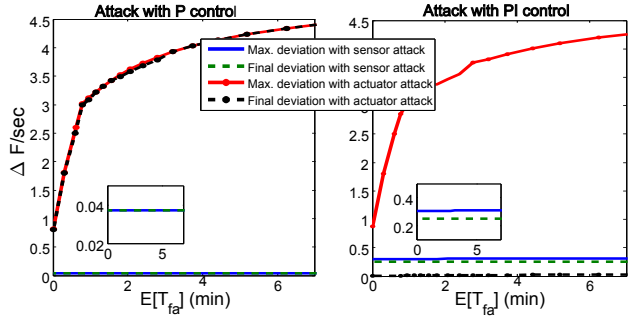

Figure 20: Differences between attacking sensors and actuators, and effects when the controller runs a P control algorithm vs. a PI control algorithm.

In all our previous examples with attacked sensors (except for the pH case), the worst possible deviation was achieved at the end of the attack, but for actuation attacks (and PI control), we can see that the controller is compensating the attack in order to correct the observed frequency deviation, and thus the final deviation will be zero: that is, the asymptotic deviation is zero, while the transient impact of the attacker can be high. Fig. 20 illustrates the difference between measuring the maximum final deviation of the state of the system achieved by the attacker, and the maximum temporary deviation of the state of the system achieved by the attacker.

As we can see, the control algorithm plays a fundamental role in how effective an actuation attack can be. An attacker that can manipulate the actuators at will can cause a larger frequency error but for a short time when we use PI control; however, if we use P control, the attacker can launch more powerful attacks causing long-term effects. On the other hand, attacks on sensors have the same long-term negative effects independent of the type of control we use (P or PI). Depending on the type of system, short-term effects may be more harmful than long-term errors. In our power plant example, a sudden frequency deviation larger than 0.5 Hz can cause irreparable damage on the generators and equipment in transmission lines (and will trigger protection mechanisms disconnecting parts of the grid). Small long-term deviations may cause cascading effects that can propagate and damage the whole grid.

While it seems that the best option to protect against actuator attacks is to deploy PI controls in all generators, several PI controllers operating in parallel in the grid can lead to other stability problems. Therefore often only the central Automatic Generation Control (AGC) implements a PI controller although distributed PI control schemes have been proposed recently [3].

Recall that we assumed the actuation attack was launched by an omniscient attacker that knows even the specific load the system is going to be subjected (i.e., it knows exactly how much will consumers demand electricity at every time-step, something not even the controller knows). For many practical applications, it will be impossible for the attacker to predict exactly the consequence of its actuation attack due to model uncertainties (consumer behavior) and random perturbations. As such, the attacker has a non-negligible risk of being detected when launching actuation attacks when compared to the 100% certainty the attacker has of not being detected when launching sensor attacks. In practice, we expect that an attacker that would like to remain undetected using actuation attacks will behave conservatively to accommodate for the uncertainties of the model, and thus we expect that the maximum transient deviation from actuation attacks will be lower.

6. CONCLUSIONS

6.1 Findings

We introduced theoretical and practical contributions to the growing literature of physics-based attack detection in control systems. Our literature review from different domains of expertise unifies disparate terminology, and notation. We hope our efforts can help other researchers refine and improve a common language to talk about physics-based attack detection across computer security, control theory, and power system venues.

In particular, in our survey we identified a lack of unified metrics and adversary models. We explained in this paper the limitations of previous metrics and adversary models, and proposed a novel stealthy and adaptive adversary model, together with its derived intrusion detection metric, that can be used to study the effectiveness of physics-based attack-detection algorithms in a systematic way.

We validated our approaches in multiple setups, including: a room-size water treatment testbed, a real large-scale operational system managing more than 100 PLCs, and sim-

ulations of primary frequency control in the power grid. We showed in Table 2 how each of these validation setups has advantages and disadvantages when evaluating the x-axis and y-axis of our proposed metric.

One result we obtained across our testbed, real operational systems, and simulations, is the fact that stateful tests perform better than stateless tests. This is in stark contrast to the popularity of stateless detection statistics as summarized in Table 1. We hope our paper motivates more implementations of stateful instead of stateless tests in future work.

We also show that for a stealthy actuator attack, PI controls play an important role in limiting the impact of this attack. In particular we show that the Integrative part of the controller corrects the system deviation and forces the attacker to have an effective negligible impact asymptotically.

Finally, we also provided the following novel observations: (1) finding spatio-temporal correlations of Modbus signals has not been proposed before, and we showed that these models are better than models of single signals, (2) while input/output models like LDS are popular in control theory, they are not frequently used in papers published in security conferences, and we should start using them because they perform better than the alternatives, unless we deal with a highly-nonlinear model, in which case the only way to limit the impact of stealthy attacks is to estimate nonlinear physical models of the system, and (3) we show why launching undetected attacks in actuators is more difficult than in sensors.

6.2 Discussion and Future Work

While physics-based attack detection can improve the security of control systems, there are some limitations. For example, in all our experiments the attacks affected the residuals and anomaly detection statistics while keeping them below the thresholds; however, there are special cases where depending on the power of the attacker or the characteristics of the plant, the residuals can remain zero (ignoring the noise) while the attacker can drive the system to an arbitrary state. For example, if the attacker has control of all sensors and actuators, then it can falsify the sensor readings so that our detector believes the sensors are reporting the expected state given the control signal, while in the meantime, the actuators can control the system to an arbitrary unsafe condition.

Similarly, some properties of the physical systems can also limit us from detecting attacks. For example, systems vulnerable to zero-dynamics attacks [61], unbounded systems [62], and highly non-linear or chaotic systems [48].

Finally, one of the biggest challenges for future work is the problem of how to respond to alerts. While in some control systems simply reporting the alert to operators can be considered enough, we need to consider automated response mechanisms in order to guarantee the safety of the system. Similar ideas in our metric can be extended to this case, where instead of measuring the false alarms, we measure the impact of a false response. For example, our previous work [10] considered switching a control system to open-loop control whenever an attack in the sensors was detected (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 control commands). As a result, instead of measuring the false alarm rate, we focused on making sure that a reconfiguration triggered by a false alarm would never drive the system to an unsafe state. Therefore maintaining safety under both, attacks and false alarms, will need to take priority in the study of any automatic response to alerts.

Acknowledgments

The work at UT Dallas was supported by NIST under award 70NANB14H236 from the U.S. Department of Commerce and by NSF CNS-1553683. The work of Justin Ruths at SUTD was supported by grant NRF2014NCR-NCR001-40 from NRF Singapore. H. Sandberg was supported in part by the Swedish Research Council (grant 2013-5523) and the Swedish Civil Contingencies Agency through the CERCES project. We thank the iTrust center at SUTD for enabling the experiments on the SWaT testbed.

Disclaimer

Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

7. REFERENCES

[1] S. Amin, X. Litrico, S. Sastry, and A. Bayen. Cyber security of water SCADA systems; Part I: Analysis and experimentation of stealthy deception attacks. *IEEE Transactions on Control Systems Technology*, 21(5):1963–1970, 2013.

[2] S. Amin, X. Litrico, S. Sastry, and A. Bayen. Cyber security of water SCADA systems; Part II: Attack detection using enhanced hydrodynamic models. *IEEE Transactions on Control Systems Technology*, 21(5):1679–1693, 2013.

[3] M. Andreasson, D. V. Dimarogonas, H. Sandberg, and K. H. Johansson. Distributed pi-control with applications to power systems frequency control. In *Proceedings of American Control Conference (ACC)*, pages 3183–3188. IEEE, 2014.

[4] K. J. Åström and P. Eykhoff. System identification—a survey. *Automatica*, 7(2):123–162, 1971.

[5] S. Axelsson. The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)*, 3(3):186–205, 2000.

[6] C.-z. Bai and V. Gupta. On Kalman filtering in the presence of a compromised sensor : Fundamental performance bounds. In *Proceedings of American Control Conference*, pages 3029–3034, 2014.

[7] C.-z. Bai, F. Pasqualetti, and V. Gupta. Security in stochastic control systems : Fundamental limitations and performance bounds. In *Proceedings of American Control Conference*, 2015.

[8] R. B. Bobba, K. M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. J. Overbye. Detecting false data injection attacks on DC state estimation. In *Proceedings of Workshop on Secure Control Systems*, volume 2010, 2010.

[9] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. N. Fovino, and A. Trombetta. A multidimensional critical state analysis for detecting intrusions in SCADA systems. *IEEE Transactions on Industrial Informatics*, 7(2):179–186, 2011.

[10] A. A. Cardenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry. Attacks against process control systems: risk assessment, detection, and response. In *Proceedings of the ACM symposium on information, computer and communications security*, pages 355–366, 2011.

[11] A. A. Cárdenas, J. S. Baras, and K. Seamon. A framework for the evaluation of intrusion detection systems. In *Proceedings of Symposium on Security and Privacy*, pages 77–91. IEEE, 2006.

[12] S. Cui, Z. Han, S. Kar, T. T. Kim, H. V. Poor, and A. Tajer. Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions. *Signal Processing Magazine, IEEE*, 29(5):106–115, 2012.

[13] G. Dán and H. Sandberg. Stealth attacks and protection schemes for state estimators in power systems. In *Proceedings of Smart Grid Commnunications Conference (SmartGridComm)*, October 2010.

[14] K. R. Davis, K. L. Morrow, R. Bobba, and E. Heine. Power flow cyber attacks and perturbation-based defense. In *Proceedings of Conference on Smart Grid Communications (SmartGridComm)*, pages 342–347. IEEE, 2012.

[15] V. L. Do, L. Fillatre, and I. Nikiforov. A statistical method for detecting cyber/physical attacks on SCADA systems. In *Proceedings of Control Applications (CCA)*, pages 364–369. IEEE, 2014.

[16] E. Eyisi and X. Koutsoukos. Energy-based attack detection in networked control systems. In *Proceedings of the Conference on High Confidence Networked Systems (HiCoNs)*, pages 115–124, New York, NY, USA, 2014. ACM.

[17] N. Falliere, L. O. Murchu, and E. Chien. W32. stuxnet dossier. White paper, Symantec Corp., Security Response, 2011.

[18] D. Formby, P. Srinivasan, A. Leonard, J. Rogers, and R. Beyah. Who's in control of your control system? Device fingerprinting for cyber-physical systems. In *Network and Distributed System Security Symposium (NDSS), Feb*, 2016.

[19] R. M. Gerdes, C. Winstead, and K. Heaslip. CPS: an efficiency-motivated attack against autonomous vehicular transportation. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, pages 99–108. ACM, 2013.

[20] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, and K. Poolla. Smart grid data integrity attacks: characterizations and countermeasures π. In *Proceedings of Smart Grid Communications Conference (SmartGridComm)*, pages 232–237. IEEE, 2011.

[21] D. Hadžiosmanović, R. Sommer, E. Zambon, and P. H. Hartel. Through the eye of the PLC: semantic security monitoring for industrial processes. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, pages 126–135. ACM, 2014.

[22] X. Hei, X. Du, S. Lin, and I. Lee. PIPAC: patient infusion pattern based access control scheme for wireless insulin pump system. In *Proceedings of INFOCOM*, pages 3030–3038. IEEE, 2013.

[23] F. Hou, Z. Pang, Y. Zhou, and D. Sun. False data injection attacks for a class of output tracking control systems. In *Proceedings of Chinese Control and Decision Conference*, pages 3319–3323, 2015.

[24] T. Kailath and H. V. Poor. Detection of stochastic processes. *IEEE Transactions on Information Theory*, 44(6):2230–2231, 1998.

[25] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys. Unmanned aircraft capture and control via gps spoofing. *Journal of Field Robotics*, 31(4):617–636, 2014.

[26] T. T. Kim and H. V. Poor. Strategic protection against data injection attacks on power grids. *IEEE Transactions on Smart Grid*, 2(2):326–333, 2011.

[27] I. Kiss, B. Genge, and P. Haller. A clustering-based approach to detect cyber attacks in process control systems. In *Proceedings of Conference on Industrial Informatics (INDIN)*, pages 142–148. IEEE, 2015.

[28] O. Kosut, L. Jia, R. Thomas, and L. Tong. Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures. In *Proceedings of Smart Grid Commnunications Conference (SmartGridComm)*, October 2010.

[29] G. Koutsandria, V. Muthukumar, M. Parvania, S. Peisert, C. McParland, and A. Scaglione. A hybrid network IDS for protective digital relays in the power transmission grid. In *Proceedings of Smart Grid Communications (SmartGridComm)*, 2014.

[30] M. Krotofil, J. Larsen, and D. Gollmann. The process matters: Ensuring data veracity in cyber-physical systems. In *Proceedings of Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 133–144. ACM, 2015.

[31] C. Kwon, W. Liu, and I. Hwang. Security analysis for cyber-physical systems against stealthy deception attacks. In *Proceedings of American Control Conference*, pages 3344–3349, 2013.

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

[33] J. Liang, O. Kosut, and L. Sankar. Cyber attacks on ac state estimation: Unobservability and physical consequences. In *Proceedings of PES General Meeting*, pages 1–5, July 2014.

[34] H. Lin, A. Slagell, Z. Kalbarczyk, P. W. Sauer, and R. K. Iyer. Semantic security analysis of SCADA networks to detect malicious control commands in power grids. In *Proceedings of the workshop on Smart energy grid security*, pages 29–34. ACM, 2013.

[35] Y. Liu, P. Ning, and M. K. Reiter. False data injection attacks against state estimation in electric power grids. In *Proceedings of ACM conference on Computer and communications security (CCS)*, pages 21–32. ACM, 2009.

[36] Y. Liu, P. Ning, and M. K. Reiter. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC)*, 14(1):13, 2011.

[37] L. Ljung. *The Control Handbook*, chapter System Identification, pages 1033–1054. CRC Press, 1996.

[38] L. Ljung. *System Identification: Theory for the User*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2 edition, 1999.

[39] L. Ljung. *System Identification Toolbox for Use with MATLAB*. The MathWorks, Inc., 2007.

[40] D. Mashima and A. A. Cárdenas. Evaluating electricity theft detectors in smart grid networks. In

Research in Attacks, Intrusions, and Defenses, pages 210–229. Springer, 2012.

[41] I. MathWorks. Identifying input-output polynomial models. www.mathworks.com/help/ident/ug/identifying-input-output-polynomial-models.html, October 2014.

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

[43] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas. Coding sensor outputs for injection attacks detection. In *Proceedings of Conference on Decision and Control*, pages 5776–5781, 2014.

[44] Y. Mo and B. Sinopoli. Secure control against replay attacks. In *Proceedings of Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 911–918. IEEE, 2009.

[45] Y. Mo, S. Weerakkody, and B. Sinopoli. Physical authentication of control systems: designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Control Systems*, 35(1):93–109, 2015.

[46] Y. L. Mo, R. Chabukswar, and B. Sinopoli. Detecting integrity attacks on SCADA systems. *IEEE Transactions on Control Systems Technology*, 22(4):1396–1407, 2014.

[47] K. L. Morrow, E. Heine, K. M. Rogers, R. B. Bobba, and T. J. Overbye. Topology perturbation for detecting malicious data injection. In *Proceedings of Hawaii International Conference on System Science (HICSS)*, pages 2104–2113. IEEE, 2012.

[48] E. Ott, C. Grebogi, and J. A. Yorke. Controlling chaos. *Physical review letters*, 64(11):1196, 1990.

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

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

[51] V. Paxson. Bro: a system for detecting network intruders in real-time. *Computer networks*, 31(23):2435–2463, 1999.

[52] S. Postalcioglu and Y. Becerikli. Wavelet networks for nonlinear system modeling. *Neural Computing and Applications*, 16(4-5):433–441, 2007.

[53] I. Sajjad, D. D. Dunn, R. Sharma, and R. Gerdes. 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)*, pages 43–53, New York, NY, USA, 2015. ACM. http://doi.acm.org/10.1145/2808705.2808713.

[54] H. Sandberg, A. Teixeira, and K. H. Johansson. On security indices for state estimators in power networks. In *Proceedings of Workshop on Secure Control Systems*, 2010.

[55] Y. Shoukry, P. Martin, Y. Yona, S. Diggavi, and M. Srivastava. PyCRA: Physical challenge-response authentication for active sensors under spoofing attacks. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1004–1015, New York, NY, USA, 2015. ACM.

[56] R. Smith. A decoupled feedback structure for covertly appropriating networked control systems. In *Proceedings of IFAC World Congress*, volume 18, pages 90–95, 2011.

[57] S. Sridhar and M. Govindarasu. Model-based attack detection and mitigation for automatic generation control. *Smart Grid, IEEE Transactions on*, 5(2):580–591, 2014.

[58] R. Tan, V. Badrinath Krishna, D. K. Yau, and Z. Kalbarczyk. Impact of integrity attacks on real-time pricing in smart grids. In *Proceedings of the SIGSAC conference on Computer & communications security (CCS)*, pages 439–450. ACM, 2013.

[59] A. Teixeira, S. Amin, H. Sandberg, K. H. Johansson, and S. S. Sastry. Cyber security analysis of state estimators in electric power systems. In *Proceedings of Conference on Decision and Control (CDC)*, pages 5991–5998. IEEE, 2010.

[60] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson. Attack models and scenarios for networked control systems. In *Proceedings of the conference on High Confidence Networked Systems (HiCoNs)*, pages 55–64. ACM, 2012.

[61] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson. Revealing stealthy attacks in control systems. In *Proceedings of Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1806–1813. IEEE, 2012.

[62] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson. A secure control framework for resource-limited adversaries. *Automatica*, 51:135–148, 2015.

[63] The Modbus Organization. Modbus application protocol specification, 2012. Version 1.1v3.

[64] D. Urbina, J. Giraldo, N. Tippenhauer, and A. Cárdenas. Attacking fieldbus communications in ics: Applications to the swat testbed. In *Proceedings of the Singapore Cyber-Security Conference (SG-CRC), Singapore*, volume 14, pages 75–89, 2016.

[65] J. Valente and A. A. Cardenas. Using visual challenges to verify the integrity of security cameras. In *Proceedings of Annual Computer Security Applications Conference (ACSAC)*. ACM, 2015.

[66] O. Vuković and G. Dán. On the security of distributed power system state estimation under targeted attacks. In *Proceedings of the Symposium on Applied Computing*, pages 666–672. ACM, 2013.

[67] Y. Wang, Z. Xu, J. Zhang, L. Xu, H. Wang, and G. Gu. SRID: State relation based intrusion detection for false data injection attacks in SCADA. In *Proceedings of European Symposium on Research in Computer Security (ESORICS)*, pages 401–418. Springer, 2014.

[68] Pandas: Python data analysis library. http://pandas.pydata.org, November 2015.

[69] M. Zeller. Myth or reality—does the aurora vulnerability pose a risk to my generator? In *Proceedings of Conference for Protective Relay Engineers*, pages 130–136. IEEE, 2011.