Title
The more the merrier

Permalink
https://escholarship.org/uc/item/7c78v21t

ISBN
9781450375610

Authors
Giraldo, Jairo
Urbina, David
Tang, CheeYee
et al.

Publication Date
2020-09-21

DOI
10.1145/3384217.3385624

Peer reviewed

The More the Merrier: Adding Hidden Measurements to Secure Industrial Control Systems

Jairo Giraldo
Electrical Engineering Department
University of Utah
jairo.giraldo@utah.edu

David Urbina
Computer Science Department
University of Texas at Dallas
david.urbina@utdallas.edu

CheeYee Tang
Engineering Laboratory
National Institute of Standards and Technology
cheeyee.tang@nist.gov

Alvaro A. Cardenas
Computer Science and Engineering Department
University of California, Santa Cruz
alvaro.cardenas@ucsc.edu

ABSTRACT

Industrial Control Systems (ICS) collect information from a variety of sensors throughout the process, and then use that information to control some physical components. Control engineers usually have to pick which measurements they are going to use and then they purchase sensors to take these measurements. However, in most cases they only need a small subset of all possible measurements that can be used. Economic and efficiency reasons motivate engineers to use only a small number of sensors for controlling a system; however, as attacks against industrial systems continue to increase, we need to study a systematic way to add sensors to the system to identify potentially malicious attacks. We propose the addition of **hidden sensor measurements** to a system to improve its security. Hidden sensor measurements are by our definition measurements that were not considered in the original design of the system, and are not used for any operational reason. We only add them to improve the security of the system and using them in anomaly detection and mitigation. We show the addition of these new, independent, but correlated measurements to the system makes it harder for adversaries to launch false-data injection stealthy attacks and, even if they do, it is possible to limit the impact caused by those attacks. When an attack is detected, we replace the compromised sensor measurements with estimated ones from the new sensors improving the risky open-loop simulations proposed by previous work.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

ICS, Security, CPS

1 INTRODUCTION

Attacks against the integrity of cyber-physical systems are a growing concern. An attacker that falsifies the data that sensors are reporting or falsifies the actions that actuators are supposed to execute, can drive the system to unsafe states, causing potential operational, economic, and safety problems. An attacker can compromise a subset of sensors and send false information to the control system. The attacks do not even have to be because of software vulnerabilities; new *transduction attacks* [10] allow the attacker to change sensor signals without compromising any device. Sensors are transducers that translate a physical signal into an electrical one, but these sensors sometimes have couplings between the property they want to measure and the analog signal that can be manipulated by the attacker. For example, sound waves can affect accelerometers and make them report incorrect movement values [20], and radio waves can trick pacemakers into disabling pacing shocks [16].

To detect these attacks, we can use our understanding of the physical evolution of the system, to see if the measurements from sensors match our predictions. There is an active community working on this type of Physics-Based Attack Detection systems (PBAD) [11]. PBAD has been explored in water control systems [1, 13], state estimation in the power grid [9, 17], chemical processes [2, 4], autonomous vehicles [6], and a variety of other cyber-physical systems [11].

All these models assume that the sensors we use for attack-detection are the same that are already present for the control algorithm. Furthermore, attack-mitigation proposals like Cardenas et al. [4] remove the sensor under attack and estimate the missing quantity with the remaining sensor measurements (they try to operate the system with less information, given the removal of this measurement). However, they do not take into account that we can gather new measurements, usually from different stages of a cyber-physical system that are correlated with each other.

We call these new measurements **hidden sensor measurements** because they are hidden from the operation of the system

under normal conditions. Furthermore, because hidden measurements are not used in regular operations, an attacker that performs reconnaissance of the industrial network will not see them (hidden sensors may remain silent and will only start reporting values based on a request by the intrusion mitigation algorithm or by the intrusion detection system based on some indicators of compromise).

For example, in the classical Tennessee-Eastman (TE) chemical process benchmark [19] (which has been used extensively in cyber-security [2, 4, 8]), the "separator cooling water outlet temperature" is not measured nor used for any purpose; however we found that this variable is highly correlated with the "product separator temperature," a measurement that is critical for safe control of the system. If an attacker falsifies or takes down the "product separator temperature," we can use the "separator cooling water outlet temperature" (with an appropriate estimation algorithm) to derive a good estimate of the attacked-sensor. Similarly the "pressure in the stripper" is a measurement that is not used at all in any control loop; however this measurement is highly correlated with the "pressure in the reactor," and therefore we can use the pressure in the stripper for security purposes.

Our contributions include:

- We introduce the concept of **Hidden Measurements**; i.e., new measurements that are not used during the normal operation of the system, but only for security purposes.
- Using new hidden sensors, we propose an anomaly detection architecture that uses the correlations between *operational* sensor measurements and *hidden* sensor measurements. Our formulation is generic and can be applied in a wide number of cyber-physical systems.
- We introduce a mitigation strategy that uses hidden sensors to respond to an attack. In particular, when an attack is detected, we generate approximate sensor signals based on redundant sensors using autoregression models. For instance, in the TE benchmark, if the reactor pressure value is compromised, we can use our added stripper pressure sensor or added separator pressure sensor (which are not used in any control loop [19]) to estimate the reactor pressure value.
- We implement our attacks and defenses in a Hardware-in-the-Loop testbed that uses a TE process simulation and is controlled with an industrial Programmable Logic Controller (PLC). In particular, we show how to launch man-in-the-middle attacks against an Open Platform Communications (OPC) server to coordinate the communication between the central controller and the field devices. The attack is able to intercept Ethernet/IP packets and falsify sensor/actuator information.

2 PROBLEM FORMULATION

In this section, we present a general mathematical model of an industrial control system (ICS) with an anomaly detection scheme. The dynamics of an ICS can be modeled using differential equations as

$$\dot{\boldsymbol{x}}(t) = F(\boldsymbol{x}(t), \boldsymbol{u}(t)),$$
$$\boldsymbol{y}(t) = H(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ corresponds to the vector with the states of the process (e.g., temperature, pressure, and water level in a chemical reaction), $\boldsymbol{u}(t) \in \mathbb{R}^m$ describes the control commands, and $\boldsymbol{y}(t) \in \mathbb{R}^p$ are the sensor readings. $F(\cdot)$ and $H(\cdot)$ are nonlinear functions that describe the system behavior.

Due to the complexity of some industrial processes, decentralized control has proven to be a practical control option in industrial plants. Decentralized control has many benefits, including easy implementation, maintenance, tuning, and robust behavior. For this reason, we assume that there are m decentralized control loops, and each one has been designed and tuned to guarantee specific performance conditions, as depicted in Figure 1; this architecture is usually implemented with Programmable Logic Controllers (PLCs), each of them controlling a subset of a larger infrastructure. In addition, we assume that there has been an adequate control configuration or input-output pairing, such that for each input u_i there is a suitable output y_i [15, 18]. We then define the controller as follows

$$u_i(t) = \mathcal{K}_i(y_i(t)), \tag{2}$$

where $\mathcal{K}_i(\cdot)$ is the function that takes sensor readings and generates control commands.
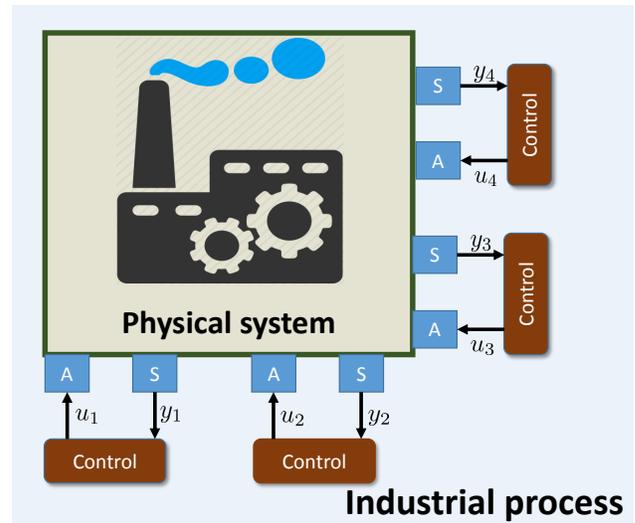
Figure 1: Decentralized control loops in an industrial control system.

Let us assume that the sensor readings and control commands can be sampled each τ seconds such that we have $y_i(k)$ and $u_i(k)$, for $k \in \mathbb{Z}_+$ the kth sampling instant. Using the historical data of the sensors and control actions, we can define an estimation function of the form

$$\hat{\boldsymbol{x}}(k+1) = \hat{F}(\boldsymbol{y}(k), \boldsymbol{y}(k-1), \dots, \boldsymbol{y}(k-T), \boldsymbol{u}(k), \boldsymbol{u}(k-1), \dots, \boldsymbol{u}(k-q))$$
$$\hat{\boldsymbol{y}}(k) = \hat{H}(\hat{\boldsymbol{x}}(k), \boldsymbol{u}(k)) \tag{3}$$

where T and q are the number of historical values we consider for the sensors and the actuators (respectively). Notice that Kalman filters, ARMA models, and neural networks can be described using Equation (3).

2.1 Detection Mechanism

Physics-based anomaly detection strategies compare current sensor readings with an estimation or prediction of the behavior of the system to detect cyber-attacks [11]. This prediction can be computed using approximated models as described in equation (3). Thus, for each sensor measurement we define the residual $r_i(k) = y_i(k) - \hat{y}_i(k)$ as the difference between the sensor reading and its corresponding prediction. An anomaly detection metric $\mathcal{D}(r(k))$ quantifies how different the historical behavior the sensor reading is from the predicted one. For instance, the χ^2-detection computes the normalized summation of all the residuals, $\mathcal{D}(r(k)) = r(k)^\top \Sigma r(k)$, for Σ the inverse of the covariance matrix. More sophisticated mechanisms accumulate the residuals in order to keep historical track of possible persistent attacks such as the CUSUM algorithm [4]. In this work, we focus our attention on the distributed bad-data detection (DBDD) mechanism defined by $\mathcal{D}_i(r_i(k)) = |r_i(k)|$. DBDD provides a detection metric for each sensor, which is useful in the context of decentralized ICS.

2.2 Attacker Model

We consider a powerful adversary that gains access to a subset of sensors and/or actuators signals. The adversary knows the detection architecture, the prediction algorithms we use, and is able to generate accurate sensor predictions. We assume first that the adversary is not aware of the hidden sensors, but we also consider the extreme case where the attacker is able to compromise hidden sensors in order to illustrate how our approach is able to limit the impact of these attacks.

3 ADDING HIDDEN SENSORS

ICS use sensor readings in order to generate control actions; however, most systems possesses physical measurements that are only used to monitor certain states but are not necessarily needed for the normal and safe operation of a system .

However, if we have a physical model of the system, we can identify variables that although not needed, might be useful for security purposes. Suppose that in our model of the system we have the following observable variables (variables for which a physical sensor can be bought and installed to measure): $\mathcal{I}^y = \{1, \ldots, p\}$. Let $\mathcal{I}^{CL} \subset \mathcal{I}^y$ be the indexes of the variables that are currently being measured and that belong to a control loop and are paired with a control input. Similarly let $\mathcal{I}^{rd} \subset \mathcal{I}^y$ be the indexes of variables that can be physically measured, but that we are not currently measuring in our physical system. Notice that $\mathcal{I}^y = \mathcal{I}^{CL} \cup \mathcal{I}^{rd}$.

According to equation (2), the control command $u_i(k)$ depends on $y_i(k)$, for all $i \in \mathcal{I}^{CL}$. As a consequence, attacks in $y_i(k)$ will affect the control command causing the system to deviate from its operation point. In order to leverage our new hidden sensors we need to find potential sensor signals that are also affected by u_i.

There are several techniques that help to quantify the input-output relationship, such as Relative Gain Array (RGA) and its variations for linear and nonlinear systems [15]; however, they depend on having a very accurate dynamic model of the system, which is difficult for nonlinear industrial control systems. We propose a simple approach that uses historical data of sensor readings and consists of calculating correlation coefficients between each

pair (y_i, y_j) for $i \in \mathcal{I}^{CL}$ and $j \in \mathcal{I}^{rd}$. The correlation coefficient determines the degree at which two variables' movements are associated. As a consequence, and because of the feedback relationship between y_i, u_i, if the pair (y_i, y_j) is highly correlated, this implies that the control action u_i affects not only y_i but also our potential hidden sensor measurement y_j.

Let $s_i \in \mathbb{R}^T$ be the signal that consists of T readings of sensor i under normal operation. The correlation coefficient is then calculated as follows

$$corr(s_i, s_j) = \frac{cov(s_i, s_j)}{\sqrt{cov(s_i, s_i)cov(s_j, s_j)}}$$

where $cov(s_i, s_j)$ denotes the covariance between s_i and s_j and correlation ranges between $-1 \leq cor(s_i, s_j) \leq 1$.

Next, we will introduce how we can use the correlation coefficient to build multi-variable anomaly detection algorithms that take advantage of hidden sensors.

3.1 Multi-Variable Anomaly Detection

Typically, centralized anomaly-detection mechanisms gather all sensor readings to construct a single prediction model in order to compute residuals and calculate a detection metric (e.g., χ^2); however, for systems with a large amount of sensors and several interconnected processes, these kind of models can be computationally expensive. Taking advantage of the decentralized nature of multiple control loops and redundant sensors, it is possible to decrease the complexity of the prediction models by constructing individual models for each control loop. Each model can be implemented locally at each control loop, and it does not only help to reduce the computational cost but also removes the single point of failure. The estimation of sensor i can be obtained according to

$$\hat{y}_i(k) = \hat{h}_i(y_i(k-1), y_i(k-2), \ldots, y_i(k-T), u_i(k-1), \ldots, u_i(k-q)).$$

The main limitation of this approach lies in the fact that an intelligent adversary can easily design stealthy attacks by only affecting a single sensor. The main idea behind multi-variable detection lies in combining some properties of both approaches, centralized and decentralized by using hidden sensors in order to limit the impact of cyber-attacks.

Our proposed Multi-Variable Detection (MVD) architecture is depicted in Figure 2 and consists on building a single prediction of sensor i based on the history of the control commands, y_i readings, and redundant sensors measurements (to ease notation we refer to redundant sensors as y_j^{rd}). The main difference with a centralized strategy is that the prediction model only depends on highly correlated sensors, instead of all sensors. As a consequence, the complexity of the model is much lower and still guarantees good accuracy.

If an adversary attacks $y_i(k)$, the controller u_i will be affected, which in turn will also affect the redundant sensors. As a consequence, the effects of the attack in y_i and in all y_j^{rd} will add up causing an error in the prediction.
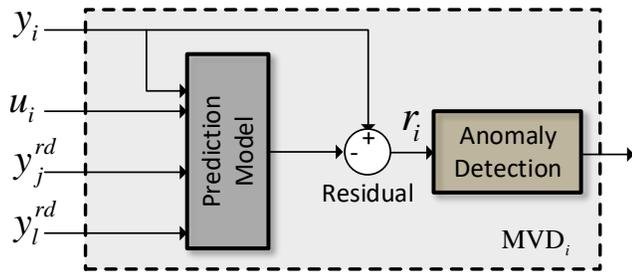
Figure 2: General architecture of the Multi-variable detection block.

3.2 System Reconfiguration for Attack Mitigation

When an attack is successfully detected, it is necessary to remove the compromised sensor readings while maintaining the system operation as close as possible to the nominal operation. In one of our earlier papers [4], we proposed to replace compromised sensor readings with estimated ones obtained from the remaining sensors. In this work, we propose a different approach that makes use of new hidden sensors. Let y_i denote sensor i and y_j^i denote one of the hidden and correlated sensors to i. Therefore, it is possible to find a mapping of $y_j^i \rightarrow \widetilde{y}_i$, where $\widetilde{y}_i$ is an approximation of the sensor reading y_i. This mapping can be computed using historical data and autoregressive models or by knowing the physical relationship between the two sensors (e.g., it is possible to compute pressure from temperature in a gas using the Gay-Lussac's Law.)

Then, when an attack associated to sensor i is detected, we replace the compromised sensor reading $y_i(k)$ with its approximation $\widetilde{y}_i(k)$ to ensure the operation of the system. The higher the correlation, the better the approximation. This is illustrated in Figure 3.
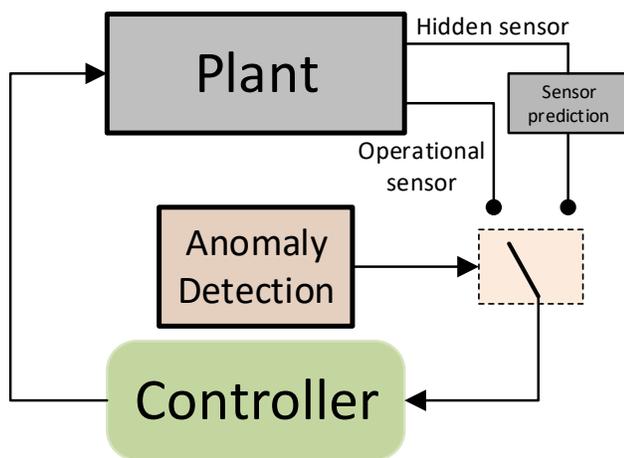
Figure 3: The system is reconfigured to use hidden sensors.

4 TESTBED

4.1 Description

The Tennessee-Eastman (TE) process was first proposed by Down and Vogel [7] and has been extensively used for the evaluation of novel control techniques, due to its complexity and large number of sensors. We were one of the first groups to use this process to study the security of industrial control systems [4, 14].

The process has five major unit operations: the reactor, the product condenser, a vapor-liquid separator, a recycle compressor, and a product stripper. The process produces two products, G and H, from four reactants A, C, D, and E. It has 41 measurements and 12 manipulated variables. The TE is open-loop unstable, which makes it very sensitive to cyber-attacks that affect the control actions.
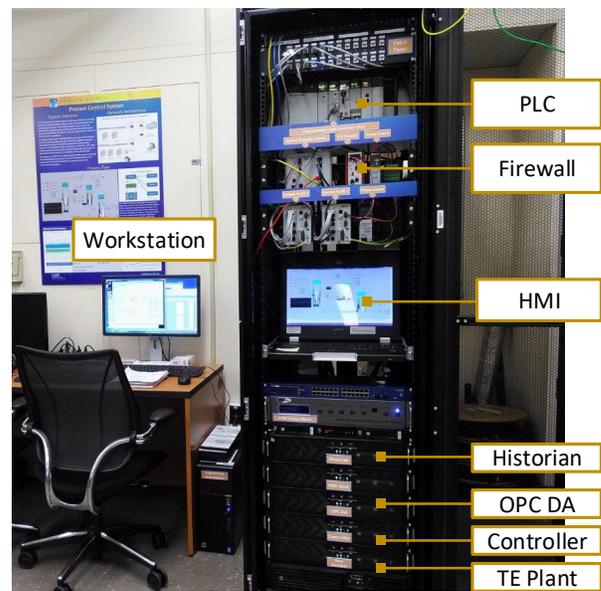
Figure 4: Tennessee-Eastman HIL Testbed.

The National Institute of Standards and Technology (NIST) Hardware-In-the-Loop (HIL) testbed for the TE process consists of 5 modules running in Microsoft Windows machines:

- simulated plant in C++,
- OLE for Process Technology (OPC) Server,
- distributed proportional-integral-derivative (PID) controller proposed in [19] where 12 control loops keep the states of the plant within operational limits while desired set-points are followed.
- Historian,
- Human-Machine Interface (HMI).

The network architecture of the testbed is illustrated in Figure 5.

The testbed also has an Allan Bradley Programmable Logic Controller (PLC). The C++ plant simulation interacts with the PLC through a Common Industrial Protocol (CIP) communication link. The PLC interacts with the OPC Server through an OPC communication link. The OPC Server communicates with the controller, the

Process Control System Network Diagram
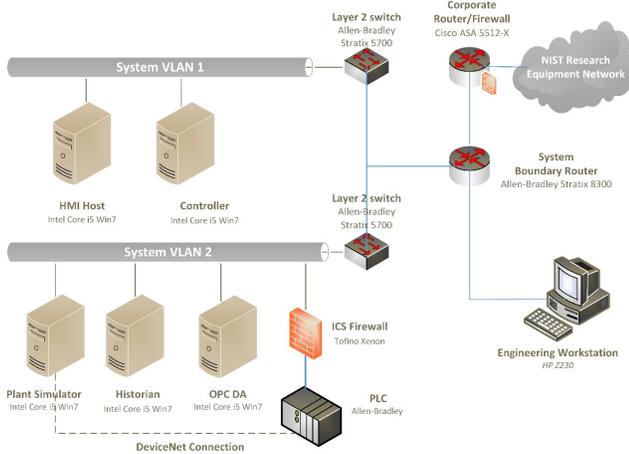
Figure 5: Network architecture of the TE testbed.
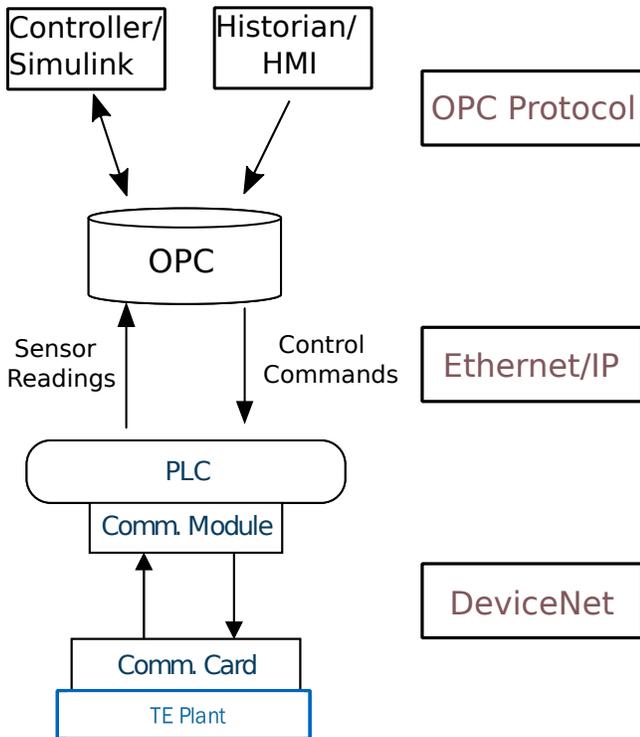
Figure 6: Logical architecture of the TE testbed.

Historian, and the HMI through the TCP/IP network. This logical interaction of components is illustrated in Figure 6.

We believe this testbed represents highly relevant aspects of an ICS. Field communications (Layer 0) are captured by the DeviceNet protocol; industrial network protocol used to connect PLCs and

workstations (Layer 1) is represented by the Ethernet/IP industrial protocol, and the widely available OPC server is used to translate among different standards and technologies. The OPC protocol is under particular interest for study as it is one of the industrial protocols that was targeted by the Industroyer malware that attacked Ukraine's power grid in 2016 [5]. OPC was also targeted by the Havex industrial espionage malware [23].

4.2 Common Industrial Protocol
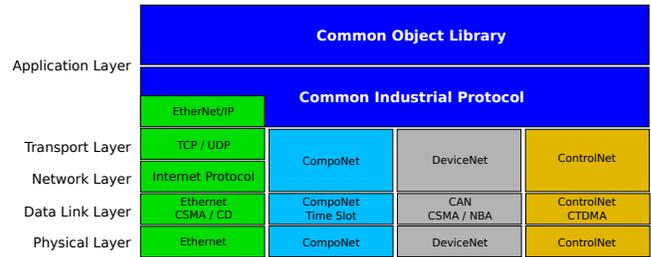
Figure 7: CIP stack and its different physical layers.

The Common Industrial Protocol (CIP) network specification library [3] was originally developed by Rockwell Automation and subsequently standardized and maintained by Open Device Vendors Association (ODVA) and ControlNet International. It aims to fulfill the main three needs of ICS systems: control, configuration, and collection of data. It defines the CIP application layer protocol as an encapsulated object-oriented protocol for transmission of connected (I/O implicit) messages between a data producer and one or more data consumer devices, and unconnected (explicit) messages between two devices in the control network. Transmissions associated with a particular connection are assigned a unique *connection ID*. While being an application layer protocol, CIP is independent of the underlying layers, and requires an encapsulation protocol, which allows abstraction from different data link and physical layers. It also includes a Common Object library defining commonly used objects, some of which are specific for a particular encapsulation protocol, and allows for extension and definition of vendor specific objects. The CIP specification library includes the definition of four different CIP stacks depending of the physical layer in use (see Figure 7): Ethernet/IP(over IEEE 802.3 Ethernet), CompoNet, DeviceNet, and ControlNet.

4.2.1 Ethernet/IP. The CIP stack introduces the Ethernet/IP protocol [3] for both, Supervisory Control and Data Acquisition (SCADA) network and fieldbus communications alike. Its specification defines the Common Packet Format (CPF) for the encapsulation of message oriented protocols, such as CIP, Modbus, and vendor proprietary messages. Ethernet/IP CPF can be stacked over UDP or TCP, in both multipoint and point-to-point connection modes. When stacking over UDP, it requires devices to select a maximum of 32 consecutive addresses from the range 239.192.1.0 to 239.192.128.255 (which belongs to the *Organizational Local Scope* [12]).
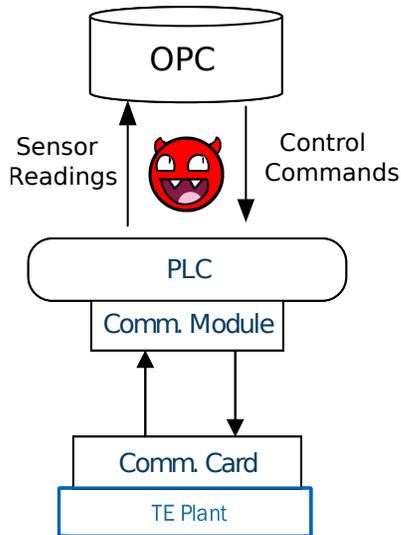
Figure 8: Man-in-the-Middle attack in the TE testbed at NIST.

Figure 9: Tags visualization and modification tool Logix5000.

4.3 False-data Injection in CIP packets

We established a Man-in-the-Middle attack (MitM) between the C++ plant simulator and the Allan Bradley PLC (Figure 8), by creating a bridge that allows us to capture all the packets coming from the sensors and the controller, modify them and send them back. This communication uses the Common Industrial Protocol (CIP) as industrial communication protocol. Although the CIP specification provides a very comprehensive library of common objects, this communication link implements a vendor-dependent extension over the protocol in order to transfer the 42 sensor measurements constantly by the plant. On the other hand, the 12 actuation commands sent from the PLC to the plant are transmitted with separate and standard CIP write-object messages containing the actuator ID number and the command value.

The CIP communication link implements a client/server mode of communication. Therefore, for the MitM to be successful in sniffing sensor measurements, the simulated controller, PLC, and OPC server must be online. In other words, the controller must request the sensor measurements for our MitM to be able to sniff the response from the plant.

We leveraged the Allan Bradley visualization tool (Logix5000 Fig. 9) installed in the workstation to program the PLC to understand at a high level the structure of the CIP extension with the 42 measurements. At first glance, it follows an array structure with every sensor measurement encoded using the Floating-Point Arithmetic Standard IEEE 754.

After developing a Scapy parser for the CIP extension, we performed initial false-data injection attacks. From the results of these attacks, we realized that some sensor measurements (such as the temperature) were being replicated in the CIP extension, and only injecting one of the instances was not enough to successfully achieve the attack, as the controller would freeze the sensor measurement to the last value before the attack started.
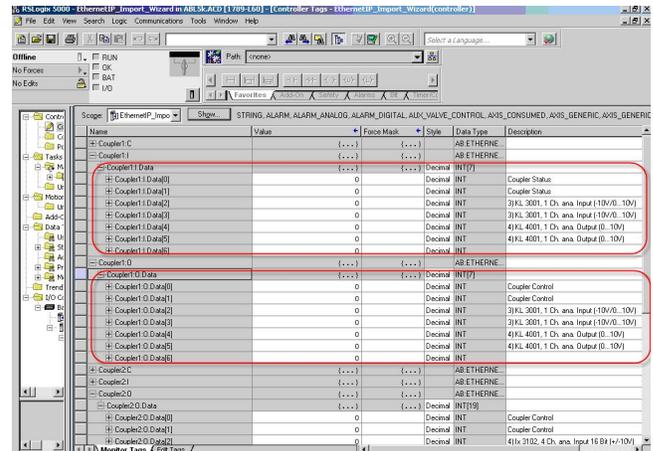
To identify the replicated sensor measurements in the CIP extension, we used a packet comparing tool similar to the process proposed by Urbina et. al. [21] to analyze packets:

(1) Using the Allan Bradley visualization tool we forced five different sensor values while sniffing a tuple (2 packets per value change) of CIP generated packets (5 tuples).
(2) We then performed diffing of packets to calculate a change-coefficient matrix representation: Any byte-change introduced by the first packet of every tuple increments its cell coefficient, while any change introduced by the second packet of the tuple decrements it.
(3) When the 5 tuples were diffed we obtained a matrix with each cell containing the coefficient of change for the corresponding byte in the CIP extension.
(4) We used the change-coefficient matrix to visualize the heatmap: the higher the coefficient the higher the heat of the cell (byte), and vice versa.

After understanding the replication on the CIP extension, we improved our parser and were able to launch false-data injection attacks on the sensor measurements.

5 EXPERIMENTAL RESULTS

Due to the correlation of the different distributed control loops, attacks that may be stealthy to one control loop might be visible (easily detectable) in other loops. Under these conditions, adversaries would need to attack all distributed loops simultaneously to remain stealthy, or decrease the impact of such attacks in a way that it does not trigger alarms in other parts of the plant.

The TE benchmark has a large number of variables where hidden sensors can be deployed. For instance, there are 11 control loops, but 42 measurements. Recall that we define two types of measurements: i) operational sensors, i.e., measurements used to generate control signals, and ii) hidden measurements—that is, measurements that provide information about the system but are not used by the controllers. The correlation among operational and hidden sensors
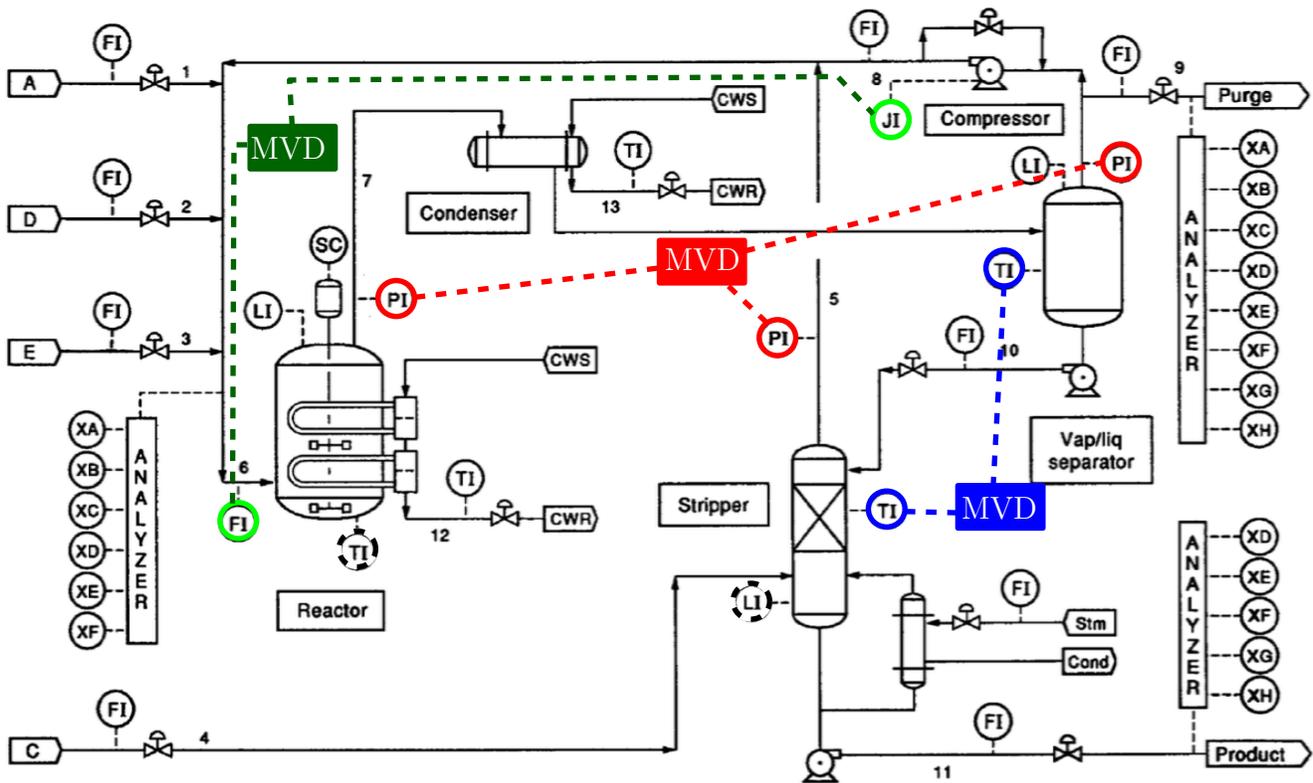
Figure 10: Tennessee Eastman Process with some Multi-variable detection (MVD). Similar colors indicate a high correlation coefficient (i.e., > 0.95). Dashed circles indicate low correlation coefficient (i.e., < 0.01). Each CD receives sensor information and the actuator signal corresponding to at least one of the sensors.

can be exploited to increase the difficulty on deploying stealthy attacks.

Figure 10 depicts the general architecture of the TE process. Using the correlation coefficient, we are able to identify the hidden sensors with the highest correlation to operational sensors.

As an example, colored circles in Fig. 10 indicate correlation and black dashed circles represent sensors without correlation. Our Multi-Variable Detection (MVD) can be located in such a way it receives information from different PLCs. The information is used to obtain detection statistics for the measurements that are not used. If an attacker wants to remain stealthy, the attacker will have to attack several sensors simultaneously from different PLCs.

Using the identification tool IDENT from Matlab, we are able to obtain individual Hammerstein-Weiner models for several control loops that estimate the input/output relationship. These models combine linear and nonlinear blocks that approximate the sensor measurement behavior for a given control input. Therefore, we are able to generate detection statistics, such as the residuals, by comparing the estimated model with the real system behavior.

As an example, let us consider the reactor pressure (called xmeas 7 in the simulation code), which is a critical parameter of the TE process and is used to control the purge rate, i.e., a valve. We construct a prediction model based on historical data from $xmeas7$ and

$xmv6$ using nonlinear ARX models. Figure 11 depicts how a simple bias attack is easily detected by the DBDD strategy. However, an attacker with enough knowledge about the system and the detection strategy is able to design optimal stealthy attacks, as it was proposed in [22], where the adversary replaces the sensor reading by $y_i^a(k) = \hat{y}_i(k) - y_i(k) + \tau_i$, for τ_i the detection threshold. Notice that the residuals under attack become $r_i(k) = |\tau_i|$, such that the detection static is never above the threshold. Now, suppose the adversary forges an attack that remains stealthy for the detector D_7. Figure 12 shows how the attack causes a shut down because the pressure reaches unsafe levels but it is never detected.

Analyzing the correlation coefficients of reactor pressure with measurements that were not being used, we found that the reactor pressure is highly correlated with the product separator pressure (xmeas 13). Therefore, we constructed a prediction model for the separator pressure using $xmv6$ and $xmeas13$ as inputs and we define a multi-variable block. Figure 12 illustrates how an stealthy attack in the reactor pressure is rapidly detected by D_{7-13} due to the high correlation between the signals. Similarly, other blocks can be constructed by using the stripper pressure sensor and other correlated measurements. As a consequence, an adversary will have to compromise all the correlated sensors in order to remain completely stealthy. Even if an attacker gains access to both sensors,
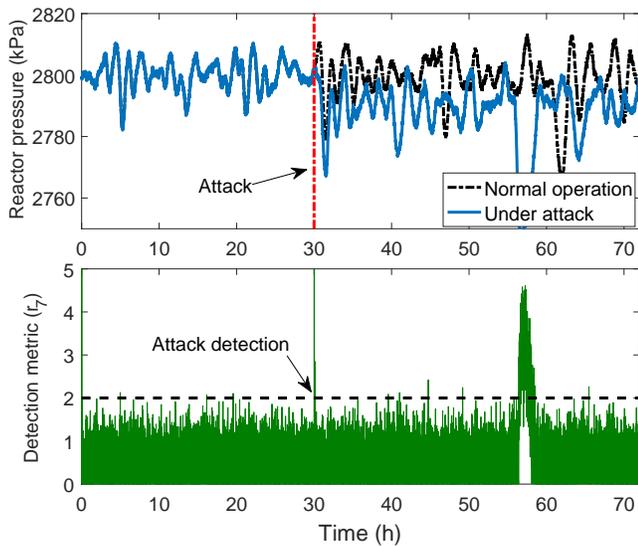
Figure 11: Reactor pressure and detection metric when an bias attack of the form $y_7^a = y_7 + 10$ is launched after 30 h. Notice that the sudden changes make the detection metric to grow rapidly and detect the attack.

the stealthy attack will not have a damaging impact in the system as depicted in Figure 13. Clearly, adding sensors is able to limit the impact of powerful attackers.

As an additional measure of security, we implemented the proposed mitigation strategy such that if an attack is detected, the compromised sensor reading is replaced by an estimation obtained from one of its correlated sensors. In our case study, the reactor pressure is highly correlated with the stripper pressure. Therefore, using an autoregressive model we can estimate the reactor pressure from the stripper pressure. Figure 14 illustrates how our proposed mitigation strategy is able to ensure the stable operation of the system even in the presence of an attack.

5.1 Comparing Detection Architectures

In order to compare how different detection mechanisms perform depending on the amount of redundant sensors included and on the type of architecture, we use the **evaluation metric for the effectiveness of physics-based anomaly detection** introduced in [22]. This metric takes into account the usability and security factors by analyzing 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).

Y-axis (Security). 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). Typically, the false alarm rate is used to measure the usability of a detection mechanism. However, it has been
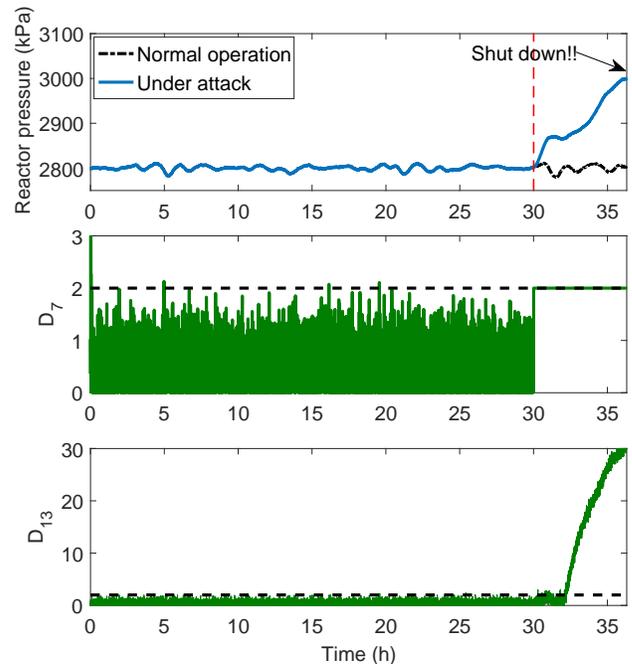
Figure 12: MVD when an optimal stealthy attack is launched only in sensor xmeas 7 (y_7). The attack remains stealthy for the detection D_7 even though the reactor pressure grows until it reaches unsafe states. However, due to the high correlation between y_7 and y_{13} (i.e., $corr(s_7, s_{13}) = 0.96$), D_{7-13} is able to detect the attack.

shown that using the expected time between false alarms $E[T_{fa}]$ offers several advantages. The usability metric can be computed by counting the number of false alarms nF_A for an experiment with a duration T_E under normal operation (without attack) for several τ. Then, for each τ we calculate the estimated time for a false alarm by $E[Tfa] = T_E/nF_A$.

As a consequence, small τ will cause small $E[Tfa]$, but it will limit the impact that an adversary can cause in order to remain stealthy.

We launched stealthy attacks for different detection strategies that involve single and multiple sensors. Figure 15 depicts the reactor pressure increase for each stealthy attack. Without attack, the reactor pressure is 2800 kPa. If it reaches 3000 kPa, the plant is shut down. Using only one sensor allows the adversary to drive the reactor pressure to dangerous values; however, using MVD with highly correlated sensors limit significantly what the attacker can do. On the other hand, when a MVD is constructed with a non-correlated sensor (sensor 19), it cannot prevent the shut down of the plant.

6 CONCLUSIONS

In this paper we have introduced the concept of hidden sensors: the idea of deploying new sensors to measure variables that are not being used by the process, in order to use them to improve the security of cyber-physical systems. We showed how these new measurements can be identified (correlation analysis) and then
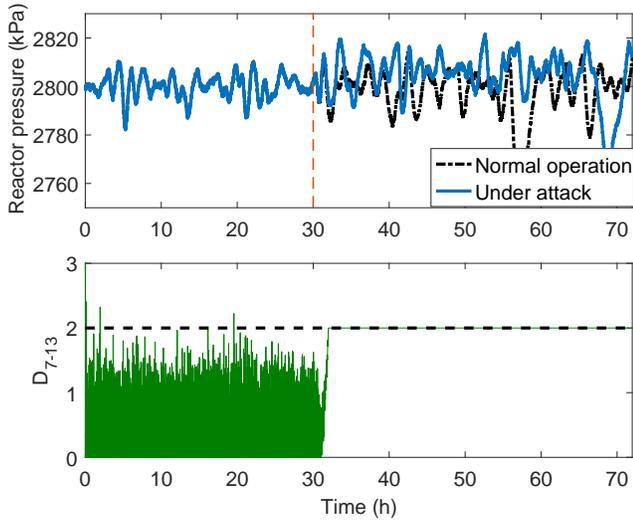
Figure 13: MVD when an optimal stealthy attack is launched in sensor y_7 with the detection D_{7-13}. Even though the attack is stealthy it causes a small deviation from the nominal operation. The attack might only be detected if we consider the correlation with other control loops.
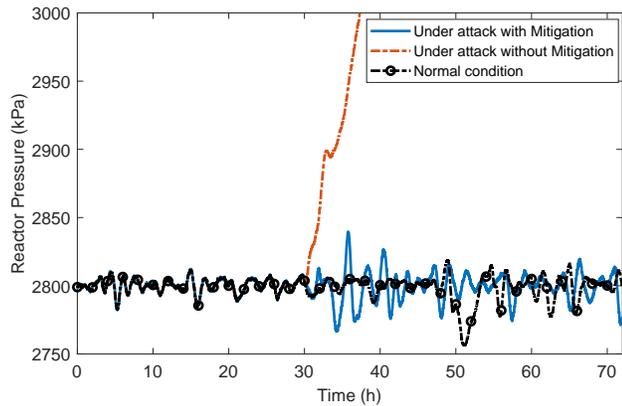
Figure 14: Reactor pressure for 3 different cases. Notice that with the proposed mitigation strategy, the system continues operating close to nominal conditions.

deployed for better attack-detection (MVD) in Figure 3 and better attack mitigation (by replacing the attacked-sensor with the new hidden sensor estimates).

We implemented our attacks and defenses in a realistic Industrial Control testbed controlling the TE process with classical industrial technologies and industrial network protocols representative of protocols that have been attacked in the real-world [5, 23].

Our results show that attacks that are not detected by previous proposals can be detected by our new hidden variables, as shown at the bottom of Figure 12. In addition we show that if the attacker becomes aware of our hidden variables, and tries to launch an attack that bypasses our hidden-variable anomaly detection algorithm,
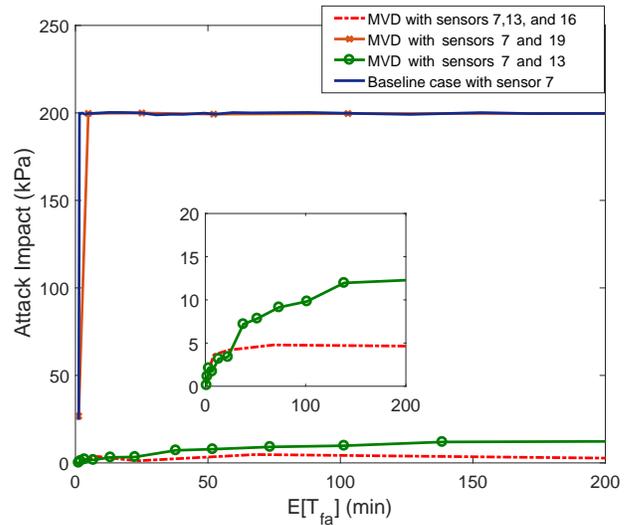
Figure 15: Comparison between different detection strategies. Sensors 13 and 16 are highly correlated with sensor 7, but sensor 19 is not. Notice that including correlated sensors in the prediction models significantly decreases the impact of stealthy attacks. On the other hand, a stealthy attack for the baseline case and by including sensor 19 is able to drive the system to dangerous pressure levels causing the plant to shut down.

then it will not succeed in driving the plant to an unsafe state, as illustrated in Figure 13. Finally, our attack-mitigation strategy (replacing the compromised sensor value with an estimate given by the hidden sensor) is also able to keep the system safe under attack, as illustrated in Figure 14.

7 ACKNOWLEDGEMENTS

REFERENCES

[1] Chuadhry Mujeeb Ahmed, Carlos Murguia, and Justin Ruths. 2017. Model-based attack detection scheme for smart water distribution networks. In *Asia Conference on Computer and Communications Security (AsiaCCS)*. ACM, 101–113.

[2] Wissam Aoudi, Mikel Iturbe, and Magnus Almgren. 2018. Truth Will Out: Departure-Based Process-Level Detection of Stealthy Attacks on Control Systems. In *Conference on Computer and Communications Security (CCS)*. ACM, 817–831.

[3] Open DeviceNet Vendors Association et al. 2013. The CIP networks Library, Volume 5, CIP Safety.

[4] 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 *Asia Conference on Computer and Communications Security (AsiaCCS)*. 355–366.

[5] Anton Cherepanov. 2017. WIN32/INDUSTROYER: A new threat for industrial control systems. *White paper, ESET (June 2017)* (2017).

[6] Hongjun Choi, Wen-Chuan Lee, Yousra Aafer, Fan Fei, Zhan Tu, Xiangyu Zhang, Dongyan Xu, and Xinyan Xinyan. 2018. Detecting Attacks Against Robotic Vehicles: A Control Invariant Approach. In *Conference on Computer and Communications Security (CCS)*. ACM, 801–816.

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

[8] Helen Durand. 2018. A nonlinear systems framework for cyberattack prevention for chemical process control systems. *Mathematics* 6, 9 (2018), 169.

[9] Sriharsha Etigowni, Dave Jing Tian, Grant Hernandez, Saman Zonouz, and Kevin Butler. 2016. CPAC: securing critical infrastructure with cyber-physical access control. In *Annual Computer Security Applications Conference (ACSAC)*. ACM, 139–152.

[10] Kevin Fu and Wenyuan Xu. 2018. Risks of trusting the physics of sensors. *Commun. ACM* 61, 2 (2018), 20–23.

[11] 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 Computing Surveys (CSUR)* 51, 4 (2018), 76.

[12] IETF Network Working Group. 2015. *Administratively Scoped IP Multicast.* http://tools.ietf.org/html/rfc2365.

[13] Dina Hadžiosmanović, Robin Sommer, Emmanuele Zambon, and Pieter H Hartel. 2014. Through the eye of the PLC: semantic security monitoring for industrial

processes. In *Annual Computer Security Applications Conference (ACSAC)*. ACM, 126–135.

[14] Yu-Lun Huang, Alvaro A Cárdenas, Saurabh Amin, Zong-Syun Lin, Hsin-Yi Tsai, and Shankar Sastry. 2009. Understanding the physical and economic consequences of attacks on control systems. *International Journal of Critical Infrastructure Protection* 2, 3 (2009), 73–83.

[15] Ali Khaki-Sedigh and Bijan Moaveni. 2009. *Control Configuration Selection of Nonlinear Multivariable Plants.* Springer Berlin Heidelberg, Berlin, Heidelberg, 139–172.

[16] Denis Foo Kune, John Backes, Shane S Clark, Daniel Kramer, Matthew Reynolds, Kevin Fu, Yongdae Kim, and Wenyuan Xu. 2013. Ghost talk: Mitigating EMI signal injection attacks against analog sensors. In *Symposium on Security and Privacy (S&P)*. IEEE, 145–159.

[17] Yao Liu, Peng Ning, and Michael K Reiter. 2009. False data injection attacks against state estimation in electric power grids. In *Conference on Computer and Communications Security (CCS)*. ACM, 21–32.

[18] Bijan Moaveni and Ali Khaki-Sedigh. 2007. Input-output pairing for nonlinear multivariable systems. *Journal of applied sciences* 7, 22 (2007), 3492–3498.

[19] N Lawrence Ricker. 1996. Decentralized control of the Tennessee Eastman challenge process. *Journal of Process Control* 6, 4 (1996), 205–221.

[20] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. 2017. WALNUT: Waging doubt on the integrity of MEMS accelerometers with acoustic injection attacks. In *European Symposium on Security and Privacy (EuroS&P)*. IEEE, 3–18.

[21] David Urbina, Yufei Gu, Juan Caballero, and Zhiqiang Lin. 2014. SigPath: A memory graph based approach for program data introspection and modification. In *European Symposium on Research in Computer Security*. Springer, 237–256.

[22] 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 *Conference on Computer and Communications Security (CCS)*. ACM, 1092–1105.

[23] Kyle Wilhoit. 2014. Havex, it is down with OPC. *Threat Research FireEye Inc. Retrieved July* 11 (2014), 2015.