

# NoiSense: Detecting Data Integrity Attacks on Sensor Measurements using Hardware based Fingerprints

Chuadhry Mujeeb Ahmed, Aditya Mathur, and Martin Ochoa,  
Address: Singapore University of Technology and Design  
E-mail: chuadhry@mymail.sutd.edu.sg

**Abstract**—In recent years fingerprinting of various physical and logical devices has been proposed with the goal of uniquely identifying users or devices of mainstream IT systems such as PCs, Laptops and smart phones. On the other hand, the application of such techniques in Cyber-Physical Systems (CPS) is less explored due to various reasons, such as difficulty of direct access to critical systems and the cost involved in faithfully reproducing realistic scenarios. In this work we evaluate the feasibility of using fingerprinting techniques in the context of realistic Industrial Control Systems related to water treatment and distribution. Based on experiments conducted with 44 sensors of six different types, it is shown that noise patterns due to microscopic imperfections in hardware manufacturing can be used to uniquely identify sensors in a CPS with up to 97% accuracy. The proposed technique can be used in to detect physical attacks, such as the replacement of legitimate sensors by faulty or manipulated sensors. We also show that, unexpectedly, sensor fingerprinting can effectively detect advanced physical attacks such as analog sensor spoofing due to variations in received energy at the transducer of an active sensor. Also, it can be leveraged to construct a novel challenge-response protocol that exposes cyber-attacks.

## 1. Introduction

A Cyber Physical System (CPS) is a distributed computing system that is sensing and actuating on the physical world [1], [2]. For instance large critical infrastructures, such as power generation and water treatment plants, are particular examples of CPS, also known as Industrial Control Systems (ICS) [3]. An ICS consists of cyber components such as Programmable Logic Controllers (PLCs), sensors, actuators, Supervisory Control and Data Acquisition (SCADA) workstation, and Human Machine Interface (HMI) elements interconnected via a communications network. The PLCs and the SCADA workstation operate in concert to control the physical process. The widespread use of such technology, and the increasing number of incidents involving it [4], [5], [6], has raised concern on the security of CPS [7].

Different from an attack against conventional IT systems, an attack on a CPS may directly result in physical damage to property (for instance due to an explosion [8], [9]) and the loss of human life (such as operators of the CPS or people depending on a critical infrastructure). Security

requirements of a CPS are thus different from those found in conventional cyber security [7], in particular the *physical* integrity of the system and its availability are often more important than confidentiality aspects [10]. Moreover, in a CPS an attacker may not only compromise the computing elements but he might partially also do so for physical components or the physical environment. This is illustrated for instance by the recent attack of [11], where a crash is induced in a drone by means of a sound signal that confuses the gyroscope, or by recent analog sensor spoofing attacks [12], [13], [14], [15]. This makes CPS security challenging, since we need to expand traditional attacker models to include physical and cyber-physical characteristics of a system [16], and consequently there is a need for novel security solutions in the intersection of these worlds.

In a CPS thus, the *veracity* [10] of sensor data is paramount to the security of the system [17]. The impact of various kinds of integrity attacks on sensor values has been studied mathematically in the control theory community, including false data injection [18], replay attacks [19], and stealthy attacks [20]. To implement such attacks in practice, attackers can spoof sensor values either by physical means, as discussed above, for instance by analog spoofing attacks [12], [13], [15], or by tampering with the communication channel between a sensor and a controller by means of a classical man-in-the-middle attack [21].

Digital or cyber-attacks can be performed by attackers who have compromised the CPS communication network and can spoof digitized sensor readings. For instance, reverse engineering of the well-known Stuxnet worm [6], [22] showed that it would attempt to perform a replay attack of normal states of a CPS while carrying out an attack to conceal it from human operators. Physical attacks can be carried out by attackers who have physical access to a CPS, such as malicious insiders [5], [23], [24], [25], or attackers that have access to a CPS distributed over a large geographical area [25]. For instance, water distribution networks or smart grids are usually spread over hundreds of miles across city, and thus it is hard to guard all the components [26], [27]. In [24] ease of physically tampering energy meters at the consumer end, without leaving any evidence, is shown. Authors in [11], [15] report sensor spoofing attacks by considering physical proximity of a few centimeters to the victim. Therefore, physical attacks are also a realistic threat model for a CPS [12], [14], [15], [23],

[26], [27].

*NoiSense*: In this work we propose a non-intrusive sensor fingerprinting method to authenticate sensors transmitting measurements to one or more PLCs. Device fingerprinting ideas based on clock skews, modulation schemes and transmission circuitry, have been reported in the literature [28], [29], [30], [31]. However, sensors in an ICS are not computationally powerful enough to exhibit the above mentioned fingerprints [26]. Thus, we seek an answer to the question, *Do sensors in a real world ICS have unique fingerprints?* It is known that hardware imperfections during the manufacturing process exhibit some unique physical behaviors that are useful for profiling and fingerprinting [32]. In particular, we observe that *noise* (imperfections in measurements), an otherwise undesirable feature of sensors, strongly depends on such manufacturing imperfections. These variations affect each device differently and thus are hard to control or reproduce [33], making it challenging for an attacker to imitate sensor noise pattern.

A technique, referred to as *NoiSense*, is designed to fingerprint sensors found in ICS. *NoiSense* creates a fingerprint for a sensor based on a set of time domain and frequency domain features that are extracted from the sensor noise. A machine learning algorithm is used to distinguish an individual sensor from others. In particular, a multi-class Support Vector Machine (SVM) is used to identify each sensor from a dataset, comprising of a multitude of industrial sensors. Experiments were performed on a total of 44 sensors including 23 low cost ultrasonic level sensors and 21 sensors of different types in an operational water treatment and distribution facility accessible for research [34], [35]. Sensor identification accuracy is observed to be as high as 97%, with a low of 90%. It is also shown that the proposed scheme is scalable for tens of sensors and that the sensor fingerprint is stable over time. True positive rate for sensor identification is observed to be 100% for most of the sensors and false positive rate as low as 0%. The cost of actual industrial scale sensors is typically a few thousand dollars, which led us to design experiments for low cost sensors. However, experiments performed on various type of industrial sensors make it a representative study for a general ICS framework.

*NoiSense* has certain advantages that make it suitable for deployment in an ICS: (a) It is *non-Intrusive*, as no modifications in CPS hardware are required. (b) It is a passive fingerprinting technique that identifies a sensor in an operational process without affecting its intended functionality. (c) It is a low cost solution that can be used at the design stage and also in an operational CPS without any significant additional cost. (d) It does not require any functional modifications to the system or control logic, other than the addition of specific code in a controller, such as in a PLC, to detect sensor tampering. One of the strongest features of the proposed scheme is that, it is able to detect attacks originating from physical (analog) as well as cyber (digital) domains.

The major contributions of this work are thus:

- A novel sensor fingerprinting framework that is based on sensor noise, and is a function of hardware characteristics of a device.
- A detailed evaluation of the proposed fingerprinting method, for a class of invasive and non-invasive physical attacks.
- Extensive empirical performance evaluation on realistic testbeds as well as using controlled lab experiments.
- A novel *challenge-response* protocol based on sensor noise fingerprinting.

This work evaluates *NoiSense* in the context of water treatment and water distribution testbeds [34], [35]. Commonly found industrial sensors are studied, but without loss of generality, these analysis are applicable to other industrial applications.

## 2. Preliminaries

### 2.1. Threat Model

In this work, we consider specific physical and cyber attacks on sensor measurements in an ICS. First, we lay down our assumptions about the attacker, followed by justification for such assumptions. Attacker's goals, objectives and attack scenarios are also explained in detail.

**Assumptions on Attacker:** We consider a strong adversary who is able to launch cyber and/or physical attacks. In an ICS sensors, actuators and PLCs communicate with each other via communication networks. An attacker can compromise these communication links in a classic *Man-in-The-Middle (MiTM)* attack [21], [36], [37], for example, by breaking into the link between sensors and PLCs. Recent studies, demonstrate malware attacks on PLCs [38], [39]. Besides false data injection in sensor readings via cyber domain, an adversary can also physically tamper a sensor, to drive a CPS into an unstable state. Therefore, we need to authenticate sensor measurements, which are transmitted to a controller. A *malicious insider*, is an attacker with physical access to the plant and thus to its devices such as sensors. An attacker can physically replace or tamper sensors. However, such an attacker does not necessarily, need to be an *insider*, because critical infrastructures, e.g., for water and power, are generally distributed across large areas [26], [27]. An *outsider*, e.g., end user, can also carry out a physical attack on sensors such as smart energy monitors. Physical attacks, invasive and non-invasive, have also been considered as a threat model in traditional IT systems [40], [41].

**Attacker's Goals ( $\mathcal{G}$ ):** An attacker may choose his goals from a set of intentions [42] such as performance degradation, disturbing a system property, and damaging a component.

**Physical Damage ( $\mathcal{G}_1$ ):** An attacker can damage devices in a plant including pumping stations and other electrical appliances. Doing so may cause injury to individuals in the plant or at a large scale by altering the devices for instance

by flooding the surroundings of the plant with wastewater as in the Marooch Shire incident [5].

**Reduction in Quality/Quantity of Product ( $\mathcal{G}2$ ):** An attacker can inject faults and defects in the product for a general industrial control system. In particular, for water networks, it can under-dose or over-dose certain chemicals to compromise the water quality. It can also reduce the production, resulting in water supply outages for the consumers.

**Utility Theft ( $\mathcal{G}3$ ):** Utility theft is usually done by the end consumer. It can be achieved by tampering different kinds of monitors/sensors [24]. One example is water theft from canals/irrigation systems [43], [44].

**Attacker's Intermediary Objectives( $\mathcal{O}$ ):**

**Concealment ( $\mathcal{O}1$ ):** An attacker wants to conceal the attack as much as possible [12]. For instance, in certain ICS [34], it has been observed that disconnecting the sensor from the controller or cutting the sensor wire to the controller, does not raise any alarm by default, making such an attack undetectable by default. In case of a stealthy cyber attack, an attacker manipulates the sensor measurements according to a precise mathematical model [20], [37], [45] to conceal it's actions.

**Inaccurate Measurements and Signal Masking ( $\mathcal{O}2$ ):** Physically replaced malicious sensors should report an inaccurate and imprecise data to disturb the product quality. An attacker can achieve the same objective of inaccurate measurements by false data injection. Also, an adversary may prevent the sensor to measure the true physical quantity. This can be achieved by data injection in SCADA workstation [21], [36] and also by masking the true quantity by overpowering the original signal [12], [14].

**Deceiving Controller ( $\mathcal{O}3$ ):** The attacker attempts to deceive the controller into believing that the data received is from a legitimate sensor.

**Attacker Characterization( $\mathcal{C}$ ):**

**Attacker Profile ( $\mathcal{C}1$ ):** Most likely attacker profile requiring physical access is that of a *malicious insider* [46]. However, for attacker goal  $\mathcal{G}3$  (utility theft), a consumer may be a malicious entity. Another strong candidate for such an attacker is *casual outsider* (malicious contractor or technician) as defined in [23]. Considering the critical nature of these utility infrastructures a nation state or a terrorist attacker profile could not be ignored [46]. An *outsider* can break into the system using cyber domain and may not necessarily need physical access to the plant.

**Technical Capabilities ( $\mathcal{C}2$ ):** We assume an adversary, (*insider or outsider*), has the complete knowledge of the workings of the plant and sensing devices in particular. An end user could employ services of a contractor, a criminal entity, for sensor tampering, to achieve the utility theft goal(  $\mathcal{G}3$ ).

**Attack Scenarios ( $\mathcal{S}$ ):** We categorize attack scenarios into two categories, namely physical domain and cyber domain.

**Physical (Analog Domain) Attacks:**

**Sensor Replacement Attack ( $\mathcal{S}1$ ):** An attacker replaces one or more sensors in the plant by new, perhaps malicious, sensors. This way the attacker can manipulate the sensor readings to drive the system to an undesirable state. The attacker can inject arbitrary sensor data that may not be detected by cyber attack detection schemes based on statistical methods such as Cumulative Sum (CUSUM) [37].

**Sensor Swap Attack ( $\mathcal{S}2$ ):** Sensor swap attack uses the legitimate control logic to achieve an attacker's goal. Rather than modifying the control logic, an attacker feeds a controller with measurements from another process. The idea is to use the existing control logic to drive the plant to an insecure state. In appendix A, an example swap attack and it's consequences on a water treatment testbed are analyzed.

**Sensor Saturation Attack( $\mathcal{S}3$ ):** Sensor saturation attack is similar to jamming attacks and is executed by injecting power to the sensor's receiver [47]. A constant value is measured by a sensor under attack and it is not able to monitor the real physical quantity. Another way to achieve a similar attack in industrial sensors is to block the physical medium (either by wireless channel jamming or cutting the wired media) between the sensor and remote  $I\emptyset$  unit. This attack also returns a constant reading to the SCADA workstation. An attacker can remain stealthy for such attacks as system raises no alarm. However, it is obvious that transmitted constant reading does not contain sensor noise component which enables *NoiSense*, to detect such attacks or faults in components.

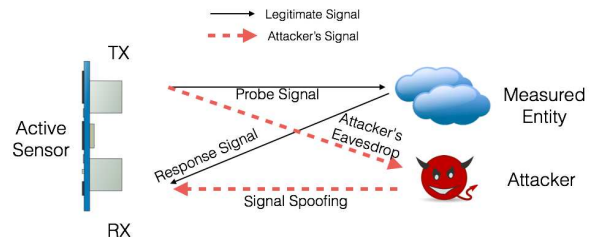


Figure 1. Analog Sensor Spoofing Attack.

**Analog Sensor Spoofing Attack( $\mathcal{S}4$ ):** The attack scenarios ( $\mathcal{S}1, \mathcal{S}2, \mathcal{S}3$ ), mentioned above, need physical access to the plant and are also invasive requiring sensor tampering. Another type of physical attack can be non-invasive where an attacker spoofs sensor measurements by disturbing the surrounding environment. This can be achieved by bringing another malicious entity (same type of device as the victim) in proximity with the victim device. One such attack is analog spoofing of the measured quantity [12] for a class of active sensors. Active sensors transmit a probe signal which interacts with the quantity to be measured and a response signal is received and analyzed to measure the

physical quantity. But if an attacker has physical access to the sensing environment, it can change the response signal before it reaches the receiver. An example of such an attacker is shown figure 1. Transmitter (TX) of the active sensor transmits a probe signal and waits for the response signal to be received. Rather than getting the legitimate response signal, an attacker transmits a fake signal. When this fake signal is received at the receiver (RX) of the active sensor (victim), it would not be able to differentiate between a legitimate and the signal generated by an attacker.

#### Network-based (Cyber Domain) Attacks:

*Digital Domain Sensor Swap Attack(S5)*: This attack is similar in concept to  $S2$  but an attacker does not necessarily need physical access to sensors. In control logic of the plant, it is possible to exchange the tags for the sensors for their respective PLCs [38].  $PLC_1$  can be made to read sensor 2, and  $PLC_2$  can be made to read sensor 1. The effects of such a swap would be same as in the case of physical sensor swap. However, such a change will not be reflected at the SCADA workstation where human operators are monitoring the process

*False Data Injection in Sensor Measurements(S6)*: This attack can be executed as *MiTM* whereby an attacker modifies the unencrypted sensor data transmitted to a PLC [21], [36]. Such a modification will change the noise fingerprint of a sensor and enable *NoiSense* to detect the attack.

*Stealthy Attacks(S7)*: An attacker modifies a sensor measurement and attempts to hide it's presence. In the literature [37], [45], [48], [49], [50] specific stealthy attacks have been designed to stay undetected for a range of statistical detectors. To achieve the *stealthiness* an attacker must modify sensor measurement in a way so that it can maximize the damage and remain undetected. To achieve this objective an attacker will inevitably change the sensor noise which will enable *NoiSense*, to detect the presence of the attacker.

*Replay and Advanced Sensor Spoofing Attacks(S8)*: In a replay attack an attacker records the system states for the normal operation of the system. Then it replays recorded states during an attack to hide itself from operators and digital domain intrusion detection systems [19], [50]. An example of such an attack is *Stuxnet* [6]. A replay attack on sensor measurements might not be detected by *NoiSense* because sensor noise will also be replayed. Another example is of a very powerful cyber attacker with an ability to learn noise fingerprint of a sensor. It can modify sensor measurements to arbitrary values and also add noise patterns for that sensor, making it strong enough to remain undetected by *NoiSense*. We extend the idea of *NoiSense* for the case of such a powerful cyber attacker by proposing a novel *challenge-response* scheme presented in section 5.

## 2.2. Sensing Technologies

In this section we explain the basic working principle of the sensing technologies under study. This insight in sensor



Figure 2. Sensors used in experiments.

construction and functionality is an aid in understanding the sources of sensor noise and fingerprints.

**Ultrasonic Level Sensors:** Water treatment testbeds use ultrasonic sensors based on a piezoelectric (PZT ceramic) material transducer. The level of water in a tank is calculated by measuring the return time of the acoustic wave after hitting the water surface. Several factors contribute to variations in the measurements obtained from ultrasonic sensors. These measurements depend on the speed of sound which changes according to the surrounding temperature [51]. Besides temperature, obstacles like tank walls reflect echo sooner than expected, contributing towards noise in the measurements. The acoustic impedance of the PZT transducers also depends on temperature thus adding another source of noise [52]. Thermal and polarization noise are the main sources of voltage fluctuation in piezoelectric ceramics [53].

**Microwave Level Sensors:** The microwave level/distance sensor, often called RADAR (Radio Distance and Ranging), works in a similar way as ultrasonic sensors. A microwave pulse is emitted by the antenna that travels at the speed of light and upon hitting the surface of the target it is reflected back and received at the same antenna. These antennae are designed to have a  $50\Omega$  resistance so that once connected with a cable of characteristic impedance of  $50\Omega$ , maximum power transfer takes place from the antenna. The sensor under consideration is designed to operate at 26 GHz with a beam angle of  $22^\circ$  and  $1\mu W$  effective radiated power [54]. However, in practice these specifications have deviation for the same type and design of an antenna due to manufacturing imperfections and installation inaccuracies. For example, antenna connection with a cable will result in impedance variations [55]. Also, beam angle and radiation pattern varies for each antenna leading to deviations from theoretical design resulting in different range resolution that is ultimately reflected in sensor noise [56].

**Electromagnetic Flow Meters:** The electromagnetic flow meters follow Faraday's law of induction according to which a voltage is induced by an electrically conductive fluid passing through a magnetic field. In an electromagnetic flow meter, the medium acts as the electrical conductor when flowing through the flow meter tube, and the induced voltage is proportional to the average flow velocity (the faster the flow rate, the higher the voltage). A commercial electromagnetic flow meter is shown in Figure 2 [57]. It's internal structure consists of a pair of coils mounted on the top and bottom of an electrically insulated flow tube. A pair of electrodes protrude through the flow tube wall perpendicular to the pipe axes and largely normal to the direction of the generated magnetic field. Noise in these sensor readings come from the area of the electrodes and size of the electro-magnets generating electromagnetic field

B. The installation and alignment of electrodes and coils will result in different stray capacitance and noise [58].

### 3. Design of *NoiSense*

Figure 3 shows the steps involved in composing a sensor fingerprint. The proposed scheme begins with data collection and then divides data into smaller chunks to extract a set of time domain and frequency domain features. Features are combined and labeled with a sensor ID. A machine learning algorithm is used for sensor classification.

#### 3.1. Data Collection

Data is collected for different types of industrial sensors listed in Table 2. We collect data for the level sensors when the process is static, i.e. tank levels are constant. For flow meters, data is collected when process is dynamic, i.e. water is flowing through the pipes and hence a non-zero flow rate is observed. The objective of data collection step is to extract sensor noise. For the case of level sensors, when the process is running, an error in sensor reading is a combination of sensor noise and process noise (water sloshing etc.). Extracting process noise from a dynamic process is a challenging task, given that the noise parameters vary with change in the process state. Therefore, a set of experiments for level sensors, are designed to obtain sensor measurements, when a process is not active. Tanks store water to provide to subsequent stages for processing. However, these processes are not always active as water demand is not constant. If there is no inflow and outflow of water, in a tank, the corresponding level sensor measurement should be a constant. Nevertheless, there are fluctuations in the sensor measurements, as a result of sensor noise or temperature variations. In the controlled lab environment, temperature is controlled, and it affects all the sensors in a similar way.

We have a limited number of level sensors in water treatment testbed, but we diversified experiments to validate the proposed idea. A total of 3 level sensors are installed on top of 3 water tanks. Each sensor is placed on all the three water tanks, to collect data for all possible sensor-tank combinations. The data is analyzed, in time and frequency domains, to examine the noise patterns, which are found to follow Gaussian distribution. Sensors are profiled using variance and other statistical features in the noise vector. The experiment is run, to obtain sensor profile, so that it can be used for later testing. A machine learning algorithm is used to profile sensors from fresh readings (test-data). Design and testing, of *NoiSense*, is feasible in settings of a water treatment plant. According to the control logic, when a tank is filled up to a specified limit, a pumping station turns OFF and the water level stays at a constant value. Since these water treatment systems, have multiple processing stages, there are instances when there is neither an incoming flow nor an outgoing flow from a tank, i.e. we have a constant water level in the tank. During these instances, we can record the data and match it with previously generated

TABLE 1. LIST OF FEATURES USED.

Feature	Description
Mean	$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$
Std-Dev	$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$
Mean Avg. Dev	$D_{\bar{x}} = \frac{1}{N} \sum_{i=1}^N  x_i - \bar{x} $
Skewness	$\gamma = \frac{1}{N} \sum_{i=1}^N \left(\frac{x_i - \bar{x}}{\sigma}\right)^3$
Kurtosis	$\beta = \frac{1}{N} \sum_{i=1}^N \left(\frac{x_i - \bar{x}}{\sigma}\right)^4 - 3$
Spec. Std-Dev	$\sigma_s = \sqrt{\frac{\sum_{i=1}^N (y_f(i))^2 * y_m(i)}{\sum_{i=1}^N y_m(i)}}$
Spec. Centroid	$C_s = \frac{\sum_{i=1}^N (y_f(i)) * y_m(i)}{\sum_{i=1}^N y_m(i)}$
DC Component	$y_m(0)$

Vector  $x$  is time domain data from the sensor for  $N$  elements in the data chunk. Vector  $y$  is the frequency domain feature of sensor data.  $y_f$  is the vector of bin frequencies and  $y_m$  is the magnitude of the frequency coefficients.

fingerprint. Fingerprints can be generated over time or at the commissioning phase of the plant.

For flow meters, if there is no water flow, we receive a value of 0 from the sensor. In the testbeds under study, the flow of water between different stages is controlled by a pump. Therefore, water flow through a pipe should be constant but flow rate data transmitted by the electromagnetic sensor is noisy. We treat these fluctuations in the sensor data as noise and collect this information during the times when process is running. For the case of low cost ultrasonic sensors, we designed an experiment where a constant water level is to be measured. Noise in each sensor is extracted and data analyzed to generate the fingerprint. Thus, by combining the level sensor and flow meter features, we are able to monitor the process, in both static (constant water level) and dynamic (water flowing) states.

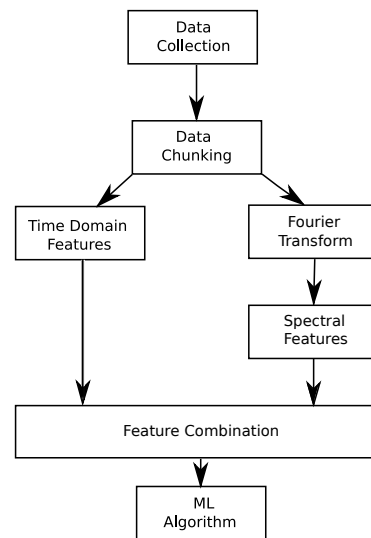


Figure 3. Components of *NoiSense*, design.

### 3.2. Feature Extraction

Data is collected from sensors at a sampling rate of one every second. Since data is collected over time, we can use raw data to extract time domain features. We used the Fast Fourier Transform (FFT) algorithm [59] to convert data to frequency domain and extract the spectral features. In total, as in Table 1, eight features are used to construct the fingerprint.

*Data Chunking:* After data collection from the sensors, the next step is to create chunks of dataset. An important question to answer is: *How many chunks of data are needed to train a well-performing machine learning model?* Also, we want to know, *how much time it will take to perform a test?* so that a decision about the attacks can be made within a specified time. To this end we create data chunks of varying size for each sensor. After dividing a sensor's data (total of  $N$  readings) into  $m$  chunks, each chunk of  $\lfloor \frac{N}{m} \rfloor$ , the feature set  $\langle F(C_i) \rangle$  for each data chunk  $i$  is calculated. For each sensor, there are  $m$  sets of features  $\langle F(C_i) \rangle_{i \in [1, m]}$ . For  $n$  sensors one can use  $n \times m$  sets of features to train the multi-class SVM. Supervised learning was used for sensor identification which has two phases—training and testing. For both the phases, chunks are created in a similar way as explained above.

*Size of Training and Testing Dataset:* An important question to address is: *How many feature sets are needed for training the classifier and how many for testing?* For a total of  $m$  feature sets for each sensor, at first half ( $\frac{m}{2}$ ) were used for training and half ( $\frac{m}{2}$ ) for testing. To analyze the accuracy of the classifier for smaller feature sets was reduced during the training phase, number of feature sets starting with  $\frac{m}{2}$ . Classification is then carried out for the following corresponding range of feature sets for Training :  $\{\frac{m}{2}, \frac{m}{3}, \frac{m}{4}, \frac{m}{5}, \frac{m}{10}\}$ , and for Testing :  $\{\frac{m}{2}, \frac{2m}{3}, \frac{3m}{4}, \frac{4m}{5}, \frac{9m}{10}\}$ , respectively. In section 4, empirical results are presented for such feature set and the one with the best performance is chosen for further analysis of the proposed scheme. For the classifier a multi-class SVM is used [60], as briefly described in Appendix B.

### 3.3. Performance Metrics

In Table 2, sensors of same type are grouped together. Each sensor is assigned a unique ID and a multi-class classification is applied to identify each sensor among those of the same type and model group. To evaluate the performance, identification accuracy is used as a performance metric. Let  $c$  be the total number of classes.  $TP_i$  as true positive for class  $c_i$  when it is rightly classified based on the ground truth. False negative  $FN_i$  is defined as the wrongly rejected, and false positive  $FP_i$  as wrongly accepted, device. True negative  $TN_i$  is the rightly rejected class. The overall accuracy ( $acc$ ) for total of  $c$  classes is defined as in the following.

$$acc = \frac{\sum_{i=1}^c TP_i + \sum_{i=1}^c TN_i}{\sum_{i=1}^c TP_i + \sum_{i=1}^c TN_i + \sum_{i=1}^c FP_i + \sum_{i=1}^c FN_i}. \quad (1)$$

TABLE 2. LIST OF SENSORS IN OUR STUDY.

Type	No. of Devices	Class
Ultrasonic level sensor	3	C1
RADAR level sensor	2	C2
HCSR-04 dual transducer 5V	15	C3
HCSR-04 dual transducer 3.3V	5	C4
Electromagnetic Flow Meter	8	C5
SRF02 single transducer Sonar	3	C6
Differential Pressure Transmitter	8	C7

The True Positive Rate (TPR) and False Positive Rate (FPR) are defined as follows,

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN}, \quad (2)$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN}. \quad (3)$$

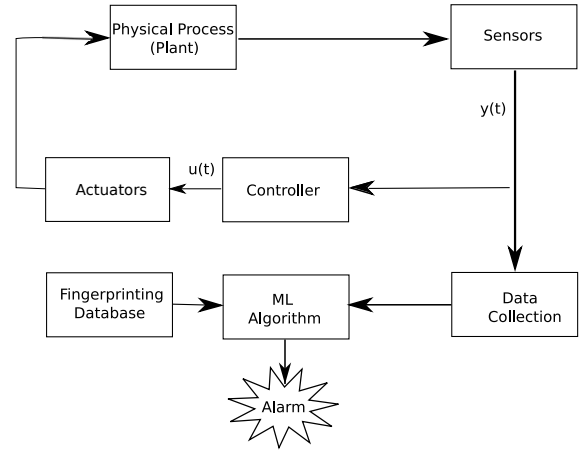


Figure 4. System Model for Proposed Methodology.

## 4. Evaluation

The underlying idea for the proposed method is to find a fingerprint from the extracted noise of sensors under consideration. Figure 4 shows the integration of the proposed scheme to any existing ICS without need of additional hardware installation. Also, the proposed method is passive and does not disrupt the functionality of the control system.

**Research Questions:** The following research questions are the focus of the remainder of this work.

- RQ1:** *Does a unique fingerprint exist for each sensor?*
- RQ2:** *How much data is needed to identify a sensor?*
- RQ3:** *How accurately can the sensors be identified?*
- RQ4:** *Is a fingerprint unique with respect to multiple sensors such as is the case in a large CPS?*
- RQ5:** *Is the sensor fingerprint stable between different runs of the experiment over a period of time?*
- RQ6:** *Is the sensor fingerprint based method able to detect analog sensor spoofing attacks?*

Sensor data is analyzed for different sets of sensors to answer the above questions. Extensive data collection and analysis is carried out to obtain the fingerprint and show that indeed this is a valid fingerprint. The following sections describe the experimentation setup as well as results obtained by applying the proposed attack detection scheme.

#### 4.1. Experimentation Setup

Experiments described in this article were conducted on an operational water distribution and water treatment systems. One testbed is a fully operational research facility and a scaled down water treatment plant. The other testbed is a water distribution network [34], [35]. Additional information on these testbeds is in the appendices C and D. Besides using the sensors in the two testbeds, 23 small ultrasonic sensors were also used to show that the proposed scheme can scale well to a larger set of sensors. Table 2 lists the sensors used in this study. These sensors are representative of those found in a range of industrial plants as fluid storage and flow is a common process [61].

**Ultrasonic Level Sensors (Water Treatment Testbed):** Experiments were performed on a portion of the six stage water treatment testbed [34]. Three ultrasonic level sensors, available in this testbed, were used in the experiments. Data was collected first from three level sensors in their original tank installation and also by mounting on the other tanks. These experiments were performed for several hours each day and over several months. An SVM model was trained over the data obtained in original positions of the sensors. For each sensor, testing data was obtained by placing the sensor over the other tanks. This experiment demonstrated: a) changing the process (tank) does not change the fingerprint, and b) fingerprint is stable and valid for long term, e.g., it is valid for data collected at different times over a period of six months.

**Flow Meters (Water Treatment Testbed):** Electromagnetic flow meters are installed in pipes to monitor the water flow. Data was collected while the plant was running. For flow meters, sensor swap is not as simple as level sensors, since these are installed in-line on a water pipe. Therefore, a 5-fold cross validation results are reported in Table 5 for the four flow meters (from each testbed) used in the study. Several measurement vectors were obtained for the water filling process for up to six days continuous run of the plant. The noise vector for each process was extracted and feature set obtained.

**RADAR Level Sensors (Water Distribution Testbed):** Two radio frequency based level sensors are available for experiments in the water distribution testbed [35]. Experiments were performed for three days and data collected. A 5-fold cross validation was performed using SVM and sensors identified with high accuracy.

**Dual Transducer Ultrasonic Sensors (HCSR04):** A limited number of industrial sensors were available to perform the experiments. Details of such sensors are given above. It is costly to acquire a lot of such sensors. Therefore, a set

of experiments was designed to test our approach on 20 low cost ultrasonic sensors. The working principle of these low cost sensors is the same as for those used in industrial scale sensors. The goal of these experiments was to explore *whether a large set of sensors can be uniquely fingerprinted and identified with high accuracy?* Among these 20 sensors, 15 are double transducer ultrasonic sensors which require 5V DC to operate, while the remaining 5 sensors require 3.3V DC to operate. This variety of operation voltages adds to the diversity in the experiments.

The setup to conduct the experiments on ultrasonic sensors is shown in Figure 10(Appendix E). It is composed of the following main components: 1) a tank is created with a small half-filled water glass, 2) breadboard to hold the ultrasonic sensor, 3) Arduino board (controller) that is connected to sensor, and 4) a server to collect and store data from sensors over WiFi. The Arduino board has a microcontroller and several input/output pins that serve as an interface for external circuitry [62]. Ultrasonic sensors can be easily mounted and removed from the breadboard. Same water level is ensured in the tank for all the sensors. Multiple rounds of experiments were performed to ensure that noise is free of the physical sensor arrangement. Sensor readings were collected over three hours with a sampling time of 1 second. Several chunks of data were extracted from these sets of readings and features extracted for each chunk. After collecting data for three hours we removed the sensor from the tank and put it back and collected data for an additional two hours. This is to demonstrate that the fingerprint can be recovered even after disturbing the sensor alignment.

**Single Transducer Sonar Sensors (SRF02):** Three small single transducer sonar range finders were also included in the experiments. The basic working principle of these sensors is the same as that of dual transducer ultrasonic sensors though with minor differences. These sensors (SRF02) use a single transducer for both transmission and reception, and the minimum range is higher than the dual transducer sensors used in the experiments. The minimum measurement range varies from around 17-18 cm. For this reason a distance measuring experiment was performed with these three sensors rather than measuring water level in small glass tanks. These experiments were controlled through the Arduino board and placed at the same place to measure distance between sensors and the ceiling. Experiments were run for 6 hours with one run of 3 hours and a second run of 3 hours to explore the stability of the sensor fingerprint.

#### 4.2. Existence of Fingerprint

**RQ1: Does a unique fingerprint exist for each sensor?** A limited number of sensors were available in the water utility testbeds. Hence, additional low cost ultrasonic sensors are included to explore the existence of fingerprints for many sensors of the same type and model. To demonstrate the existence of fingerprint, ten dual transducer ultrasonic sensors (HCSR04) from the same manufacturer were used. All ten sensors were mounted on the same water tank. Data

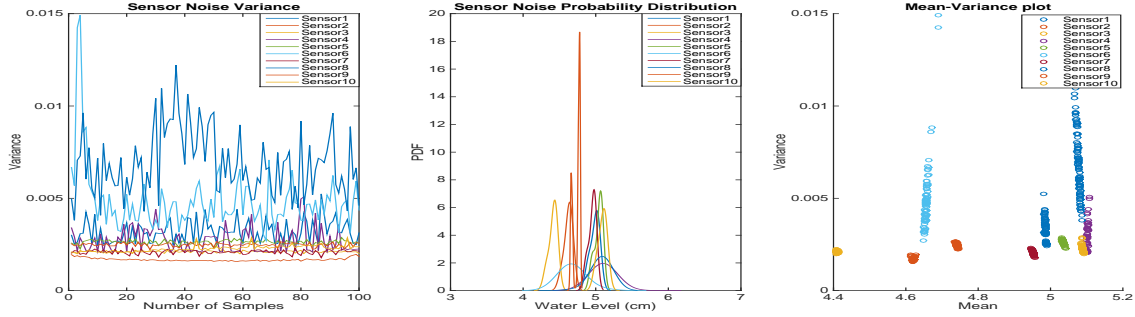


Figure 5. Noise data from ten ultrasonic sensors of same type (HC SR04). Left: Variance for each sensor noise vector is shown for different data chunks of each sensor. Middle: Distribution of each sensor noise vector is shown for a data chunk. Right: Including mean of the noise improves the separation for individual sensors.

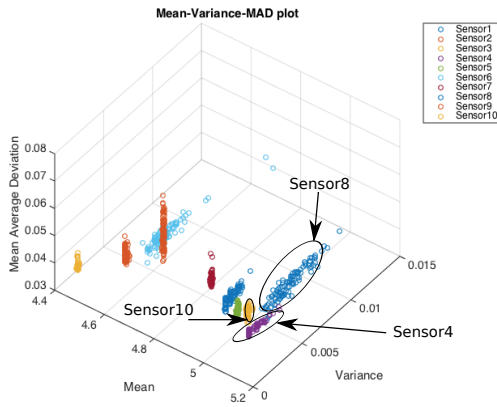


Figure 6. Three features have improved the clustering and unique identification of the sensors as compared to using only the variance or the mean. This is a 3D representation of the right-most plot in Figure 5.

was collected for 3 hours and many chunks of the collected data taken for analysis. Each chunk consists of 300 readings from the sensor. Figure 5 shows results for the collected data. The plot on the left shows the variance of noise vector from each sensor for all chunks. It is observed that some of these sensors have a unique noise variance and can be distinguished from each other but there remain few sensors that have similar noise pattern in terms of noise variance. The middle pane is a plot of the distribution of the noise vector from each sensor. It also shows that sensors can be distinguished based on noise statistics. However, there remain overlaps among some sensors. Right pane shows 2-D clustering of the sensors. Sensors can be distinguished more precisely by using one more feature of sensor's noise i.e. mean value. The scatter plot on the right hand side clusters each chunk with its respective mean and variance. The separation is quite clear but there remain overlaps, e.g., sensor4, sensor8 and sensor10. We need additional features to further eliminate such overlaps. In Figure 6, by adding one more feature, i.e. mean average deviation, sensor4, sensor8 and sensor10 can be distinguished. In the following sections we show that by using additional features it is possible to achieve high accuracy for sensor identification. Details of

feature set used are in Table 1.

### 4.3. Sensor Identification Accuracy

**RQ2:** How much data do we need to identify a sensor? We start our analysis with 20 dual transducer small ultrasonic sensors. The goal now is to find how much training data is adequate to identify sensors with high accuracy. Also, we are interested in knowing the chunk size of data to extract features. Table 4 shows results from our analysis. The first row shows the size of training and testing data set, respectively, where  $m$  is total number of chunks for a sensor. The first column in the table shows chunk size starting from 60 seconds to 1000 seconds. It is observed that too small a chunk size does the accuracy is slightly lower as compared to chunks in the middle range. A large chunk size, as for example 1000, limits the chunks and hence leads to a small a feature set. Doing so results in lower identification accuracy. If we move towards right in the row, the size of the training data set is reducing, which also results in lower accuracy. This result is intuitive. However, we select a chunk size of 120 seconds which is not too small that it could not capture sensor noise statistics and not too large that we have to wait too long before reaching a decision about authenticity of the sensor. Also, it was decided to divide the entire sensor data set in three parts and used one third for training for further experiments. For 20 low cost ultrasonic sensors, this choice of chunk size and training data set provided 96.43% accuracy for sensor identification as shown in Table 4.

For a chunk size of 120 seconds, and one third of data set for training, the SVM classifier, a multi-class SVM classification for 20 dual transducer ultrasonic sensors was carried out. It was possible to distinguish sensors with an accuracy of 96.43%. Figure 12 (Appendix D) shows the same result visually with a plot of the confusion matrix as a heat map for these 20 sensors. The horizontal axis represents actual sensor ID, while the vertical axis represents the predicted sensor ID by the SVM classifier. We note that nearly all sensors are accurately identified, and hence, on diagonal of the confusion matrix, the prediction accuracy is close to 100%.



TABLE 3. IDENTIFICATION EVALUATION FOR EACH SENSOR. **FPR**: FALSE POSITIVE RATE, **TPR**: TRUE POSITIVE RATE.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
TPR	1	0.87	1	1	1	0.97	1	1	0.99	0.98	1	0.98	0.99	0.98	0.86	0.97	1	0.94	0.97	1
FPR	$3.4e^{-3}$	$0.9e^{-3}$	0	0	0	0	0	0	0	$6.2e^{-3}$	0	$1.2e^{-3}$	$2.2e^{-3}$	$1.2e^{-3}$	$7.4e^{-3}$	$2.2e^{-3}$	0	$0.6e^{-3}$	0	0

TABLE 4. CLASSIFICATION ACCURACY FOR 20 SMALL ULTRASONIC SENSORS.  $m$  IS THE TOTAL NUMBER OF READINGS RECORDED FROM EACH SENSOR.  $\frac{m}{3}, \frac{2m}{3}$  MEANS ONE THIRD OF THE DATA IS USED FOR TRAINING AND TWO THIRD IS USED FOR TESTING RESPECTIVELY. THE FIRST COLUMN IN THE TABLE SHOWS CHUNK SIZE OF DATA USED TO EXTRACT FEATURES.

Chunk Size (s) / Data Segmentation	$\frac{m}{2}, \frac{m}{2}$	$\frac{m}{3}, \frac{2m}{3}$	$\frac{m}{4}, \frac{3m}{4}$	$\frac{m}{5}, \frac{4m}{5}$	$\frac{m}{10}, \frac{9m}{10}$
60	95.13%	95.56%	95.14%	93.56%	88.52%
<b>120</b>	96.43%	<b>96.43%</b>	95.6%	93.86%	92.61%
250	97%	95.82%	94.74%	94.37%	91.62%
500	96.72%	95.59%	95.20%	94.13%	85.09%
700	96.74%	95.66%	91.51%	90.55%	77.08%
1000	96.88%	93.25%	90.62%	83.75%	69.25%

TABLE 5. OVERALL RESULT

Sensor	Type and Model	Number of Sensors	Identification Accuracy
Ultrasonic Level Sensor	iSOLV LevelWizard II	3	90%
Electromagnetic Flowmeter (SWaT)	iSOLV EFS803/CFT183	4	96%
Dual Transducer Ultrasonic Level Sensor	HC-SR04 (5V)	15	97.65%
Dual Transducer Ultrasonic Level Sensor version 2	HC-SR04 (3 5V)	5	97.36%
Sonar: Single Transducer Range Finder	SRF02	3	90%
Electromagnetic Flowmeter (WaDI)	iSOLV EFS803/CFT183	4	98.2%
Differential Pressure Transmitter	iSOLV SPT 200	8	92.5%
RADAR: RF Level Sensor	iSOLV RD700	2	99%

Table 3 shows the TPR and FPR for each of the sensors. Eq. 2, gives us the percentage of rightly classified sensor (TPR), while Eq. 3 gives the percentage of mis-classification of sensor (FPR). Ideally,  $FPR = 0$  and  $TPR = 1$  for perfect classification. Table 3 shows the results for 20 ultrasonic sensors labeled as  $S1, S2, \dots, S20$ , where the TPR is 100% and FPR is 0% for nearly all sensors.

**RQ3: How accurately can we identify sensors?** Table 5 shows the sensor identification accuracy for all 44 sensors used in our study. This table lists the type and model of sensors, number of sensors, and identification accuracy obtained in the experiments. It is noted that the lowest identification accuracy is 90% for 5 sensors and, the remaining 27 sensors have identification accuracy above 97% for their sensor type, respectively. These results highlight the significance of *NoiSense*.

#### 4.4. Scalability of the *NoiSense*

**RQ4: Is a fingerprint unique against many sensors such as is the case in large plants?** Data from the experiments reported here does not provide a clear answer to this question as we had limited sensors in the lab to test the idea of fingerprinting. We also do not have any theoretical proof of scalability. However, we experimented on tens of small ultrasonic sensors, which gives an indication of scalability of *NoiSense*. The experiment was designed using a set of 20 dual transducer ultrasonic sensors. 5 sensors were selected randomly out of these 20 sensors and identification

accuracy calculated using SVM. Then, 10 sensors were selected randomly, followed by 15 and ultimately all 20 sensors were selected. The idea is to explore if identification accuracy drops as the number of sensors to be identified is increased. The results are shown in Table 6. Identification accuracy does not drop by increasing the number of sensors in the identification pool. This result is intuitive as the idea of *NoiSense* results from the assumption that each device has unique noise characteristics due to manufacturing imperfections. By increasing the number of devices, the confusion for a classifier does not necessarily increase as each device brings its own unique fingerprint. This shows that the idea of *NoiSense* is scalable for a larger number of sensors.

#### 4.5. Stability of *NoiSense*

**RQ5: Is the sensor fingerprint stable between different runs of the experiment over a period of time?** Several experiments were performed to answer this question. Data collection phase for the level sensors installed in the water treatment testbed was spread over a period of six months. Results shown in Table 5, for the case of *iSOLV Level Wizard 2 (level sensor in SWaT)* are representative of this experiment. From these results we conclude that sensor fingerprints are stable over long periods of time (i.e. six months in our study).

Besides collecting data over long periods of times on the water treatment testbed, another experiment was designed to

TABLE 6. RANDOMLY SELECTED SENSORS VS. ACCURACY.

Number of Sensors	Accuracy (Chunk Size (s) = 120)
5	96.97%
10	97.92%
15	97.65%
20	96.43%

test with low cost ultrasonic sensors. To see if the sensor fingerprint is stable between different runs of the *NoiSense*, data was collected for 3 hours for each sensor and then remove those from the tank and Arduino board (we refer to it as "first run"). Then, in the second run, the sensor was placed back on the tank and connected with the Arduino board. Features from the collected data are compared across in two runs. A visual representation of this experiment is shown in Figure 7. Here the first run is on the left hand and second on the right hand column. The top pane shows the time domain data plotted for a sample of two sensors. The time domain data is the same from first run to second run. However, shape and variance of time domain signal looks similar for the two sensors. To further distinguish and to motivate the need for using spectral features, frequency response (Fourier Transform) was plotted in the middle and bottom pane for sensor1 and sensor2, respectively. From the middle and bottom panes, we can see the frequency bins ( $x$ -axis) for peak magnitude is different for both the sensors even though the signal might look similar in time domain, but they have distinguishable spectral features. In Figure 7, if we look horizontally we can see that these features are present from one run to the second, thus showing the stability of sensor fingerprints.

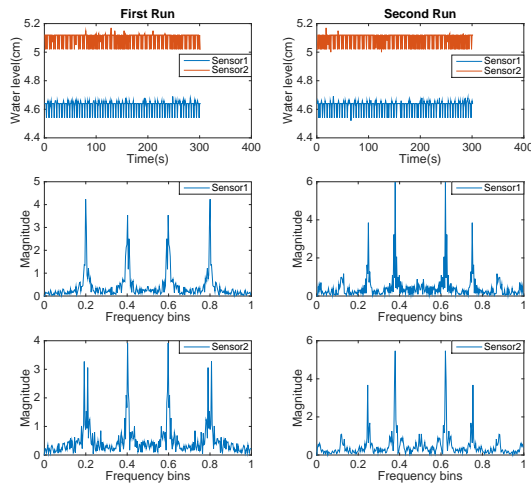


Figure 7. Stability of the *NoiSense* for the case of two small ultrasonic sensors. Run 1 corresponds to data collected for three hours, run 2 for two hours after removal and re-installation of sensor. In time domain both sensors time series look similar but frequency response is distinguishable and also stable for different runs.

TABLE 7. ANALOG SENSOR SPOOFING ATTACK DETECTION.

Number of Sensors	Attack Detection Accuracy
10	100.0%
7	99.64%
3	99.28%

#### 4.6. Analog Sensor Spoofing Attack Detection

RQ6: *Is the sensor fingerprint based method, able to detect analog sensor spoofing attack?* We extend attack detection performance of the proposed sensor fingerprinting method to detect analog sensor spoofing attacks. Experiments are designed, on attacks reported in the literature [12]. Such an attacker model is shown in Figure 1 where an attacker spoofs the reading before it reaches the sensing device. Experimental setup is shown in Figure 11, in Appendix E. The experiment was designed to measure the distance from a wall using active ultrasonic sensors. An active sensor has a transmitter (TX) that transmits a probe signal, and a receiver (RX) that listens to a response signal based on which the physical quantity is measured. It can also be applied in a similar way for the case of water level measurement in a water treatment plant. A legitimate sensor is placed at a fixed distance from the wall and an attacker is brought to transmit a response signal that deceives the sensor into believing that the spoofed signal is the actual echo from the wall. Data was collected before and after the start of the attack. Results are shown in Table 7. For 20 ultrasonic sensors, the attack is detected soon after it is launched. Data was collected and labeled in attack and attack-free scenarios. SVM was used to classify the presence of an attacker based on ground truth (labeled data). The intuition behind sensor fingerprint is that it is a unique characteristic for a pair of transmitter (TX) and receiver (RX) for an active sensor. Since, during this attack, RX is actually getting a sound wave from another TX, violating that fingerprint. This observation is intuitive, as presence of an active attacker in victim’s vicinity, raise the energy (impeding sound waves on transducer) in the environment [15] and changes the noise pattern for the sensor’s receiving transducer [63].

**Theoretical Proof:** In the following, theoretical guarantee is provided for detection of analog sensor spoofing attacks. To understand this, we need look at how these sensors work. For example, in ultrasonic sensors, sound waves vibrate the diaphragm of the transducer and sound energy is converted into electrical signal and appears at the output of the transducer. This produced voltage is proportional to the strength of sound vibrations and is the signal of interest. This analog signal has sensor noise effects in it. In the literature [63], *signal to noise ratio (SNR)* is used to analyze the effects of noise on a signal.

**Definition 1.** *Noise floor: The magnitude of noise in a sensing device is referred to as ‘noise floor’ [63].*

**Definition 2.** *Energy of a time domain signal ( $S$ ) can be*

calculated as:

$$\mathcal{S} = \int_0^T V^2 dt \quad (4)$$

where  $V$  is the voltage level of the signal, and  $T$  is time window for which  $\mathcal{S}$  is measured [15].

**Definition 3.** *Signal to Noise Ratio (SNR): It is defined as a ratio of signal energy ( $\mathcal{S}$ ) to noise energy ( $\mathcal{N}$ ) [63].*

**Theorem 1.** *Let  $SNR_s = \frac{\mathcal{S}_s}{\mathcal{N}_s}$ , be sensor's signal to noise ratio. Under analog sensor spoofing attack, noise floor changes and  $SNR_s$  deviates from the one under normal operation, and the attack is detected.*

*Proof.* The proof of the theorem follows directly from Definitions 1-3 and a similar analog signal contamination argument made in [15]. Suppose the signal to noise ratio at a sensor is given as  $SNR_s = \frac{\mathcal{S}_s}{\mathcal{N}_s}$ , where  $\mathcal{S}_s$  is signal strength and  $\mathcal{N}_s$  is sensor noise at the output of the transducer of the sensor. During the attack signal to noise ratio becomes  $SNR_a = \frac{\mathcal{S}_s + \mathcal{S}_a}{\mathcal{N}_s}$ , where  $SNR_a$  is signal to noise ratio under attack and  $\mathcal{S}_a$  is the attacker's signal energy. Since  $\mathcal{S}_s + \mathcal{S}_a \neq \mathcal{S}_s$ , hence  $SNR_a \neq SNR_s$ . From this result we can see that in the presence of an attacker, *noise floor* is changed and since *NoiSense*, is based on sensor's noise pattern, attack will be detected as sensor's fingerprint could not be authenticated.  $\square$

## 4.7. Discussion

**Accuracy:** The proposed method is intended to complement the existing intrusion detection systems. Legacy intrusion detection systems are either host based or network based [64], and hence those may not detect physical or analog sensor spoofing attacks. The method proposed in this work is based on the physical structure and hardware of the devices, making it an intrusion detection system for physical attacks. However, because of the very nature of the proposed scheme, feature vector has some randomness and few sensors occasionally have overlapping feature sets. This leads to an identification accuracy less than 100%. Nevertheless, in the majority of the cases, sensor identification accuracy more than 97% was achieved which makes our proposed method comparable to those proposed in the literature.

**Scalability:** Results in the previous section indicate that *NoiSense* scales well with an increase in the number of sensors to be fingerprinted. However, in comparison with the device fingerprinting work in CPS domain, authors in [26], created fingerprints for 2 relay devices, from two different vendors, with an accuracy of 92%. In comparison, our method performs better with an accuracy of 97% for tens of different sensors of same model.

**Robustness against Forgery:** Even though an adversary had learned the sensor noise, we believe each new device has it's own noise, so when an adversary replaces a device, it can not modify it's hardware to get the same noise pattern as another device without affecting its performance or intended application. Physically tampering a the device

would change the noise pattern. If we assume a powerful attacker who can learn the noise distribution and reproduce it, our proposed scheme is still helpful for raising the bar for attackers. Learning hardware characteristics, for all the hardware devices in the network, is time consuming and will likely raise suspicion and ultimately reveal the presence of an attacker to system operators.

**Application in Real-World CPS:** We have collected and tested the proposed method for a data set collected over a period of six months in a real water treatment testbed and for a couple of week data from a water distribution testbed. The results are promising for such a time period. However, it is recommended to train the classifiers after every plant maintenance cycle. Moreover, being used in a testbed for six months is different from being used in a real world production system of physical plants with possibly more harsh environment especially for the case of level measurements including rivers, dams etc. Although the testbeds used in the reported experiments imitates real water treatment plants as close as possible but we believe the sensors and actuators wear out with time, rendering them less accurate. There is a possibility that those environmental effects may change the fingerprint but according to our hypothesis each sensor will be affected in a distinct way and, if retrained, will possess a unique fingerprint. As far as the ambient noise or interference is concerned that would affect all the devices in a similar manner, letting us to cancel out those effects from sensor fingerprint.

## 5. Towards Physical Quantity based Challenge-Response Protocol

The proposed *NoiSense* works well in the case of physical manipulation or physical layer signal spoofing in the sensor. However, it is challenged by attackers who can learn and inject the correct sensor noise in the digital domain while spoofing the real measurements. For example, in the case of a replay attack, sensor noise from previous readings also gets replayed while preserving the sensor noise fingerprint. Another example is a traditional *man-in-the-middle* attacker who can learn the sensor noise pattern and, while injecting fake sensor measurements, also adds the sensor noise fingerprint making it hard for *NoiSense* to detect such an attack. We propose a novel *challenge-response* protocol in which a challenge is produced in the physical quantity to be measured. Traditionally, a challenge is generated in the digital domain [12], [22] and its effect is observed on sensor measurements. In our scheme, the challenge originates from the physical/analog domain. The challenger, will be between the physical quantity and the receiver (RX) of the sensor. We send the challenge as a fake measurement and observe it's effect in sensor's output. This proposed *challenge-response* protocol, enables *NoiSense*, to detect a strong cyber adversary. Essentially, a special sort of analog spoofing attack is launched which, as discussed in the previous sections, is expected to be reflected and identifiable

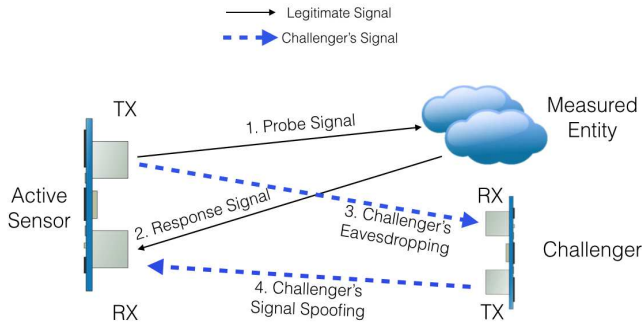


Figure 8. Challenge-Response Protocol.

by the sensor readings, as this attack will change the noise fingerprint.

Consider the attack scenario *Replay and Advanced Sensor Spoofing Attack (S8)* as explained in section 2.1. The proposed *challenge-response* protocol would detect replay attack as the attacker would not be able to replicate the effect of a fresh challenge. Now, consider an even stronger attacker who can actively receive sensor data and tries to discover the presence of the *challenge*. We train the machine learning algorithm on *legitimate sensor-challenger pair fingerprint* in addition to training it for sensor noise fingerprint. With the *challenge-response* protocol, *NoiSense* tests sensor data for the presence of the sensor's as well as the challenger's fingerprint. In the presence of a challenge, an active attacker can observe the change in the sensor fingerprint but it needs time to learn the challenger's fingerprint. Machine learning based identification works on a chunk of data, and it takes time to collect that data and to make a decision. Therefore, an attacker needs some time and a chunk of data to learn challenger's fingerprint. After learning the challenger's fingerprint, an attacker might try to add the effect of challenger's noise in sensor measurements but it will appear with a delay (due to the time it took to learn the change), which will likely expose the attacker. There is an important consideration in designing such a *challenge-response* protocol; so that it should not effect the normal functionality of the industrial control system.

**Design of Challenge-Response Protocol:** In Figure 8, a proposed setup for *challenge-response* protocol is shown. A *challenger* is a device similar to the one used for sensing. This *challenger* is placed between the active sensor and the measured entity. While active sensor waits for response to the probe signal, the *challenger* transmits a signal to spoof the fake reading. This is similar to analog sensor spoofing attack as explained in section 2.1. Unlike analog spoofing attack, the *challenger* needs to spoof a reading that is close to the reading that would be received by a real sensor. A naive *challenger* would send a random reading that is not close to the real quantity and we would be able to observe its effect on the sensor's data but it might disturb the physical process as our *challenger* itself is attacking sensor

measurements. One important consideration in the design of *challenger* is to not disturb the control logic because of the added *challenge* in the sensor data. Receiver (RX) is used as the challenger to receive the probe signal from the active sensor's transmitter (TX). Based on the probe signal, the *challenger* can calculate a quantity to be spoofed. For example, if the sensor is an ultrasonic sensor used to measure the distance, then by listening to sensor's probe signal, the *challenger* is able to calculate the exact quantity and it can generate the new spoofed signal which is close to the real quantity. Thus the spoofed signal will not trigger any wrong actuation due to the added *challenge* in the sensor but will change the noise fingerprint of the sensor. Such a *challenger* may not work in all cyber physical systems but it is practical in the use cases studied in this work. In the case of an ultrasonic level sensor installed in the testbeds used in this work, the *challenger* can be placed, for example, on water surface on a buoy, and it would be able to calculate the challenge signal which is close to the real system state. Following are the steps to create a *challenge* as shown in Figure 8:

- 1) Transmitter (TX) of the active sensor sends a probe signal towards the quantity to be measured.
- 2) Response signal arrives at the sensor's receiver (RX) resulting in measurement of the physical quantity.
- 3) Receiver (RX) of the *challenger* is passively listening to probe signals. This eavesdropping will help it to calculate the amount of signal to be spoofed so that it does not disturb the control logic.
- 4) Transmitter (TX) of the *challenger* spoofs a signal to the receiver (RX) of the active sensor. The amount of fake measurement to be transmitted is calculated in step 3. This will keep the measurement close to the real quantity but change the noise fingerprint of the sensor.

**Security argument:** The proposed challenge is an instance of an analog spoofing attack, which has been shown to alter the noise profile of the sensor under test in the previous subsection. Therefore, by challenging a sensor at a random time  $t$  for a random  $\delta$  period, we expect the values of the sensor (as for instance reflected in the ICS historian) to have the (anomalous) noise fingerprint of an analog spoofing attack. If not, we can suspect a cyber attacker to be spoofing the historian values. An attacker who is aware of this *challenge-response*, but at the same time is spoofing the real values, needs thus to consistently reflect the anomalous noise profile starting at time  $t$  and for  $\delta$  seconds. However, if the challenge is close to the real physical measurement, an attacker needs to wait  $\Delta$  seconds in order to recognize that the noise profile has changed, at the beginning and at the end of the challenge. Therefore, he can at most react consistently with the expectations of the challenger at time  $t + \Delta$  and stop at  $t + \delta + \Delta$ . As is shown in the previous sections,  $\Delta$  needed to detect a change in the noise profile is significant (around 1 minute) and can be leveraged to detect incoherent responses to the challenge.

## 6. Related work

*Device Fingerprinting:* The approach presented in this article is inspired by the idea of using sensor noise as a fingerprint for camera identification [65]. In [65], images are taken by a camera and filtered to obtain noise components and averaged for all images. This resultant noise vector acts as a reference pattern for test images. An image is tested against reference patterns for all cameras being studied and matched with one having the highest correlation with image's noise vector. In [66] prospects of sensor identification, based on [65], are studied. The authors analyzed the effects of varying number of images to get a reference pattern. It is shown that the noise fingerprinting method as in [65], can be used to counteract injection attacks at the time of checks in biometric systems as well as to establish evidence in a criminal incidence. In [67] a method for image authentication from flatbed desktop scanners is presented. This method is similar to that used in earlier works for identifying digital cameras using pattern noise of the imaging sensor. In [68] seven techniques for extracting unique signatures from NAND flash devices are studied. These techniques are experimentally evaluated using thirteen different flash memory devices. The techniques can help identify and authenticate electronic devices that use flash devices. The idea of fingerprinting a device remotely based on its hardware is presented in [28]. Small microscopic deviations in device's clock [69], [70] are used as fingerprint for the particular device. In [71] inter arrival time of packets is analyzed to fingerprint devices on a small campus network. In [29], 50 RFID smart cards from the same manufacturer and type are tested for fingerprints.

*Smartphone Fingerprinting:* In [32] hardware imperfections during the sensor manufacturing process are exploited as a fingerprint. Accelerometers in smartphones are used to create fingerprints. Experimental results show that when the properties of these imperfections are extracted, the device, and ultimately a user, can be identified. In [72] a smartphone speaker is used as a fingerprint component. During fabrication, subtle imperfections arise in device speakers which induce anomalies in produced sounds. Experimental results show high accuracy of the proposed method for speakers from same as well as different vendors. Another work [73] used sensor fingerprinting to identify a mobile device. In [73] the speakerphone-microphone system and the accelerometer in a typical smartphone were used to identify the mobile device. The article listed types of sensors and the related imperfections which can be useful in identifying systems. [74] analyze techniques to mitigate device fingerprinting either by calibrating the sensors to eliminate the signal anomalies, or by adding noise that obfuscates the anomalies.

*Software based Fingerprinting:* A technique is proposed [75], that identifies the wireless device driver running on an IEEE 802.11 compliant device by passive monitoring. In [76] unique devices over a Wireless Local Area Network (WLAN) are fingerprinted in a passive manner through

the timing analysis of 802.11 probe request frames. Nmap [77], uses active fingerprinting by sending specific requests to determine operating systems and server versions. On the other hand p0f [78], is a tool which can determine operating system and browser version of a client by passively monitoring TCP and HTTP header fields. The version and configuration information of the web browser will be transmitted to websites upon request [79], which can be used to fingerprint the devices on which these browsers are running. In [80], indirect history data, such as information about categories of visited websites can also be effective in fingerprinting users. In [81], 25 applications on cloud are fingerprinted based on cache access pattern.

*RF Fingerprinting:* RF fingerprints have been proposed using a measured temporal link signature to uniquely identify the link between a transmitter and a receiver [30], [82]. A similar approach based on received signal strength is shown for the case study of a smart water treatment plant [83]. Modulation domain of RF signals is also studied to generate unique device fingerprints [84]. Researchers have shown that signal waveforms can also be used for device fingerprinting amid manufacturing inconsistencies [31], [85]. Besides wireless fingerprinting research has shown that it is possible to fingerprint devices based on Ethernet NICs analog signals [33].

*CPS Device fingerprinting:* In [26] authors focus on the idea of device fingerprinting in ICS. One approach in [26] is based on traditional network traffic monitoring and observing message response times, while the second approach is based on physical operation times of a device. Analysis is carried out on 2 latching relays based on their operation timings. Another related work presented a preliminary study, on the idea of sensor fingerprinting in [86], using 2 sensors, based on correlation analysis with an accuracy of 86%. However, to the best of our knowledge, this article present the first attempt to present a rigorous analysis on a multitude of sensors specific to ICS. Our machine learning based sensor fingerprint successfully identifies sensors with an accuracy as high as 99%. This is first work to present a holistic scheme to detect physical and cyber attacks using physical quantity based *challenge-response* protocol.

## 7. Conclusions

In this paper we have presented a sensor identification technique for ICS. The proposed method creates a fingerprint of each sensor based on sensor's noise statistics. The feature set used is extracted from sensor noise and is leveraged for sensor identification. Machine learning is used to identify the installed sensors. Uniqueness in the fingerprint is the result of manufacturing imperfections even for the same type and model of the device. The results in this article point to an effective method for hardware identification without affecting the performance of the host system. The results reported show that it is possible to identify whether test data is generated by the installed sensor

or by a replaced sensor. Out of a 44 total sensors, 31 were identified with an accuracy more than 96% and 13 with an accuracy above 90%. Also, using true positive and false positive rate as a metric, we got a true positive rate of 100% for most of the sensors and a false positive rate of 0%. Analog sensor spoofing attacks are detected with an accuracy of 100%. Moreover, a *challenge-response* protocol is proposed that is able to detect advanced cyber attackers by means of sensor fingerprinting.

In the future, we plan to extend the proposed technique to fingerprint actuators as well. Also we plan to extend the feature set considered for fingerprinting based on the physical working principles of sensors.

## References

- [1] Rajeev Alur. *Principles of cyber-physical systems*. MIT Press, 2015.
- [2] NIST. Cyber-physical systems. <https://www.nist.gov/el/cyber-physical-systems>, 2014.
- [3] E. A. Lee. Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369, May 2008.
- [4] Defense Use Case. Analysis of the cyber attack on the ukrainian power grid. 2016.
- [5] J. Slay and M. Miller. Lessons learned from the maroochy water breach. *Springer 620 US, Boston, MA*, pages 73–82, 2008.
- [6] N. Falliere, L.O. Murchu, and E. Chien. W32 stuxnet dossier. symantec, version 1.4. [https://www.symantec.com/content/en/us/enterprise/media/security\\_response/whitepapers/w32\\_stuxnet\\_dossier.pdf](https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf), Feb. 2011.
- [7] Alvaro Cardenas, Saurabh Amin, Bruno Sinopoli, Annarita Giani, Adrian Perrig, and Shankar Sastry. Challenges for securing cyber physical systems. In *Workshop on future directions in cyber-physical systems security*, page 5, 2009.
- [8] CNN. Staged cyber attack reveals vulnerability in power grid. <http://edition.cnn.com/2007/US/09/26/power.at.risk/index.html>, year = 2007.
- [9] Wired. A cyberattack has caused confirmed physical damage for the second time ever. <https://www.wired.com/2015/01/german-steel-mill-hack-destruction/>, 2015.
- [10] Dieter Gollmann and Marina Krotofil. *Cyber-Physical Systems Security*, pages 195–204. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
- [11] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. Rocking drones with intentional sound noise on gyroscopic sensors. In *Proceedings of the 24th USENIX Conference on Security Symposium, SEC'15*, pages 881–896, Berkeley, CA, USA, 2015. USENIX Association.
- [12] Yasser Shoukry, Paul Martin, Yair Yona, Suhas Diggavi, and Mani Srivastava. Pycra: Physical challenge-response authentication for active sensors under spoofing attacks. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15*, pages 1004–1015, New York, NY, USA, 2015. ACM.
- [13] S. Yasser, M. Paul, T. Paulo, and S. Mani. Non-invasive spoofing attacks for anti-lock braking systems. In *CHES, Springer Link*, volume 8086, pages 55–72, oct. 2013.
- [14] Hocheol Shin, Yunmok Son, Youngseok Park, Yujin Kwon, and Yongdae Kim. Sampling race: Bypassing timing-based analog active sensor spoofing detection on analog-digital systems. In *Proceedings of the 10th USENIX Conference on Offensive Technologies, WOOT'16*, pages 200–210, Berkeley, CA, USA, 2016. USENIX Association.
- [15] D. F. Kune, J. Backes, S. S. Clark, D. Kramer, M. Reynolds, K. Fu, Y. Kim, and W. Xu. Ghost talk: Mitigating emi signal injection attacks against analog sensors. In *2013 IEEE Symposium on Security and Privacy*, pages 145–159, May 2013.
- [16] Marco Rocchetto and Nils Ole Tippenhauer. CPDY: extending the dolev-yao attacker with physical-layer interactions. *CoRR*, abs/1607.02562, 2016.
- [17] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A. R. Sadeghi, M. Maniatakos, and R. Karri. The cybersecurity landscape in industrial control systems. *Proceedings of the IEEE*, 104(5):1039–1057, May 2016.
- [18] Yilin Mo and Bruno Sinopoli. Integrity attacks on cyber-physical systems. In *Proceedings of the 1st International Conference on High Confidence Networked Systems, HiCoNS '12*, pages 47–54, New York, NY, USA, 2012. ACM.
- [19] Y. Mo and B. Sinopoli. Secure control against replay attacks. In *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 911–918, Sept 2009.
- [20] Gyorgy Dan and Henrik Sandberg. Stealth attacks and protection schemes for state estimators in power systems. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 214–219. IEEE, 2010.
- [21] David I Urbina, Jairo A Giraldo, Alvaro A Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. Limiting the impact of stealthy attacks on industrial control systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1092–1105. ACM, 2016.
- [22] 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, Feb 2015.
- [23] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero. An experimental security analysis of an industrial robot controller. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 268–286, May 2017.
- [24] Stephen McLaughlin, Dmitry Podkuiko, and Patrick McDaniel. *Energy Theft in the Advanced Metering Infrastructure*, pages 176–187. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [25] A. Humayed, J. Lin, F. Li, and B. Luo. Cyber-physical systems security – a survey. *IEEE Internet of Things Journal*, PP(99):1–1, 2017.
- [26] David Formby, Preethi Srinivasan, Andrew Leonard, Jonathan Rogers, and Raheem Beyah. Whos in control of your control system? device fingerprinting for cyber-physical systems. In *NDSS*, April 2016.
- [27] S. Sridhar, A. Hahn, and M. Govindarasu. Cyber physical system security for the electric power grid. *Proceedings of the IEEE*, 100(1):210–224, Jan 2012.
- [28] T. Kohno, A. Broido, and K. C. Claffy. Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2(2):93–108, April 2005.
- [29] Boris Danev, Thomas S. Heydt-Benjamin, and Srđjan Čapkun. Physical-layer identification of rfid devices. In *Proceedings of the 18th Conference on USENIX Security Symposium, SSYM'09*, pages 199–214, Berkeley, CA, USA, 2009. USENIX Association.
- [30] Daniel B. Faria and David R. Cheriton. Detecting identity-based attacks in wireless networks using signalprints. In *Proceedings of the 5th ACM Workshop on Wireless Security, WiSe '06*, pages 43–52, New York, NY, USA, 2006. ACM.
- [31] K. A. Remley, C. A. Grosvenor, R. T. Johnk, D. R. Novotny, P. D. Hale, M. D. McKinley, A. Karygiannis, and E. Antonakakis. Electromagnetic signatures of wlan cards and network security. In *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.*, pages 484–488, Dec 2005.

- [32] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi. Accelerprint: Imperfections of accelerometers make smartphones trackable. In *Network and Distributed System Security Symposium (NDSS)*, 2014.
- [33] Ryan M. Gerdes, Thomas E. Daniels, Mani Mina, and Steve F. Russell. Device identification via analog signal fingerprinting: A matched filter approach. In *NDSS*, 2006.
- [34] A. P. Mathur and N. O. Tippenhauer. Swat: a water treatment testbed for research and training on ics security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, pages 31–36, April 2016.
- [35] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P. Mathur. Wadi: A water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, CySWATER '17*, pages 25–28, New York, NY, USA, 2017. ACM.
- [36] Sridhar Adepu and Aditya Mathur. Distributed detection of single-stage multipoint cyber attacks in a water treatment plant. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, ASIA CCS '16*, pages 449–460, New York, NY, USA, 2016. ACM.
- [37] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen. Cyber security of water scada systems x2014;part i: Analysis and experimentation of stealthy deception attacks. *IEEE Transactions on Control Systems Technology*, 21(5):1963–1970, Sept 2013.
- [38] Naman Govil, Anand Agrawal, and Nils Ole Tippenhauer. On ladder logic bombs in industrial control systems. *CoRR*, abs/1702.05241, 2017.
- [39] Luis Garcia, Ferdinand Brasser, Mehmet H. Cintuglu, Ahmad-Reza Sadeghi, Osama Mohammed, and Saman A. Zonouz. Hey, my malware knows physics! attacking plcs with physical model aware rootkit. In *24th Annual Network & Distributed System Security Symposium (NDSS)*, February 2017.
- [40] Sergei Petrovich Skorobogatov. *Semi-invasive attacks: a new approach to hardware security analysis*. PhD thesis, University of Cambridge Ph. D. dissertation, 2005.
- [41] Ross Anderson and Markus Kuhn. Tamper resistance: A cautionary note. In *Proceedings of the 2Nd Conference on Proceedings of the Second USENIX Workshop on Electronic Commerce - Volume 2, WOEC'96*, pages 1–1, Berkeley, CA, USA, 1996. USENIX Association.
- [42] S. Adepu and A. Mathur. Generalized attacker and attack models for cyber physical systems. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 283–292, June 2016.
- [43] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen. Cyber security of water scada systems-part i: analysis and experimentation of stealthy deception attacks. *IEEE Transactions on Systems Technology*, pages 1963–1970, 2013a.
- [44] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen. Cyber security of water scada systems-part ii: Attack detection using enhanced hydrodynamic models. *IEEE Transactions on Systems Technology*, pages 1679–1693, 2013b.
- [45] C. Murguia and J. Ruths. Characterization of a cusum model-based sensor attack detector. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1303–1309, Dec 2016.
- [46] Marco Rocchetto and Nils Ole Tippenhauer. *On Attacker Models and Profiles for Cyber-Physical Systems*, pages 427–449. Springer International Publishing, Cham, 2016.
- [47] Youngseok Park, Yunmok Son, Hocheol Shin, Dohyun Kim, and Yongdae Kim. This ain't your dose: Sensor spoofing attack on medical infusion pump. In *10th USENIX Workshop on Offensive Technologies (WOOT 16)*, Austin, TX, 2016. USENIX Association.
- [48] Chuadhry Mujeeb Ahmed, Carlos Murguia, and Justin Ruths. Model-based attack detection scheme for smart water distribution networks. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ASIA CCS '17*, pages 101–113, New York, NY, USA, 2017. ACM.
- [49] Qadeer R., Murguia C. and Ahmed C.M., and Ruths J. Multistage downstream attack detection in a cyber physical system. In *Cyber-ICPS Workshop 2017, in conjunction with ESORICS 2017*, Sep. 2017.
- [50] C. M. Ahmed, S. Adepu, and A. Mathur. Limitations of state estimation based cyber attack detection schemes in industrial control systems. In *2016 Smart City Security and Privacy Workshop (SCSP-W)*, pages 1–5, April 2016.
- [51] Jenny T., Edin T., Romesh N., and Muhammad A. *Ultrasonic Fluid Quantity Measurement in Dynamic Vehicular Applications: A Support Vector Machine Approach*. Springer, 2013.
- [52] F. Coutard, E. Tisserand, and P. Schweitzer. The temperature influence on the piezoelectric transducer noise, measurements and modelling. In *IEEE Ultrasonics Symposium*, volume 3, pages 1652–1655, 2005.
- [53] S. Petr, M. Jiri, and S. Josef. Noise in piezoelectric ceramics at the low temperature. In *Radio Engineering*, volume 20, 2011.
- [54] Flotech. RD700 2-wire radar level transmitter. <http://www.flotech.com.sg/downloads/rd700-radar-level-transmitter.pdf>, 2016.
- [55] Indumart. Accuracy of the radar measurements. <http://www.indumart.com/Level-measurement-3.pdf>, 2012.
- [56] F. stner, E. Aydemir, E. Gle, M. Iarlsan, M. elebi, and E. Demirel. Antenna radiation pattern measurement using an unmanned aerial vehicle (uav). In *2014 XXXIth URSI General Assembly and Scientific Symposium (URSI GASS)*, pages 1–4, Aug 2014.
- [57] Flotech. Electromagnetic flowmeter. <http://www.unhas.ac.id/rhiza/arsip/iwormee2009/old-archieive/Spec2016>.
- [58] David Lincoln. *AN INVESTIGATION INTO AN ELECTROMAGNETIC FLOWMETER FOR USE WITH LOW CONDUCTIVITY LIQUIDS*. PhD thesis, University of Wales, Cardiff, Sep. 2006.
- [59] Peter Welch. The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73, 1967.
- [60] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [61] Bela G. Liptak. *Instrument Engineer's Handbook*, volume 1. CRC Press, 4 edition, June 2003.
- [62] Arduino. Arduino. <http://www.arduino.org/>, 2016.
- [63] D. Buchla and W. McLachlan. *Applied Electronic Instrumentation and Measurement*. Maxwell Macmillan international editions in engineering. Merrill, 1992.
- [64] Robert Mitchell and Ing-Ray Chen. A survey of intrusion detection techniques for cyber-physical systems. *ACM Comput. Surv.*, 46(4):55:1–55:29, March 2014.
- [65] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2), 2006.
- [66] B. Nick, K. Nathan, C. Bojan, and R. Arun. Identifying sensors from fingerprint images. In *10th IEEE Computer Vision and Pattern Recognition Workshops*, pages 78–84, 2009.
- [67] N. Khanna, A.K. Mikkilineni, G.T.C. Chiu, J.P. Allebach, and E.J. III Delp. Scanner identification using sensor pattern noise. In *SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents*, volume 6505, 2007.

- [68] P. Prabhu, A. Akel, L. M. Grupp, W.-K. S. Yu, G. E. Suh, E. Kan, and S. Swanson. Extracting device fingerprints from flash memory by exploiting physical variations. In *4th international conference on Trust and trustworthy computing*, pages 188–201, 2011.
- [69] S. B. Moon, P. Skelly, and D. Towsley. Estimation and removal of clock skew from network delay measurements. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 227–234 vol.1, Mar 1999.
- [70] Vern Paxson. On calibrating measurements of packet transit times. In *Proceedings of the 1998 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '98/PERFORMANCE '98*, pages 11–21, New York, NY, USA, 1998. ACM.
- [71] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah. Gtid: A technique for physical device and device type fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 12(5):519–532, Sept 2015.
- [72] A. Das and N. Borisov. Poster: Fingerprinting smartphones through speaker. In *IEEE Security and Privacy Symposium*, 2014.
- [73] H. Bojinov, D. Boneh, Y. Michalevsky, and G. Nakibly. Mobile device identification via sensor fingerprinting. In <http://arxiv.org/abs/1408.1416>, March 2016.
- [74] Anupam Das, Nikita Borisov, and Matthew Caesar. Exploring ways to mitigate sensor-based smartphone fingerprinting. *CoRR*, abs/1503.01874, 2015.
- [75] Jason Franklin, Damon McCoy, Parisa Tabriz, Vicentiu Neagoie, Jamie Van Randwyk, and Douglas Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, USENIX-SS'06, Berkeley, CA, USA, 2006. USENIX Association.
- [76] Loh Chin Choong Desmond, Cho Chia Yuan, Tan Chung Pheng, and Ri Seng Lee. Identifying unique devices through wireless fingerprinting. In *Proceedings of the First ACM Conference on Wireless Network Security, WiSec '08*, pages 46–55, New York, NY, USA, 2008. ACM.
- [77] G. Lyon. Nmap network mapper. <http://www.nmap.org/>, 2011.
- [78] M. Zalewski. p0f v3. <http://lcamtuf.coredump.cx/p0f3/>, 2016.
- [79] Peter Eckersley. How unique is your web browser? In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies, PETS'10*, pages 1–18, Berlin, Heidelberg, 2010. Springer-Verlag.
- [80] Lukasz Olejnik, Claude Castelluccia, and Artur Janc. Why Johnny Can't Browse in Peace: On the Uniqueness of Web Browsing History Patterns. In *5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2012)*, Vigo, Spain, July 2012.
- [81] Berk Gulmezoglu, Thomas Eisenbarth, and Berk Sunar. Cache-based application detection in the cloud using machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ASIA CCS '17*, pages 288–300, New York, NY, USA, 2017. ACM.
- [82] Neal Patwari and Sneha K. Kasera. Robust location distinction using temporal link signatures. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, MobiCom '07*, pages 111–122, New York, NY, USA, 2007. ACM.
- [83] J. Prakash and C. M. Ahmed. Can you see me on performance of wireless fingerprinting in a cyber physical system. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pages 163–170, Jan 2017.
- [84] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, MobiCom '08*, pages 116–127, New York, NY, USA, 2008. ACM.

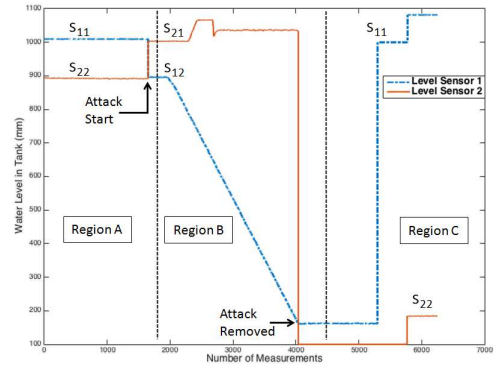


Figure 9. Impact of swapping two sensors on the measurements received by corresponding PLCs.  $s_{ij}$  represents  $i^{th}$  sensor in  $j^{th}$  tank. At the point of attack start  $PLC_1$  will start receiving values from sensor 1 in tank 2 ( $s_{12}$ ) and  $PLC_2$  will receive  $s_{21}$ .

- [85] J. Hall, M. Barbeau, and E. Kranakis. Radio frequency fingerprinting for intrusion detection in wireless networks. In *Defendable and Secure Computing*, 2005.
- [86] C. M. Ahmed and A. P. Mathur. Hardware identification via sensor fingerprinting in a cyber physical system. In *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 517–524, July 2017.
- [87] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Good practice in large-scale learning for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):507–520, March 2014.

## Appendix A. Physical Sensor Swap Attack in SWaT Testbed

*An Example Attack:* An attack scenario is created, comprising attacker goal of physical damage ( $\mathcal{G}_1$ ), an insider with the physical access ( $\mathcal{C}_1$ ) and with an objective of deceiving the controllers ( $\mathcal{O}_3$ ). *Sensor swap attack*, as explained in Section 2.1, can achieve attacker's goal as shown in following attack, which is launched on a real water treatment testbed. During the normal operation of the plant, if  $s_{11}$  indicates a low water level in tank 1,  $PLC_1$  issues a command to stop the outlet pump, to avoid dry running of the pump (which can result in physical damage). Similarly, if the tank 1 is full (based on  $s_{11}$  values), the inlet to tank 1 should be closed to avoid flooding. But if we swap sensor  $s_{11}$  with sensor  $s_{22}$ , the new sensor configuration is  $s_{21}$  which means sensor 2 is placed on top of tank 1 (but this sensor is still transmitting data to  $PLC_2$ ) and  $s_{12}$  which represents sensor 1 on tank 2 (but this sensor is still transmitting data to  $PLC_1$ ). Now, as soon as we start the attack,  $s_{21}$  indicates that the tank level is high (which indeed is high as it is placed over tank 1 instead of tank 2),  $PLC_2$  will keep running the outlet pump ultimately resulting in pump running dry and damage (assuming that there is no mechanical interlock that prevents dry running).

Figure 9 illustrates the impact of a *sensor swap attack*. The attack start and end times are as labeled in Figure 9.



When the attack starts, the level readings from  $s_{11}$  and  $s_{22}$  get exchanged. As the sensors are physically swapped, they keep sending data to their respective PLCs but those measurements are no longer coming from the same normal process. If the attack is not removed, the pump at stage two would be damaged, as it would continue to run without any water in the tank. The pump will stay ON because the corresponding level sensor is transmitting high water level to PLC. Note that as programmed in water treatment testbed, the PLC logic has a low water level limit below which it is dangerous to keep pumping. However, due to this attack, the PLC can be deceived. Regions A, B and C in Figure 9 show, respectively, the sensor readings before, during, and after the attack. A detailed analysis on such an attack and preliminary results based on sensor fingerprint, is presented in form of a short paper [86].

## Appendix B. Support Vector Machine Classifier

SVM is a data classification technique used in many areas such as speech recognition, image recognition and so on [87]. The aim of SVM is to produce a model based on the training data and give classification results for testing data. For a training set of instance-label pairs  $(x_i, y_i), i = 1, \dots, k$  where  $x_i \in \mathbb{R}^n$  and  $y \in \{1, -1\}^k$ , SVM require the solution of the following optimization problem:

$$\begin{aligned} & \underset{w, b, \zeta}{\text{minimize}} && \frac{1}{2} w^T w + C \sum_{i=1}^k \zeta_i \\ & \text{subject to} && y_i (w^T \phi(x_i) + b) \geq 1 - \zeta_i, \\ & && \text{where } \zeta_i \geq 0. \end{aligned} \quad (5)$$

The function  $\zeta$  maps the training vectors into a higher dimensional space. In this higher dimensional space a linear separating hyperplane is found by SVM, where  $C > 0$  is the penalty parameter of the error term. For the kernel function in this work we use the radial basis function:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0. \quad (6)$$

In our work, we have multiple sensors to classify. Therefore, multi-class SVM library LIBSVM [60] is used.

## Appendix C. Water Treatment Testbed

It is a fully operational (research facility) scaled down water treatment plant producing 5 gallons/minute of doubly filtered water, this testbed mimics large modern plants for water treatment. Following is the brief overview of the testbed, for further details, please refer to [34].

**Water Treatment Process:** The treatment process consists of six distinct stages each controlled by an independent Programmable Logic Controller (PLC). Control actions are taken by the PLCs using data from sensors. Stage P1

controls the inflow of water to be treated by opening or closing a motorized valve MV-101. Water from the raw water tank is pumped via a chemical dosing station (stage P2, chlorination) to another UF (Ultra Filtration) feed water tank in stage P3. A UF feed pump in P3 sends water via UF unit to RO (Reverse Osmosis) feed water tank in stage P4. Here an RO feed pump sends water through an ultraviolet dechlorination unit controlled by a PLC in stage P4. This step is necessary to remove any free chlorine from the water prior to passing it through the reverse osmosis unit in stage P5. Sodium bisulphate ( $\text{NaHSO}_3$ ) can be added in stage P4 to control the ORP (Oxidation Reduction Potential). In stage P5, the dechlorinated water is passed through a 2-stage RO filtration unit. The filtered water from the RO unit is stored in the permeate tank and the reject in the UF backwash tank. Stage P6 controls the cleaning of the membranes in the UF unit by turning on or off the UF backwash pump.

**Communication Network and Vulnerabilities:** Each PLC in the testbed obtains data from sensors associated with the corresponding stage, and controls pumps and valves in its domain. PLCs communicate with each other through a separate network. Communications among sensors, actuators, and PLCs can be via either wired or wireless links. Attacks that exploit vulnerabilities in the protocol used, and in the PLC firmware, are feasible and could compromise the communication links between sensors and PLCs, PLCs and actuators, among the PLCs, and the PLCs themselves. Having compromised one or more links, an attacker could use one of several strategies to send fake state data to one or more PLCs.

## Appendix D. Water Distribution Testbed

It is an operational testbed supplying 10 US gallons/min of filtered water. It represents a scaled-down version of a large water distribution network in a city. It contains three distinct control processes labeled P1 through P3, each controlled by its own set of PLCs.

**Stages in WADI:** Water distribution process is segmented into the following sub-processes: P1: Primary grid, P2: Secondary grid, P3: Return water grid.

**Primary grid:** The primary grid contains two raw water tanks of 2500 liters each, and a level sensor (1-LIT-001) to monitor the water level in the tanks. Water intake into these two tanks can be from the water treatment plant, from Public Utility Board inlet, or from the return water grid. A chemical dosing system is installed to maintain adequate water quality. Sensors are installed to measure the water quality parameters of the water flowing into and out of the primary grid.

**Secondary grid:** This grid has two elevated reservoir tanks and six consumer tanks. Raw water tanks supply water to the elevated reservoir tanks and, in turn, these tanks supply water to the consumer tanks based on a pre-set demand pattern. Once consumer tanks meet their demands, water

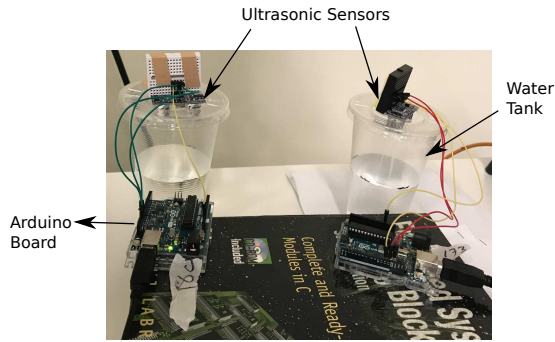


Figure 10. Experiment setup with small ultrasonic sensors (HCSR04). A glass of water is filled up to a marked level to simulate the water tank. Sensors are controlled by Arduino Uno and data is collected.

drains to the return water grid. Return water grid is equipped with a tank.

**Communications Infrastructure:**

The communication network contains layer-0 (L0), layer-1 (L1) and layer 2-(L2). L0 is at process level and connects actuators/sensors and I/O modules via RS485-Modbus protocol. L1 is the plant control network where all PLCs are connected to a central node in a star topology. Communication among PLCs and RTUs takes place over Ethernet switches using NIP/SP based on TCP and High Speed Packet Access (HSPA) cellular gateways using GPRS modem. L2 is a communication network between a touch panel Human-Machine Interface (HMI) and the plant control network. This network is implemented using star topology and consists of PLCs and RTUs. A firewall isolates the enterprise network from the plant control network. A SCADA workstation provides an interface between the plant operators and PLCs for remote monitoring and control.

**Appendix E.  
Supporting Figures/Tables**

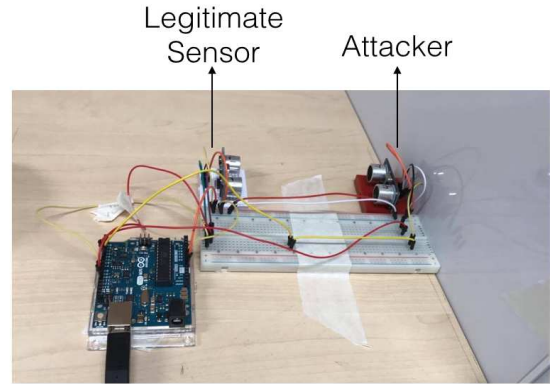


Figure 11. Experiment setup with small ultrasonic sensors (HCSR04). An experiment is set to measure a fixed distance between the sensor and the wall. Sensors are controlled by Arduino Uno and data is collected.

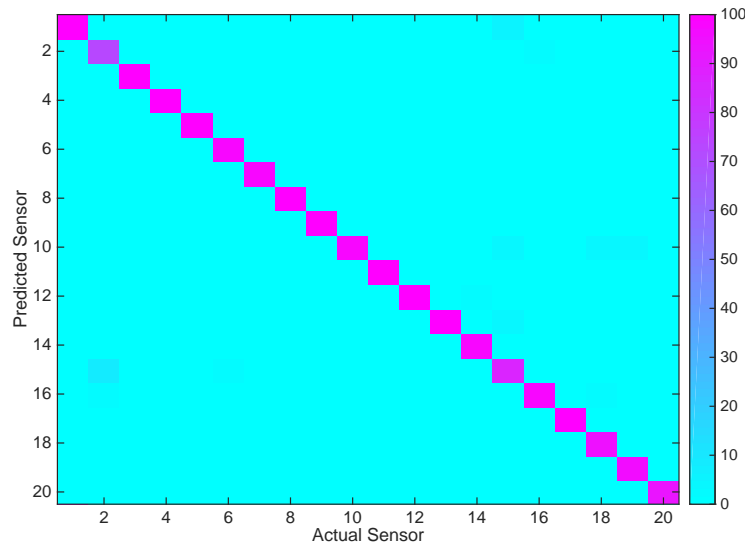


Figure 12. Confusion matrix for 20 small ultrasonic sensors.