

# Detection of Electromagnetic Interference Attacks on Sensor Systems

Youqian Zhang  
University of Oxford  
youqian.zhang@cs.ox.ac.uk

Kasper Rasmussen  
University of Oxford  
kasper.rasmussen@cs.ox.ac.uk

**Abstract**—Sensor systems are used every time a microcontroller needs to interact with the physical world. They are abundant in home automation, factory control systems, critical infrastructure, transport systems and many, many other things.

In a sensor system, a sensor transforms a physical quantity into an analog signal which is sent to an ADC and a microcontroller for digitization and further processing. Once the measurement is in digital form, the microcontroller can execute tasks according to the measurement. Electromagnetic interference (EMI) can affect a measurement as it is transferred to the microcontroller. An attacker can manipulate the sensor output by intentionally inducing EMI in the wire between the sensor and the microcontroller. The nature of the analog channel between the sensor and the microcontroller means that the microcontroller cannot authenticate whether the measurement is from the sensor or the attacker. If the microcontroller includes incorrect measurements in its control decisions, it could have disastrous consequences.

We present a novel detection system for these low-level electromagnetic interference attacks. Our system is based on the idea that if the sensor is turned off, the signal read by the microcontroller should be 0V (or some other known value). We use this idea to modulate the sensor output in a way that is unpredictable to the adversary. If the microcontroller detects fluctuations in the sensor output, the attacking signal can be detected. Our proposal works with a minimal amount of extra components and is thus cheap and easy to implement.

We present the working mechanism of our detection method and prove the detection guarantee in the context of a strong attacker model. We implement our approach in order to detect adversarial EMI signals, both in a microphone system and a temperature sensor system, and we show that our detection mechanism is both effective and robust.

## I. INTRODUCTION

A sensor is an interface between the physical world and an electronic circuit, and it is the device that can convert physical quantities such as temperature, gravity, and sound into electrical signals in the form of analog voltages. Sensors are widely applied in our daily lives. For example, in our smartphones, an ambient light sensor measures light so that the brightness of the screen can be adjusted accordingly; an accelerometer can monitor motion of the smartphone, and thus the phone can track user's steps. A microphone is also a sensor that collects audio signals such as voice commands. Sensors can also be found in critical applications such as automobiles and nuclear plants. For example, a light detecting and ranging (LiDAR) sensor helps the automobile to see the surroundings, and a temperature sensor can monitor a temperature of a cooling system of a nuclear reactor. Sensors are highly integrated into

our infrastructure and modern life in general, and hence it is essential to be concerned with the security and correctness of sensor measurements.

In a sensor system, a sensor transforms a physical quantity into an analog signal which is sent to a microcontroller. Without an authentication scheme, the microcontroller has no choice but to trust the measurement. The wire that connects the sensor to the microcontroller is subject to electromagnetic interference (EMI). An attacker can use EMI to remotely, using easily available radio equipment, inject an attacking signal into the sensor system and change the sensor output, regardless of the sensor type. We cover this process in detail in Section II. As a result, the attacker can manipulate the microcontroller into believing that a measurement was obtained by the legitimate sensor. For example, an air conditioner can adjust the temperature of the air according to the room temperature. Suppose an attacker remotely sends an attacking signal to hold the sensor output at a level that corresponds to a low temperature, the air conditioner is deceived into continuously expelling hot air. As a result, the room becomes warmer and warmer. This might seem rather harmless, but a similar attack can be done to the cooling system of a nuclear power plant, or the pitch control of a fly-by-wire helicopter.

To protect a sensor system from attacks, existing defense strategies such as shielding and EMI filters have been well studied. Although shielding and EMI filters can significantly attenuate EMI, they do not fully block interference, nor do they provide the system with an ability to detect an attacking signal. In this paper we propose a novel defense method to detect an attack. Our method is based on the idea that when the sensor has its power switched off, the output of the sensor should be “quiet”. If an attacking signal is maliciously induced into the sensor system during the “quiet” period, the microcontroller can detect this.

We summarize our contributions as follows:

- We propose a novel method to detect EMI attacks by modulating the sensor power, and monitoring the output.
- We analyze the security of the detection method and prove that our method can be bypassed only with a negligible probability.
- We deploy the detection method on an off-the-shelf microphone module as well as a thermistor, to demonstrate the feasibility and robustness of discovering an attacking signal for both constant and non-constant signals.

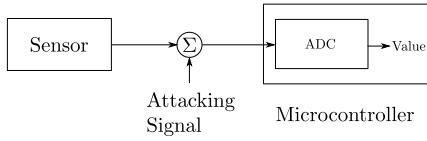


Fig. 1: A sensor system consists of a sensor and a microcontroller.

In the following sections, we first briefly present some background on EMI attacks and explain how to remotely inject a malicious signal into a sensor system in Section II. In Section III, we present an overview of our detection scheme and introduce the system and adversarial model. In Section IV, we present in detail how our defence method works and we analyze the security of the method. Then, in Section V, we show how to still maintain some security guarantee even if the measured quantity becomes non-constant (in the measuring period). Implementations of the detection method in a microphone system and a temperature sensor system are described in Section VI. We discuss a few additional points in Section VII and summarize related work in Section VIII. Finally, the whole work is concluded in Section IX.

## II. BACKGROUND ON ELECTROMAGNETIC INTERFERENCE ATTACK AGAINST SENSOR SYSTEM

In recent years, sensor systems have been widely deployed in different applications such as smart devices and automobiles. Attackers can exploit electromagnetic interference (EMI) to modify sensor readings, and such attacks may threaten users' privacy and safety. In this section, we show a general model of sensor systems, and we explain how to inject a malicious signal into the sensor system remotely.

### A. A Model of Sensor Systems

As shown in Figure 1, a sensor system consists of two essential modules: a sensor and a microcontroller. The sensor outputs a measurement to the microcontroller through a wire. An attacker can interfere with the sensor output by injecting an attacking signal into the sensor system (see details in Section II-B). When the attacking signal enters the sensor system, it is superimposed with the sensor output. The malicious sensor output is digitized by an analog-to-digital converter (ADC) in the microcontroller, and finally, an incorrect digitized sensor output is processed by the microcontroller.

### B. Injecting Malicious Signals into Sensor Systems

EMI attacks can be categorized into two types: high-power EMI attacks and low-power EMI attacks. The high-power EMI attacks refer to disruption, jamming and burning to the victim system. Sabath [22] summarizes a series of criminal uses of high-power EMI tools that result in degradation or loss of the main function of the victim's system, where technical defects, economic losses, and disasters occur. Various defense methods against the high-power EMI attacks have been studied thoroughly in previous studies [1], [2], [4], [12], [15], [18], [19], [28].

In this paper, we focus on low-power EMI attacks, in which the attacker manipulates the sensors of a victim to report the values that the attacker wishes. Examples of low-power EMI attacks can be found in prior work [10], [14], [16], [24].

To change sensor readings successfully, the attacker relies on two features of a sensor system: one is that the wire connecting the sensor and the microcontroller acts as an unintentional antenna; the other one is nonlinearity of electronic components or undersampling of an ADC. The attacker's objective is adding a malicious signal to the sensor output. The attacker generates an attacking signal by modulating a high-frequency carrier signal. This signal is picked up by the wire connecting the sensor to the microcontroller and will cause the microcontroller to read a false value [9], [11], [16], [24]. Many researchers, including [7], [8], [10], [14], [16], [21], [27], [29], [30], exploit the nonlinearity of electronic components to inject arbitrary data into sensors. This data can be amplitude, frequency or phase modulated (AM, FM, or PM) onto the carrier. By injecting a signal with a frequency that exceeds the sampling rate of the ADC, the ADC will undersample the attacking signal at a specified interval and skip high-frequency oscillations [16], [17]. This means that the ADC can be abused to work as a demodulator for the attacking signal. As a result, the malicious signal is superimposed with the legitimate sensor output.

## III. OUR APPROACH

In this section, we briefly introduce three classes of sensors on which our method is effective before explaining the core idea of our approach. The details of our defence scheme, and a careful security analysis are presented in Section IV. In this section we also, present the system- and adversarial models.

We classify sensors into three main types: active sensors, powered passive sensors, and non-powered passive sensors. An active sensor consists of an emitter and a receiver. The emitter sends out a signal to be reflected by a measured entity, and the receiver gathers information from the reflected signal. Examples of active sensors are ultrasonic sensors and infrared sensors. A powered passive sensor or a non-powered passive sensor has no emitter, and the sensor directly senses the physical phenomenon such as vibration or radiation of the measured entity. A powered passive sensor needs an external excitation signal or a power signal when it works. Examples of such sensors are microphones, light dependent resistors, and thermistors. A non-powered passive sensor does not need any external power signal. When the non-powered passive sensor is exposed to an entity that is expected to be measured, the sensor generates an output, which can be a voltage signal or a current signal. Sensors such as piezoelectric sensors, photodiodes, and thermocouples are non-powered passive sensors. Our approach modifies the way that the powered/non-powered passive sensor works; since the receiver of an active sensor is a powered/non-powered passive sensor, our approach also works for the active sensor. To simplify our exposition, in the rest of the paper, we use the powered passive sensor as an example to explain our approach. In Section VII-C, we will further illustrate how to suit our approach to the non-powered passive sensor. Unless

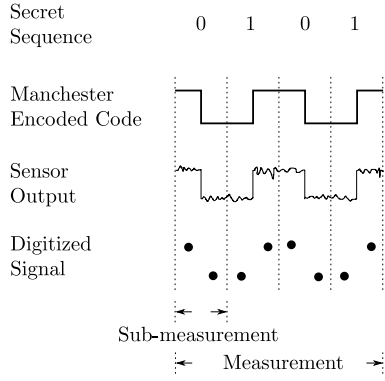


Fig. 2: An  $n$ -bit ( $n = 4$ ) secret sequence of zeros and ones is converted to a Manchester encoded code, which is toggled between a high voltage level and a low voltage level (0 V). The sensor output carries the information of the physical quantity and the noise. After digitization, a digitized signal is obtained.

otherwise stated, sensor/sensors represent powered passive sensor/sensors hereafter.

#### A. Randomized Sensor Output

Before introducing our approach, we briefly recap how an attacker can change a sensor output of a sensor system. A sensor system consists of two essential modules: a sensor and a microcontroller (see details in Section II-A). The sensor readings are transmitted to the microcontroller through a wire connecting the output of the sensor and the input of the microcontroller. Unfortunately, the wire is sensitive to electromagnetic interference (EMI), and EMI can affect the sensor system by inducing voltages on the wire. An attacker can utilize the wire to inject an attacking signal into the sensor output to change the sensor readings.

We turn the sensor on and off. Turning on means that the sensor is biased at a high voltage; turning off means that the sensor is biased at 0 V (or other known voltage levels). When the sensor is on, the sensor measures the physical quantity and the sensor output carries the information of the physical quantity. As the sensor is off, the sensor output becomes a constant signal at a specific voltage level. Suppose that the attacker injects an attacking signal to the sensor system when the sensor is off, a disturbance will appear in the flat sensor output. The microcontroller can easily detect such disturbances, and hence the attacking signal is discovered. If the sensor system can randomly turn off the sensor, the attacker has to guess when the sensor is off so that she can avoid sending an attacking signal to the sensor system; otherwise, a mistake of causing an uneven sensor output when the sensor is off will directly unveil the attacker herself to the sensor system.

We require that the microcontroller can measure the physical quantity and monitor the attacking signal by turns, and hence the sensor should be switched between the on and the off states. We use a Manchester encoded code [3] as the bias voltage for the sensor, because the Manchester encoded code toggles between a high voltage level and 0 V at the midpoint of each

clock cycle (see Figure 2). In our approach, the Manchester encoded code is encoded from an  $n$ -bit randomized secret sequence of zeros and ones. Because the secret sequence is randomized, the sensor is switched on and off randomly, and hence the sensor output has a randomized on-and-off pattern. In our approach, we assume that the physical quantity is constant (see details in Section III-B). Since the physical quantity is constant, as shown in Figure 2, the waveform of the sensor output is similar to the Manchester encoded code.

A built-in ADC digitizes the sensor output, and the microcontroller decides whether an attack occurs by checking the digitized sensor output. As shown in Figure 2, the secret sequence has  $n$  bits, and thus the Manchester encoded code has  $n$  clock cycles. Accordingly, the sensor output has  $n$  clock cycles. We define each clock cycle of the sensor output as a sub-measurement, and all  $n$  sub-measurements form a measurement. Further, each sub-measurement is digitized into two samples by the ADC: one is sampled when the sensor is biased at the high voltage, and the value of the sample is non-zero volt; the other sample is digitized when the sensor is biased at 0 V, and the value of the sample is 0 V. The microcontroller can align the digitized signal with the secret sequence precisely, and hence, given any sample, the microcontroller knows whether it should be zero or non-zero. Hereafter, based on the microcontroller's knowledge of the secret sequence, a sample that should be non-zero is called as a "non-zero sample", and a sample that should be zero is called as a "zero sample".

Under an attack, either a zero or a non-zero sample in a sub-measurement can be influenced by the attacking signal. If the attacker alters a zero sample, the microcontroller can spot the attack immediately, as the voltage level of the zero sample is not 0 V. Conversely, if the attacker alters a non-zero sample, she will also be detected quickly. This is because that the physical quantity should remain unchanged during a measurement, and all non-zero samples should be equal; however, the changed non-zero sample has a different voltage level from the other non-zero samples, and hence the attack is detected. Our detection approach are detailed in Section IV.

If the sensor system does not detect any attacking signal, the quantification of the physical quantity is the value of a non-zero sample. In practice, noise must be considered. As shown in Figure 2, since the sensor output is noisy, the non-zero samples vary slightly in a small range. Thus, the quantification is an average of all non-zero samples. To simplify the exposition, noise is ignored in Section IV and Section V. How to handle noise will be detailed in Section VI.

Note that researchers [26] have proposed a defense strategy named PyCRA, which detects sensor spoofing attacks by turning off the emitter in an active sensor. Details of the working principle of PyCRA and a comparison between our approach and PyCRA are presented in Section VII-D.

#### B. System Model

Figure 3 presents a system model of the sensor system that is equipped with our detection method. The system model consists of a sensor and a microcontroller. The sensor is driven

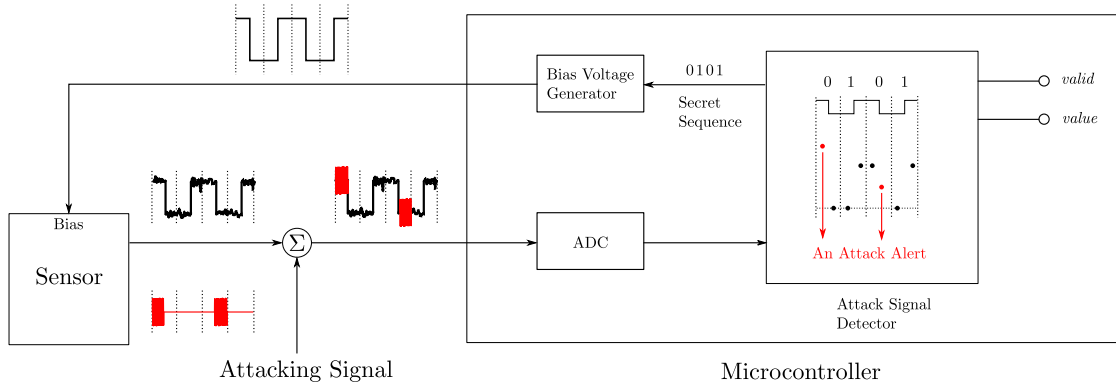


Fig. 3: A sensor system that is equipped with the detection method consists of a sensor and a microcontroller. The bias voltage of the sensor is controlled by the microcontroller. In the attack signal detector, unequal non-zero samples imply an attack. Also, a changed zero sample indicates an attack.

by a bias voltage that is controlled by the microcontroller. An output of the sensor is used to send a measurement to the microcontroller, which checks the existence of attacking signals and recovers the physical quantity from the measurement.

The microcontroller has three blocks including a bias voltage generator, an ADC, and an attack signal detector. The bias voltage generator encodes an  $n$ -bit secret sequence into a Manchester encoded code, which is the bias voltage for the sensor. The ADC digitizes the sensor output and transmits the digitized data to the attack signal detector to check whether an attacking signal exists. The attack signal detector has two outputs: *value* represents a measurement of the physical quantity; *valid* indicates whether *value* is ready to be read. If no attacking signal is detected, the measurement is assigned to *value*, and then *valid* is set to true. Hence the sensor system knows that *value* is valid to be further processed. However, if an attacking signal is detected in a measurement, *valid* is set to false throughout that measurement, which means that *value* is invalid to be read. Also, the microcontroller will be alerted that the sensor system is under an attack.

In our system model, we assume that the physical quantity remains unchanged in a measurement. Even though the physical quantity varies, if the duration of a measurement is short enough, we can also regard the physical quantity as constant. An example of a constant physical quantity is room temperature. The temperature changes slowly over a long period; however, in a short time such as 0.01 s, the temperature is unchanged.

For each measurement, the microcontroller generates  $n$ -bit secret sequence, and accordingly, the Manchester encoded code has  $n$  clock cycles. Two samples are digitized from each clock cycle or sub-measurement, and hence the sampling rate of the ADC is two times larger than the clock rate of the Manchester encoded code. In practice, the sampling rate of the ADC has an upper limit, and thus the clock rate of the Manchester encoded code also has a maximal value, which is a half of the fastest sampling rate. The shortest duration of  $n$  clock cycles is determined by the fastest sampling rate of the ADC. To apply

our detection method, it is essential to ensure that the physical quantity is unchanged within the  $n$  clock cycles.

### C. Adversarial Model

The objective of the attacker is manipulating the waveform of the sensor output without being detected by the sensor system. We suppose that the attacker cannot access the sensor system physically. Also, we assume that the attacker has no information about the  $n$ -bit secret sequence. Given any sub-measurement, we assume that the attacker knows voltage levels, but she does not know whether the voltage level transitions from the high voltage to 0 V or from 0 V to the high voltage in the midpoint of the sub-measurement (see Figure 2). Thus, the attacker has to guess the direction of the voltage level transition in each sub-measurement. Moreover, the attacker can deliberately inject a crafted signal into the sensor system, and hence the attacker can change the waveform of the sensor output as she wishes. Also, the attacker knows when the sensor module starts and stops transmitting the measurement, and she can align the crafted signal with the sensor output precisely.

## IV. ATTACK DETECTION

After receiving the digitized sensor output, the attack signal detector aligns it with the corresponding secret sequence. As shown in Figure 2, each digit in the secret sequence corresponds to two samples in the digitized sensor output. A digit 1 means that the corresponding two samples are zero and non-zero in a consecutive order; a digit 0 indicates a non-zero sample and a zero sample in a consecutive order. Thus, the microcontroller knows the order of all samples. When no attacking signal exists, the digitized sensor output satisfies two requirements:

- 1) All non-zero samples are equal.
- 2) All zero samples are zero.

Once an attack occurs, either sample in a sub-measurement can be altered. The attack signal detector first checks non-zero samples. As shown in Figure 4, if the attacker only changes several non-zero samples in the measurement, the signal formed by all non-zero samples become non-constant.

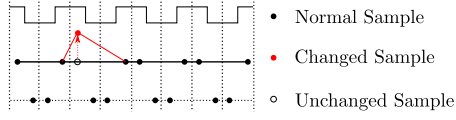


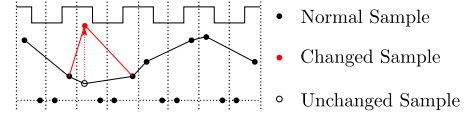
Fig. 4: A sensor output of a constant physical quantity. An attacker shifts one non-zero sample, and the signal formed by all non-zero samples becomes non-constant.

Unequal non-zero samples imply that an attack occurs. To bypass the detection, the attacker is forced to increase or decrease all non-zero samples to the same voltage level. It is possible for the attacker to make a mistake and change a zero sample. Once a zero sample is altered by the attacker accidentally, the attack will be detected.

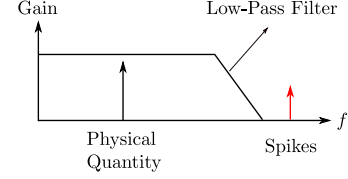
After checking the digitized sensor output, if an attack is discovered, the measurement is discarded. In contrast, if no attacking signal is detected, a quantification of the physical quantity can be obtained. As it is discussed in Section III-A, the quantification is the value of a non-zero samples; however, in practice, considering the existence of noise, it can be calculated by averaging all non-zero samples.

A smart attacker must guess whether a sample is zero or non-zero. To avoid being detected, the attacker must not affect any zero sample, and she must alter all non-zero samples to keep them the same. In Figure 3, we present an example of detecting an attacking signal in the sensor system. The attacker aims to alter the first and the third sub-measurements of the sensor output. In the first sub-measurement, the attacker makes a correct guess, and a high-frequency signal is added to the non-zero half cycle. However, in the third sub-measurement, the attacker makes a wrong guess and adds the high-frequency signal to the zero half cycle. After digitization, two samples are shifted up: the non-zero sample in the first sub-measurement and the zero sample in the third sub-measurement. Compared with other non-zero samples, the non-zero sample in the first sub-measurement has a different value, and the attack signal detector can discover the attack immediately. In the third sub-measurement, the second sample should have been zero; however, it is shifted to a non-zero value, and the microcontroller can notice the change. As a result, the attacking signal can be detected.

**Interfering with the Bias:** As described above, the detection method is used to spot attacking signals that are injected into the sensor system through the wire connecting the sensor output and the ADC. However, in practice, the wire controlling the bias of the sensor may also be an unintentional antenna. An attacking signal that is injected into this wire may alter the voltage levels of several specific periods of the Manchester encoded code. Further, the corresponding periods of the sensor output are impacted. For example, some periods that should have been at a certain voltage level are at other voltage levels; some periods that should have been 0 V are not zero. After digitizing the sensor output, the microcontroller may spot that non-zero samples are unequal and some zero samples are lifted.



(a) A sensor output of a non-constant physical quantity.



(b) A digital low-pass filter removes the spikes.

Fig. 5: The attacker alters an non-zero sample in the digitized sensor output.

Therefore, our method can also detect attacks affecting the bias. For simplicity, we only regard the wire connecting the sensor and the ADC as the injection point of an attacking signal hereafter.

#### A. Security Analysis

Only when the attacker changes all non-zero samples without influencing any zero sample, can she avoid being detected by the sensor system. In this section, we prove that the attacker can bypass our detection method with a negligible probability.

For a constant physical quantity, all non-zero samples in a measurement have the same voltage level. To avoid being detected by the sensor system, the attacker must change all non-zero samples to the same voltage level. Thus, the attacker must correctly guess the order of the zero and the non-zero samples in every sub-measurement. There are two combinations of the order of samples in a sub-measurement, and the probability of correctly guessing the order is  $\frac{1}{2}$ . Considering a measurement with  $n$  sub-measurements, the probability of correctly guessing the orders is  $\frac{1}{2^n}$ . In other words, the probability of bypassing the detection method in one measurement is  $\frac{1}{2^n}$ , which is negligible. The larger the  $n$  is, the more difficult it is for the attacker to achieve the attack.

#### V. NON-CONSTANT PHYSICAL QUANTITY

In the previous section, we describe our approach regarding constant physical quantities. However, there are physical quantities such as sounds that oscillate rapidly; even though the sampling rate of an ADC reaches the maximum, the digitized non-zero samples may have different values in a measurement. We call such a physical quantity as a non-constant physical quantity, and an example is shown in Figure 5a.

If the attacker affects either a non-zero sample or a zero sample in a constant physical quantity, our approach can detect the attack (see details in Section IV). For a non-constant physical quantity, unequal non-zero samples do not indicate an attack anymore. This means that, if the attacker plans to alter one sample only, she can bypass the detection with a probability of  $\frac{1}{2}$ . For example, as shown in Figure 5a, the

attacker wants to affect the third clock cycle: if she changes the non-zero sample, she succeeds; otherwise, changing the zero sample still leads to an alert of the attack. Compared with the detection method for a constant physical quantity, the one for the non-constant source gives a weak security guarantee. In order to achieve a strong security guarantee, the sampling rate of the ADC must be large enough so that the physical quantity can be regarded as constant, and thus the approach for a constant source applies.

However, in practice, a sensor system may have to handle non-constant scenarios subject to multiple limitations (e.g., sampling rates of ADCs). Then, it is necessary to revise the approach for non-constant physical quantities to detect attacks affecting either non-zero or zero samples. In this section, we describe the revised method. Also, we show that the negative impacts that are caused by attacking signals can be mitigated. We analyze the security of our detection method. Finally, we discuss an additional requirement for the ADC.

#### A. Attack Detection for Non-constant Physical Quantities

An attacker can change any numbers of non-zero samples. Without loss of generality, we assume that the attacker plans to change  $k$  ( $1 \leq k \leq n$ ) out of  $n$  samples. She can achieve the modification without being detected with a probability of  $\frac{1}{2^k}$  (see details in Section V-B). When a few samples are changed, as shown in Figure 5a, the modified sample leads to a spike in the measured signal. Without knowing any information about the measured signal, we can do nothing to detect the change. However, if we know concrete characteristics that can describe the behavior of the non-constant signal, we can recognize modified samples as outliers. As depicted in Figure 5b, if we know the bandwidth of the measured signal, we can recognize the sample that causes a spike beyond the band as an outlier. Moreover, if we have a model of the measured signal, we can recognize the sample that fails to fit the model as an outlier. Despite that a few modified samples form spikes in the measured signal, the major information of the physical quantity may be still retained. For example, regarding an audio signal, a spike in the measured signal sounds like a chirp; however, a listener can still understand the information that is conveyed in the audio signal. A digital low-pass filter can be used to filter out the spike so that the negative impacts can be mitigated.

If the attacker changes many samples, the modified samples dominate, and she may bypass the detection of outliers. However, the probability of avoiding affecting zero samples is  $\frac{1}{2^k}$ , which exponentially decreases with the number of samples that the attacker wishes to change. Therefore, changing more samples increases the difficulty of bypassing the detection.

#### B. Security Analysis

We have assumed that the attacker plans to change  $k$  ( $1 \leq k \leq n$ ) out of  $n$  non-zero samples. When  $k = n$ , the probability of bypassing the detection method is the same as the one for a constant physical quantity. When  $1 \leq k < n$ , the attacker needs to guess the orders of samples in  $k$  sub-measurements. The probability of bypassing the detection

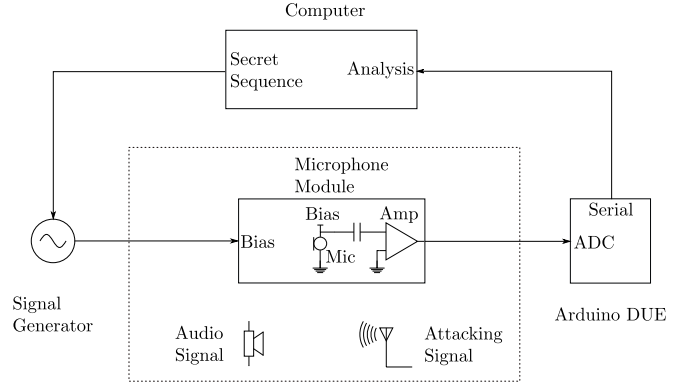


Fig. 6: A testbed is built to test a microphone system. A signal generator, which is controlled by a computer, provides the microphone module with a bias voltage. An Arduino DUE is used to collect the signal from the microphone module. The computer is used to analyze the signal.

method is  $\frac{1}{2^k}$ , which is negligible. If  $k$  is small, the attacker can easily achieve an attack, but the impacts of the modified samples are small; while  $k$  is large, it is difficult for the attacker to bypass the detection method.

#### C. The Sampling Rate of the ADC

To ensure that the measurement contains complete information of the physical quantity, according to the Nyquist-Shannon sampling theorem, the clock rate of the Manchester encoded code should be at least twice larger than the bandwidth of the non-constant physical quantity. Since the sampling rate of the ADC is twice larger than the clock rate of Manchester encoded code, the sampling rate is at least four times larger than the bandwidth of the physical quantity.

### VI. IMPLEMENTATION

In this section, we implement our approach on two sensor systems: a microphone system (see Section VI-A) and a temperature sensor system (see Section VI-B). In each sensor system, we first show how an attacker can remotely modify sensor readings by EMI, and then we present the effectiveness and robustness of our detection method.

#### A. Microphone System

A microphone can convert sound into an electrical signal. At present, microphones can be found in many different devices such as smartphones, headphones, and laptops. In a microphone system, a wire is used to connect a microphone module and a microcontroller, and hence the attacker can exploit the wire to inject an attacking signal into the microphone system. For example, an attacker can inject voice commands into a smartphone through EMI, and the voice assistant system can be asked to execute malicious tasks in the smartphone. Note that human cannot hear any EMI, and hence the user cannot notice the attacking signal.



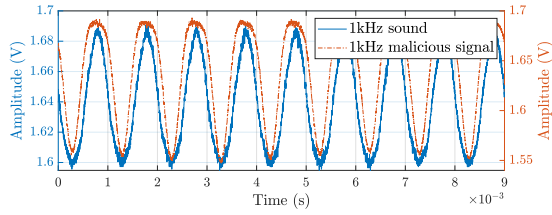


Fig. 7: One 1 kHz signal is the sound, and the other 1 kHz signal is from the attacker, who injects the 1 kHz malicious signal into the microphone system by EMI. The similarity of these two signals is above 0.93.

1) *Setup*: In Figure 6, the setup of the microphone system is presented. The microphone system consists of a computer, a signal generator, an off-the-shelf microphone module, and an Arduino DUE. The computer controls a RIGOL DG4062 signal generator to generate a bias voltage for the microphone. The microphone converts the sound into a voltage signal, which is further amplified by the amplifier. The output of the amplifier is biased at 1.65 V. Then, the output of the microphone module is digitized by a built-in ADC in the Arduino DUE at a sampling rate of 666.8 kHz. Next, the Arduino DUE sends the digitized data to the computer through a serial port. Finally, we can use the computer to analyze the digitized signal.

Note that the sampling rate we choose is higher than the minimum theoretical sampling rate required. According to Section V-C, the sampling frequency should be at least four times larger than the bandwidth of the physical quantity. Since the microphone in our experiment can measure up to 20 kHz, the sampling frequency is 80 kHz in theory. However, in practice, we need to consider samples that are digitized from signal edges, and hence the sampling rate is higher than the theoretical one. Details are discussed in Section VI-A3.

There are two signal sources: one is a legitimate sound from a speaker of a Motorola XT1541 Moto G3 smartphone, and the other is an attacking signal from the attacker. The attacker uses an R&S SMC 100A signal generator to amplitude-modulates a malicious signal on a 144 MHz carrier signal to form the attacking signal. Then, the attacking signal is radiated through a 144 MHz omnidirectional vertical antenna. The reason why 144 MHz is chosen as the carrier frequency of the attacking signal is that, by experiment, the 144 MHz signal can be received by the unintentional antenna in the microphone module effectively. Both the antenna and the speaker are placed 10 cm away from the microphone module.

2) *Without the Detection Method*: Without the detection method, the microphone system cannot determine whether the signal is legitimate or malicious. In the following parts, we will show that the attacker can remotely inject a malicious signal that is similar to the audio signal into the microphone system.

The signal generator is configured to output a constant 300 mV signal, and thus the microphone is biased at 300 mV. We first play a 1 kHz audio signal through the speaker of the mobile phone at the maximal volume. Next, we turned off

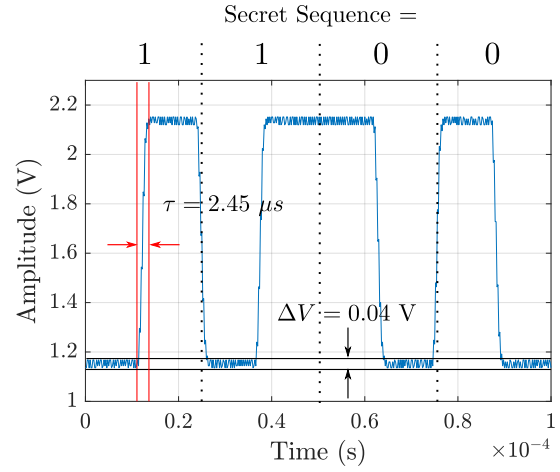


Fig. 8: Measure the bound of zero samples and the time of the signal edges by an oscilloscope with a sampling frequency of 2 GHz.

the speaker, and an attacking signal, which is generated by modulating the 1 kHz malicious signal on a 144 MHz carrier signal, is emitted through the antenna at  $-5$  dBm. The attacking signal is demodulated by the nonlinear electronic components (e.g., amplifiers and ADCs) in the microphone system, and a 1 kHz digitized malicious signal is obtained.

In Figure 7, two 1 kHz signals that are reconstructed by the computer are presented: one is the signal from the speaker; the other one is induced by the attacker. It can be observed that, without our detection method, it is difficult to tell whether a received signal is from the speaker or the attacker: both the sound and the malicious signal are 1 kHz, and they have similar amplitudes. It is known that Pearson's correlation coefficient (PCC) can be used to measure the linear correlation of two signal [5], [23], and PCC is a suitable metric to show the similarity of two signals in our experiments. The PCC of the 1 kHz audio signal and the 1 kHz malicious signal is above 0.93, which means that these two signals have a high similarity. Above all, the attacker can control the output of the microphone system and deceive the microcontroller.

3) *Applying the Detection Method*: From the experimental results above, the microphone system may regard the malicious signal as the legitimate audio signal. In this part, we illustrate how to deploy the detection method to the microphone system to detect the attacking signal.

When the detection method is applied to the microphone system, the computer repeatedly transmits a secret sequence of [1100] to the signal generator, and the signal generator encodes the secret sequence into a Manchester encoded code with a clock rate of 40 kHz. The Manchester encoded code toggles between 0 mV and 300 mV. Note that the bias voltage is for the microphone, which is denoted as "Mic" in Figure 6, instead of

the amplifier<sup>1</sup>. In Figure 8, without any audio signal or attacking signal, we present the output of the microphone module that is captured by a RIGOL DS2302A Digital Oscilloscope, which has a sampling frequency of 2 GHz.

When the computer receives the digitized signal from the Arduino DUE, three practical challenges in the microphone system need to be considered before checking the existence of an attack. The first challenge is synchronizing the digitized signal with the secret sequence. Each digit in the secret sequence corresponds to one sub-measurement, and the value of the digit decides the direction of the voltage level transition at the midpoint of the sub-measurement. Only if the digitized signal is aligned with the secret sequence precisely will the computer know whether a specific sample is zero or non-zero. In practice, we configure the signal generator so that there is always a voltage level transition from high to low at the beginning of the first sub-measurement so that we can identify the start point of the digitized signal. Further, it is easy to align the digitized signal with the secret sequence.

Another practical challenge is how to handle samples from the rising or the falling edges of the output of the microphone module. The samples from the edge can lead to a false positive alert of attack or an inaccurate measurement of the physical quantity. As shown in Figure 8, the time of the signal edge is  $\tau = 2.45 \mu\text{s}$ . The sampling period of the ADC is  $\frac{1}{f_s} = \frac{1}{666.8 \text{ kHz}} \approx 1.50 \mu\text{s}$ , and hence at most two samples emerge from the signal edge. Also, given the sampling rate and the clock rate, we can find that there are 16 samples in each sub-measurement. Thus, to eliminate the negative impacts of the edge samples, we remove the first and the last samples in each half cycle.

The third practical challenge is determining the voltage level of zero samples. Because the output of the microphone module is centered at 1.65 V, the zero samples are shifted to a non-zero level. As shown in Figure 8, the mean value of the zero samples is 1.15 V. However, it can be observed that the zero samples fluctuate around 1.15 V, and the range of the fluctuation is  $\Delta V = 0.04 \text{ V}$ . Note that  $\Delta V$  is also the noise tolerance of zero samples. When there is no attacking signal, the zero samples are within a range of  $[1.15 \text{ V} - \frac{1}{2}\Delta V, 1.15 \text{ V} + \frac{1}{2}\Delta V] = [1.13, 1.17] \text{ V}$ . If a zero sample is outside  $[1.13, 1.17] \text{ V}$ , the microphone system will be alerted with an attack.

After obtaining a measurement from the microphone module, the computer synchronizes the corresponding secret sequence with the measurement, and removes samples from edges. According to the bounds of zero samples, which is  $[1.13, 1.17] \text{ V}$ , the computer can determine whether an attack occurs in the measurement. To evaluate the performance of our detection method, we consider the following three cases:

**Case 1:** A 1 kHz audio signal is played from the speaker at its maximal volume, and there is no attacking signal. In

<sup>1</sup>If the Manchester encoded code is used to bias the amplifier, when the amplifier is off, an attacking signal that is injected before the amplifier does not affect the output of the amplifier. This means that attacks that affect zero samples cannot be detected.

TABLE I: Detection results of Case 2 and 3.

Case No.	Sound	Attacking Signal (modulating signal, carrier)	True-positive Rate
2	-	(1 kHz, 144 MHz)	100%
3	1 kHz	(5 kHz, 144 MHz)	100%

Figure 9a, the amplitude envelope that is formed by non-zero samples of the digitized sequence represents the 1 kHz component. Since no attacking signal exists, this case is a reference for the following two cases.

**Case 2:** Turn off the speaker, and the attacker transmits an attacking signal at  $-5 \text{ dBm}$ . To inject a 1 kHz signal into the microphone system, the attacking signal is generated by modulating the 1 kHz signal on a 144 MHz carrier. As Figure 9b shows, it can be noticed that both zero and non-zero samples carry the information of the 1 kHz signal.

**Case 3:** Turn on the speaker, and the attacker radiates an attacking signal at the same time. The frequency of the audio signal is still 1 kHz, and volume is unchanged. To insert a 5 kHz signal into the system, the attacker modulates the 5 kHz signal on a 144 MHz carrier, and the transmission power of the attacking signal is  $0 \text{ dBm}$ . As it is shown in Figure 9c, the 5 kHz signal dominates in both zero and non-zero samples.

In each case, 100 measurements are recorded. Because the physical quantity is non-constant in a measurement, we use our detection criteria of non-constant physical quantity to check whether an attacking signal exists in each measurement. Accordingly, in Case 2 and Case 3, we can calculate the true-positive rate of detecting the attacking signal. The detection results are presented in Table I. In Case 2 and Case 3, the computer finds that some zero samples are outside the bounds, and thus the attacking signal can be detected. The true-positive rates of detecting the attack are 100% in both Case 2 and Case 3. The results mean that the attacking signals exist in every measurement in these two cases.

Our experiments also show that, when there is no attacking signal (Case 1), all zero samples are within the bounds, and our detection method does not give any false positive alarm of an attack. Once the attacker accidentally increases or decreases the value of the zero sample to a value that is outside the bounds (e.g., Case 2 and 3), the attack is detected immediately.

Note that, in Case 2 and 3, the attacker initiates “dumb” attacks, which mean that the attacker does not guess when the sensor is on or off. In other words, the dumb attacking signal affects every sample in the measurement. This is the reason why the true-positive rate is 100% for these two cases. In practice, it is difficult to conduct “smart” attacks that allow the attacker to do the guessing and align the attacking signal with the sensor output. In the experiment of a temperature sensor system in Section VI-B, smart attacks are simulated from real sensor data.

**4) Signal Reconstruction:** When no attack is detected, the final step is to recover the physical quantity. Because measurements in Case 2 and 3 are detected with attacking signals, we cannot recover the physical quantity from these two cases. In Case 1, no attacking signal is detected, and we can



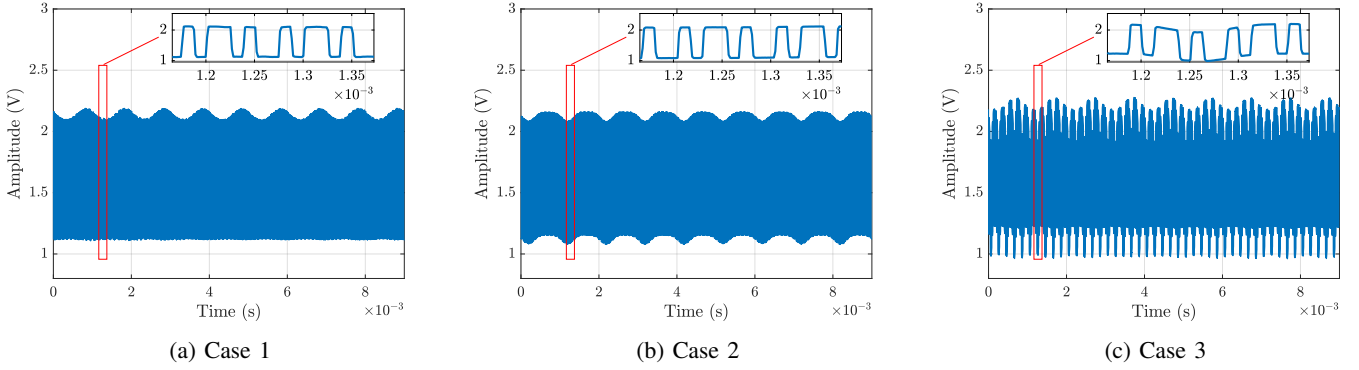


Fig. 9: When detection method is applied, (a) the speaker plays a 1 kHz tone; (b) the attacker transmits an attacking signal, which is generated by modulating 1 kHz signal on a 144 MHz carrier signal at the power of  $-5$  dBm; (c) the attacker transmits an attacking that is generated by modulating a 5 kHz signal on a 144 MHz carrier signal at a transmission power of 0 dBm, and the speaker plays 1 kHz tone at the same time.

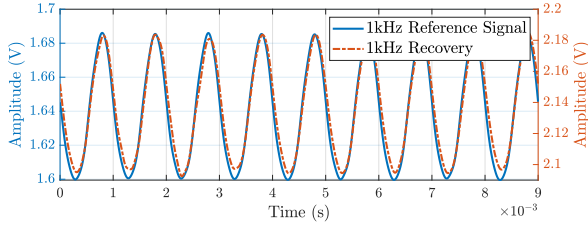


Fig. 10: Remove zero samples and edge samples to reconstruct the 1 kHz audio signal. As a comparison, the 1 kHz reference signal is presented.

recover the 1 kHz signal by excluding zero samples and edge samples in the measurement. Then, we use a digital second-order Butterworth low-pass filter with a cut-off frequency of 5 kHz to get rid of high-frequency components in the digitized signal. The recovered 1 kHz signal is shown in Figure 10. As a comparison, we also digitize 1 kHz audio signal with the same ADC as a reference signal, and it is filtered by the same low-pass filter. The reference signal is depicted in Figure 10.

We analyze the quality of the recovered signal in two aspects: similarity and Signal-to-Noise Ratio (SNR). As discussed in Section VI-A2, PCC can be used to measure the similarity between two signals. We calculate the PCCs between 100 recovered signals and the reference audio signal. The averaged PCC in Case 1 is above 0.99, which implies that the recovered signal is similar to the audio signal in the time domain. The averaged SNR of all 100 recovered signals in Case 1 is  $30.6 \text{ dB} \pm 0.1 \text{ dB}$  at a 99% confidence level; the SNR of the reference signal is 29.9 dB. It can be concluded that the recovered signal has a equivalent quality as the reference signal.

### B. Temperature Sensor System

We use a thermistor to build a temperature sensor system. The thermistor is a resistor that varies its resistance according to temperature. In our experiment, we choose a thermistor with

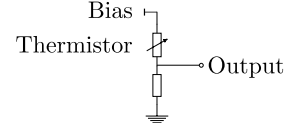


Fig. 11: A thermistor circuit is a voltage divider. When the temperature increases, the output voltage of the circuit increases accordingly.

a negative temperature coefficient (NTC), which means that the resistance of the thermistor increases with decreasing the temperature. In order to present experimental results properly, we define that the temperature measuring range is from  $0.0^\circ\text{C}$  to  $50.0^\circ\text{C}$ , which is within the allowable measuring range of the thermistor.

In the following sections, we first introduce the setup of the temperature sensor system. Then, we demonstrate how an attacking signal affects a sensor reading. Finally, we show that our detection method can detect the attacking signal.

1) *Setup*: In Figure 11, we present a diagram of a thermistor circuit. The thermistor circuit is a voltage divider, which is formed by connecting an NTC thermistor and a resistor in series. The output voltage of the thermistor circuit increases with increasing the temperature. We test the thermistor circuit using the setup shown in Figure 6, and we replace the microphone module with the thermistor circuit. This setup is placed in a laboratory with a constant temperature at around  $25.0^\circ\text{C}$ . Since the room temperature can be regarded as a constant physical quantity, digitized samples that should be non-zero are supposed to be approximately equal. The sampling rate is set to 284 Hz, which is much lower than the one in the microphone system.

The attacking signal has a frequency of 144 MHz, and it is radiated from a 144 MHz omnidirectional antenna. The antenna is placed 1 cm away from the thermistor circuit. Note that the distance between the antenna and the thermistor circuit is small

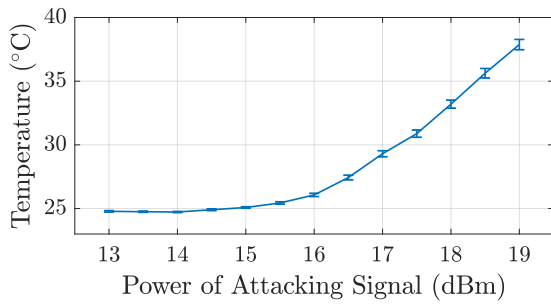


Fig. 12: The power of attacking signal is increased from 13 dBm to 19 dBm with a step of 0.5 dBm. Under the attack, the temperature is changed from 24.9 °C to 37.9 °C.

because we want to realize the remote injection with a low power of the R&S SMC 100A signal generator.

2) *Without Detection Method:* The thermistor circuit is biased at 1 V. When no attacking signal is radiated, the temperature sensor system outputs 24.9 °C ± 0.1 °C at a 99% confidence level.

Next, the attacker radiates an attacking signal, and the power of the attacking signal is increased from 13 dBm to 19 dBm with a step of 0.5 dBm. For each power level, 100 temperature measurements are recorded. We calculate the 99% confidence interval around the mean of the 100 measurements, and results are presented in Figure 12. Below 14 dBm, the attacking signal has no significant effect on the temperature measurement. When the power of the attacking signal is increased above 14 dBm, the temperature measurements increase. The 19 dBm attacking signal results in a temperature measurement of 37.9 °C ± 0.4 °C, which is approximately 13 °C higher than the true room temperature of 24.9 °C ± 0.1 °C. The curve in Figure 12 shows that the attacker can change the temperature reading of the sensor to any values as she wishes. Without any detection method, the temperature sensor system cannot detect the existence of the attacking signal.

3) *Applying Detection Method:* The secret sequence we use is also [1100], and the clock rate of the Manchester code is set to 20 Hz. We use oscilloscope to measure the time of signal edge, and the width of signal edge is around 2 ms. Regarding that the sampling period is  $\frac{1}{284 \text{ Hz}} = 3.5 \text{ ms}$ , at most one sample is digitized from signal edges. In order to eliminate the negative influence caused by samples from signal edges, the first and the last sample in each half clock cycle are abandoned.

We use the oscilloscope to measure the bound of non-zero samples, which is 0.03 V; the bound of zero samples has the same value. When no attacking signal is radiated, fluctuations of non-zero samples are within 0.03 V; note that zero samples swing between 0 V and  $\frac{1}{2} \times 0.03 \text{ V} = 0.015 \text{ V}$ , as the ADC in the microcontroller can only read positive voltages. Because the room temperature is a constant physical quantity, we can concrete the requirements as follows:

- The standard deviation of all non-zero samples is smaller than or equal to  $\frac{1}{2} \times 0.03 \text{ V} = 0.015 \text{ V}$ .
- All zero samples are within [0, 0.015] V.

In the following parts, a reference case (Case 1) is presented, in which no attacking signal exists. A dumb attack (Case 2) is conducted on the temperature sensor system, and then a smart attack (Case 3) is simulated from data that are collected from Case 1 and 2. In the following parts, the thermistor circuit's voltage outputs are converted into temperature. Note that when the bias voltage is 0 V, the output is also 0 V. Since 0 V corresponds to a temperature that is beyond the measurement range of the thermistor circuit, this temperature is denoted as  $T_{ref}$  (see Figure 13).

**Case 1:** No attacking signal is radiated from the antenna, and the microcontroller records the output of the thermistor circuit. In Figure 13a, a measurement is presented. The measured temperature is 25.5 °C ± 0.1 °C.

**Case 2:** In order to change the sensor reading to a significant high temperature, the antenna radiates an attacking signal with a power of 19 dBm. The microcontroller records the output of the thermistor circuit. A measurement is shown in Figure 13b. Note that such an attack is a dumb attack, as the attacker radiates the attacking signal continuously. The mean of the non-zero samples corresponds to a temperature of 38.3 °C ± 0.1 °C, which is around 13 °C higher than the true room temperature. The zero samples are lifted to 27.4 °C ± 0.1 °C, which indicates an attack.

**Case 3:** (A simulation of a smart attack) The attacker has a fair coin that has a probability of 50% showing a head and 50% showing a tail every time it is tossed. The attacker selects a measurement from Case 1, and each measurement contains 4 clock cycles or 8 half clock cycles (see Figure 13a). For each clock cycle, the attacker tosses the coin to decide whether to send an attacking signal. A head means that the attacker radiates an attacking signal in the first half cycle and remains silent in the second half cycle. Accordingly, the first half cycle is replaced by a half cycle that corresponds to 38.3 °C ± 0.1 °C from Case 2. Conversely, a tail means that the attacker remains silent in the first half cycle and radiates an attacking signal in the second half cycle. Accordingly, the second half cycle is replaced by a half cycle that is 27.4 °C ± 0.1 °C from Case 2. After tossing the coin for all four clock cycles, we have a new measurement that is affected by a smart attack (see Figure 13c).

As shown in Figure 13c, except for the third clock cycle, the attacker's guesses in the other three clock cycles are correct. The attacker accidentally radiates the attacking signal during the second half cycle of the third cycle: the temperature of that half cycle is enhanced from  $T_{ref}$  to 27.4 °C. After digitization, non-zero samples form a non-constant signal, and thus an attack can be detected. Also, since samples that should be  $T_{ref}$  in the third clock cycle are lifted, the attack is alarmed.

In each case, 100 measurements are recorded. In Case 2, the true-positive rate is 100%, which implies that an attacking signal is detected in each measurement. Also, we repeat the simulation of smart attacks 100 times, and the true-positive rate is 93%. In theory, since the number of digits of the secret is four, the attacker has a probability of  $\frac{1}{2^4}$  guessing the secret of each measurement correctly. Among 100 measurements, the

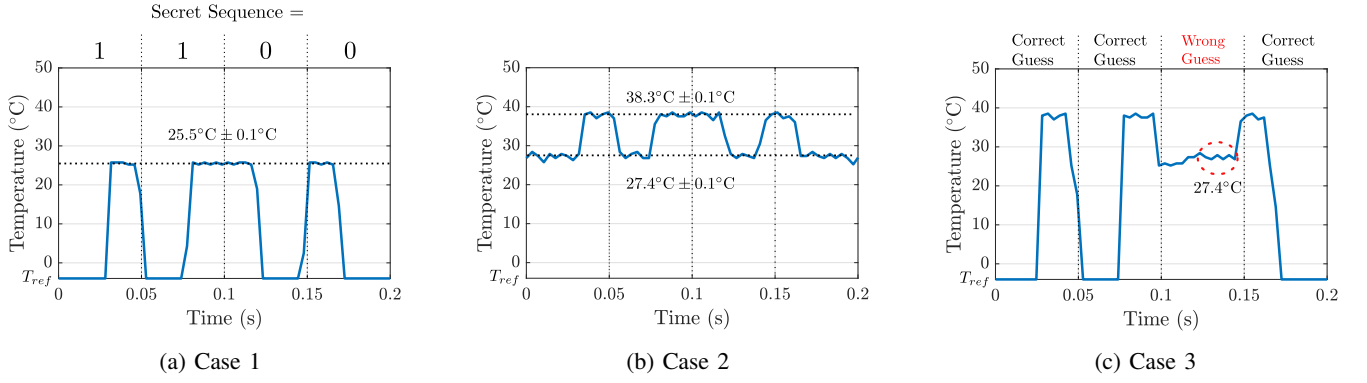


Fig. 13: Our detection method is deployed to the temperature sensor system, and the outputs of the thermistor circuit are presented. In (a), no attacking signal exists, and the non-zero samples are approximately equal, which indicates a temperature of 25.5°C. In (b), a dumb attacking signal is radiated, and the non-zero samples indicate a room temperature of 38.3°C, and the zero samples corresponds to a temperature of 27.4°C. In (c), a smart attack is simulated, and a wrong guess is made in the third clock cycle.

expectation of correct guesses is  $\frac{100}{2^4}$ . Therefore, the theoretical true-positive rate is  $1 - \frac{100}{2^4}/100 = 93.75\%$ . The real true-positive rate is approximately equal to the theoretical one.

## VII. DISCUSSION

### A. Guaranteeing the Security with Small $n$ for Constant Physical Quantities

In Section IV-A, we have discussed that increasing the length of the secret sequence  $n$  leads to increasing the difficulty of bypassing the detection method. A larger  $n$  results in a more secure system. Given a fixed duration of a measurement, a larger  $n$  requires a faster sampling rate of the ADC. Because of the hardware limitations, the sampling rate has an upper limit, and thus  $n$  also has a maximal value. Although the sampling rate reaches the highest, it is possible that  $n$  is a small number (e.g.,  $n = 8$ ). However, in our detection method, a small  $n$  can also guarantee the security of the sensor system.

For each measurement, the number of combinations of  $n$ -bit secret sequence is  $2^n$ , and the attacker can find the correct secret sequence to bypass the detection method by trying all combinations. However, in practice, it is impossible for the attacker to try  $2^n$  times, and the attacker has only one chance to change the measurement. The probability of successfully attacking the measurement without being detected is  $\frac{1}{2^n}$ , and this means that the expected number of successful attacks in attacking  $2^n$  measurements is only one. In the other  $2^n - 1$  measurements, the attacking signal is discovered by the sensor system. Imagine that the microcontroller receives  $2^n - 1$  invalid measurements before one valid measurement. Because the  $2^n - 1$  invalid measurements imply that the sensor system is currently under an attack, the valid measurement is still untrustworthy, and hence the microcontroller rejects to further processing the valid measurement.

In general, we suggest using a large  $n$  (e.g.,  $n = 128$ ) to guarantee the security of the sensor system. However, limited by

the sampling rate, although a substantial  $n$  may be impractical, a relatively small  $n$  is still effective to prevent an attacker from bypassing the detection method, and further, the security of the sensor system is guaranteed.

### B. Trade-off between Security and Speed

In some applications, the sampling rate of an ADC is fixed. To increase the security, we can lengthen the duration of one measurement, and thus more sub-measurements are included. If the physical quantity keeps constant after lengthening the measurement, the number of sub-measurements that the attacker must change increases. As a result, it is more difficult for the attacker to change all sub-measurements without being detected. For non-constant physical quantities, to change the waveform of the sensor output effectively, the attacker has to alter more sub-measurements after lengthening the measurement. Consequently, the difficulty of bypassing the detection method also increases. Above all, without changing the sampling rate of the ADC, the security of the sensor system can be further improved at the cost of lengthening the measurement.

In summary, to achieve a more secure sensor system, we have to sacrifice the speed, which is either the speed of sampling or the speed of obtaining a measurement. In real applications, designers need to consider the constraints of their sensor systems to choose the proper option to enhance the security.

### C. Our Approach for Non-Powered Passive Sensors

In order to deploy our approach to a sensor system with a non-powered passive sensor, a switch can be added between the sensor output and the ADC. Figure 14 depicts a configuration for the non-powered passive sensor. The microcontroller can “turn on” and “turn off” the sensor by controlling the switch. When the microcontroller “turns on” the sensor, the switch connects the sensor output and the ADC; thus, the microcontroller can read the sensor output. When the microcontroller “turns off” the sensor, the switch disconnects the ADC from the sensor

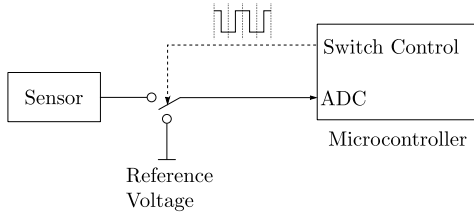


Fig. 14: A switch that is controlled by the microcontroller is added between the output of the sensor and the ADC.

output. When the ADC is disconnected from the sensor output, in order to ensure that inputs to the ADC settle at a specific level, the ADC should be connected to a reference voltage.

Note that the switch must be installed very close to the sensor output. The wire between the switch and the ADC must act as an unintentional antenna so that the security is the same as powered passive sensors. Otherwise, if the wire between the switch and the sensor output is long enough, this wire may work as an unintentional antenna, which is also an injection point. When the ADC is disconnected from the sensor output, the readings of the ADC will not be affected by the attacking signal. Since the zero samples will not be affected by the attacking signal, no attack will be detected.

#### D. Difference between PyCRA and Our Approach

Shoukry et al. [26] proposed a generalizable sensor spoofing detection method named PyCRA for sensors such as ultrasonic sensors and infrared sensors, which consist of emitters and receivers. As described in Section III, the emitter sends a challenge signal to the measured entity, and the receiver gather information from the reflected signal. In a spoofing attack, an attacker manipulates the reflected signal. To detect such attacks, PyCRA turns off the emitter randomly, and hence the receiver should receive nothing during the shutdown of the emitter; if a reflected signal is received when the emitter is off, an attack is detected.

Our approach differs from PyCRA in the following aspects. In this paper, we show that our approach works for powered/non-powered passive sensors. Because the powered/non-powered passive sensors are the receivers of active sensors, our approach also applies to the active sensors. Hence, our approach is applicable to all three types of sensors that we define in Section III. In PyCRA, since an emitter is necessary, this method is designed for active sensors only. Thus, our approach outperforms PyCRA as our approach covers two more types of sensors.

PyCRA counts on the secrecy of the timing of voltage level changes in the challenge signal. In PyCRA, for an attacker in real life, there is a non-zero physical delay between capturing the challenge signal and radiating an attacking signal. This means that the attacker cannot align the attacking signal with the reflected signal. Researchers [25] showed that PyCRA could be entirely bypassed: suppose that the attacker has a faster sampling rate than the sensor system, when the challenge signal starts falling, the attacker can quickly spot the change and stop

generating attacking signals. Because the attacker does not influence the periods that are used to detect attacks, she will not be noticed by PyCRA. However, such an attacker cannot bypass our detection method. In our scenario, the attacker has full information of the timing as it is assumed in Section III-C. In other words, our approach allows the attacker to precisely align the attacking signal with the legitimate sensor output. Even so, the attacker still must guess whether the sensor turns on or off, and a wrong guess will expose the attacker herself to the sensor system.

Regarding the threat model, in our approach, the attacker can stay far away from the sensor system, as the attacker uses EMI to remotely interfere with the sensor readings. In PyCRA, the attacker must stay in a specific area near the sensor system and the measured entity so that she can capture the challenge signal and produce a malicious reflected signal. Therefore, our approach has a stronger threat model.

For the working principle, our method detects attacks by examining both non-zero and zero samples; however, PyCRA monitors attacking signals by checking zero samples only. In other words, PyCRA cannot recognize attacks affecting non-zero samples.

## VIII. RELATED WORK

Recent work on the defense methods against the low-power EMI attacks can be classified into three categories: hardware methods, software methods, and hybrid methods. The hardware methods use specific materials or electronic components to mitigate attacking signals. There are several common strategies such as shielding, differential comparators, and filters. Regarding shielding, specific materials are used to dampen the received electromagnetic radiation. Shielding is recommended in previous studies [14], [16], [20]. Additionally, we can use a differential comparator to remove the common mode interference in the sensor signal and ground, and thus the attacking signal can be mitigated. Also, a low-pass filter can attenuate the signal outside the sensor's baseband, and hence the attacking signal at high-frequency band can be filtered out. However, researchers [13] presented that the parasitics in surface mount components convert the low-pass filter into a band-stop filter, which allows the attacking signal to pass. In order to solve this problem, an alternative is using an EMI filter. Although these methods effectively attenuate attacking signals, they do not have the function of detection.

The software methods detect or attenuate the attacking signal by examining the measurement at a software level. The microcontroller knows the model of the measurement, and anomalies found in the measurement may imply the existence of an attacking signal.

A hybrid method is a combination of hardware methods and software methods. In hybrid methods, microcontrollers handle attacking signals through specific channels. Researchers [16] proposed that a specialized component in the victim device could be chosen to capture the attacking signal. The captured attacking signal can be an input to an adaptive noise canceling system (which can be realized in software), and

hence the victim system can attenuate the attacking signal in the sensor signal. Also, they showed that the microcontroller in a cardiac implantable electrical device (CIED) could use its direct connection to the cardiac tissue to discern between a measured signal and an induced signal. Fujimoto et al. [6] proposed a detection method against the attacking signal in the cryptographic integrated circuit by monitoring the built-in voltage variation of the power supply using the on-chip voltmeter. These hybrid methods are devised for specific applications. In other words, they are not universal for different devices. However, our approach is designed for sensor systems that match our model, and it can be quickly deployed to the sensor systems.

## IX. CONCLUSION

In this paper, we propose a novel method to detect EMI attacks for sensor systems that match our model. In our detection method, a sensor system turns off the sensor to monitor the attacking signal in the sensor output. Our detection method can prevent the sensor system from processing an attacking signal: once the microcontroller detects an existence of an attacking signal, the microcontroller rejects to handling the sensor output further. Compared with other detection methods, our approach is not only low-cost and space-saving but also can be quickly deployed.

Regarding the security of the sensor system, we proved that our detection method can be bypassed with a negligible probability. The security of the sensor system is based on that the  $n$ -bit secret sequence is unknown to the attacker. The longer the secret sequence is, the more secure the sensor system is. Also, our detection method can guarantee the security with a small  $n$ .

In practice, we deploy the detection method to a microphone system and a temperature sensor system. The high true-positive rates show that our detection method is effective and robust in detecting the attacking signal.

## ACKNOWLEDGMENTS

This work was supported in part by China Scholarship Council – University of Oxford Scholarships.

## REFERENCES

- [1] C. Adami, C. Braun, P. Clemens, M. Joester, S. Ruge, M. Suhrke, H. Schmidt, and H. Taenzer, "HPM detector system with frequency identification," in *2014 International Symposium on Electromagnetic Compatibility (EMC Europe)*. IEEE, 2014, pp. 140–145.
- [2] C. Adami, C. Braun, P. Clemens, M. Suhrke, H. Schmidt, and A. Taenzer, "HPM detection system for mobile and stationary use," in *EMC Europe 2011 York*. IEEE, 2011, pp. 1–6.
- [3] *Manchester Coding Basics*, Atmel Corporation, 2325 Orchard Parkway, San Jose, CA 95131, USA, sep 2009.
- [4] C. E. Baum and D. McLemore, "Damping Transmission-Line and Cavity Resonances," *Interaction Note*, vol. 503, pp. 239–244, 1994.
- [5] J. Benesty, J. Chen, and Y. Huang, "On the importance of the Pearson correlation coefficient in noise reduction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 4, pp. 757–765, 2008.
- [6] D. Fujimoto, Y.-i. Hayashi, A. Beckers, J. Balasch, B. Gierlichs, and I. Verbauwhede, "Detection of IEMI fault injection using voltage monitor constructed with fully digital circuit," in *2018 IEEE International Symposium on Electromagnetic Compatibility and 2018 IEEE Asia-Pacific Symposium on Electromagnetic Compatibility (EMC/APEMC)*. IEEE, 2018, pp. 753–755.
- [7] J. Gago, J. Balcells, D. González, M. Lamich, J. Mon, and A. Santolaria, "EMI susceptibility model of signal conditioning circuits based on operational amplifiers," *IEEE Transactions on Electromagnetic Compatibility*, vol. 49, no. 4, pp. 849–859, 2007.
- [8] H. Ghadamabadi, J. J. Whalen, R. Coslick, C. Hung, T. Johnson, W. Sitzman, and J. Stevens, "Comparison of demodulation RFI in inverting operational amplifier circuits of the same gain with different input and feedback resistor values," in *1990 IEEE International Symposium on Electromagnetic Compatibility*. IEEE, 1990, pp. 145–152.
- [9] I. Giechaskiel and K. B. Rasmussen, "Taxonomy and challenges of out-of-band signal injection attacks and defenses," *arXiv preprint arXiv:1901.06935*, 2019.
- [10] I. Giechaskiel, Y. Zhang, and K. B. Rasmussen, "A framework for evaluating security in the presence of signal injection attacks," *arXiv preprint arXiv:1901.03675*, 2019.
- [11] Y. Hayashi, N. Homma, T. Sugawara, T. Mizuki, T. Aoki, and H. Sone, "Non-invasive EMI-based fault injection attack against cryptographic modules," in *2011 IEEE International Symposium on Electromagnetic Compatibility (EMC)*. IEEE, 2011, pp. 763–767.
- [12] R. Hoad and I. Sutherland, "The forensic utility of detecting disruptive electromagnetic interference," in *ECIW2008-7th European Conference on Information Warfare and Security: ECIW2008*. Academic Conferences Limited, 2008, p. 77.
- [13] R. Hurley, *Design Considerations for ESD/EMI Filters: II Low Pass Filters for Audio Filter Applications*, ON Semiconductor, 2007.
- [14] C. Kasmi and J. L. Esteves, "IEMI threats for information security: Remote command injection on modern smartphones," *IEEE Transactions on Electromagnetic Compatibility*, vol. 57, no. 6, pp. 1752–1755, 2015.
- [15] R. Krzikalla and J. Ter Haseborg, "HPEM protection on HF transmission lines," *Advances in Radio Science*, vol. 2, no. E. 1, pp. 79–82, 2005.
- [16] 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 (SP)*. IEEE, 2013, pp. 145–159.
- [17] M. Mishali and Y. C. Eldar, "From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 375–391, 2010.
- [18] T. A. Nappholz and R. Whigham, "EMI detection in an implantable pacemaker and the like," Jun. 16 1998, US Patent 5,766,227.
- [19] W. A. Radasky, C. E. Baum, and M. W. Wik, "Introduction to the special issue on high-power electromagnetics (HPEM) and intentional electromagnetic interference (IEMI)," *IEEE Transactions on Electromagnetic Compatibility*, vol. 46, no. 3, pp. 314–321, 2004.
- [20] K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, and S. Capkun, "Proximity-based access control for implantable medical devices," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 410–419.
- [21] N. Roy, H. Hassanieh, and R. Roy Choudhury, "Backdoor: Making microphones hear inaudible sounds," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2017, pp. 2–14.
- [22] F. Sabath, "What can be learned from documented Intentional Electromagnetic Interference (IEMI) attacks?" in *General Assembly and Scientific Symposium, 2011 XXXth URSI*. IEEE, 2011, pp. 1–4.
- [23] P. Sedgwick, "Pearson's correlation coefficient," *Bmj*, vol. 345, p. e4483, 2012.
- [24] J. Selvaraj, G. Y. Dayanikli, N. P. Gaunkar, D. Ware, R. M. Gerdes, M. Mina et al., "Electromagnetic Induction Attacks Against Embedded Systems," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. ACM, 2018, pp. 499–510.
- [25] H. Shin, Y. Son, Y. Park, Y. Kwon, and Y. Kim, "Sampling race: Bypassing timing-based analog active sensor spoofing detection on analog-digital systems," in *10th USENIX Workshop on Offensive Technologies*. USENIX, 2016.
- [26] Y. Shoukry, P. Martin, Y. Yona, S. Diggavi, and M. Srivastava, "Pycra: Physical challenge-response authentication for active sensors under spoofing attacks," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1004–1015.
- [27] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu, "WALNUT: Waging doubt on the integrity of MEMS accelerometers with acoustic injection attacks," in *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2017, pp. 3–18.

- [28] D. J. Yonce and L. Babler, "EMI detection for implantable medical devices," Jun. 12 2007, US Patent 7,231,251.
- [29] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "CommanderSong: A Systematic Approach for Practical Adversarial Voice Recognition," *arXiv preprint arXiv:1801.08535*, 2018.
- [30] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "DolphinAttack: Inaudible voice commands," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 103–117.