UNIVERSITY OF CALIFORNIA,
IRVINE

Cross-Layer Security in Cyber-Physical Systems (CPSs)

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Anomadarshi Barua

Dissertation Committee:
Professor Mohammad Abdullah Al Faruque, Chair
Professor Pramod P. Khargonekar
Assistant Professor Zhou Li

2023

DEDICATION

To
My loving parents Dr. Nishi Ranjan Talukdar and Dr. Paramita Barua
&
My amazing sister Nandita Barua
&
My wonderful wife Tilottoma Barua and dearest son Aradhyo Praggo Barua
&
My Ph.D. supervisor Dr. Mohammad Abdullah Al Faruque and every teacher from my nursery grade to the graduate level, who have helped me to shape my thoughts and for not letting me walk alone in dark
&
To all the good people of this beautiful world whose constant efforts are making this world a better place to live

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my gratitude to my academic advisor and committee chair, Professor Mohammad Abdullah Al Faruque, for all of his help and support throughout my research. Without his constant guidance and valuable suggestions, I would not have been able to complete this thesis today. He has been there for me through thick and thin.

I would also like to express my heartfelt appreciation to my committee members, Professor Pramod P. Khargonekar and Professor Zhou Li, for dedicating their valuable time to review my work and provide insightful comments.

I would like to extend my sincere thanks to colleagues in the Autonomous and Intelligent Cyber-Physical Systems (AICPS) laboratory, particularly Dr. Sujit Rokka Chhetri, Dr. Sina Faezi, Arnav Vaibhav Malawade, Mohanad Odema, Luke Chen, Trier Mortlock, Yonatan Gizachew Achamyeleh, and Rozhin Yasaei with whom I had the pleasure to discuss and collaborate with. I also thank my other collaborators, Dr. Deepan Muthirayan and Dr. Francesco Regazzoni, for their support in my research.

I would like to thank UCI's Department of Electrical Engineering and Computer Science for their support which allowed me to thrive on this exciting journey of research. I would also like to thank National Science Foundation (NSF) for partially funding my research under grants National Science Foundation (NSF) under awards ECCS-2028269. My research work is also partially supported by the University of California, Office of the President award LFR-18-548175. Any opinions, findings, conclusions, or recommendations expressed in this thesis are those of the author and do not necessarily reflect the funding agencies' views.

Part of this dissertation is a reprint of the materials as they appear in — Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS'22), Proceedings of the 38th Annual Computer Security Applications Conference (ACSAC 2022), IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES 2022), Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2022), Proceedings of the 29th USENIX Conference on Security Symposium (USENIX 2020), IEEE International Symposium on Hardware Oriented Security and Trust (HOST 2023) and IEEE Transactions on Dependable and Secure Computing (TDSC), Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD 2022), and IEEE Transactions on Information Forensics and Security (TIFS).

VITA

Anomadarshi Barua

EDUCATION

Doctor of Philosophy in Electrical and Computer Engineering **2023**
University of California, Irvine *Irvine, California*

Joined European Masters in Embedded Computing Systems (EMECS) **2016**
University of Southampton *United Kingdom*
Norwegian University of Science and Technology *Norway*

Bachelor of Science in Electrical and Electronic Engineering **2012**
Bangladesh University of Engineering and Technology (BUET) *Dhaka, Bangladesh*

RESEARCH EXPERIENCE

Graduate Research Assistant **2018–2023**
University of California, Irvine *Irvine, California*

PhD Graduate Intern **2022**
Solidigm *Folsom, California*

PhD Graduate Intern **2017**
Intel Corporation *Folsom, California*

TEACHING EXPERIENCE

Teaching Assistant **2021–2023**
University of California, Irvine *Irvine, California*

REFEREED JOURNAL PUBLICATIONS

Hierarchical temporal memory based one-pass learning for real-time anomaly detection and simultaneous data prediction in smart grids **2022**
IEEE Transactions on Dependable and Secure Computing

Brain-Inspired Golden Chip Free Hardware Trojan Detection **2021**
IEEE Transactions on Information Forensics and Security

Tool of Spies: Leaking your IP by Altering the 3D Printer Compiler **2021**
IEEE Transactions on Dependable and Secure Computing

REFEREED CONFERENCE PUBLICATIONS

MagHop: Magnetic Spectrum Hopping for Securing Voltage and Current Magnetic Sensors **2023**
To appear at IEEE International Symposium on Hardware Oriented Security and Trust (HOST 2023)

A Wolf in Sheep's Clothing: Spreading Deadly Pathogens Under the Disguise of Popular Music **2022**
29th ACM Conference on Computer and Communications Security (CCS)

BayesImposter: Bayesian Estimation Based .bss Imposter Attack on Industrial Control Systems **2022**
Annual Computer Security Applications Conference (ACSAC)

PreMSat: Preventing Magnetic Saturation Attack on Hall Sensors **2022**
IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)

HALC: A Real-time In-sensor Defense against the Magnetic Spoofing Attack on Hall Sensors **2022**
25th International Symposium on Research in Attacks, Intrusions, and Defenses (RAID)

Sensor Security: Current Progress, Research Challenges, and Future Roadmap **2022**
41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD)

Hall Spoofing: A Non-Invasive DoS Attack on Grid-Tied Solar Inverter **2020**
29th USENIX Security Symposium (USENIX Security 2020)

Hierarchical Temporal Memory Based Machine Learning for Real-Time, Unsupervised Anomaly Detection in Smart Grid **2020**
ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)

Noninvasive Sensor-Spoofing Attacks on Embedded and Cyber-Physical Systems **2020**
IEEE 38th International Conference on Computer Design (ICCD)

ABSTRACT OF THE DISSERTATION

Cross-Layer Security in Cyber-Physical Systems (CPSs)

By

Anomadarshi Barua

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2023

Professor Mohammad Abdullah Al Faruque, Chair

The definition of cyber-physical systems (CPSs) is that they integrate sensing, actuation, storage, computation, control, decision-making, and networking into physical systems and objects, connecting them to the Internet and to each other. With the advancement of complex hardware and software technologies and the prevalence of the Internet of Things (IoTs), interactions of cyber and physical components open a "Pandora's Box" of unknown threats that can come from unconventional ways. CPSs have tight integration of cyber and physical components, and complex interactions happen between this cyber and physical layer which may affect the safety and controllability of closed-loop control from sensing to actuation. In most cases, researchers put significant efforts into improving the efficiency and responsiveness of the cyber and physical interactions in CPSs. However, the CIA triad - textitconfidentiality, *integrity*, and *availability*, is often absent while designing the interactions between cyber and physical parts. As a result, attacks and vulnerabilities are lurking between cyber and physical intersections. The cyber and physical layers in CPSs are termed Cross-layers in this thesis.

Most attacks on CPSs can be propagated into this cross-layer, i.e., from the physical domain to the cyber domain or vice-versa, and hence, can be termed as cross-domain attacks. To understand these cross-domain attacks and to address the challenges that exist in the cross-

layer, a very different set of methodologies and tools are needed. Moreover, as these cross-domain attacks involve hardware and software layers, defenses against these vulnerabilities also demand new hardware/software co-design approaches to detect, contain and isolate vulnerabilities in CPSs.

The first half of the thesis addresses some interesting and unconventional attack models and vulnerabilities in CPSs, particularly focusing on the smart power grid systems, bio-safety labs, and industrial control systems (ICSs). This thesis addresses how attacking a single hall-effect sensor of a solar inverter using an attack signal from the magnetic spectrum can compromise a weak microgrid in smart grid systems. Next, this thesis explores the use of a different attack signal other than the magnetic spectrum. In doing so, this thesis investigates how the use of simple music as an attack signal can fool a building management system of a bio-safety lab and can facilitate the leaking of deadly pathogens from the bio-research facilities. Next, this thesis explores the vulnerabilities of industrial control systems (ICSs) in cloud settings and provides how combining memory deduplication and rowhammer attack can compromise a programmable logic controller (PLC) in ICSs.

The remaining half of the thesis provides defenses for the unconventional vulnerabilities discussed in the first half of the thesis. This part of the thesis focuses on different sensor defense techniques working against false data injection and spoofing attacks in CPSs. Please note that the defense techniques that exist in the literature have the following limitations: (i) they don't work against attack signals having a frequency equal to the frequency of original signals, (ii) they don't work against attack signals having zero frequency, and (iii) they don't work in the saturation region of the sensor. This thesis begins to fill this gap by providing defense techniques against false data injection into sensors using hardware-software co-design techniques. At first, we demonstrate a defense named HALC, which can detect and contain all types of strong and weak magnetic attack fields, such as constant, sinusoidal, and pulsating magnetic fields, injected into hall sensors in real-time. Next, this

thesis provides another defense named PreMSat, which can work in the saturation region of the hall sensors. Last, we present MagHop, which can prevent electromagnetic interference (EMIs) from being injected into magnetic sensors. All three defense techniques proposed here achieve better performance compared to state-of-the-art works and can contain the attack in real-time without hampering the normal data processing speed of sensors.

Chapter 1

Introduction

1.1 Cyber-Physical System

A cyber-physical system (CPS) is a type of system that integrates physical and computational components to create a network of interacting devices, sensors, and machines that can interact with the physical world [1, 2]. CPS combines the power of data processing, communication, and control to improve the efficiency and reliability of physical systems using the computational components that exist in the cyber domain. CPS broadly has two parts: computational components in the cyber domain and physical processes in the physical domain. These two components are explained below.

1.1.1 Cyber Component in Cyber-Physical Systems

The computational and network elements that provide communication, calculation, and control are referred to as the "cyber component" of a CPS. It consists of hardware and software elements that provide physical component connectivity and enable the system to

evaluate, process, and take action on data [3].

Sensors, actuators, embedded systems, network protocols, data analytics, and software systems are frequently found among a CPS's cyber components. Together, these parts gather and interpret data from the system's physical parts, then utilize that information to decide what to do and how to make the parts behave.

A CPS's cyber components are in charge of carrying out operations like data gathering, processing, communication, and analysis. For instance, in a smart grid CPS, the cyber components would be in charge of gathering information on energy consumption, evaluating that information to spot patterns and trends, and utilizing that analysis to maximize energy use and minimize waste.

1.1.2 Physical Component in Cyber-Physical Systems

The physical systems and equipment that communicate with the physical world are referred to as the physical component of a CPS. It consists of hardware elements that gather data and carry out physical operations, such as sensors, actuators, controllers, and machines [4].

Depending on the particular application, a CPS's physical components can differ significantly. For instance, the physical parts of a manufacturing CPS might be robotic arms, conveyor belts, and sensors that track the production process. The physical elements of a smart building CPS may include of HVAC systems, lighting controls, and occupancy sensors.

Performing physical actions, perceiving the physical environment, and responding to commands from the cyber components of the system are all responsibilities of the physical components of a CPS. For instance, if a sensor detects a change in temperature, the system's cyber component would analyze the information and instruct an actuator to alter the HVAC system appropriately.

Autonomous vehicles, smart grids, manufacturing processes, and medical equipment are a few examples of CPS. These systems integrate their physical and digital components in order to sense, assess, and react to real-world events, frequently in real-time.

CPS technology is becoming increasingly important in areas such as manufacturing, transportation, healthcare, and energy management. The integration of physical and digital systems enables more efficient and effective use of resources, improved safety and security, and better decision-making [5].

1.2 Cross-Layer Security in Cyber-Physical Systems

Cross-layer security is a strategy for safeguarding Cyber-Physical Systems (CPS) that involves integrating security mechanisms across several system layers. Cross-layer security is crucial in CPS because of how interconnected these systems are and how many different levels of hardware, software, and communication protocols are used.

Integrating security controls across the system's many layers, including the physical layer, data link layer, network layer, transport layer, and application layer, is known as cross-layer security (CPS). Security measures like encryption, authentication, access control, and intrusion detection can all be included in this integration.

The requirement for close coordination across several system levels (cite: zhu2020cross) is one of the major obstacles to cross-layer security implementation in CPS. The Internet Protocol (IP) suite and other standardized communication protocols, as well as security measures that are put in place at every level of the system, can be used to achieve this coordination.

Another difficulty is finding a compromise between security needs and additive manufacturing system performance, energy consumption, sensor security, blockchain, and side-channel

analysis [6–10]. Additional security measures may occasionally complicate systems and slow down performance, which can be troublesome for real-time CPS applications.

Cross-layer security experts and researchers are creating fresh methods for protection that can be effective while having a little negative influence on system performance in order to address these issues. These methods include data-driven security, sensor fusion, real-time security monitoring and response systems, energy-efficient cryptographic algorithms, and lightweight security mechanisms [9, 11–16].

1.3 Cross-Layer Vulnerability in CPSs

Cross-layer vulnerabilities in Cyber-Physical Systems (CPS) are flaws that can be exploited by attackers to undermine the security of the system and exist across various layers of the system. These flaws are brought about by the intricate relationships between the various hardware, software, and communication protocol layers in CPS.

A time attack [17] is an illustration of a cross-layer vulnerability in CPS, where an attacker takes advantage of the system's timing features to extract sensitive data. Timing attacks can happen at any one of the system's tiers, including the physical, data connection, and network layers.

Another illustration of a cross-layer vulnerability is a protocol defect [18], which allows an attacker to bypass security precautions and obtain unauthorized access to the system by taking advantage of a flaw in the design or implementation of a communication protocol. Data link, network, transport, and application layers are only a few of the system layers where protocol defects might appear.

In addition to these particular instances, cross-layer vulnerabilities might develop as a result

of a lack of coordination and integration across various system layers. For instance, security mechanisms put in place at higher layers of the system, like the network layer or application layer, may not be able to identify or mitigate a vulnerability in the physical layer of the system.

It is crucial to adopt a comprehensive security strategy that incorporates multiple layers of defense, such as access control, intrusion detection, hardware trojan detection, and secure communication protocols to address cross-layer vulnerabilities in CPS [13, 19–23]. Additionally, it is crucial to regularly scan the system for any flaws and apply the necessary patches and upgrades as soon as these flaws are identified [24–26].

1.4 Thesis Contribution

This thesis has two types of contributions. The first contribution is that this thesis addresses the following vulnerabilities that exist in the cross-layer of CPSs:

- How attacking a single Hall sensor of a grid-tied solar inverter with magnetic fields can compromise a weak microgrid.
- How attacking pressure sensors used in biosafety labs with music can compromise the integrity of bio-research labs and help to leak deadly pathogens.
- How attacking the DLL files of cloud protocols can be combined with memory deduplication and rowhammer attack to compromise industrial control systems (ICSs).

To tackle the aforementioned vulnerabilities, the second contribution of this thesis is that this thesis also provides defenses against the above vulnerabilities that exist in the corresponding CPS:

- Propose a defense HALC that can detect and contain all types of strong and weak magnetic spoofing, such as constant, sinusoidal, and pulsating magnetic fields, in real-time.

- Propose a defense named PreMSat against the saturation attack on passive Hall sensors.

- Propose a defense named MagHop that can prevent electromagnetic interference in the voltage and current magnetic sensors (VCMSs).

1.5 Thesis Structure

This thesis is structured as follows:

- *Chapter 2* demonstrates a noninvasive attack that could come by spoofing the Hall sensor of an inverter in a stealthy way by using an external magnetic field. We demonstrate how an attacker can camouflage his/her attack tool and place it near a target inverter. In doing so, he/she can intentionally perturb grid voltage and frequency and can inject false real and reactive power to the grid.

- *Chapter 3* demonstrates a non-invasive and stealthy attack on Negative pressure rooms (NPRs) by spoofing a differential pressure sensor (DPS) at its resonant frequency. Our contributions are: (1) We show that DPSs used in NPRs typically have resonant frequencies in the audible range. (2) We use this finding to design malicious music to create resonance in DPSs, resulting in an overshooting in the DPS's normal pressure readings. (3) We show how the resonance in DPSs can fool the BMSs so that the NPR turns its negative pressure to a positive one, causing a potential *leak* of deadly microbes from NPRs.

- *Chapter 4* shows a new attack primitive - *BayesImposter*, which points out that the attacker can duplicate the .bss section of the target control DLL file of cloud protocols using the *Bayesian estimation* technique. Our approach results in less memory (i.e., 4 KB compared to GB) and time (i.e., 13 minutes compared to hours) compared to the brute-force approach used in recent works.

- *Chapter 5* explores a new defense HALC that can detect and contain all types of strong and weak magnetic spoofing, such as constant, sinusoidal, and pulsating magnetic fields, in real-time. HALC works up to ∼9000 G of external magnetic spoofing within a frequency range of 0 - 150 kHz, whereas existing defenses work only when the spoofing signals have a separate frequency from the original signal being measured. HALC utilizes the analog and digital cores to achieve a constant computational complexity O(1). Moreover, it is low-power (∼1.9 mW), low-cost (∼$12), and can be implemented in the sensor hardware.

- *Chapter 6* provides a defense named PreMSat against the saturation attack on passive Hall sensors. The core idea behind PreMSat is that it can generate an internal magnetic field having the *same* strength but in *opposite polarity* to external magnetic fields injected by an attacker. Therefore, the generated internal magnetic field by PreMSat can nullify the injected external field while preventing: (i) intentional spoofing in the sensor's *linear region*, and (ii) saturation attack in the *saturation region*.

- *Chapter 7* provides a defense named MagHop against an intentional EMI or external magnetic field injection into voltage and current magnetic sensors. The core idea of our defense is to shift the frequency spectrum of the magnetic field, which is used as the transduction medium of the sensor, to another spectrum unknown to an attacker. In addition, the frequency spectrum, which carries the magnetic field in the transduction medium, is varied in a pseudo-random fashion so that the attacker will not be able to track it to inject any EMI into it.

- *Chapter 8* concludes the dissertation with some remarks on the contributions and discussion on future directions.

Chapter 2

A Noninvasive DoS Attack on Grid-Tied Solar Inverter

2.1 Abstract

Grid-tied solar inverters continue to proliferate rapidly to tackle the growing environmental challenges. Nowadays, different smart sensors and transducers are tightly integrated with the grid-tied inverter. This integration opens the "Pandora's Box" of unknown threats that could come from very unconventional ways. This paper demonstrates a noninvasive attack that could come by spoofing the Hall sensor of an inverter in a stealthy way by using an external magnetic field. We demonstrate how an attacker can camouflage his/her attack tool and place it near a target inverter. In doing so, he/she can intentionally perturb grid voltage and frequency and can inject false real and reactive power to the grid. We also show the consequences of the attack on a scaled-down testbed of a power grid with a commercial 140 W grid-tied inverter from Texas Instruments. We are able to achieve a 31.52% change in output voltage, 3.16x (-6dB to -11dB) increase in low-frequency harmonics power, and

3.44x increase in real power. Moreover, we introduce a duty-cycle variation approach for a noninvasive adversarial control that can change the inverter voltage up to 34% and real power up to 38%. We discuss the feasibility of using a 100 kW inverter through discussion. This provides insights behind the generalization of the attack model. In addition, the commercial power system simulation tool Etap 19.0.1 is used to simulate the impact of the attack on a 2.3 MW power grid. To the best of our knowledge, this is the first methodology that highlights the possibility of such an attack that might lead to grid blackout in a weak grid. The findings in this chapter have been published in [27].

2.2 Introduction

Cyber-physical systems (CPSs) in power grids comprise sophisticated control mechanisms. These mechanisms may produce multidisciplinary security issues capable of compromising the *Availability* and *Integrity* [28–30] of the power grids. Examples of such attacks on power CPSs include cyberattacks on the Ukrainian power grid [31], DoS attacks on anonymous western utilities in the U.S. power sector [32], the Slammer worm attack on Ohio's Davis-Besse nuclear power plant [33], the Stuxnet malware attack on Iran's nuclear facilities [34], etc. The results of these attacks are very serious, including region-wise blackouts affecting more than 230,000 residents [35] and monetary losses [36].

Nowadays, distributed energy sources are proliferating rapidly and a substantial portion of these sources are highly efficient grid-tied solar inverters[1] [37, 38] equipped with Hall sensors. These Hall sensors, however, can be cleverly spoofed to orchestrate a noninvasive attack on the grid. The attack in question can perturb the normal operation of a power system and may cause grid failures in a weak grid. It is important to note that a strong grid gradually becomes weak due to the continuous integration of distributed energy sources [39]. Strong grids may also behave as weak grids at a particular time of a day (e.g., peak hours).

[1] In this paper, grid-tied solar inverter are used interchangeably with inverter.

Moreover, micro-grids [40] also behave as weak grids when connected over long cables to a utility grid. A detailed background of strong and weak grids is provided in Section 2.4.1.

This paper shows that a smart attacker can inject measurement errors into the Hall sensors of an inverter using a noninvasive magnetic spoofing technique with adversarial control. The injected errors can propagate from the compromised Hall sensor to the internal controllers of the inverter and eventually compromise the inverter itself. The compromised inverter can hamper the grid stability and may cause grid failures in a weak grid scenario. This method is similar to the false data injection approach. But in this case, the injection is coming from the physical domain by exploiting the physics of the Hall sensor. *We show that the attacker can intelligently control the false data injection by applying distinct types of external magnetic fields, such as constant, sinusoidal, and square pulsating magnetic fields, on the Hall sensors.* This may perturb the inverter output voltage, frequency, real and reactive power. This perturbation can propagate through the cyber domain and finally impact the physical domain. Hence, this can be termed as an attack from **P**hysical-to-**C**yber-to-**P**hysical (P-2-C-2-P) domain [41]. In power CPSs, this type of cross-domain attack is yet to be explored in depth by the security community.

Technical Contributions: Our technical contributions are listed as follows that are elaborated in the following sections:

i. A new attack model (**Section 2.5**) that describes how the availability of the grid-tied inverter is stealthily breached.

ii. Algorithms and a potential design for the relevant attack tool (i.e., Embedded Hall Spoofing Controller) and mathematical models of an inverter's control blocks (**Section 2.6**).

iii. A testbed (**Section 2.7**) with a scaled-down model of a power grid, on which the attack model is validated and adversarial control is demonstrated (**Section 2.8**).

iv. The attack model is further evaluated (**Section 2.9**) using an industry-standard commercially used *Electrical Power System Analysis Software (Etap 19.0.1)* on a medium-sized 2.3 MW (equivalent to approx. $\sim$ 150 houses) grid.

v. Defense (**Section 2.10.1**) is proposed and justified, and limitations (**Section 2.10.2**) of this attack are noted.

2.3 Related Work

We discuss here different attacks on analog sensors, inertial sensors, and on power systems that exist in the literature.

Attacks on Analog Sensors: Kune et al. [42] spoofed sensors by electromagnetic interference (EMI) to induce defibrillation shocks on implantable cardiac devices. Park et al. [43] used infrared to trigger a medical infusion pump to deliver overdose to patients. Davidson et al. [44] reported how spoofing optical sensors of an unmanned aerial vehicle (UAV) can compromise complete control of the lateral movement. Yan et al. [45] published a contactless attack on self-driving cars using ultrasound and EMI. Shin et al. [46] showed a spoofing attack on LiDar to create illusions of objects appearing closer in automotive systems. Zhang et al. [47] injected inaudible commands into a microphone using ultrasonic carriers. Lastly, Shoukry et al. [48] used an external magnetic field to spoof the Antilock Braking System (ABS) to change the wheel speed of a vehicle. There are a few fundamental differences between our work and [48]. First, the attacker requires access to place the electromagnetic actuator near the ABS wheel speed sensor and must strongly secure the attack object *ABS Hacker* to the vehicle body, likely with a nut and bolt. Second, the original magnetic field of the vehicle must be shielded before spoofing. The space to place this extra shield near the ABS sensor is critical. Third, the *ABS Hacker* comprises expensive heterogeneous processors. Fourth, the adaptive controller of [48] requires complex tuning of its closed-loop poles

and zeros. In contrast to [48], our attack can be noninvasively executed on a cheap Arduino board and does not require strong physical mounting or extra shielding.

Attacks on Inertial Sensors: Son et al. [49] used high power sound noise to compromise the gyroscope of a drone to make it uncontrollable. Wang et al. [50] used a sonic gun to demonstrate acoustic attacks on different inertial sensors. Trippel et al. [51] showed fine-grained adversarial control over MEMS accelerometers using acoustic signals to damage digital integrity. Tu et al. [52] also demonstrated adversarial control over embedded inertial sensors to trigger the actuation of different control systems. In contrast to their methods (e.g., biasing attack, sample rate drifts, etc.), our paper introduces a duty-cycle variation approach for adversarial control that is novel in our attack model in the power CPSs.

Attacks on Modern Power Systems: There are quite a lot of works on traditional **C**yber-to-**P**hysical domain (C-2-P) attacks in the literature, such as malicious false data injection [53], flooding [54], arbitrary command injection [55], time-delay input attack [56], load distribution attack [57]. Ilge Akkaya et al. [58] used GPS spoofing on *Phase Measurement Units* (PMUs) to lead a substation to an erroneous state. In contrast to these works, our work demonstrates an unconventional P-2-C-2-P attack in the power CPSs.

Our work shows how an attacker can cause damage (e.g., blackout) to the connected power grid by intelligently applying constant, sinusoidal, and square pulsating magnetic fields. Moreover, in contrast to the prior works, this paper models the vulnerable blocks of the controller of an inverter and mathematically proves the underlying principle of propagation of attack from sensors to the internal controllers. Our attack impact is more realistic, has more economically damaging effect, and can impact a large region.

2.4 Background

2.4.1 Strong and Weak Grid in the Power CPSs

The grid where voltage and frequency are stable and do not vary during load connection/disconnection is known as a strong grid. Historically, rotational generators are present in the power systems. Rotational generators have prime movers to convert rotational kinetic energy into electrical energy. Rotational energy stored in the prime mover of these generators acts as an *inertia* against any sudden change of load in the system; therefore, the voltage/frequency does not vary abruptly within a limit in the grid when a small load is disconnected from the grid. *It is important to note that a strong grid is not ideally strong all the time. The voltage/frequency of a strong grid may vary abruptly if the change of the load is large compared to the generation capacity, or if the rotational energy stored in the prime mover is not sufficient to compensate for the sudden change in the grid. Hence, a strong grid can behave as a weak grid. A weak grid refers to a grid wherein its voltage is highly sensitive to any variation in the load [59].*

Due to the continuous integration of distributed solar/wind inverters, the modern grid is shifting from centralized to distributed generation resulting in poor control and lack of inertia (i.e., rotational turbines). This causes grid weakening over time [39], which is already a concern in the community. In this scenario, an attacker can perturb the grid voltage/frequency using an inverter and this perturbation may disrupt the entire system. Moreover, low generation, long transmission lines, etc., can also contribute to weak grids. We can also find weak grids in isolated places like Baja, Mexico; parts of Alaska; or under-developed areas between strong grids.

2.4.2 Real Power, Reactive Power and Phase

An inverter can inject real power and reactive power into the grid. *Real power is related to grid frequency and reactive power is related to grid voltage [60].* If the generation of real power is lower than the real power demand, the grid frequency may fall. Whereas, if the generation of reactive power is lower than the required, the grid voltage may fall. Real power is the amount of power in watts (W) being dissipated, and reactive power results from inductive/capacitive loads measured in volt-ampere reactive (VAR). The phase is the position of a point of a wave in a time instant. Three-phase voltages are 120^o phase apart from each other.

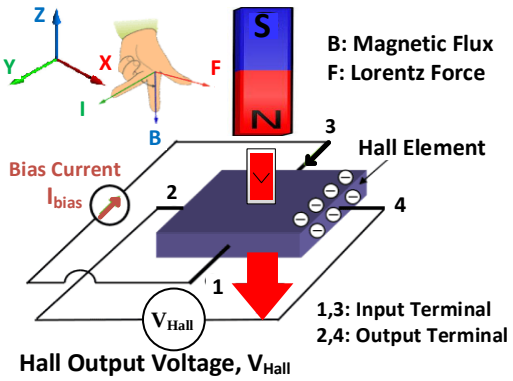


Figure 2.1: Working principle of a typical Hall sensor.

2.4.3 Working Principle of a Hall Sensor

Fig. 2.1 shows the working principle of a typical Hall sensor. It comprises a Hall element, which is made of a thin piece of p-type semiconductor material (e.g. Gallium Arsenide, etc.). Let us assume that a bias current I_{bias} is flowing in +ve Y direction (terminal 1, 3) of the Hall element having thickness d. This Hall element is placed within an applied magnetic field B whose direction is -ve Z-axis. The charge carriers inside the Hall sensors feel a force along +ve X-axis. This force is known as the Lorentz force F. Due to this Lorentz force, the charge carriers will be deflected along the +ve X-axis and a voltage V_{Hall} will be generated across the Hall element. The generated voltage V_{Hall} may be expressed as:

$$V_{Hall} = k(\frac{I_{bias}}{d} \times B) \tag{2.1}$$

where k is the hall coefficient, which depends upon the properties of the hall element. If d, I_{bias}, and k are constant, V_{Hall} depends only on applied B. This B is proportional to the current/voltage to be measured. *Any external perturbation of B can change V_{Hall}. And this change can give a **false** sense of voltage/current measurement that can propagate to the inverter controller and hamper its normal operation.*

2.4.4 Why is a Hall Sensor Used in an Inverter?

Inverters measure grid voltage, current, and their phase angles for important control applications. Four methods [61] are mainly used to measure voltage/current: i) Resistive drop/divider method, ii) Magneto-resistance method, iii) A voltage/current transformer, and iv) A Hall effect sensor.

A resistive drop/divider is not suitable for high voltage/current measurement because of the following reasons: high power loss in the resistor itself, inability to measure small DC current in the presence of large AC current, and absence of proper isolation. A magneto-resistive material is nonlinear and temperature-dependent, therefore, it is not suitable for accurate high current measurement. A voltage/current transformer is not suitable for simultaneous AC/DC measurement and is bulky. It also requires an external resistance to convert current into voltage and has a low efficiency for core loss. *In contrast, the Hall effect sensor has excellent accuracy, high efficiency, very good linearity, low thermal drift, and low response time. It is lightweight, compact, and suitable for simultaneous large AC/DC voltage/current measurement with galvanic isolation.* Therefore, Hall sensors are pervasive in high power

inverter applications.

To show the prevalence of Hall sensors in inverters, we investigate six industry-designed inverters (small to medium range) and a large 100 kW inverter. All these inverters (Table 2.1 and Section 2.9) have similar functional blocks, and Hall effect sensors are present in the measurement unit. This is because inverters are optimized for ***efficiency and accuracy***, but not for security from this type of unconventional spoofing attack.

Table 2.1: Presence of Hall sensors in different inverters.

Manufacturer	Inverter Series	Sensor	Power
Texas Instr. [62]	TMDSOLARUINVKIT	Hall	0.14 kW
Texas Instr.[63]	TIDA-01606	Hall	10 kW
STMicro. [64]	STEVAL-ISV003V1	Hall	0.25 kW
Microchip [65]	Grid Connected Inverter	Hall	0.215 kW
SMA[66]	Sunny Boy	Hall	5 kW
SOLAX [67]	SL-TL5000T	Hall	3 kW

2.5 Attack Model

Fig. 2.2 depicts our proposed attack model, which can affect the availability of an inverter by spoofing Hall sensors. The components of our attack model are described as follows:

Attacker's Intent: The attacker wants to disrupt the normal operation of a power system by spoofing an inverter noninvasively and wants to cause grid failures in a weak grid.

Attacker's Capabilities: The attacker can surreptitiously place a small box near the target inverter. This box contains a powerful electromagnet integrated with an electronic spoofing controller (i.e., *Embedded Hall Spoofing Controller*). This box is small enough to be camouflaged within a small container, such as flower vase, coffee cup. Placing the camouflaged attack tool near the inverter requires a *brief one-time* access. The box has wireless controls allowing for remote communication. Therefore, the attacker can remotely control the timing of the attack and can pick a vulnerable time (e.g., at peak hour, etc.) to impact the connected power grid. The authorities of the target inverter may not be aware of this attack model and would possibly neglect the security implications of any small

camouflaged box placed near an inverter.

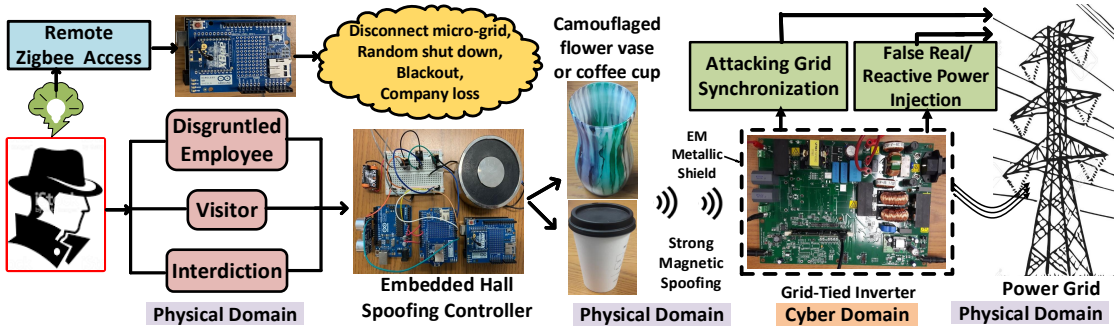


Figure 2.2: Brief overview of the Hall spoofing attack methodology.

Attacker's Access Level: The access near the inverter needed for the attack can be possible in at least three scenarios. **First** (most likely), a malicious employee or a guest, who has access near the inverter, may place the camouflaged attack tool near the inverter. An incident similar to this has already been reported in past news [68]. A disgruntled ex-employee of an electric utility in Texas posted a note in a hacker journal indicating that his insider knowledge of the system could be used to shut down that region's power grid. Moreover, solar plants are usually located in an isolated place with less security [69]. Getting a *brief one-time* access near the isolated solar plants may not be difficult. Staggs et al. [69] demonstrated how easily an attacker can access a wind plant in the middle of a remote field and can invasively place an attack tool inside of the wind turbine. Our attack model is stronger compared to [69] because of its noninvasive nature. **Second**, the manufacturer may introduce the malicious electromagnet with controllers inside of the solar inverter. **Third** is interdiction, which has been rumored to be used in the past [70–73] and has been recently proven to be feasible [74]. During interdiction, a competitor can intercept the inverter during delivery or installation and may modify the inverter by placing an electronic device inside and then proceed with delivery or installation to the customer.

Stealthy Nature: The attacker can remotely perturb the inverter by camouflaging the tiny attack tool and can choose the timing of the attack to remain unidentified to maximize the impact. Fig. 2.3 is an example that shows how the attacker can place the camouflaged

attack tool near the target inverter.

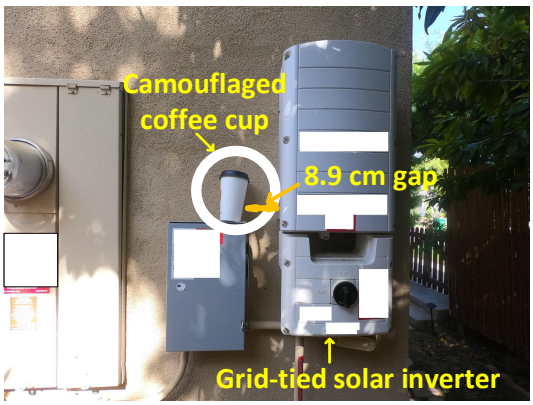


Figure 2.3: Demonstration of access near a typical inverter.

Outcome of this Attack: The attacker may cause grid failures if the power grid is weak. And for weakly protected systems, the attacker can fry the internal circuitry of the inverter itself. By spoofing the Hall sensor, the attacker can give a false impression that the conditions required for synchronization of the inverter with the grid have been achieved when they have not. This improper grid synchronization may shut down the inverter (Section 2.8.1). *A micro-grid is a group of interconnected loads and distributed energy resources, which can operate in both grid-connected or island mode* [40]. The attacker can disconnect the micro-grid from the utility grid at a random time or can prevent it from disconnecting even when it is supposed to disconnect (e.g., in the case of an outage). The attacker can choose the timing of the attack and can remotely shut down the inverter in peak hours to create a shortage of real/reactive power with no prior notice to the authority. This scenario can be significant in a weak grid and a micro-grid. As the timing of the attack can be remotely controlled, the attacker can cause a security breach by randomly shutting down the local solar power supply of any important organization, remote airport, army base, etc. The attacker can prevent the inverter from starting and can cause a repetitive shutdown. Simply pressing the restart button of the inverter may not solve the problem until the attack tool is removed. As this attack is stealthy, it can remain unidentified. This trick, which may cause grid instability, can be used to ask for ransom or to blackmail the utility.

Attacker's Safety: As inverters handle high voltage, it is unsafe for the attacker to invasively manipulate them. In this sense, our attack model is safe for the attacker as it enables the attacker to control the operation of the inverter noninvasively.

Attacker's Resources: We assume that the attacker has domain knowledge of the inverter controllers with some high school knowledge of electromagnetism.

Cost: The design cost of the *Embedded Hall Spoofing Controller* and the electromagnet is less than $50. The electronic parts are readily available from Amazon and Digikey.

2.6 Attack Model Design

This section explains how an attacker can design the attack tool (i.e., *Embedded Hall Spoofing Controller*). This section also mathematically models important basic blocks of an inverter irrespective of the inverter size.

2.6.1 Embedded Hall Spoofing Controller

The *Embedded Hall Spoofing Controller* consists of an electromagnet, an Arduino Uno, few MOSFETs, a Zigbee RF module, an Ultrasonic Sensor, and Energizer A23 Batteries. A small (height 3.8 cm, radius 3.5 cm) but powerful electromagnet (WF-P80/38) is used as a source of magnetic field. An electromagnet can also be built by winding wires around a strong neodymium (NIB) magnet, which is easily found in a computer hard disk [75]. An ultrasonic sensor (HC-SR04) is interfaced with the Arduino board to measure the distance between the electromagnet and the inverter shield. This distance helps to calculate the required strength of the *Magneto-Motive Force* (MMF) to influence the Hall sensors and stops oversupply of power to extend the battery lifetime. MMF measures the strength of the generated magnetic flux. A Metal Oxide Semiconductor Field Effect Transistor (MOSFET), P7N20E is used to toggle the electromagnet ON and OFF with variable frequencies using a

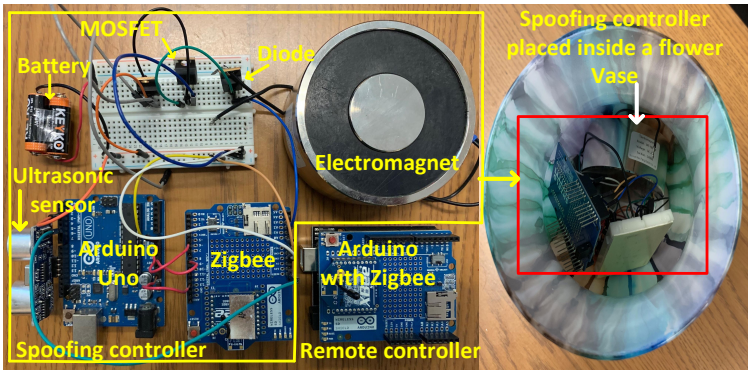


Figure 2.4: The Embedded Hall Spoofing Controller.

Pulse Width Modulation (PWM) technique. This PWM helps to generate variable-frequency electromagnetic flux and controls the power input to the electromagnet depending upon the attacker's need and intention. To protect the MOSFET from an inductive surge (due to the switching of the large electromagnet), a free-wheeling diode (U1620G) is connected across the electromagnet.

2.6.2 Controller Compromising Algorithm

The algorithm, which compromises the inverter controller, runs on the Arduino Uno (Algorithm 1). It is computationally inexpensive and may run on the Arduino for a long period with the battery pack mentioned in Section 2.6.1. It controls the ultrasonic sensor, Zigbee modules, and ADC, PWM, RX-TX peripherals of the Arduino Uno. After initializing the necessary modules and peripherals, the algorithm first checks for battery voltage level to see whether it is above the threshold. Otherwise, it returns `ErrorCode` after informing the attacker about this issue through Zigbee. Then the distance from the inverter is calculated using the ultrasonic module. If it is outside of the range, it notifies the attacker (`ErrorCode`) through Zigbee. Otherwise, it activates the MOSFET switching block and generates PWM frequency depending upon the attacker's need and intention for different attack scenarios. The attacker can also enable adversarial control and provide duty-cycle to the attack tool through the Zigbee. Depending upon the provided duty-cycle (see Section

2.8.4), the `PowerController` supplies the required amount of power to the electromagnet. This algorithm also checks for `MagnetCurrent`, which is flowing through the electromagnet. If it is less than the required amount, the algorithm also notifies the attacker.

2.6.3 Modelling Grid-Tied Inverters

This section mathematically models the basic blocks of the inverter controller. A grid-tied solar inverter can be single-phase or three-phase. Fig. 2.5 shows the basic blocks of a three-phase inverter. Let us denote the balanced abc-phase (*phase a, b, c*) grid voltages by e_a, e_b, and e_c, which are 120^0 phase apart. These abc-phase grid voltages may be represented by a grid voltage space vector $\vec{S}_{abc}$ as follows:

$$\vec{S}_{abc}(t) = \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} = \begin{bmatrix} E\cos\omega t \\ E\cos(\omega t - 120^0) \\ E\cos(\omega t + 120^0) \end{bmatrix} \tag{2.2}$$

where E is the amplitude and ω is the angular frequency of the grid voltage. Terms e_a, e_b, and e_c are sensed by three Hall effect voltage sensors *(we name these as grid sensors)* and then are sampled by the Digital Signal Processing (DSP) unit.

The abc-to-dq Transformation Block: This block transforms abc-phase grid voltage $\vec{S}_{abc}$ into direct-quadrature (dq) axis components, which are direct current (DC) quantities. This transformation facilitates the designing of a simple controller, such as the Proportional-Integral (PI) controller, in DC domain [76]. We know the axis of the rotor flux of a rotating machine is known as direct (d) axis, and the quadrature (q) axis lags d axis by 90^o. The *abc-to-dq* transformation is done in two steps: a *Clarke Matrix* (CM) transforms $\vec{S}_{abc}$ into alpha-beta component vector $\vec{S}_{\alpha\beta}$ (e_α and e_β), and a *Park Matrix* (PM) transforms $\vec{S}_{\alpha\beta}$ into dq component vector $\vec{S}_{dq}$ (e_d and e_q). The term $\vec{S}_{dq}$ can be given by:

Algorithm 1: Solar Inverter Controller Compromising Algorithm.

Input: Control variables: $\{Attack\_level, Adversarial\_control, Duty\_cycle\}$
Output: Pulse Width Modulation Frequency: PWM_{freq}

$n \leftarrow$: $Timesteps$
$ADC\_arduino, PWM\_arduino, RX\_TX\_arduino \leftarrow Initialize$
$Zigbee\_module, Ultrasound\_module \leftarrow Initialize$
for $i \leftarrow 1$ ***to*** n **do**
    $batteryVoltage \leftarrow ADC\_Channel\_1$
    if $batteryVoltage < VoltageThreshold$ **then**
        Inform_attacker (battery_voltage_low)
        return $ErrorCode\_BatteryVoltageLow$
    end
    else
        Inform_attacker (battery_voltage_sufficient)
    end
    $ultrasound\_setup \leftarrow Activate$
    $Distance \leftarrow Ultrasound\_Measurements$
    if $Distance > Distance\_threshold$ **then**
        Inform_attacker (distance_threshold_exceed)
        return $ErrorCode\_DistanceThresholdExceed$
    end
    $PowerController \leftarrow (Duty\_cycle = 100\%)$
    if $Attack\_Level = Constant\_MMF$ **then**
        $MosfetGate \leftarrow PulledUp$
    end
    else if $Attack\_Level = Pulsating\_MMF\_1Hz$ **then**
        $MosfetGate \leftarrow PulledUp$
        $PWM_{freq} \leftarrow 1$
    end
    else if $Attack\_Level = Pulsating\_MMF\_2Hz$ **then**
        $MosfetGate \leftarrow PulledUp$
        $PWM_{freq} \leftarrow 2$
    end
    else
        $MosfetGate \leftarrow PulledDown$
    end
    if $Adversarial\_control = Enable$ **then**
        $PowerController \leftarrow (Duty\_cycle, Distance)$
        Inform_attacker (adversarial_control_enabled)
    end
    else
        $PowerController \leftarrow (Duty\_cycle = 100\%)$
        Inform_attacker (adversarial_control_disabled)
    end
    $MagnetCurrent \leftarrow ADC\_Channel\_2$
    if $MagnetCurrent < CurrentThreshold$ **then**
        Inform_attacker (battery_Charge_low)
        return $ErrorCode\_BatteryChargeLow$
    end
end

$$\vec{S}_{dq} = \begin{bmatrix} e_d \\ e_q \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{3}{2}}E \\ 0 \end{bmatrix} \tag{2.3}$$

where e_d and e_q are the d and q axis components of the abc-phase grid voltages, respectively and they are DC quantities. Please note that $e_q = 0$ for balanced grid voltage.

Let us denote the three-phase inverter output voltages $[u_a, u_b, u_c]$ and output currents $[i_a, i_b, i_c]$ as vectors $\vec{U}_{abc}$ and $\vec{I}_{abc}$, respectively. The inverter output current $\vec{I}_{abc}$ is similarly sensed and sampled by three Hall effect current sensors (*we name these as grid sensors*) and the DSP unit, respectively.

$\vec{U}_{abc}$ and $\vec{I}_{abc}$ vectors are also sinusoidal quantities, and they are converted into their dq axis components using a Clarke and a Park matrix. Let us denote $\vec{U}_{dq}$ and $\vec{I}_{dq}$ as the dq transformations of $\vec{U}_{abc}$ and $\vec{I}_{abc}$, respectively. The term $\vec{U}_{dq}$ comprises u_d and u_q where u_d and u_q are the d and q axis components of $\vec{U}_{abc}$. The term $\vec{I}_{dq}$ similarly comprises i_d and i_q where i_d and i_q are the d and q axis components of $\vec{I}_{abc}$.

A loop filter with inductance L is present between $\vec{S}_{abc}$ and $\vec{U}_{abc}$ for signal smoothing. The relation between $\vec{S}_{abc}$ and $\vec{U}_{abc}$ can be simplified using their dq axis components (e_d, e_q and u_d, u_q) and finally can be expressed as:

$$u_d = e_d + L\frac{di_d}{dt} - \omega L i_q \tag{2.4}$$

$$u_q = L\frac{di_q}{dt} + \omega L i_d \tag{2.5}$$

Generation of Reference Currents (i_d^*, i_q^*): Two reference points, which are i_d^* and i_q^*,

control the real and reactive power set points of the inverter. The solar panel output voltage V_T and current I_T are sensed by *two separate Hall voltage and current sensors* (*we name these as solar panel sensors*). V_T and I_T are given as inputs to a Maximum Power Point Tracking (MPPT) block that generates reference point i_d^* to track the maximum available real power from the panel. The other reference point i_q^* is generated from the reference reactive power Q^*, which is provided by the facility's energy management systems using a Wide/Local Area Network [77].

Proportional-Integral (PI) Current Controllers: Two separate PI current controllers force the dq axis components i_d and i_q to track the reference set points i_d^* and i_q^*. This tracking generates fractional DC voltages u_d^p and u_q^p as follows:

$$u_d^p = K_p(i_d^* - i_d) + K_i \int (i_d^* - i_d) \tag{2.6}$$

$$u_q^p = K_p(i_q^* - i_q) + K_i \int (i_q^* - i_q) \tag{2.7}$$

where K_p and K_i are the proportional and integral constants of the PI controllers. The term i_d^* is related with real power and i_q^* is related with reactive power. By tracking these two quantities, PI controllers control the correct injection of real and reactive power into the grid (Eqn. 2.6, 2.7).

Space Vector Pulse Width Modulation (SVPWM) Block: The SVPWM block, which generates appropriate *pulse width modulated* signals, controls the MOSFET switches and generates appropriate 3-phase inverter output voltages u_a, u_b, and u_c. The SVPWM block uses two reference signals u_d^* and u_q^*, which are generated by putting Eqn. 2.6, 2.7 into Eqn. 2.4, 2.5:

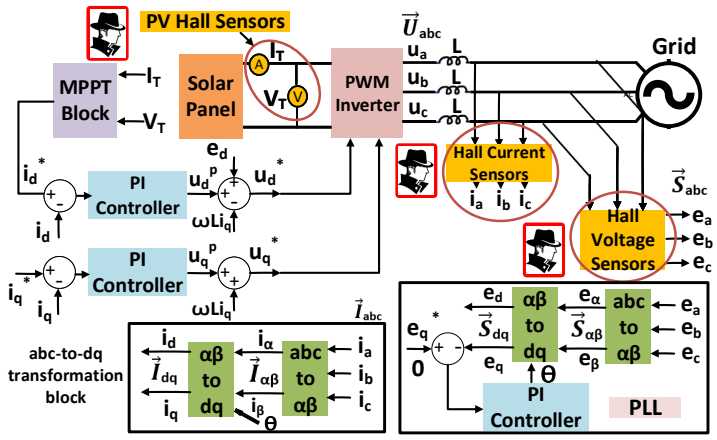


Figure 2.5: Typical controllers inside of a 3-phase inverter.

$$u_d^* = e_d + u_d^p - \omega L i_q \tag{2.8}$$

$$u_q^* = u_q^p + \omega L i_d \tag{2.9}$$

Note that, the reference voltages u_d^* and u_q^* depend on reference currents i_d^* and i_q^*, dq components of grid currents i_d and i_q, angular frequency ω, and filter inductance L.

Phase Locked Loop (PLL) Block: PLL synchronizes the inverter output frequency with the grid frequency by implementing the following equation [78]:

$$\begin{bmatrix} e_d \\ e_q \end{bmatrix} = \begin{bmatrix} \cos\theta^* & \sin\theta^* \\ -\sin\theta^* & \cos\theta^* \end{bmatrix} \begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix} = k \begin{bmatrix} \cos(\theta - \theta^*) \\ \sin(\theta - \theta^*) \end{bmatrix} \tag{2.10}$$

where k is a constant, θ and θ^* are the instantaneous phase angles (i.e., frequency) of the grid and inverter output voltage, respectively. The PI controller of the PLL tries to equal e_q with e_q^*. Therefore, if the reference value e_q^* is set to 0 (generated internally), e_q in Eqn. 2.10 will be also close to 0. This causes $\sin(\theta - \theta^*) = 0$ (i.e., $\theta = \theta^*$) in Eqn. 2.10. This results in grid-synchronization, because the inverter output voltage $\vec{U}_{abc}$ has the same phase

(i.e., $\theta = \theta^*$) as the grid voltage $\vec{S}_{abc}$.

Single Phase Grid Controllers: A single-phase grid-tied inverter has similar blocks as the three-phase, except it does not have Clarke matrix transformation, but it uses *Phase Shifters*. As it has a similar controller, an adversary can similarly affect it using the same attack methodology.

2.7 Experimental Setup

2.7.1 A Scaled-Down Testbed of a Power Grid

To avoid safety concerns related to high voltage and high power experiments, we have created a scaled-down version of a real grid in our lab (Fig. 2.6) to validate our attack model. A 140 W grid-tied inverter kit (part# = TMDSOLARUINVKIT) from Texas Instruments Inc. is used. This is a scaled-down version of a practical solar inverter. This inverter has a Piccolo-B control card (C2000 microcontroller) that implements all the controller blocks (e.g., PLL, Park & Clarke transformations, PI controllers, MPPT, SVPWM, etc.). The supported software kit is downloaded from *ControlSUITE*, then compiled using *Code Composer Studio 9.1.0* IDE, and then flashed into the solar inverter kit. The Solar panel is emulated by a DC Power source (Part# = PSB 2400L2). An isolated and stable grid is created using another inverter (Part# = BESTEK) with a 300 W load. The 140 W target solar inverter is connected with this stable grid to emulate a weak grid. Oscilloscopes (Part# = Tektronix TDS2022C) with differential probes (Part# = Yokogawa 700924 Probe 1400V / 100 MHz) and multimeters are used to measure the inverter output voltage, current, and power before and after the attack. In order to assist the understanding for readers, attack demonstration and results are shown in a video in the following link: `https://sites.google.com/view/usenix-spoofing/home`

2.7.2 Feasibility Analysis of the Attack

The feasibility of this attack methodology depends upon the following three key factors: (i) The location of the Hall sensors, (ii) The barrier and EM shielding around the inverter, and (iii) The amount of MMF required to overcome the barrier and influence the Hall sensors.

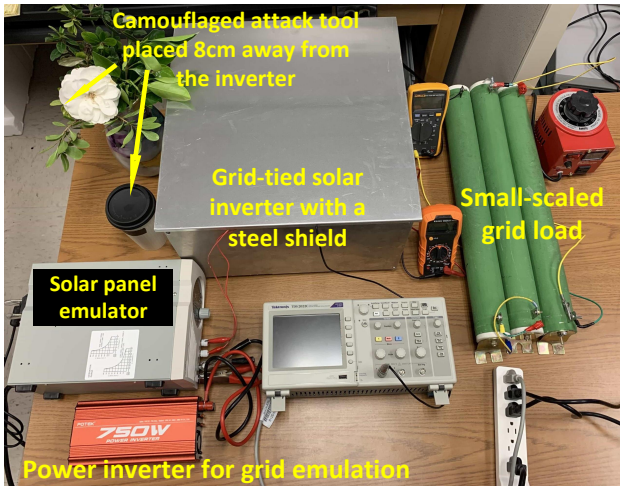


Figure 2.6: A scaled-down testbed of a power grid.

As Hall sensors measure the voltage/current, they normally are placed nearby where the solar panel and the grid voltage cables enter the inverter board. Therefore, the *PV Connection side* and the *Grid Connection side* are two suitable locations to place the camouflaged attack tool near the inverter (Fig. 2.7). The Hall sensors are within 4 cm from the board edge for our experimental inverter. This information regarding the location of the Hall sensors is essential to optimal placement of the attack tool and thus maximizing the attack's impact.

The generated MMF by the electromagnet should be strong enough to overcome the following two barriers: (i) The air gap between the body of the inverter and the electromagnet, and (ii) The metallic shield around the inverter.

Most of the generated MMF is used to overcome the air gap barrier because air has a very high magnetic reluctance. The more the air gap (the distance between the inverter and the electromagnet) is, the more MMF is required to overcome the distance. After penetrating

the air, the remaining MMF is used to penetrate the shield around the solar inverter. If the shield is non-magnetic (e.g., aluminum, tin, brass, stainless steel, etc.) or non-metallic (e.g., plastic, polycarbonate, etc.), the remaining MMF can easily penetrate the shield. If the shield is made of ferromagnets (e.g., steel, etc.), the remaining MMF should be strong enough to saturate the ferromagnetic shield, so that its magnetic shielding property gets diminished [79]. For example, 0.6 Tesla magnetic flux density is sufficient to saturate steel shield [80].

Is it possible to generate that much MMF by our *Embedded Hall Spoofing Controller*? We discuss some comparative numbers here to answer this question. It is possible to make a 0.1 Tesla to 2 Tesla powerful lab magnet with 500-9000 turns on an iron core [81]. A coin-sized neodymium magnet has 0.5-1.25 Tesla [82] and a typical loudspeaker magnet has 1-2.4 Tesla [83] magnetic strength. Our experimental electromagnet has approx. ∼4000 turns that can generate up to 0.8 Tesla with the mentioned battery pack. This is sufficient to spoof the Hall sensors of an inverter from at most 10 cm distance. Here we consider a steel shield around the inverter. By investing more money (¿$50) on the magnetic core (e.g., neodymium–iron–boron ($Nd_2Fe_{14}B$) rare earth magnet [82]), we can shrink the size of the electromagnet and make it stronger to spoof from 10+ cm distance.

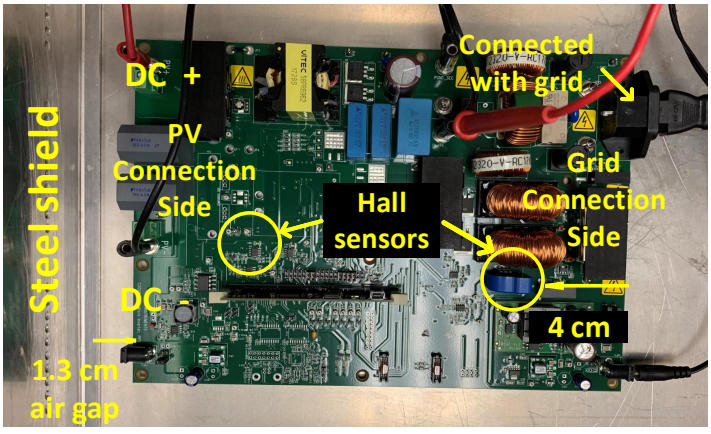


Figure 2.7: Typical locations of Hall sensors inside an inverter.

2.8 Attack Model Validation

In this section, we validate our proposed attack model, which is explained in Section 2.5, in our lab testbed for 5 different scenarios. We also explain how the attack propagates from the sensor to the inverter controller by using suitable equations.

It is clear from Section 2.6.3 that grid voltage $\vec{S}_{abc}$ can control the inverter output voltage $\vec{U}_{abc}$ (Eqn. 2.8, 2.9) and phase angle θ (Eqn. 2.10); inverter output voltage $\vec{U}_{abc}$ and real power P depend on output current $\vec{I_{abc}}$ (Eqn. 2.8, 2.9), solar panel voltage V_T, and current C_T; and inverter reactive power Q depends on output current $\vec{I_{abc}}$ and reference i_q^*. The above dependency information is important from the attacker's perspective and can be formulated mathematically as follows:

$$\begin{aligned} \theta &= f(\vec{S}_{abc}); \quad \vec{U}_{abc} = f(\vec{S}_{abc}, \vec{I}_{abc}, V_T, I_T) \\ P &= f(\vec{I}_{abc}, V_T, I_T); \quad Q = f(\vec{I}_{abc}, i_q^*) \end{aligned} \tag{2.11}$$

where $f(.)$ is the function notation.

2.8.1 Attacking Grid Synchronization

Two conditions must be satisfied to synchronize the inverter with the grid [60]: (i) inverter output voltage $\vec{U}_{abc}$ must be slightly higher than the grid voltage $\vec{S}_{abc}$, and (ii) inverter voltage phase θ must be same as the grid voltage phase.

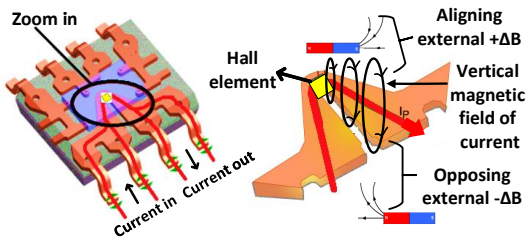


Figure 2.8: Aligning and opposing spoofing into Hall sensors.

Scenario 1: Let us assume the attacker spoofs only the grid voltage ($\vec{S}_{abc}$) sensors with a constant ±MMF (aligning and opposing polarity). Therefore, the attacker considers injecting magnetic field $\pm\Delta B$ into the Hall grid voltage sensors. The term $+\Delta B$ means that the applied +MMF aligns vertically in the same *direction* of the Hall sensor measurement axis, and $-\Delta B$ means that the applied -MMF aligns vertically in the *opposite* direction of the Hall sensor measurement axis (Fig. 2.8). An injection of $\pm\Delta B$ results in a ***false*** Hall voltage V_{Hall}^{f}; therefore Eqn. 2.1 may be expressed as follows:

$$V_{Hall}^{f} = k\{\frac{I_{bias}}{d} \times (B \pm \Delta B)\} \tag{2.12}$$

V_{Hall}^{f} causes injection of false voltages, which include $\pm\Delta E_a, \pm\Delta E_b$, and $\pm\Delta E_c$ (± for ±MMF), into grid voltage vector $\vec{S}_{abc}$. Therefore, Eqn. 2.2 is changed as follows:

$$\vec{S}_{abc}^{false}(t) = \begin{bmatrix} e_a \pm \Delta E_a \\ e_b \pm \Delta E_b \\ e_c \pm \Delta E_c \end{bmatrix} \tag{2.13}$$

where $\pm\Delta E_a, \pm\Delta E_b$, and $\pm\Delta E_c$ may be different from each other. The low-pass filter of the DSP unit cannot filter out these false voltages. So, $\vec{S}_{abc}^{false}$ propagates to the following *abc-to-dq* transformation block. This affects Eqn. 2.3 as follows:

$$\vec{S}_{dq}^{false} = \begin{bmatrix} e_d \\ e_q \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{3}{2}}E \\ 0 \end{bmatrix} \pm PM \times \begin{bmatrix} \Delta e_{\alpha} \\ \Delta e_{\beta} \end{bmatrix} \tag{2.14}$$

where $PM \times \begin{bmatrix} \Delta e_{\alpha} \\ \Delta e_{\beta} \end{bmatrix}$ is a time-varying quantity. Terms Δe_{α} and Δe_{β} are the errors prop-

agating from the Clarke matrix transformation block. Therefore, $\vec{S}_{dq}^{false}$ is no longer stable, and as a result, e_d and e_q change with time (i.e., $e_q \neq 0$). This influences the *Right-Hand Side* (R.H.S) of Eqn. 2.8 and 2.9. As a result, reference voltages u_d^* and u_q^* are perturbed. This will force SVPWM to create a false inverter output voltage vector $\vec{U}_{abc}^{false}$. It is possible to generate a larger or smaller $\vec{U}_{abc}^{false}$ than allowed. A larger $\vec{U}_{abc}^{false}$ than the grid voltage $\vec{S}_{abc}$ can cause high transient current to be pushed into the grid. If $\vec{U}_{abc}^{false}$ is smaller than $\vec{S}_{abc}$, the inverter acts as a load and current flows into the inverter from the grid. Both cases can shut down the inverter or may damage the inverter by frying the electronics.

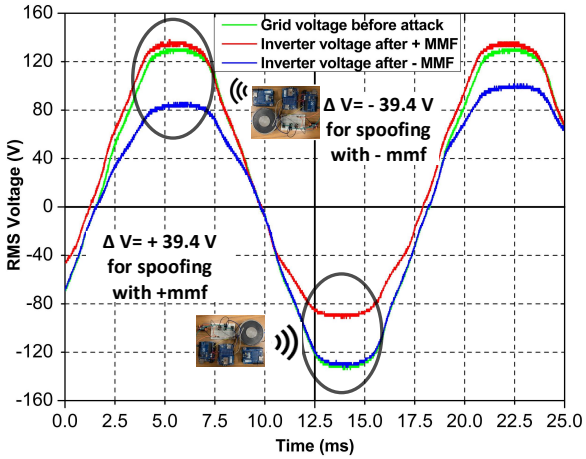


Figure 2.9: Spoofing grid-tied inverter output voltage.

Scenario 2: Let us assume the attacker spoofs only the grid current ($\vec{I}_{abc}$) sensors with a constant ±MMF. An injection of ±MMF results in a ***false*** Hall voltage V_{Hall}^f, which causes an injection of $\pm\Delta I_a, \pm\Delta I_b, \pm\Delta I_c$ measurement errors into $\vec{I}_{abc}$. This causes a false output current $\vec{I}_{abc}^{false}$. The low-pass filter of the DSP unit cannot filter out this false signal. This propagates to the following *abc-to-dq* transformation block and creates a false current $\vec{I}_{dq}^{false}$. This affects Eqn. 2.6 and 2.7 as follows:

$$u_d^f = K_p(i_d^* - i_d^{false}) + K_i \int (i_d^* - i_d^{false}) \tag{2.15}$$

$$u_q^f = K_p(i_q^* - i_q^{false}) + K_i \int (i_q^* - i_q^{false}) \tag{2.16}$$

Generated false voltages u_d^f and u_q^f influence the R.H.S of Eqn. 2.8 and 2.9. As a result, reference voltages u_d^* and u_q^* are perturbed. This will force SVPWM to create false inverter output voltage vector $\vec{U}_{abc}^{false}$. Similar to the consequences in Scenario 1, this may shut down the inverter.

The attack Scenario 2 is demonstrated in our testbed by spoofing a grid current sensor using 0.8 Tesla from a 7.8 cm distance (Fig. 2.9). The attacker causes an increase in the inverter output voltage from -125 V to -85.6 V (ΔV = + 31.52%) by +MMF spoofing and causes a decrease from +125 V to +85.6 V (ΔV = - 31.52%) by -MMF spoofing. This creates a sudden mismatch between the inverter output voltage and the grid voltage. This mismatch forces the inverter to shut down.

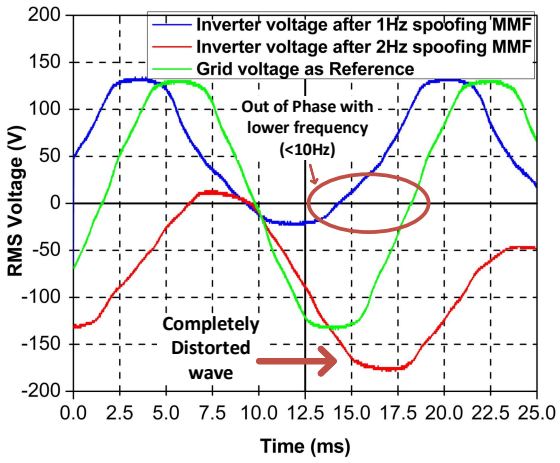


Figure 2.10: Spoofing grid-tied inverter output frequency.

Scenario 3: Let us assume the attacker spoofs only the grid voltage ($\vec{S}_{abc}$) sensors with a sinusoidal MMF (*note that the last two scenarios are for constant MMF*). An injection of a sinusoidal MMF results in a ***false*** Hall voltage $V_{Hall}^f(t)$, which causes an injection of $\Delta E_a(t)$, $\Delta E_b(t)$, $\Delta E_c(t)$ measurement errors into $\vec{S}_{abc}$. Therefore, Eqn. 2.2 is changed as follows:

$$\vec{S}_{abc}^{false}(t) = \begin{bmatrix} e_a + \Delta E_a(t) \\ e_b + \Delta E_b(t) \\ e_c + \Delta E_c(t \end{bmatrix} = \begin{bmatrix} E_{1a}^f \cos(\omega t + \theta_a^f) \\ E_{2a}^f \cos(\omega t + \theta_b^f) \\ E_{3a}^f \cos(\omega t + \theta_c^f) \end{bmatrix} \tag{2.17}$$

where E_{1a}^f, E_{2a}^f, E_{3a}^f and θ_a^f, θ_b^f, θ_c^f are false amplitudes and phase angles, respectively. Thus $\vec{S}_{abc}^{false}$ has different phase angles and amplitudes than $\vec{S}_{abc}$. The low-pass filter of the DSP unit cannot filter out this injected low frequency (< 2Hz) error, and the error propagates to the following PLL block of the controller. Hence, the R.H.S of the Eqn. 2.10 is given by:

$$\begin{bmatrix} e_d \\ e_q \end{bmatrix} = \begin{bmatrix} \cos\theta^* & \sin\theta^* \\ -\sin\theta^* & \cos\theta^* \end{bmatrix} \begin{bmatrix} e_\alpha^f \\ e_\beta^f \end{bmatrix} = k \begin{bmatrix} \cos(\theta^f - \theta^*) \\ \sin(\theta^f - \theta^*) \end{bmatrix} \tag{2.18}$$

where e_α^f and e_β^f are propagated errors that cause false phase angle θ^f of the grid voltage. The PLL of the inverter tries to lock with the attacker provided phase angle θ^f (i.e., $\theta^* = \theta^f$). This causes a frequency mismatch between the grid and the inverter voltage. This frequency mismatch causes frequency oscillations and may cause grid failures in weak grids.

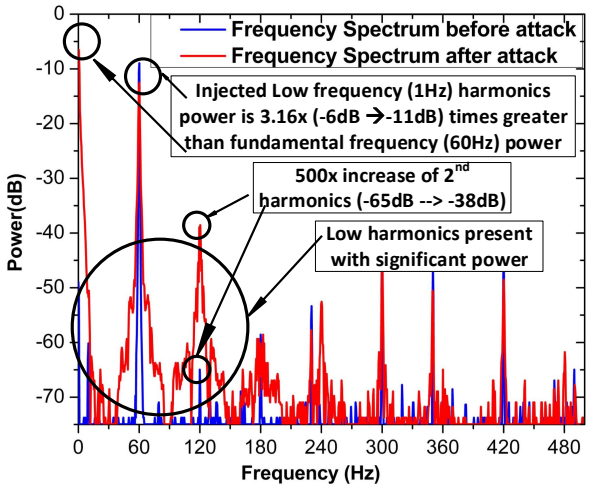


Figure 2.11: The frequency spectrum of the inverter output voltage before and after the attack Scenario 3.

The attack Scenario 3 is demonstrated in our testbed and the outcome is shown in Fig. 2.10. The attacker injects 0.8 Tesla magnetic pulse (1Hz) from a 7.8 cm distance into the grid voltage sensors. This causes the inverter output frequency to go out of phase. The output voltage shape is completely distorted when the attack tool is placed within 1 cm of the inverter (extreme scenario). Fig. 2.11 shows the *Fast Fourier Transform (FFT)* analysis of the inverter output voltage. The frequency spectrum reveals the strong presence of low-frequency components (<10Hz) and indicates that low frequency (1Hz) power is 3.16x (-6dB to -11dB) more than the fundamental frequency (60Hz) power during the attack. This distorted output wave shuts down the inverter, and blackout occurs in the testbed.

2.8.2 False Real/Reactive Power Injection

The attacker can attack $\vec{I}_{abc}$, V_T, or I_T sensor depending upon his resources to perturb the real power or reactive power injection (Eqn. 2.11). Note that, three current sensors are placed in the AC section of the inverter to measure $\vec{I}_{abc}$, and one voltage sensor and one current sensor are placed in the DC section of the inverter (*note that we name these as solar panel sensors*) to measure the solar voltage V_T or the current I_T.

Scenario 4: Let us assume the attacker wants to perform a real power injection attack; therefore, the attacker considers attacking either V_T or I_T sensor by spoofing with a constant MMF (a.k.a. exerting external ΔB). This may create a false Hall voltage V_{Hall}^f. The false V_{Hall}^f causes a false solar panel voltage V_T^f or a current I_T^f as follows:

$$V_T^f = V_T + \Delta V_T \; and \; I_T^f \; = I_T + \Delta I_T \tag{2.19}$$

where ΔV_T or ΔI_T are due to the attacker's false MMF injection into the sensor. This false signal V_T^f or I_T^f is fed into the MPPT algorithm. Several algorithms [84], such as Perturb and Observe, Incremental Conductance, Parasitic Capacitance, and Constant Voltage are

used as MPPT algorithms and none of these can filter out the injected error $\Delta V_T/\Delta I_T$. As a consequence, the MPPT block generates a false reference current i_d^{*f}. The PI current controller (Section 2.6) tracks (Eqn. 2.6) the false i_d^{*f} and generates false u_d^f as follows:

$$u_d^f = K_p(i_d^{*f} - i_d) + K_i \int (i_d^{*f} - i_d) \tag{2.20}$$

u_d^f can change the input reference voltage of the SVPWM (Eqn. 2.8, 2.9) causing more or less injection of real power than required into the grid. This phenomenon may alter the demand response of the grid and can be critical in a weak grid. The results of this scenario are discussed in detail in Section 2.8.3.

Scenario 5: Let us assume the attacker wants to perform a reactive power injection attack; therefore, the attacker considers attacking the $\vec{I}_{abc}$ sensors (Eqn. 2.11). The attacker can use pulsating square ($\sqcap$) MMF (as a square wave generation is easier than the sine wave generation) to spoof the $\vec{I}_{abc}$ sensors. It creates pulsating perturbation $\Delta I_{\sqcap}(t)$ with frequency $\omega_{\sqcap}$, which may be expressed as: $\Delta I_{\sqcap}(t) = sgn(\sin(\omega_{\sqcap} t))$, where sgn is the signum function. The pulsating error $\Delta I_{\sqcap}(t)$ may cause pulsating voltage $V_{Hall}^{f\sqcap}(t)$ (Eqn. 2.12). This false $V_{Hall}^{f\sqcap}(t)$ results in an injection of pulsating $\Delta I_{a\sqcap}(t)$, $\Delta I_{b\sqcap}(t)$, $\Delta I_{c\sqcap}(t)$ measurement errors into $\vec{I}_{abc}$ as follows:

$$\vec{I}_{abc}^{false}(t) = \begin{bmatrix} I\ \cos\omega t + sgn(\sin(\omega_{\sqcap} t)) \\ I\ \cos(\omega t - 120^0) + sgn(\sin(\omega_{\sqcap} t)) \\ I\ \cos(\omega t + 120^0) + sgn(\sin(\omega_{\sqcap} t)) \end{bmatrix} \tag{2.21}$$

The pulsating false current $\vec{I}_{abc}^{false}(t)$ creates a pulsating q-axis current $i_q^{\sqcap}$ after the *abc-to-dq* transformation (Section 2.6). PI current controller cannot properly track the i_q^* due to the pulsating nature of $i_q^{\sqcap}$. As a result, a pulsating error voltage is produced (Eqn. 2.7) that

causes a pulsating push of reactive power into the grid. This may cause fluctuation in the grid voltage. And for a weak grid scenario, this fluctuation for a long time may be detrimental for the grid health. As our setup does not have reactive power injection capability, we have shown the impacts of this scenario via simulation using the commercially used software Etap (Section 2.9.2).

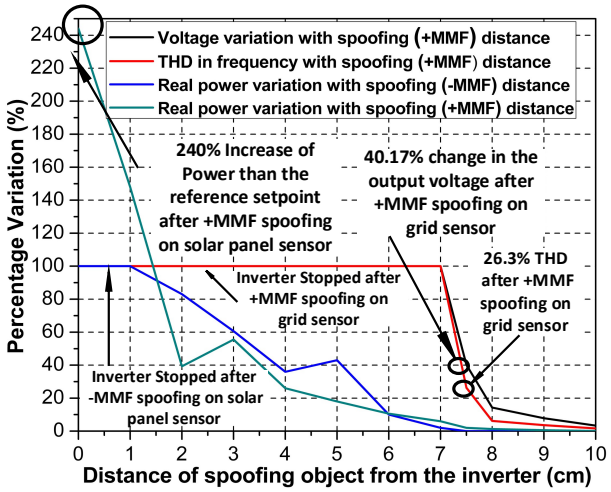


Figure 2.12: Attack effects with different spoofing-distance.

2.8.3 Attack-Impact with Spoofing-Distance

Fig. 2.12 shows the impact of the attack scenarios for different spoofing-distances for 0.8 Tesla magnetic field. Here, spoofing-distance means the distance between the electromagnet and the sensor. Note that attack scenarios 1, 2, 3 are created by spoofing the grid voltage/current sensors, and scenario 4 is created by spoofing the solar panel voltage/current sensors. For scenarios 2 and 3, 40.17% output voltage variation and 26.3% Total Harmonic Distortion (THD) in output frequency are noted, respectively, for 7.5 cm of spoofing-distance. *The THD value refers to the magnitude of harmonics (i.e., due to injected errors) present in the frequency.* The inverter is shut down if the spoofing-distance is less than 7 cm for scenarios 2 and 3. This is shown as a flat line (100% variation) in Fig. 2.12. For attack scenario 4, real power injection increases from 45 W to 155 W (240% increase) for +MMF spoofing, and the inverter is shut down for -MMF spoofing for 1cm spoofing-distance. The attack impact

prevails up to 10 cm for scenarios 2, 3 and up to 8 cm for scenario 4 in our experimental setup. Note that MMF follows the inverse square law with distance ($MMF \propto 1/distance^2$). However, inverter power, voltage, and frequency may not change by following the inverse square law. The reason for this is that the relevant controllers are nonlinear and they may add higher order poles and zeros. Fig. 2.12 supports this claim. It shows that real power, voltage, and frequency change in inverse of higher order (greater than inverse square) with distance. Moreover, voltage and frequency vary significantly compared to power. This indicates that voltage and frequency are more sensitive than power to distance.

2.8.4 Controlling Inverter Voltage and Power

The generated MMF from the electromagnet depends upon power, and this power is supplied by the battery pack. The attacker can remotely send adversarial commands (i.e., duty-cycle) using the Zigbee to control the input power to the electromagnet (i.e., *spoofing-power*). The *Embedded Hall Spoofing Controller* can vary the *spoofing-power* according to the received adversarial command. This results in varying MMF exerted to the inverter. As our attack model is noninvasive, the direct feedback from the compromised Hall sensor to the *Embedded Hall spoofing Controller* is absent. Rather, the ultrasonic sensor provides specific information about the distance between the inverter and the attack tool. This information acts as a weak feedback to control the *spoofing-power* and this can be utilized to control the inverter voltage and power from a specific distance.

Duty-Cycle Variation: The *spoofing*-power can be controlled from a specific distance by using a PWM technique. PWM is used to vary the duty-cycle (i.e., active/on-time) of the relevant MOSFET. Fig. 2.13 shows that by varying the duty-cycle of a signal of 100Hz from 0% to 100%, the attacker can change the power input to the electromagnet from 0 W to 50 W and can control the output voltage and the real power of the inverter (Eqn. 2.12, 2.15, 2.16, and 2.20 give more insights). This experiment is conducted by placing the electromagnet 5

cm away from the sensors. When the magnetic field is applied to grid sensors, the output voltage of the inverter changes in sub-linear fashion from 0% to 34%, up to 32 W of input power to the electromagnet. The inverter stops working after this point, and this is shown as a flat line (100% variation). When the magnetic field is applied to solar panel sensors, the real power output of the inverter changes in sub-linear fashion from 0% to 38%, up to 50 W of input power to the electromagnet. The battery pack can provide this amount of power as this power is required only for a few seconds. Fig. 2.13 shows that the 35 W power applied to grid sensors may turn off the inverter, but the same power applied to solar panel sensors may not do the same. This indicates that the inverter is more sensitive to its grid voltage variation than its real power variation.

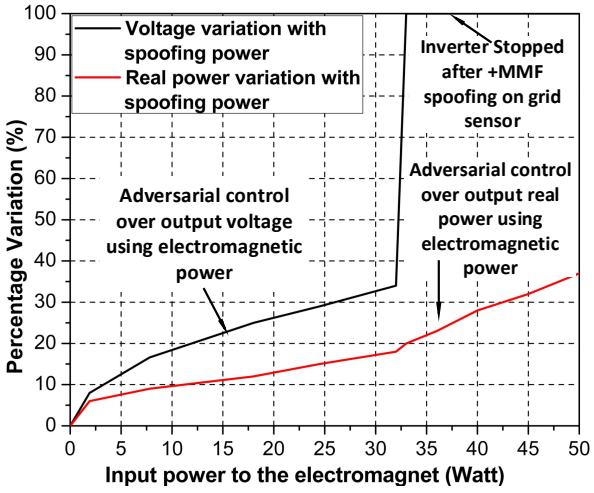


Figure 2.13: Attack effects with different spoofing-power.

2.9 Attack Evaluation in a Practical Grid

In Section 2.8, different attack scenarios are demonstrated using a 140 W inverter in our testbed. However, in this section, an industry used software, the *Electrical Power System Analysis & Operation Software Etap 19.0.1,* is used to show the impacts and the consequences of the previously explained attack scenarios in the context of a large grid.

The IEEE 13 bus test grid is used to model a medium-sized isolated grid with 2.3 MW and

1.536 MVar distributed loads (typical size of a substation/micro-grid representing approx. $\sim$ 150 houses) to demonstrate the attack consequences (Fig. 2.14). The test grid has five distributed generators and a lumped solar inverter. The generators and the inverter have ranges of 1000 MW, 500 kW, and 100 kW generation rating. Let us assume that the attacker has chosen the comparatively small 100 kW inverter (Gen 5) to show how attacking a small generation could eventually collapse the entire grid. It is important to note that a single inverter can bring down the entire network if the grid is weak, the inverter size is large compared to other generators, or the grid does not have the inertia to compensate for the sudden load change. Usually, residential inverters (0.1 kW-10 kW) are too small to bring down the entire network. Rather, in this section, we address the impact of compromising a larger inverter (e.g., 100 kW) in detail.

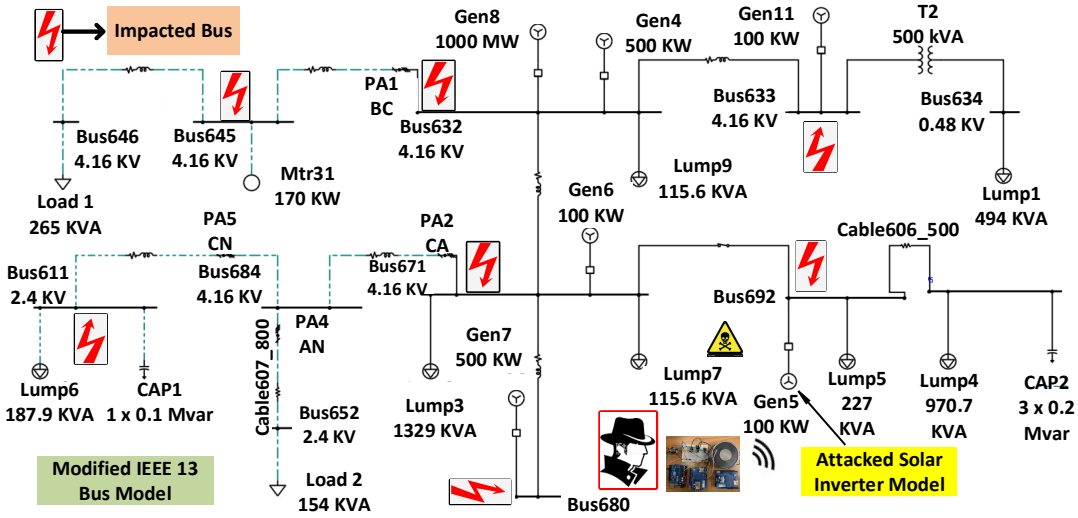


Figure 2.14: IEEE 13 bus model simulation in Etap to demonstrate the attack impacts in a large system.

Feasibility Analysis of using a 100 kW Inverter: Large inverters (e.g., 100 kW) normally exist as the central inverter in solar/industrial plants or shopping malls. To the best of our knowledge [85–87], the inverters have abc-to-dq transformation blocks, PI controllers, PLLs, MPPT, SVPWM in common, irrespective of their sizes (see Section 2.6.3). These high power central inverters are normally connected with high voltage DC (> 600V) and AC

(∼480V) lines, and overall good efficiency (>98%) is a critical requirement of these inverters. To increase efficiency, they are designed as an iron-core transformerless system. However, this way of design increases the injection of DC voltage/current and circulating current into the grid. These injections of unwanted signals can cause overloading in the distribution transformer. Therefore, tight control is necessary to overcome these shortcomings, and accurate measurement is the key to obtaining this control. Thus, designers commonly use Hall sensors because of their lower measurement error, better linearity, higher efficiency, and better galvanic isolation. Hall sensors are used to find DC current injection and measure ground leakage current and circulating current in the inverter's power stage [85] [88]. Fig. 2.15 is a teardown of a 100 kW inverter, which is obtained by contacting the designers of the relevant inverter [85]. *This figure clearly shows the presence of Hall voltage and current sensors inside of it and gives a strong insight of using a 100 kW inverter in our simulation.* The PV and grid voltage sensors are LV 25-P, and the leakage current sensors are CT 0.4-P, the circulating current sensors are HO-6P, and the grid current sensors are LA 100-TP. These sensors are present within 4.2 cm from the edge, therefore, these sensors are within the attack range. The enclosures of these inverters are made of steel, aluminum, or non-metallic poly-carbonate. Metallic enclosures often get hot due to sunlight, and it is detrimental for the inverter. Therefore, manufacturers prefer non-metallic poly-carbonate [89] as an enclosure, which is heat-resistant but more fragile to our attack model. *As we can't access a high voltage inverter for safety reasons, our experiments use the miniature inverter having core functionalities similar to an industry-standard inverter. It is clear from Table 2.1 and the above discussion that highly efficient small, medium, and large grid-tied inverters have Hall sensors. This gives a strong intuition behind the generalization of our attack model.*

2.9.1 Grid Synchronization Attack Evaluation

Inverters are typically connected with the power grid using protective relays at the point-of-interface (POI). These protective relays have under/over frequency, rate of change of

frequency , under/over voltage detection schemes. If the frequency/voltage changes fast or goes beyond the threshold set by the standard (e.g., IEEE 1547, IEEE 2030), the relays trip out the corresponding inverters/loads from the POI.

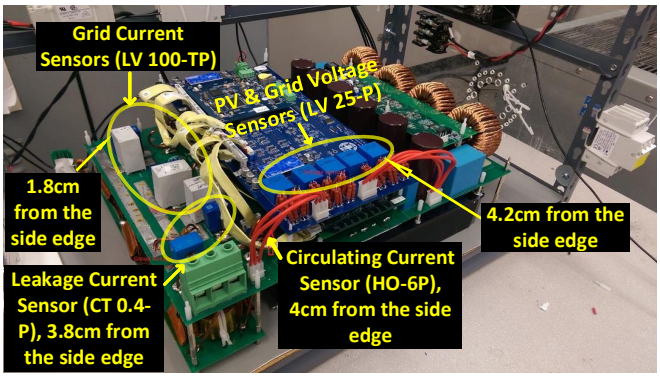


Figure 2.15: Feasibility analysis of using a 100 kW inverter.

The attacker can perturb output voltage, phase, and frequency of the 100 kW (Gen 5) target inverter by using our attack model (Scenario 1, 2, 3 of Section 2.8). This can lead to any of the following consequences: the inverter can be damaged, it can be shut down, or connected protective relays can trip it out from the connected grid. Any of these consequences can result in a sudden loss of 100 kW power from the grid.

Explanation of Cascading Grid Collapse [90]: The grid power generation should be equal to the sum of power consumption and loss. This balance needs to be maintained for a stable grid health. As the 100 kW inverter stops working without *prior notice, anticipation, or preparation*, it will shift its 100 kW load to nearby generators. Those nearby generators will be overloaded and will shift their loads onto other generators in a cascading manner in a very short time, eventually causing grid collapse. This effect can be extreme during peak hours when the generators are already running at maximum capacity and may be unable to compensate for this 100 kW sudden mismatch between generation and demand. Moreover, when the 100 kW inverter stops working, the adjacent generator's *governor set point* is also changed to push kinetic energy into the grid to catch up with this power disparity. When generators adjust their governors, power system frequency falls and blackout is required in

the affected part to preserve the power system. *Due to the grid weakening, this frequency fluctuation is an important issue, and the attacker can leverage this vulnerability by using our attack model.*

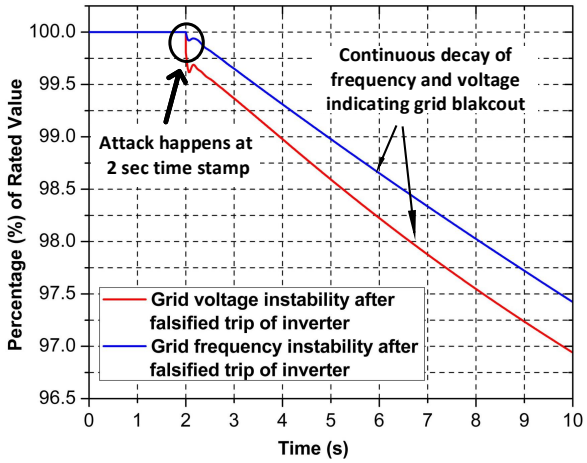


Figure 2.16: Grid voltage and frequency instability in IEEE 13 bus model after the grid synchronization attack.

This is demonstrated in Fig. 2.16 by simulating in Etap 19.0.1. The simulation is run for a 10-second window. The attacker attacks the inverter at $t = 2$ second. After this point, the grid voltage and frequency start continuously decaying and fall to 97% of the rated values within 8 sec. IEEE 1547 standard [91] indicates that the grid will shut down as the grid frequency is out of this range: 59.3 Hz $<$ frequency $<$ 60.5 Hz. This may result in a blackout in the region.

2.9.2 Real and Reactive Power Injection Attack

Section 2.8.4 explains that the attacker can force the inverter to inject more or less real/reactive power into the grid by duty-cycle variation. Let us consider a scenario where the grid is balanced (i.e., generation = consumption) and the 100 kW inverter (Gen 5) is running in under-rated condition (i.e., sending less power into the grid than the rated maximum amount). Suddenly, the inverter (Gen 5) is compromised and pushes excess real/reactive power into the grid because of +MMF spoofing. This sudden push of power (i.e., adversar-

ial control) forces the other nearby generators to regulate their own *governor* set-points to absorb the excess power. As frequency and voltage depend on the set-points of the governors, the sudden swing of the governors can cause temporary grid voltage and frequency dip. This scenario is shown in Fig. 2.17. The adversary attacks the inverter at $t = 2$ second by injecting real/reactive power. This injection causes frequency to fall to 68% and voltage to fall to 15% of the rated value. The attacker can also force the inverter to push less power than the inverter set-point by -MMF spoofing (Section 2.8). If the attacker keeps injecting more/less power into the grid in a periodic fashion (Scenario 5), the nearby generators will continuously change their *governor set-points* and this may create oscillations in grid voltage and frequency. This can cause transient instability and may result in a blackout in the region because of the reasons already described in Section 2.9.1.

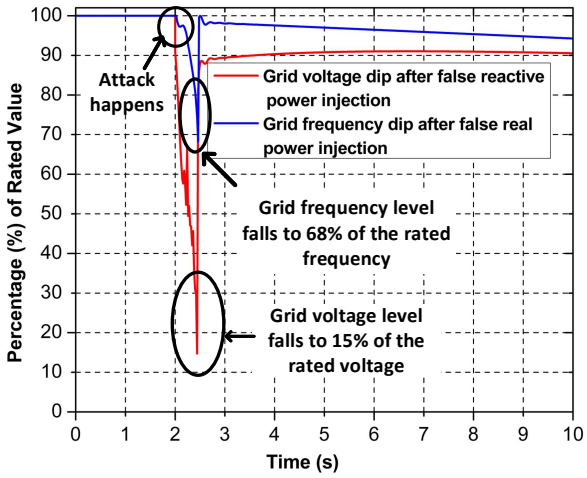


Figure 2.17: Impact of false real and reactive power injection.

2.9.3 Attacking Utility Connected Micro-Grid

Section 2.9.1 and Section 2.9.2 show the impacts of our attack on an isolated grid. Let us consider a scenario where this isolated grid is connected with the utility grid forming a medium-sized micro-grid. Normally, a utility grid having rotational generators is considered as a strong grid, and any grid (i.e., the micro-grid) connected with this strong grid is also considered as strong. A small amount of power and frequency fluctuation in the micro-

grid can be absorbed by the connected strong utility grid. However, a micro-grid becomes weaker as its distance from the utility grid increases. A long transmission line acts as a large impedance between the micro-grid and the utility grid. Voltage/frequency fluctuation in the micro-grid cannot ride through to the utility grid because of this large impedance. As a result, disparities in the micro-grid may not be absorbed by the connected strong utility grid. In large countries like the U.S.A. or China, this far away micro-grid can be easily found (e.g., Borrego Springs, 90 miles east of San-Diego [92]; 6.8 GW Gansu province wind farm project, 1000 miles from the industrial east coast in China [93]; Blue Lake Rancheria, 300 miles north of San Francisco [94], etc.).

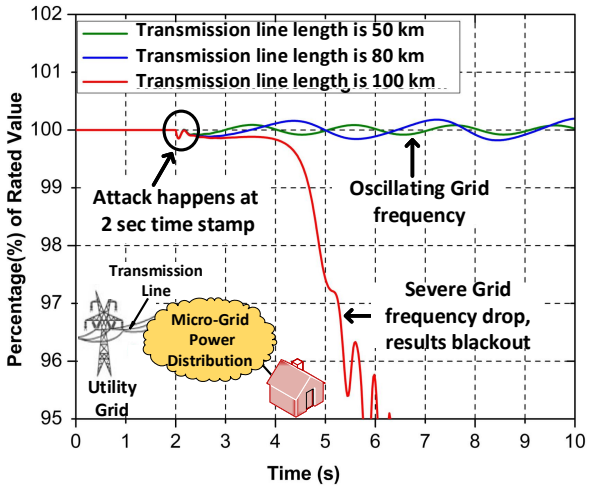


Figure 2.18: Frequency instability in a weak micro-grid.

Etap 19.0.1 simulation in Fig. 2.18 shows that if the transmission line length between the utility and the micro-grid increases, the micro-grid becomes weaker. Scenario 1, 2, 3, 4 can cause the grid frequency to drop in our IEEE 13 Bus model if the transmission line length is *more* than 100 km (i.e., micro-grid is 100 km away from the utility grid). If the distance is less, the micro-grid remains strong and a negligible frequency fluctuation can be present after the attack.

2.10 Defense and Limitations

2.10.1 Defense

The defense against this type of unconventional attack should consider the following four practices together:

Sensing Presence of External Magnetic Field: The first practice is to put a magnetic flux sensor as a guard near the Hall voltage/current sensor device to measure the presence of an external magnetic field. This idea is similar to the presence of a temperature sensor near a MOSFET to shut it down at a higher temperature. Most high power devices use this method to protect a MOSFET from over temperature. Sensing of a high external magnetic field by the guard magnetic sensor can be used to relay the information to the operator about the possible attack situation. It is noteworthy that this guard sensor has a very low chance of getting influenced by the external magnetic field generated by a nearby current-carrying conductor as this magnetic field is very low. For example, a 500 A current-carrying conductor in the power system can generate only 1 mT at 10 cm distance [95], and the attacker's external magnetic field is much greater (¿ 0.8 T) than this. Therefore, the additional magnetic sensor can safely separate the attacker's high spoofing magnetic field from the magnetic field usually present in the power grid.

Secured Surrounding Environment: The second practice is to prevent any visitor or unauthorized personnel from going near the grid-tied solar inverter. Any unauthorized object found near the inverter should be considered as a security breach. Furthermore, any authorized electronic device, which has magnetic capabilities placed near the inverter, should be carefully examined. However, this countermeasure *alone* may fail in a few scenarios that involve large countries where solar plants are usually found in an isolated place with less security. Staggs et al. [69] demonstrated how easily this countermeasure can be defeated and an attacker can access a wind plant in the middle of a remote field.

Shielding: Shields redirect the magnetic fields from sensitive devices. Presence of multiple lamination layers in the shield can increase the robustness against the strong magnetic field. High saturation magnetic flux density material (HB), non-magnetic material (NM) and amorphous alloy material (AM) can be used as lamination layers of the shield [96]. Aluminum and poly-carbonates are not good for shielding and should never be used. The thickness of the shields also matters. *We have increased the thickness of the shield from 2mm to 4 mm and the impact of the attack is reduced by approx. 40%.* The thickness of the shield can also increase the weight making it more inconvenient. Alloys, such as CO-NETIC-AA, NETIC S3-6, and MuMETAL, can be used as shields [97] but they are costlier. However, we must remember that having only a good shield is not enough, as any shield can be compromised with a stronger magnetic field.

Robust Sensors: Differential Hall effect sensors can be used because they are robust to external common-mode magnetic interference. The differential Hall effect sensor has two Hall elements, which are closely placed together to cancel out common-mode noises [98]. Sensor-shielding can be added to the Hall sensor to make it insensitive to a small external magnetic field ($< \sim 30$mT) [99]. Moreover, a field concentrator can be added to a Hall sensor to make it robust to an external magnetic field. However, a field concentrator causes magnetic hysteresis, which introduces an additional source of error in the measurement [99].

2.10.2 Limitations

In this paper, the introduced adversarial control does not offer fine-grained control compared to [51, 52]. The reason for this is that the direct feedback from the compromised Hall sensor to the attacker is absent. However, the attack is strong enough to perturb the connected power grid. Our adversarial attack offers limited control over the inverter voltage within a limited range (Section 2.8.4) and exceeding this range can result in a DoS attack as the inverter is very sensitive to output voltage variation. Moreover, close access near the inverter, short-

attacking range, finding the weak grid scenario, and the prior knowledge on the timing of the attack (i.e., peak hours) are also the limitations. Furthermore, the attacker can not inject high frequency (¿2Hz) pulsating MMF, because the inductive property of the electromagnet filters it out.

2.11 Summary

We have proposed and presented a noninvasive attack using the magnetic field on the grid-tied solar inverter. The presence of the Hall sensors in the inverters leaves them vulnerable to be spoofed from the outside. We have illustrated the integrity and availability risks of an inverter by proper mathematical modeling of the basic blocks of the inverter controller. This shows how the false data injection into a Hall sensor can compromise the inverter controller. We have identified five attack scenarios by which the attacker can compromise the inverter and also the connected grid. Moreover, we have introduced a duty-cycle variation approach for adversarial control that can alter the inverter voltage and real power noninvasively. We have tested the attack scenarios in our scaled-down testbed of the power grid and demonstrated our proof of concept. We discuss the feasibility of using a 100 kW inverter and this gives insights behind the generalization of our attack model. We have used industry-standard software Etap 19.0.1 to show the consequences of our attack in a large power grid. This attack can lead to a grid blackout in a weak grid. Our work is an example of a noninvasive attack that originates in the physical domain following some physical laws, compromises the cyber domain, and again finally impacts the physical domain. This can cause financial loss to the power companies. Hence, this attack is novel in power CPSs and it can draw attention to the security community for further research.

Chapter 3

Spreading Deadly Pathogens Under the Disguise of Popular Music

3.1 Abstract

A Negative Pressure Room (NPR) is an essential requirement by the Bio-Safety Levels (BSLs) in biolabs or infectious-control hospitals to prevent deadly pathogens from being leaked from the facility. An NPR maintains a negative pressure inside with respect to the outside reference space so that microbes are contained inside of an NPR. Nowadays, differential pressure sensors (DPSs) are utilized by the Building Management Systems (BMSs) to control and monitor the negative pressure in an NPR. This paper demonstrates a non-invasive and stealthy attack on NPRs by spoofing a DPS at its resonant frequency. Our contributions are: (1) We show that DPSs used in NPRs typically have resonant frequencies in the audible range. (2) We use this finding to design malicious music to create resonance in DPSs, resulting in an overshooting in the DPS's normal pressure readings. (3) We show how the resonance in DPSs can fool the BMSs so that the NPR turns its negative pressure to a positive one, causing a potential *leak* of deadly microbes from NPRs. We do experiments on 8 DPSs from 5 different manufacturers to evaluate their resonant frequencies considering the sampling tube length and find resonance in 6 DPSs. We can achieve a 2.5 Pa change in negative pressure from a ∼7 cm distance when a sampling tube is not present and from a

$\sim$2.5 cm distance for a 1 m sampling tube length. We also introduce an interval-time variation approach for an adversarial control over the negative pressure and show that the *forged* pressure can be varied within 12 - 33 Pa. Our attack is also capable of attacking multiple NPRs simultaneously. Moreover, we demonstrate our attack at a real-world NPR located in an anonymous bioresearch facility, which is FDA approved and follows CDC guidelines. We also provide countermeasures to prevent the attack. The findings in this chapter have been published in [100].

3.2 Introduction

A Bio-Safety Level (BSL) [101, 102] is a set of strict regulations assigned to a biolab or hospital facility to prevent deadly pathogens from being leaked from the facility. The BSL is ranked from BSL-1 (lowest safety level) to BSL-4 (highest safety level) depending on the microbes that are being contained in a laboratory or hospital setting. The Centers for Disease Control and Prevention (CDC) sets BSLs to exhibit specific controls for the containment of microbes to protect the surrounding environment and community.

BSLs require that the isolation rooms in a biolab or infectious-control hospital maintain negative pressure with respect to the outside hallway [101]. Therefore, the room is known as the Negative Pressure Room (NPR). An NPR ensures that potentially harmful microbes cannot leak from the facility through airflow by maintaining negative pressure inside. Therefore, an NPR is critical in preventing deadly bioaerosols from escaping from the facility.

With rising concerns of bioterrorism, an NPR must maintain a certain *negative pressure* following strict regulations established by the CDC, ASHRAE, or other authorities [103, 104]. The Differential Pressure Sensors (DPSs) are commonly used in NPRs to measure the negative pressure in the facility [105]. The DPSs provide the pressure data to the Heating, Ventilation, and Air Conditioning (HVAC) systems, which *maintains* the negative pressure

by controlling the airflow into NPRs [106]. In addition, a Room Pressure Monitoring (RPM) system is also present in NPRs to *monitor* the room pressure [107]. The RPM system also depends on the reading from the DPSs installed in an NPR. Both RPM and HVAC systems are connected with the Building Management Systems (BMSs) for automated control and monitoring of the negative pressure in an NPR.

A DPS has an elastic diaphragm working as a pressure force collector. Therefore, a DPS can be modeled as a second-order dynamic system with a resonant frequency [108]. We demonstrate by thorough experiments that the resonant frequencies of DPSs used in NPRS are typically in the audible range. In addition, we show that the DPS with a sampling tube can be modeled as a Helmholtz resonator, and the resonant frequency of a DPS with a sampling tube still falls within the audible range. This finding is important because an attacker, who has an intention to change the negative pressure in an NPR, may use an audible sound having a resonant frequency to create resonance in a DPS and generate a *forged* pressure to perturb the normal readings of a DPS located in an NPR.

However, a sound having a single-tone resonant frequency will create a "beep"-ish sound, which makes the attack easily identifiable by the authority. Moreover, the HVAC and RPM systems cannot be fooled by a simple resonance in DPS because these systems have a slower response time compared to a resonance. Therefore, a simple resonance in DPS is not enough to turn NPR's negative pressure into a positive pressure to leak airborne pathogens from an NPR.

To solve the above problems, this paper adopts a smart strategy by *disguising* the resonant frequency band inside popular music. The resonant frequencies are inserted as a *segment* into the music for a certain duration in every specific interval. Every inserted segment of the resonant frequency is ended at its peak. Therefore, the corresponding pressure wave inside a DPS also ends at its peak. As a DPS with a sampling tube is a second-order oscillating system [109], the pressure wave does not instantly fall to zero from the peak value. Instead,

the pressure wave starts to attenuate from its peak exponentially. If the interval between two consecutive segments is small, the pressure wave never falls below a certain value. Therefore, a forged pressure is always present inside a DPS having an average value greater than zero. As a result, the malicious music injected into the DPS can fool the controller of HVAC and RPM systems connected with BMSs to change the negative pressure of an NPR into a positive one. Moreover, the segments of resonant frequency are camouflaged in the malicious music so that the attack is not identifiable by the authority. Therefore, we name this attack as "*the wolf in sheep's clothing*" since this strategy ensures stealthiness.

The consequences of changing a negative pressure into a positive one can be catastrophic. If the NPR has an infectious patient admitted or an ongoing bioresearch, the attacker can control the timing of the attack to *leak* a deadly pathogen *from* the NPR. Moreover, an abnormal change in NPR's pressure triggers an alarm that may create chaos in the facility. An attacker can use this chaos to initiate a stronger attack, such as stealing deadly microbes from the NPR or physically attacking the biosafety cabinets in an NPR. Therefore, our attack model is strong and impactful and has the potential to cause tremendous losses in human lives and monetary resources.

Contributions: We have the following technical contributions:

(1) We evaluate eight industry-used pressure sensors from five different manufacturers to show that the pressure sensors used in NPRs have resonant frequencies in the audible range.

(2) We design malicious music disguising the resonant frequencies of DPSs inside of the music to fool the HVAC and RPM systems of an NPR. We show through experiments that this strategy can change the negative pressure of an NPR to a positive one.

(3) We show that the attacker can adversarially control the forged pressure in DPSs by using the malicious music. Moreover, we show that the attacker can also *simultaneously* attack *multiple* NPRs in a facility using our attack model.

(4) We demonstrate our attack model at a real-world NPR located in an anonymous bioresearch facility. The NPR is approved by the Food and Drug Administration (FDA) and follows CDC guidelines. We also provide countermeasures to prevent the attack on NPRs.

Demonstration: The demonstration of the attack is shown in the following link: https://sites.google.com/view/awolfinsheepsclothing/home

3.3 Background

3.3.1 NPR and its importance

An NPR [110] maintains lower pressure inside with respect to the outside reference space. As air typically travels from higher pressure areas to lower pressure areas, NPR ensures that clean air is drawn into the room so that contaminated particles inside the room are not able to escape. This is why NPRs are present in hospitals and biosafety labs as they prevent airborne particles like bacteria and viruses from spreading out from the facility. NPRs are also present in safety-critical facilities, such as pharmacies and clean rooms.

Importance: The safety of NPRs is paramount as spreading airborne microbes from NPRs may result in catastrophic consequences. For example, a deadly fungus belonging to the genus *Aspergillus* is an airborne pathogen that can cause Aspergillosis disease resulting in acute pneumonia and abscesses of the lungs and kidneys [111]. It has a mortality rate of $\sim$100% for people with neutropenia (i.e., low neutrophils). Respiratory tract infections, such as influenza, swine flu, and COVID-19, are great examples of airborne pathogens that result in a worldwide pandemic. Recently, a conspiracy theory has been rumored about the leakage of the COVID-19 as bioweapons from a biolab [112]. In this context, *imagine* an attacker with the intention of spreading infectious disease as bioweapons may target NPRs, where either infected patients are admitted for isolation or research is carried out on deadly pathogens. Therefore, the security of NPRs is critical and is regulated with strict guidelines.

3.3.2 Regulations for NPRs

With rising concerns about bioterrorism and emerging infectious diseases, there has been a greater emphasis on the proper regulations of NPRs. NPRs must follow requirements established by the CDC [103], ASHRAE [104], and healthcare design construction guidelines [113] to correctly manage airborne infections. Different authorities follow their own regulations [114–117] to maintain a certain negative pressure in NPRs (see Table 3.1). For example, CDC requires that NPRs must maintain a negative pressure differential of at least ∼2.5 Pa (i.e., 0.01 inch water column) in a hospital or biolabs and change the air at least 12 times per hour [103]. Moreover, exhaust from NPRs must be allowed to exit directly outside without contaminating exhaust from other locations. In addition, all exhaust air must be discharged through a High-Efficiency Particulate Air (HEPA) filter to prevent any contamination in the environment.

Table 3.1: Regulations for a Negative Pressure Room (NPR).

Country	**Taiwan**	**CDC(USA)**	**AIA(USA)**	**Australia**
Negative pressure	-8 Pa	-2.5 Pa	-2.5 Pa	-15 Pa
Air change per hour (ACH)	8 -12	> 12	> 12	> 12

3.3.3 Types of pressure sensors used in NPRs

Traditionally, hot-wire anemometers [118] and ball pressure sensors [119] were used to measure pressure in NPRs. However, they have limitations, such as they are highly sensitive to dust, require periodic maintenance, and cannot be connected to a BMS or RPM for real-time control. Therefore, transducer-based pressure sensors (TBPSs) are replacing hot-wire and ball pressure sensors in NPRs since TBPSs are more accurate, reliable, require low maintenance, and can be connected to BMS or RPM for real-time monitoring.

Physics of TBPSs: A force collector and a transducer are two fundamental components of TBPSs. A force collector, such as an elastic diaphragm, is combined with a transducer to

generate an electrical signal [120] proportional to the input pressure.

Types of TBPSs: In general, TBPSs work in one of three modes: absolute, gauge, or differential measurement. Absolute pressure sensors use vacuum pressure, and gauge sensors use local atmospheric pressure as the static reference pressure. On the other hand, **Differential Pressure Sensors (DPSs)** measure the difference between any two pressure levels using two input ports (see Fig. 3.1). Therefore, DPSs are naturally suitable in such applications where the *pressure difference* is required to be measured, such as in NPRs [121]. As a DPS has *high sensitivity* to differential pressure and is deployed in NPRs, we focus on DPSs in next sections.

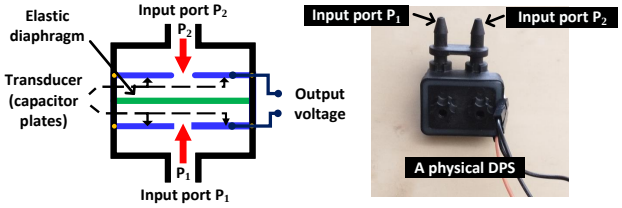


Figure 3.1: Basics of a DPS having two input ports.

3.3.4 Types of differential pressure sensors

DPSs typically have a elastic diaphragm placed in between two pressure input ports P_1 and P_2 (see Fig. 3.1). The diaphragm senses the differential pressure P_1 - P_2 applied to the pressure input ports by changing its shape. The diaphragm's shape change is converted to a proportional output voltage by using a transducer. DPSs either use a *capacitor*, or a *piezoresistor*, or *thermal mass-flow* as a transducer. A DPS is named after the type of transducer it has.

Fig. 3.1 shows a capacitive DPS as an example. The diaphragm is placed in between rigid capacitor plates. A differential pressure applied to the diaphragm generates a proportional change in the capacitive transducer resulting in a proportional voltage at the sensor output.

3.3.5 Differential pressure sensors used in NPRs

DPSs are highly sensitive to a small differential change in the low pressure range (i.e., Pa range) and are naturally suitable to measure a pressure difference. Therefore, DPSs are a *natural choice* to be used in most RPM/BMS systems to control the negative pressure. To prove the prevalence of DPSs in NPRs, we investigate six industry-used RPM systems designed by popular manufacturers. All of these RPM systems use different types of DPSs that are shown in Table 3.2.

Table 3.2: Differential pressure sensors used in NPRs

Sl.	RPM/DPS part#	Type	Technology	Manufacturer
1	Series RSME [122]	Capacitive	Differential	Dwyer
2	SRPM 0R1WB [107]	Capacitive	Differential	Setra
3	One Vue Sense [123]	Unknown	Differential	Primex
4	RSME-B-003 [124]	Piezoresistive	Differential	Dwyer
5	Siemens 547-101A [125]	Unknown	Differential	Siemens
6	Series A1 [126]	Piezoresistive	Differential	Sensocon
7	GUARDIAN [127]	Unknown	Differential	Paragon Con.

3.3.6 Resonant frequency of a DPS and resonance

Resonant frequency: As mentioned in Section 3.3.3 and 3.3.4, typically, DPSs have a diaphragm/membrane and a transducer. Therefore, the pressure transducer system in DPS is considered as a second-order dynamic system, analogous to a bouncing ball [108]. Hence, the transducer system in a DPS has its own resonant frequency, f_r, which depends on the mass and stiffness of the diaphragm and mass of the pressure medium as Eqn. 3.1 [128].

$$f_r = \frac{1}{2\pi}\sqrt{\frac{\text{stiffness of a diaphragm}}{\text{mass of the pressure medium and diaphragm}}} \tag{3.1}$$

Resonance: Resonance occurs when the frequency of the input pressure wave matches the resonant frequency of the driven transducer system in a DPS, resulting in oscillations [129] in the transducer at large amplitude. This results in significant error by overshooting the

peaks and troughs in the actual pressure wave, with an overestimation/underestimation of the actual reading. Therefore, users ensure that a DPS typically operates below its resonant frequency to prevent the resonance. A thumb's rule is 20% of the resonant frequency is typically used as the usable frequency limit for a given DPS [130]. This concept is illustrated in Fig. 3.2.

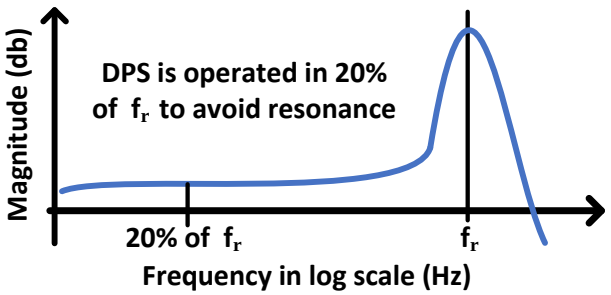


Figure 3.2: Resonant frequency in a DPS.

3.3.7 Electronics inside of a DPS

DPSs have a signal conditioning block in addition to a transducer (see Fig. 3.3). The signal conditioning block has differential amplifiers, low-pass filters (LPFs), and analog-to-digital converters (ADCs). A differential amplifier amplifies the output after removing the common-mode noises. An LPF with an ADC digitizes the measured value. Both analog and digital DPSs are available on the market. Analog DPSs output the analog signals from the differential amplifier directly, while digital DPSs contain the LPF and ADC.

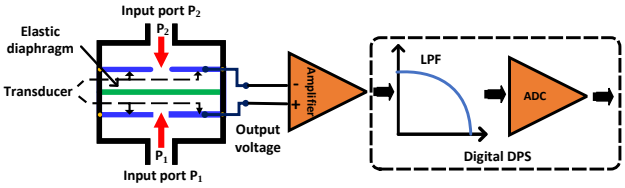


Figure 3.3: Different components inside of a DPS.

3.4 Basics of an NPR

This section explains the construction of an NPR, where and how the DPSs are deployed in an NPR, and how the output from the DPS controls the NPR's control system.

3.4.1 Components of a real-world NPR

The components of an NPR vary depending upon the requirements of different facilities. However, the core components are more or less the same for most NPRs. Here, we detail the components of an anonymous NPR where we have visited and experimented with to validate our attack model. **Please note that the target NPR evaluated in this paper is located in a clean room in an anonymous bioresearch facility. This NPR is also approved by the FDA and follows CDC guidelines.**

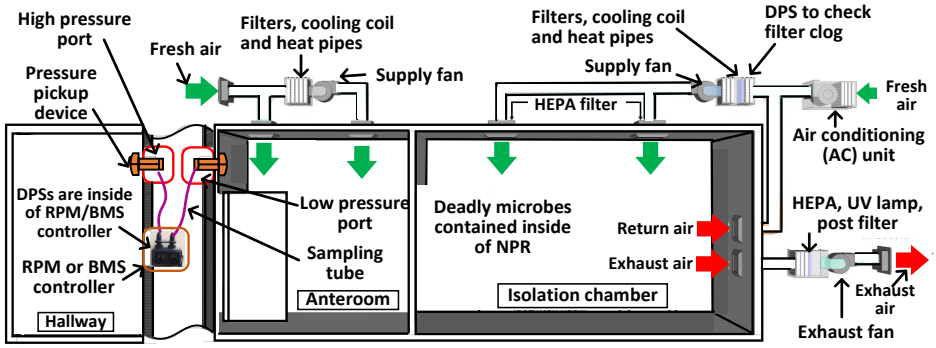


Figure 3.4: Different components of a real-world NPR.

A typical construction of an NPR is shown in Fig. 3.4. An NPR has an HVAC system, which includes fresh air inlet ports. The fresh air from the outside is treated with multistage filters and then supplied to the isolation chamber of an NPR, including the anteroom, through an air conditioning (AC) unit. The AC has a Variable Air Volume (VAV) controller, which can increase or decrease the *supply* fan speed, controlling the fresh airflow to the NPR. An exhaust fan continuously moves the contaminated air out from the NPR through a HEPA filter using an exhaust pipe. The polluted air is further treated with a post-filtration unit

having an Ultraviolet (UV) lamp. The room is maintained as airtight as possible. An RPM system is installed at the wall and integrated with the BMSs.

3.4.2 How DPSs are deployed in an NPR

The HVAC system *ensures* a negative pressure in the NPR by controlling the fresh air and exhaust airflow using the supply and exhaust fan. An RPM system continuously *monitors* the negative room pressure. The RPM and HVAC systems use DPSs to monitor and control negative pressure in an NPR. The DPS is typically located inside of RPM or BMS controller. Commonly, the input ports of a DPS are connected with pressure ports using sampling tubes (see Fig. 3.4 and 3.5). The pressure port located inside an NPR is known as a *low pressure port*. The pressure port located outside an NPR in a hallway/reference space is known as a *high pressure port*. The sampling tube is connected with a pressure pickup device in the pressure ports. The pressure pick-up device increases the surface area of the sampling tube to pick up the target pressure accurately.

The low and high pressure ports are *exposed* and typically installed in *eyesight* near the door wall or on the ceiling of an NPR. There are other DPSs used in the HVAC system to indicate whether the filters of the HVAC are clogged or not. Typically they are not installed in the eyesight. Therefore, they are not accessible.

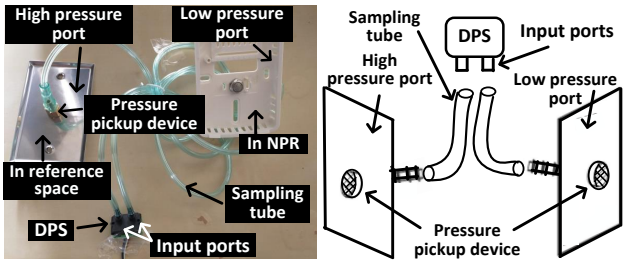


Figure 3.5: Pressure ports and sampling tube of a DPS.

3.4.3 Pressure control algorithm in an NPR

A pressure control algorithm running on the BMS controls the HVAC system of the NPR to maintain a constant negative pressure. A simplified control algorithm 2 is provided below. Algorithm 2 shows that the pressure readings from DPSs are used to control the speed of the supply fan and exhaust fan when the negative pressure increases or decreases from a reference value in the NPR, maintaining the negative pressure close to the reference value. The rest of the control algorithm 2 is self-explanatory.

Algorithm 2: Pressure control algorithm in an NPR.

```
   Input: Pressure measurement data from DPSs
   Output: Send control signals to the HVAC system
 1 for t ← 1 to ∞ do
 2 |   Track differential pressure reading from DPS's pressure ports
 3 |   if Negative differential pressure increases from a reference value then
 4 |   |   Reduce the supply fan speed of the AC to control the fresh airflow
 5 |   |   Increase the exhaust fan speed to increase the exhaust airflow
 6 |   end
 7 |   else if Negative differential pressure decreases from a reference value then
 8 |   |   Increase the supply fan speed of the AC to control the fresh airflow
 9 |   |   Reduce the exhaust fan speed to reduce the exhaust airflow
10 |   end
11 |   else
12 |   |   Maintain the same state of the controller
13 |   end
14 end
```

3.5 Attack Model

Fig. 3.7 shows the different components of our attack model associated with NPRs. We discuss the components of the attack model below in a point-by-point fashion.

Attacker's intent: The attacker creates a forged resonance in the DPSs used in NPRs with malicious music having a frequency equal to the resonant frequency of the DPSs. As a result. the overshooting occurs in the actual pressure reading, resulting in a change in the negative pressure maintained in NPRs by the BMSs.

Target system: The attacker targets a facility where NPRs are used to contain deadly microbes and infectious airborne particles. Such facilities include isolation rooms, clean rooms and pharmacies in infectious-control hospitals, and biolabs in bioresearch facilities.

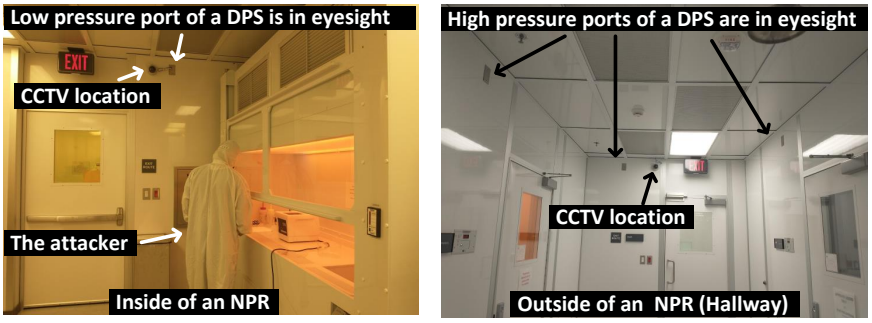


Figure 3.6: Pressure ports of DPSs are in eyesight in NPRs.

Attacker's capabilities: The attacker can surreptitiously place an attack tool near the target pressure ports of a DPS used in an NPR. The attack tool has an *audio source*. The audio source plays malicious music having a frequency equal to the resonant frequency of a DPS mounted in a target NPR. The audio source can be a simple cellphone or a speaker from an entertainment unit, such as televisions and radios, or CCTVs, placed in the vicinity of the pressure port of a target DPS. The low and high pressure ports are often mounted in eyesight, and placing the audio source near the target pressure port requires a *brief one-time access*. Moreover, audio sources, such as televisions or CCTVs with speakers, are often installed in NPR facilities near the pressure ports (see Fig. 3.6). The audio source may have wireless controls allowing for remote communication. Therefore, the attacker can remotely control the timing of the attack and can pick a vulnerable time (e.g., infectious patient admitted in an NPR, ongoing bioresearch, etc.) for a maximal consequence. The authority of the target NPR may not be aware of the attack model and would possibly neglect the security implications of any audio source placed near the pressure ports in an NPR.

Attacker's access level: The access near the pressure port of a DPS needed for the attack can be possible in at least two scenarios. **First** (most likely), a malicious employee or a guest or a maintenance person, who has access to an NPR, may place the audio source near

the pressure port. Though an NPR is restricted for unauthorized personnel, getting brief one-time access near the pressure port may not be difficult for an attacker in disguise of a guest or a maintenance person. **Second** is interdiction, which has been rumored to be used in the past [70–73] and has been recently proven to be feasible [74]. During interdiction, a competitor can intercept the DPS during delivery or installation and may modify the DPS by placing an audio source inside and then proceed with delivery or installation to the NPR facility.

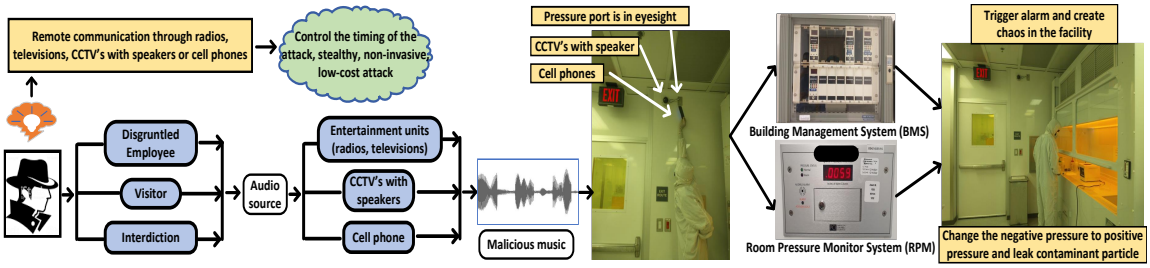


Figure 3.7: A brief overview of the attack model - A Wolf in Sheep's Clothing.

Playing malicious music: The attacker can play the malicious music in speakers to inject sound into DPS in the following three ways. **First**, the attacker can use a standard phishing attack to trick the authority into playing malicious music via email or a web page with autoplay audio enabled in CCTVs or televisions. **Second**, the attacker can play the malicious music using public radios. If some individuals place their radio near a pressure port, there is a good chance that the attack will be effective. **Third**, a physical proximity attack can happen if an attacker plays the music via a cell phone.

Outcomes of the attack: The attacker changes the actual pressure reading of DPSs and fools the BMS to turn the negative room pressure into a positive pressure or reduce the negative pressure from a reference value. This will trigger an alarm and create chaos in the facility. Moreover, the NPR cannot work properly for what it is intended to design for and may not contain the deadly microbes. The intentional leak of deadly microbes from NPRs may result in bioterrorism. The potential for mass destruction by bioterrorism is evident from a report from the U.S. Office of Technology, which predicted that the release of 100

kg of anthrax spores in Washington, DC, would cause 130,000 to 3 million deaths, matching the lethal potential of a hydrogen bomb [131]. The CDC reviewed potential microbes, such as smallpox and viral hemorrhagic fever, as airborne bioweapons [132]. An intentional leak of these bioweapons from an NPR by an attacker can trigger a worldwide pandemic with a tremendous loss of human lives and monetary resources.

Non-invasiveness: The spoofing attack is non-invasive and is performed without making physical contact with the target DPS. The attacker don't need to directly access or physically touch the sensor readings. However, we expect that attackers can examine the behavior of a similar sensor subjected to acoustic impacts before launching an actual attack.

Attacker's resources and cost: We assume that the attacker knows how the HVAC system works in NPRs and has a high school knowledge of resonance in DPSs. Moreover, a simple cell phone with a price of $60 - $100 can play the malicious music with a proper resonant frequency to attack the NPR.

3.6 Threats in an NPR

Here, we find the resonant frequency of DPSs used in NPRs by thorough experiments and explain how the resonance can be affected by different factors in an NPR.

3.6.1 Sound wave as a threat to DPSs

Sound wave: Sound is frequently referred to as a pressure wave since it is made up of a repeating pattern of high and low-pressure regions traveling across a medium [133].

Threat to DPSs: As a result, when sound waves collide with the diaphragms of DPSs, the diaphragm starts vibrating with the same frequency of sound. Therefore, having the above knowledge, a smart attacker can use a sound with a frequency equal to the resonant

frequency of the DPS to create a *resonance* and artificially displace the diaphragm in its maximal amplitude. The forged displacement of the diaphragm can change the pressure reading of a DPS by introducing overshooting in the actual pressure waveform.

3.6.2 Modeling sound effects on DPSs

We develop a model for how a sound wave perturbs the reading of a DPS. We measure the pressure as a linear combination of the original/equilibrium pressure $P_o(t)$ without a sound, and the external sound pressure $P_s(t)$. After a sound played at a frequency f, with an amplitude A_0, velocity v, and phase ϕ from a distance d, the total measured pressure $P(t)$ by a DPS can be modeled as:

$$\begin{aligned} P(t) &= P_o(t) + P_s(t) \\ &= P_o(t) + h(d, f) \cdot A_0 cos(2\pi f t + d/v + \phi) \end{aligned} \tag{3.2}$$

where $h(d, f)$ represents the attenuation of a sound wave, which depends on distance d and frequency f of the audio source. If the frequency f of the sound wave is equal to the DPS's resonant frequency f_r, the impact $P_s(t)$ will be maximum for a target DPS.

It should be clear from the above explanation that the attacker, at first, needs to identify the resonant frequency f_r of the DPS to orchestrate an attack. However, datasheets of the pressure sensors used in NPRs do not provide information related to their resonant frequencies. Therefore, we use thorough experiments to find the resonant frequency discussed in detail in the next sections.

3.6.3 Experimental setup

Figure 3.8 depicts the experimental setup to evaluate the resonant frequency of TBPSs. We produce a single-tone sound wave at different frequencies from an audible value of 50 Hz to an inaudible value of 40 kHz with the following three different audio sources.

1. Source 1: We use a Samsung Galaxy S10 smartphone [134] to generate frequencies within 50 Hz to 13 kHz. We use an app named Function Generator to sweep frequencies within the specific frequency range using the smartphone, which has a *sound pressure level (SPL)* of $\sim$ 80 dB [135] at its maximum volume at 1-inch distance.

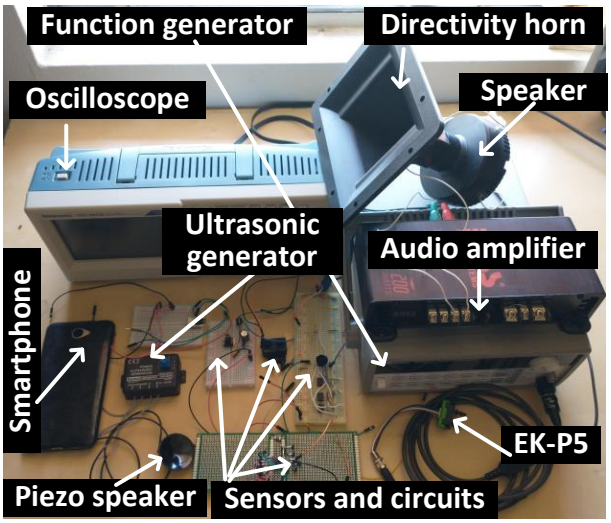


Figure 3.8: Experiment setup for different DPSs.

2. Source 2: We use a function generator [136], a 200 W audio amplifier (part# BOSS Audio Systems R1002 [137]), a speaker (part# Goldwood Sound Module [138]), and a directivity tweeter horn (part# GT-1188 [139]) to generate frequencies within 100 Hz to 18 kHz. The directivity horn is connected with the speaker to direct the sound to the target sensor. This setup can generate an SPL up to $\sim$ 95 dB at 1-inch distance. The reason for using audio *source 2* when we have the audio *source 1* is to test the sensors with a higher SPL. We use an app named Sound Meter [140] to measure the SPL.

3. Source 3: We use an ultrasound generator (part# Kemo Electronic M048N [141]), a piezo speaker (part# ToToT Ultrasonic Speaker [142]) to generate frequencies within a range of 15 kHz to 40 kHz.

We test 8 industry-used TBPSs from 5 different manufacturers including analog and digital types (see Table 3.3). Out of the 8 sensors, 6 of them are DPSs, and 2 of them are gauge pressure sensors (see Section 3.3.3). We use gauge sensors to identify that not only the DPSs but also the gauge pressure sensors have resonant frequencies that can be utilized by an attacker. This supports the idea that if an NPR uses a gauge pressure sensor instead of DPSs, an attacker can also target those NPRs. Therefore, our attack model will work for any TBPSs irrespective of gauge pressure sensors and DPSs.

Table 3.3: Summary of the resonant frequencies of Transducer Based Pressure Sensors (TBPSs) *without* a sampling tube.

Sl.	Sensor	Manufac.	Type	Transducer	Pressure range	Interface	Resonant freq.
1	P1K-2-2X16PA [143]	Sensata	Differential	Piezoresistive	0 to 500 Pa	Analog	790 - 800 Hz
2	MPVZ5004GW7U [144]	Freescale	Gauge	Piezoresistive	0 to 3.92 kPa	Analog	1750 - 1800 Hz
3	SDP810-250PA [145]	Sensirion	Differential	Thermal mass-flow	±250 Pa	Digital	760 - 780 Hz
4	SDP810-500PA [145]	Sensirion	Differential	Thermal mass-flow	±500 Pa	Digital	870 - 890 Hz
5	TBPDPNS100PG [146]	Honeywell	Gauge	Piezoresistive	0 to 689 kPa	Analog	not found
6	P993-1B [147]	Sensata	Differential	Capacitive	±248 Pa	Analog	740 - 750 Hz
7	NSCSS015PDUNV [148]	Honeywell	Differential	Piezoresistive	±103 kPa	Analog	not found
8	A1011-00 [149]	Sensocon	Differential	Piezoresistive	0 to 60 Pa	Digital	680 - 690 Hz

The experimental setup is placed inside an acoustic isolation chamber to avoid external noise. To read and log the pressure measurements, we utilize an oscilloscope for analog TBPSs and a Ek-P5 [150] test kit connected with our laptop for digital DPSs.

Please note that a few pressure sensors require a separate unique circuit for testing, data

collection, and signal conditioning. Therefore, we build a separate signal conditioning circuit for each of the sensors that requires it. As an example, a signal conditioning circuit using an instrumentation amplifier to collect data from a DPS with part# NSCSSNN015PDUNV is shown in Fig. 3.9.

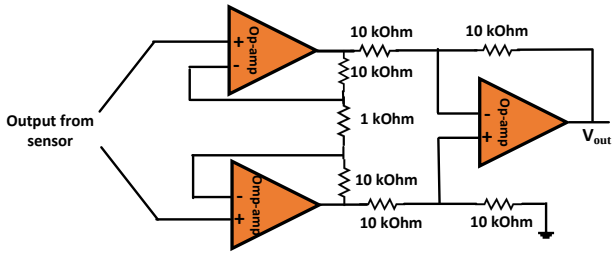


Figure 3.9: Instrumentation amplifier.

3.6.4 Evaluating the resonant frequency

A single tone sound having a frequency between 50 Hz to 40 kHz with a 10 Hz increment is applied to *one of the two ports* of a DPS or to a single port of a gauge pressure sensor in our testbed *without* a sampling tube. We vary the frequency every 3 ms and record the data for every frequency using an oscilloscope for analog gauge/DPSs or using the Ek-P5 test kit for digital DPSs. We maintain the SPL within $\sim$ 35 - 95 dB from 2 cm in our experiments.

We examine the difference in the sensor readings with and without sound signal. When there is no sound wave present, the two input ports of a DPS or a single input port of a gauge pressure sensor measure the unperturbed pressure from the environment. As a result, the intended output of the sensor should be zero in the absence of the single tone sound wave. When the single tone sound is applied to an input port of a DPS or a gauge sensor, the output of the target sensor starts oscillating. The oscillations reach a peak value at a resonant frequency of the target pressure sensor.

Two examples are shown in Fig. 3.10 as a proof-of-concept to support our observations on resonant frequencies. The outputs from an analog DPS with part# P993-1 and a digital DPS with part# SDP810-500Pa are shown in Fig. 3.10 (left) and (right), respectively. The

blue color is the sensor output before applying the sound, and the red color is the sensor output after applying the sound. It is clear from Fig. 3.10 that the sensor output has the largest perturbations within 740-750 Hz for an analog DPS with the part# P993-1B and within 870-890 Hz for a digital DPS with the part# SDP810-500Pa.

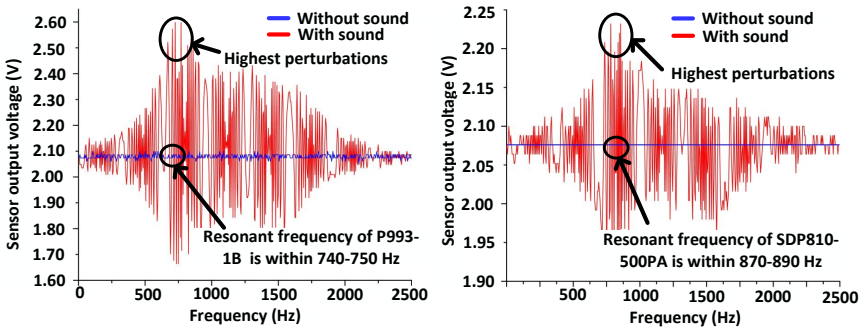


Figure 3.10: Sound injection effect on (left) P993-1B and (right) SDP810-500PA pressure sensors for different frequencies.

Table 3.3 summarizes the experiment's findings on resonant frequencies. According to our findings, 6 of the 8 pressure sensors resonated in response to the applied sound wave. We find that the detected resonant frequencies range from ∼600 Hz to ∼1800 Hz, which are in the audible range.

We are unable to detect the resonant effect in 2 of the 8 sensors: part# TBPDPNS100PGUCV and NSCSS015PDUNV. We observe from Table 3.3 that with the increase of the pressure range, the value of the resonant frequency increases. The reason behind this is that the sensors, which work in high pressure range, have more stiff diaphragms compared to those sensors, which work in low pressure range. For example, MPVZ5004GW7U has a higher resonant frequency than P1K-2-2X16PA because of its higher pressure range. Therefore, it is possible that the resonant frequencies of TBPDPNS100PGUCV and NSCSS015PDUNV may fall outside of 40 kHz, which is the highest test frequency we use in our experiments.

3.6.5 Why resonant frequencies in audible range?

An interesting observation from Table 3.3 is that all resonant frequencies of the DPSs used in NPRs fall in the audible range. We only experimented with 8 sensors used in NPRs. Can we conclude from our experiments that most of the sensors used in NPRs typically have resonant frequencies in the audible range? The answer is *Yes*.

Reason: Table 3.1 shows that NPRs need to maintain a low negative pressure within 2.5 Pa to 15 Pa. Therefore, DPSs used in NPRs are selected to have high sensitivity in the low Pa range for an accurate measurement. The sensors working in the low pressure range have less stiff diaphragms compared to those sensors working in the high pressure range [151]. Eqn. 3.1 indicates that resonant frequency decreases in a square-root fashion with the decrease of stiffness of the diaphragms. Therefore, the DPSs working in a pressure range of few Pa, typically have less stiff diaphragms with low resonant frequencies typically in audible range (i.e., < 20 kHz).

3.6.6 Factors influencing the resonant frequency

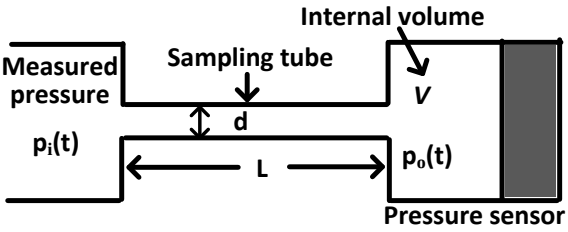


Figure 3.11: Modeling sound pressure inside of a DPS having a sampling tube as a Helmholtz resonator.

We measure resonant frequencies in Table 3.3 by directly applying the sound wave to the input ports of a pressure sensor. However, sampling tubes and a pressure pick-up device are often connected with the pressure ports of a DPS (see Fig. 3.4 and 3.5) to pick up the pressure from a target location. The *geometric properties* of the sampling tube affect the characteristics of the DPS's transducer systems. As a result, the resonant frequency of DPSs

with sampling tubes differs from the value *without* sampling tubes.

Helmholtz resonators: A pressure sensor with a sampling tube can be modeled as Fig. 3.11. Let's denote the internal volume of the sensor by V, and the internal diameter and length of the tube by d and L, respectively. As the sensor's internal volume and the connecting tube are similar to a structure having a cavity with a narrow neck, a pressure sensor with a tube is a basic form of discrete Helmholtz fluid resonator [109, 152]. The fluid in the tube acts as the oscillator mass, while the compressible fluid in the cavity acts as the oscillator spring. The Helmholtz resonator can be simplified by a second-order dynamic system (see Section 3.3.6), which yields the following relation between the sampling tube inlet pressure $p_i(t)$ and the sensor output pressure $p_o(t)$:

$$\frac{d^2 p_o}{dt^2} + 2\xi\omega_h \frac{dp_o}{dt} + \omega_h^2 p_o = \omega_h^2 p_i \tag{3.3}$$

where $\omega_h = 2\pi f_h$, f_h is the overall resonant frequency of the sensor with a tube, and ξ is the damping ratio. The resonant frequency f_h of the sensor with a tube can be expressed as:

$$f_h = \frac{1}{2\pi} v \sqrt{\frac{AS}{LVM}} \tag{3.4}$$

where v is the sound velocity in air, A is the internal cross-sectional area of the tube, S is the stiffness of the diaphragm, M is the mass of the pressure medium and diaphragm. Eqn. 3.4 indicates that the resonant frequency of a DPS with a tube increases with the increase of the tube's internal cross-sectional area A and decreases with the increase of the tube length L. As the DPS used in NPRs has a standard diameter of its input ports, the diameter of the sampling tube is somewhat fixed. Therefore, we focus on the effect of sampling tube length on our attack model in the next section.

3.6.7 Resonance with sampling tube in NPRs

Fig. 3.4 and Fig. 3.5 show how the sampling tube is connected with the DPS's ports. For good sensitivity and error-free measurement, the DPS is placed close to the high and low pressure ports. Therefore, the length of the sampling tube is typically < 2 m. Therefore, we vary the length of the sampling tube up to 2 m with a 0.4 m increment for a diameter of 5/16 inch and calculate resonant frequencies for each of the 6 DPSs (i.e., having valid resonant frequency) from Table 3.3. Fig. 3.12 shows the results. We notice that with the increase of the sampling tube length, the sensor's overall resonant frequency f_h reduces, supporting Eqn. 3.4.

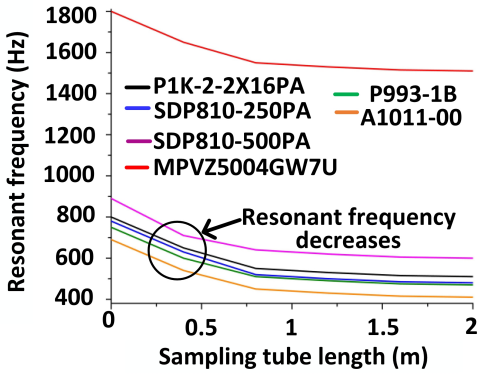


Figure 3.12: Resonant frequency decreases with tube length.

3.6.8 A wolf in sheep's clothing

It is evident from Section 3.6.7 that the resonant frequencies of DPSs even with the sampling tube fall within audible range. Attacking DPSs with a sound just having resonant frequencies would make the attacker immediately identifiable because resonant frequencies will generate a "beep"-ish sound, raising a concern to the authority.

We came up with a solution explained in **Section 3.7.1** to *disguise* the resonant frequencies inside a popular music so that the attack will not be identifiable. Once the attacker injects the malicious music into DPSs, he/she can successfully create resonance in DPSs. This is

referred to as putting *"the wolf in sheep's clothing"* since it is the resonant frequency that has been disguised inside music.

3.7 Attacking a negative pressure room

As mentioned in Section 3.4.2, the low pressure port of the DPS is exposed to the negative pressure room and the high pressure port is linked to a hallway, which is a reference space. If the pressure at the low pressure and high pressure port is denoted by P_L and P_H, respectively, the DPS measures the differential pressure, P_D as:

$$P_D = P_L - P_H \tag{3.5}$$

As mentioned in Section 3.4.1, an NPR has an HVAC and an RPM system. There can be the following two scenarios depending on how the HVAC and RPM systems use the DPSs in NPRs.

First, the HVAC and RPM systems in NPRs use the *same* DPS to control and monitor the negative pressure in an NPR. This scenario exists in modern facilities where both HVAC and RPM systems are automated and integrated with the BMS.

Second, the HVAC uses a DPS to maintain the negative pressure, and the RPM uses a *separate* DPS to monitor the differential pressure in an NPR. Here, the RPM system only gives an alarm if the negative pressure falls below a threshold but is not responsible for maintaining a negative pressure in an NPR.

We discuss the above two scenarios below.

3.7.1 When HVAC and RPM use the same DPS

This scenario is easier for the attacker as he/she can attack both the HVAC and RPM systems of an NPR just by attacking a single DPS. The attacker can either inject sound to the low pressure port of the DPS if he/she is inside of the NPR and find that it is comparatively easier to access the low pressure port. Otherwise, the attacker can inject sound to the high pressure port of the DPS.

A simple resonance is not enough: If the attacker creates resonance either by attacking the low pressure or high pressure port of the target DPS in the NPR, the resonance changes the original pressure reading by overshooting the original pressure level in both upward and downward directions (see Fig. 3.10). Therefore, the differential pressure reading P_D in the DPS (Eqn. 3.5) starts fluctuating. As a result, the *supply fan* and the *exhaust fan* immediately track the DPS's pressure fluctuations and vary their fan speed to maintain a static negative pressure inside of the NPR, following the algorithm 2. However, the rate of change in the pressure reading because of the resonance is high for a mechanical fan to track. Therefore, the supply fan and the exhaust fan cannot vary their speed with the high fluctuating rate. As a result, the negative pressure in the NPR only fluctuates a little bit and truly does not change on a large scale from the reference value. Moreover, the attacker does not have any adversarial control over it. Therefore, the attack can not induce any noticeable effect in the target NPR.

A wolf in sheep's clothing: To create a maximal change in the NPR's negative pressure, a smart strategy is adopted in addition to simply *disguising* the resonant frequency band inside of music. The strategy is illustrated in Fig. 3.13. The resonant frequency is inserted into the music as a segment in a specific interval for a certain duration. Let us denote the interval by T_I and duration by T_D. Every inserted segment of resonant frequency is *ended at its peak* after duration T_D, and the same segment is inserted again in every interval T_I.

When the inserted segment is ended at its peak, the corresponding pressure wave inside the DPS's transducer system also ends at its peak (see Fig. 3.13). As a DPS with a sampling tube is a second-order oscillating system (i.e., Helmholtz resonator), the pressure wave does not instantly fall to zero from the peak value. Instead, the pressure wave starts to attenuate from its peak exponentially following Eqn. 3.6 of a damped 2^{nd} order system [128].

$$p(t) = p_o e^{-\omega_h t} + (\omega_h p_o + v_o) t e^{-\omega_h t} \tag{3.6}$$

where p_o and v_o are the initial pressure and velocity at peak, respectively, and ω_h is the angular resonant frequency. The term v_o depends on the viscosity and density of the pressure medium.

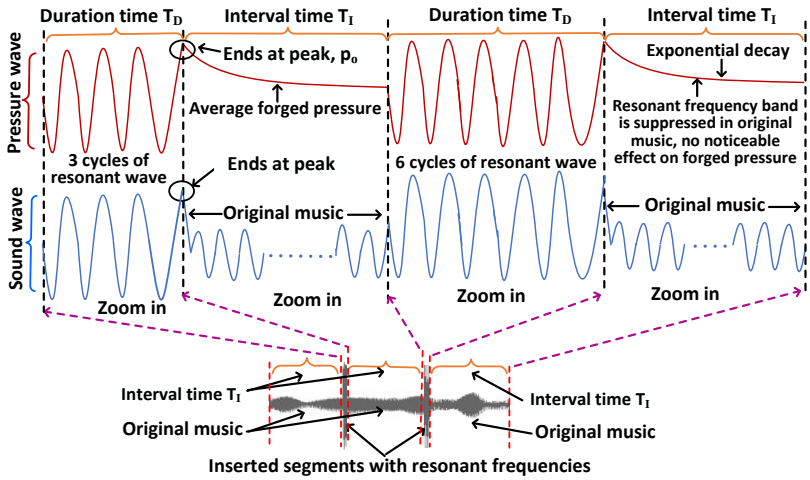


Figure 3.13: Turning a popular music into an attack tool.

The interval time T_I is selected in such a way that the pressure wave never falls to zero. Therefore, there is always an *average forged* pressure present inside the DPS's transducer system, originating from the injected music by the attacker. As the generated forged pressure has an *average* value greater than zero and changes *slowly*, the *supply fan* and the *exhaust fan* can track the pressure change in DPS, and they can vary their fan speed according to the pressure reading of the DPS. Therefore, this time the attack can induce a noticeable effect in the target NPR.

Between two consecutive inserted segments of resonant frequency (i.e., in the interval time T_I), the original music is inserted by suppressing its resonant frequency components. Therefore, the original music does not have a noticeable effect on the forged pressure present in the interval T_I. Moreover, the inserted segment with the resonant frequency has ∼3.8x increased power density compared to the original music. Fig. 3.14 shows this phenomena for SDP810-500PA, which has resonant frequency within 700 - 900 Hz (see Fig. 3.12). Therefore, the inserted segment can create a maximal effect in the NPR by turning a negative pressure into a positive one.

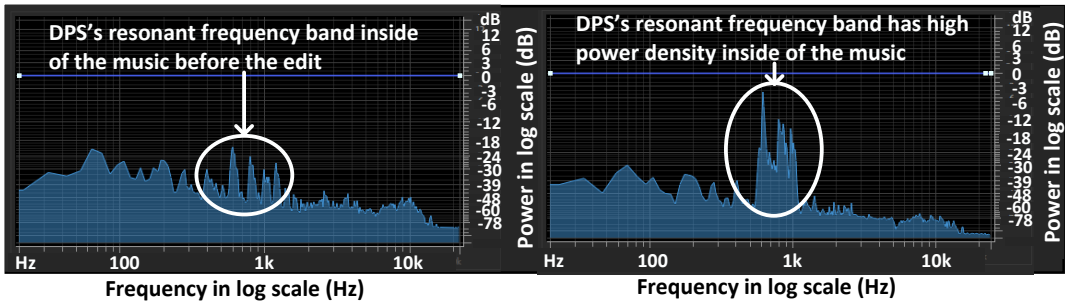


Figure 3.14: High power density of resonant frequencies inside of a music because of the inserted segments.

Adversarial control: The attacker can control the average forged pressure in the DPS's transducer system by controlling the interval time T_I and duration time T_D.

The duration T_D cannot be too small as a small T_D cannot provide the inserted segment enough time to impact the DPS. The T_D cannot be too large because the inserted segment with large T_D can badly distort the music so that the attack might be identified. The duration of T_D should be equal to or larger than the period of the resonant frequency so that at least one cycle of the resonant wave is accommodated inside of the duration T_D (i.e., inserted segment).

With a small interval T_I, the average forged pressure is increased. However, a small T_I results in a large number of inserted segments that may distort the music significantly. We measured the forged pressure for a T_I between 15 ms to 60 ms for a ∼65 dB sound for a

DPS (part# A1011-00) with a 1 m sampling tube. The sound is applied at 0.2 cm from the pressure port. The result is shown in Fig. 3.15 for a duration time T_D = 1.47 ms, which is equal to the period of the resonant wave of part# A1011-00 (i.e., part# A1011-00 has resonant frequency 680 Hz from Table 3.3; 1/680 Hz = 1.47 ms).

As mentioned earlier, the resonant frequency can vary within a range depending upon the sampling tube length. As the attacker may not know the exact length of the sampling tube, the attacker may need to vary the duration time T_D within a range to accommodate at least one cycle of the variable resonant wave for a maximal impact (Fig. 3.15). The attacker can also vary the number of cycles in the duration T_D from one inserted segment to another inserted segment. For example, the first inserted segment in Fig. 3.13 has 3 cycles, whereas the second segment has 6 cycles within the duration T_D.

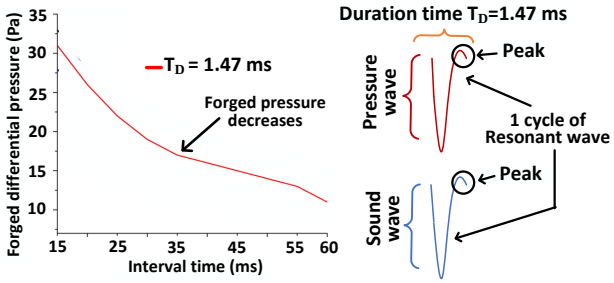


Figure 3.15: Adversarial control using malicious music.

Tools for a malicious music: The attacker selects music and inserts segments of resonant frequencies within the music in a way already explained in Section 3.7.1 using a software named *Adobe Audition*. Though someone who has listened to the music many times before may identify the change in the music, the vast majority of people will either be oblivious of the change or will incorrectly ascribe the change in the music to a speaker issue. For example, we pick a popular song *Hello* by *Adele* and convert it into a malicious music in a way explained in Section 3.7.1 for T_D = 2 ms and T_I = 15 ms. The malicious music is uploaded in the following link: https://sites.google.com/view/awolfinsheepsclothing/home

Injecting music into the low pressure port: Let us give an example to elaborate on the

result of injecting music into the low pressure port. Suppose, before an attack, the pressure at a low pressure port P_L = 10 Pa and at a high pressure port P_H = 12.5 Pa. Therefore, the differential pressure from Eqn. 3.5 is P_D = 10 - 12.5 = - 2.5 Pa, which is the reference differential pressure in the NPR. Suppose the forged pressure resulting from the injected malicious music into the low pressure port is 8 Pa. Now, after the attack, P_D = (10 + 8 = 18) - 12.5 = 5.5 Pa. Therefore, the HVAC system will reduce the NPR's pressure from 18 Pa to 10 Pa to keep the differential pressure at -2.5 Pa. The reduction of 8 Pa in the NPR will result in a *true* differential pressure of P_D = (10 - 8 = 2) - 12.5 = -10 Pa. The injection of music into the low pressure port results in more negative differential pressure (i.e., - 2.5 Pa to - 10 Pa), which is actually good for keeping deadly microbes in the NPR. However, the abnormal change in negative pressure may trigger an alarm by the RPM system and create chaos in the facility. An attacker can use this chaos to initiate a stronger attack, such as stealing deadly microbes from biosafety cabinets as he is already inside of the NPR.

Injecting music into the high pressure port: Let us use the previous example to elaborate on the effect of injecting music into the high pressure port. If the forged pressure resulting from the injected music into the high pressure port is 8 Pa, the P_D after the attack is 10 - (12.5 + 8) = -10.5 Pa. Therefore, the HVAC system will increase the NPR's pressure from 10 Pa to 18 Pa to keep the differential pressure at -2.5 Pa. The increase of 8 Pa in the NPR will result in a *true* differential pressure of P_D = (10 + 8 = 18) - 12.5 = 5.5 Pa, which is positive. The consequences of turning a negative pressure into a positive one in an NPR can be catastrophic as the NPR cannot contain the deadly microbes anymore, causing a potential leak of microbes from the compromised NPR. Moreover, an abnormal change in the negative pressure may trigger an alarm by the RPM system and create chaos in the facility. An attacker can use this chaos to initiate a stronger attack, such as entering the NPR and stealing deadly microbes from the biosafety cabinets.

3.7.2 When HVAC and RPM use separate DPSs

When the HVAC and RPM systems use separate DPSs, and if the attacker has a *single* audio source, he/she should attack the high or low pressure port of the DPS connected with the HVAC system to change the negative pressure in an NPR. Because the HVAC system maintains the negative pressure in an NPR. However, if the attacker attacks the high or low pressure port of the DPS connected with the RPM system, only an alarm may be triggered, and chaos will be created in the facility, but it will not change the NPR's pressure. The attacker can use the attack model already explained in Section 3.7.1 either to attack the HVAC or RPM system of an NPR.

A stronger attacker: Suppose we consider a stronger attacker, who can use *multiple* audio sources to attack the RPM and HVAC systems simultaneously. In that case, he/she can avoid the alarm triggered by the RPM system in the following way.

Let us explain this attack model using the same example from Section 3.7.1. Let us assume the attacker injects the same *forged* pressure of 8 Pa by music to the high pressure port of the DPS connected with the HVAC system. Therefore, the HVAC system similarly will increase the NPR's pressure from 10 Pa to 18 Pa, resulting in a positive differential pressure of 5.5 Pa. The RPM system will trigger an alarm for this abnormal change in the NPR's pressure. To prevent the alarm from being triggered, the attacker must need to inject the same 8 Pa *forged* pressure to the high pressure port of the DPS connected with the RPM system. As a result, the RPM will measure differential pressure of 18 - (12.5 + 8) = -2.5 Pa, which is equal to the NPR's reference pressure. Therefore, the RPM system will not trigger any alarm, and the attack will remain unidentified, resulting in a stronger attack model.

However, if both of the high pressure ports of the RPM and HVAC systems are in *close proximity*, the attacker can use a *single* audio source to attack the NPR without triggering the alarm.

3.7.3 Attacking multiple NPRs simultaneously

It is possible to simultaneously attack multiple NPRs just by injecting music into *a single* high pressure port of the DPS connected with the HVAC or RPM systems. As we mentioned earlier, the high pressure port is located in hallway to measure the reference pressure, and the NPR maintains a negative pressure inside with respect to the reference pressure. If there are multiple NPRs in a facility and if all the NPRs use a common place (e.g., hallway) as their reference pressure, it is a common practice to connect all the high pressure ports from all the NPRs into *one common* high pressure port to reduce cost. This is shown in Fig. 3.16. As multiple NPRs share a common high pressure port, the attacker can simultaneously attack multiple NPRs just by attacking the common high pressure port in the facility. It can trigger a combined leak of deadly microbes from multiple NPRs and can create chaos in different parts of the facility.

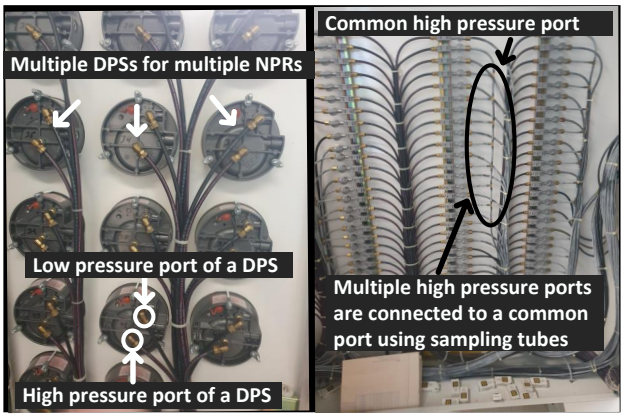


Figure 3.16: Multiple high pressure ports are connected together to a common high pressure port.

3.8 Attack model demonstration

We demonstrate our attack at an FDA-approved NPR located in an anonymous bioresearch facility. The demonstration is shown in Fig. 3.17. This facility uses separate DPSs for the HVAC and RPM systems. The location of the DPS connected with the RPM system is close

to the exit door. The DPS connected with the HVAC is at the sidewall of the wet bench. The wet bench stores sensitive particles inside of it under negative pressure. The authority did not permit us to attack the DPS connected with the HVAC system due to safety protocols. Therefore, we only demonstrate the attack on the DPS connected with the RPM system.

We use a Samsung Galaxy S10 smartphone from a 0.1 cm distance with an SPL of $\sim$ 65 dB to inject the malicious music into the low pressure port of the RPM system for a room #1422. We check the differential pressure for room #1422 before the attack from a logbook. We can see that the negative pressure stays within a range of 0.0278 - 0.0325 inch water column (i.e., 6.9 - 8 Pa). After injecting music from the smartphone, the negative pressure reading in the RPM system changes to a positive pressure of 0.0005 inch water column (i.e., 0.12 Pa). That is a 7 - 8 Pa pressure reading change in the RPM system due to an attack. A video demonstrating the attack model in the NPR is posted at the following link: https://sites.google.com/view/awolfinsheepsclothing/home

Though we are not permitted to attack the DPS connected with the HVAC system, according to the authority, our attack on the DPS connected with the HVAC system would create the same pressure change in the NPR.

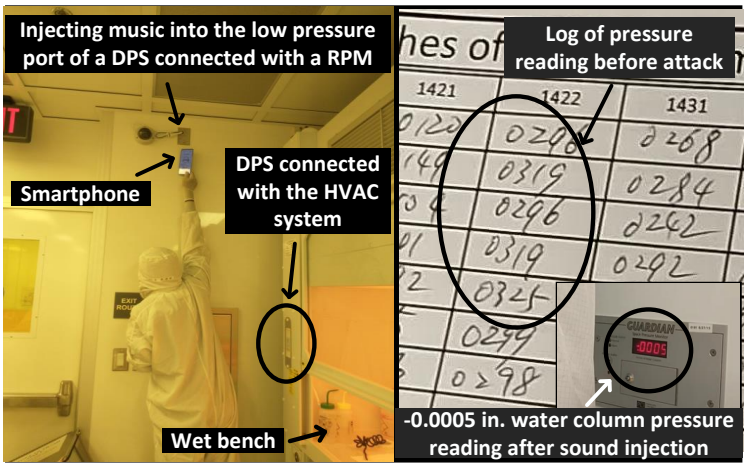


Figure 3.17: Attacking a practical NPR in a bioresearch facility.

3.9 Attack model evaluation

We already evaluate resonant frequencies of DPSs in Section 3.6 in detail. Here, we evaluate our attack model further for other parameters related to the DPSs and NPRs.

3.9.1 Experimental setup

We already show our attack at an FDA-approved NPR in a bioresearch facility in Section 3.8. As it is not *permitted* to vary different parameters of the DPS's transducer system located in the bioresearch facility, we prepare a testbed to evaluate our attack model. We use an industry used DPS from Sensocon with part# A1011-00 [126], two vinyl sampling tubes having inner diameters of 3/16" and 5/16" [153], a pressure pickup device with part# A-417A [154] and an oscilloscope in the testbed (see Fig. 3.18).

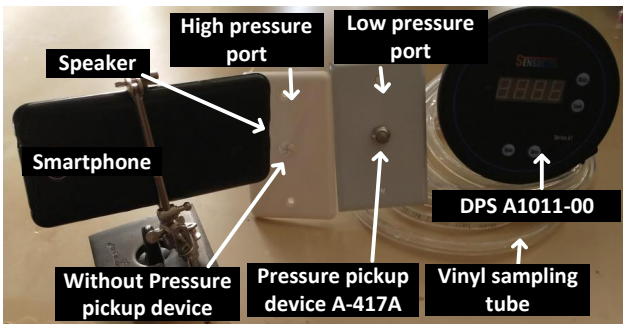


Figure 3.18: Experimental setup for evaluating attack model.

3.9.2 Varying the tube length and diameter

We vary the sampling tube length from 1 m to 5 m with a 1 m increment for two inner diameters of 3/16" and 5/16". We connect the sampling tube and pressure pickup device with the input ports of the A1011-00 sensor and inject sound into one of the pressure ports with the Samsung Galaxy S10 smartphone from a 0.1 cm distance. The result is shown in Fig. 3.19 (left). With the increase of the sampling tube length and the decrease of the sampling tube inner diameter, the sound damping inside the tube increases. Therefore, the

forged differential pressure originated from the injected music reduces for larger length and smaller diameter.

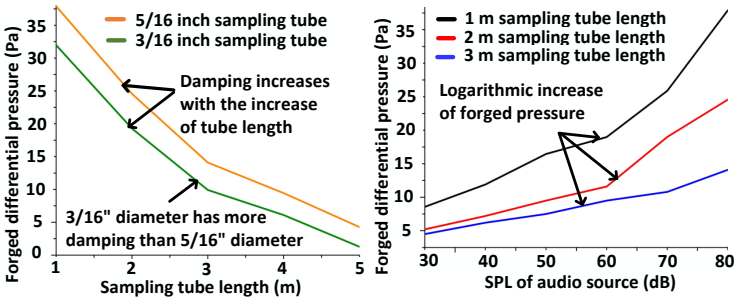


Figure 3.19: (left) Impact of sampling tube length and diameter. (right) Impact of the SPL of the audio source on the attack

3.9.3 Varying the SPL of the audio source

A logarithmic scale known as Sound Pressure Level (SPL) is used to measure the loudness of a sound. SPL is measured in decibels (dB). We vary the SPL of the audio source (i.e., Samsung Galaxy S10) from 30 dB to 80 dB with a 10 dB increment for 1m, 2 m, and 3 m of sampling tube (5/16" diameter) lengths for a 0.1 cm distance from the pressure pickup device. The result is shown in Fig. 3.19 (right). As with the increase of the SPL, the sound pressure from the audio source logarithmically increases. Therefore, the forged differential pressure also increases logarithmically. As sound damping increases with the increase of sampling tube length, the shorter sampling tube causes higher forged differential pressure.

3.9.4 Varying the distance of the audio source

We vary the distance of the audio source (i.e., Samsung Galaxy S10) from the pressure pickup device for 0 m (no sampling tube), 1 m, 2 m, and 3 m of sampling tube (5/16" diameter) lengths. The result is shown in Fig. 3.20. In acoustics, the SPL of a sound wave radiating from a point source decreases as the distance increases following the inverse-proportional law [155]: $SPL \propto 1/distance$. Therefore, the forged differential pressure also decreases with the increase of audio source distance from the pressure pickup device. Fig. 3.20 (right)

shows that an audio source has more impact on the DPS without a sampling tube (i.e., no dampening) with a saturated output.

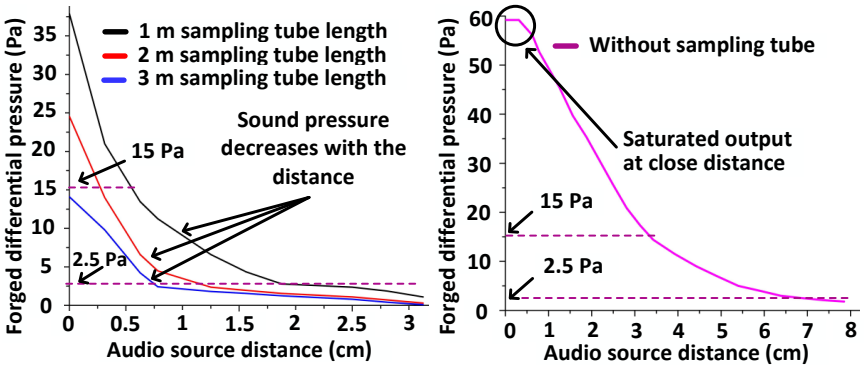


Figure 3.20: Impact of audio source distance on the attack.

3.9.5 With and without a pressure pickup device

A pressure pickup device is connected with the other end of the sampling tube and installed at the high and low pressure ports, mounted on the wall. The pressure pickup device increases the exposed area of the sampling tube end. Therefore, a small change in pressure can be sensed without an error. It is possible that some NPRs don't use pressure pickup devices; instead, a simple hole is mounted at the pressure ports. To evaluate the effect of the pressure pickup device, we inject music from a 0.1 cm distance into the pressure port with and without the pressure pickup device and vary the sampling tube length from 1 m to 5 m with a 1 m increment. We see from the results in Fig. 3.21 that the forged pressure is lower with a pressure pickup device. Because a pressure pickup device has foam gasket inside, which dampens the injected sound into it.

3.10 Feasibility of the Attack

1. Audio source distance: Section 3.9 indicates that the sampling tube's length and audio source's distance can restrict the effectiveness of the attack. Moreover, Table 3.1 indicates that the negative pressure has to be maintained between -2.5 Pa to -15 Pa for country-specific

requirements. Now, Fig. 3.20 (left) indicates that the audio source should be less than 0.6 cm away (1 m tube length) from pressure ports to generate a 15 Pa forged pressure, which can turn a -15 Pa negative pressure into a positive pressure. Fig. 3.20 (left) also indicates that the audio source should be less than 2.5 cm away from the pressure port to generate a 2.5 Pa forged pressure, which can turn a -2.5 Pa negative pressure into a positive pressure. *This indicates that the CDC guidelines (i.e., -2.5 Pa) in Table 3.1 can be impacted from a larger audio source distance compared to the guidelines adopted in Taiwan and Australia.*

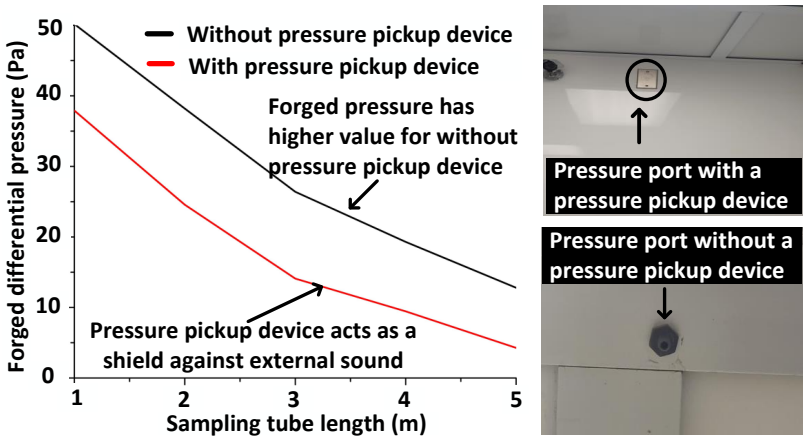


Figure 3.21: Impact of the pressure pickup device on the attack.

However, the audio source needs to be in close proximity to the pressure ports to have a feasible attack. CCTV's with speakers and entertainment units are often located in such close proximity to the pressure ports. Moreover, Fig. 3.20 indicates that the attacker can use an audio source from a larger distance if the sampling tube length is shorter or no sampling tube is present. For example, the audio source can generate a 2.5 Pa forged pressure at 7 cm far from the pressure ports without a sampling tube (Fig. 3.20 (right)). Sampling tube length depends on the location of DPSs from the pressure ports. Depending upon different locations of DPSs, the sampling tube length can be very short, or even no sampling tube can be present. The attacker can target those DPSs for greater impact.

2. LPF and the resonant frequency: Section 3.3.7 mentions that a DPS has an LPF. Therefore, simply filtering the resonant frequency using an LPF can prevent the resonance

in DPS. However, manufacturers don't use the LPF to filter out the resonant frequency because the resonant frequency of a DPS is not constant. A resonant frequency not only depends on the transducer and diaphragm of the DPS but also depends on the sampling tube's length and diameter, the fluid's viscosity and density inside of the sampling tube (see Sections 3.6.6 and 3.6.7). Therefore, it **varies** within a *band* for different transducer systems depending upon different applications. Moreover, manufacturers also don't filter out the whole *band* where the resonant frequency may belong. The reason is that a DPS is not only used in NPRs but also used in other *dynamic pressure sensing* applications where removing a frequency band might remove important information from the input data.

We can find a simple proof of this concept in Table 3.3. Both of the digital DPSs in Table 3.3 have $\sim$ 2.1 kHz sampling frequency and 760-890 Hz resonant frequency. If the LPF in the DPS filtered out the resonant frequency, we would not find the resonance.

3.10.1 Limitations

In this paper, the introduced adversarial control does not offer fine-grained control compared to [51, 52]. The reason behind this is that the direct feedback from the compromised NPR to the attacker is absent. Because, typically, the audio sources, such as cellphones, radios, televisions, and CCTVs, which inject malicious music, do not have pressure sensors to measure the pressure after the attack and send it back to the attacker. However, the attack is strong enough to change the negative pressure in an NPR. Moreover, close access near the pressure ports in an NPR, short-attacking range, and prior knowledge of the NPR are also the limitations of our attack model.

3.11 Countermeasures

The following techniques should be adopted together to prevent our attack - a wolf in sheep's clothing.

Dampening of the music: The simplest method of preventing resonance originating from the malicious music is to dampen the music. The smart way to dampen the music is to use a long sampling tube with the DPS's port. Even if the pressure port is very close to the DPS and the DPS would not require the sampling tube, we still suggest using a long sampling tube with the DPS. We find that a tube length greater than 7 m can completely dampen music having an SPL of 90 dB. The long tube can be coiled if space is limited for the mounting (see Fig. 3.22). However, a long sampling tube reduces the sensitivity of the DPS, resulting in a measurement error.

Enclosure around the pressure port: A box-like enclosure should enclose the pressure pickup device mounted in the pressure port (see Fig. 3.22). The box-like enclosure should be filled with sound damping foam to dampen the malicious music. However, this method also reduces the sensitivity of the DPS.

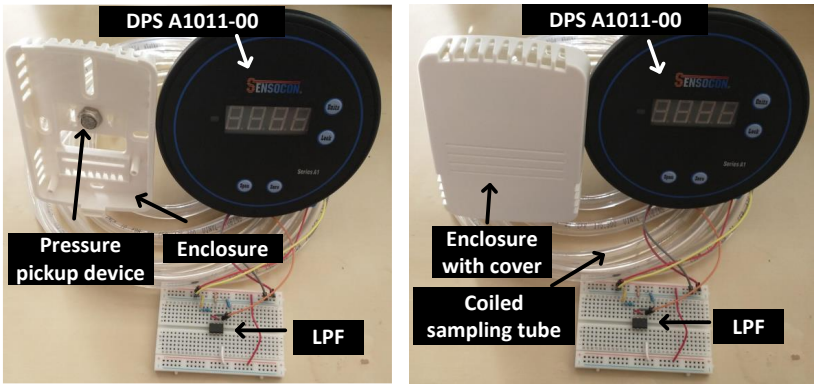


Figure 3.22: Different countermeasures to prevent the attack.

Filtering the resonant frequency: Though the DPSs don't use their LPFs to remove the resonant frequency, the authority of the NPR facility can ask the company, that install the RPM system or BMS, to cascade an LPF just after the DPS. The LPF must have a

lower cut-off frequency, such as a frequency ∼20% of the resonant frequency of a DPS (see Fig. 3.2). Therefore, the variability of the resonant frequency will not impact the safety of the DPS. For example, we use a first-order LPF built with an Op-Amp having a cut-off frequency of ∼120 Hz with the A1011-00 DPS from Sensocon (see Fig. 3.22). A low cut-off frequency of an LPF will not hamper the normal operation of a DPS in an NPR as the pressure does not change in high frequency in an NPR. Another complex approach is to use a microphone to sense the music first and then filter out the music from the pressure reading using an LPF. Similar techniques are found here [156–159]. Moreover, a guideline should be adopted by CDC or other authorities that NPRs should strictly use LPFs to protect from the resonance in DPSs.

Increasing the reference negative pressure: The CDC or other authorities should have a guideline to maintain a negative pressure higher than -2.5 Pa, such as at least 20 Pa. An attacker may find it difficult to turn a high negative pressure into a positive pressure through malicious music.

Removing audio sources: Any audio source should be removed from the close proximity to the DPS. Even CCTVs should be mounted at least 3 m away from the pressure ports in an NPR.

3.12 Related Work

To the best of our knowledge, there is no work in the literature that shows an attack on an NPR facility using malicious music by exploiting the resonant frequency of a DPS. We compare our work with the state-of-the-art works in the following four categories.

Attacks on pressure Sensors: Rouf et al. [160] used unauthenticated wireless transmission to spoof a tire pressure sensor using a radio frequency (RF) channel and attacked a moving vehicle from a close distance. Tu et al. [161] showed a deliberate EMI attack on an

inflation pump's pressure sensor while inflating a car tire and studied the attack impacts on the system's actuation. Yan et al. [162] did a formal analysis of semantic attacks on pressure sensors *without* mentioning how the pressure sensors can be attacked.

Attacks with acoustic signals: Wang et al. [50] used an ultrasonic gun to create resonance at membranes of different inertial sensors, such as MEMS accelerometers and gyroscopes and spoofed the inertial sensors to create havoc in the connected systems. Son et al. [49] used a high-power acoustic signal in audible range to compromise the gyroscope of a drone creating a resonance and made it uncontrollable. Trippel et al. [51], and Tu et al. [52] showed an adversarial control over MEMS accelerometers and gyroscopes using audible acoustic signals at their resonant frequencies. Yan et al. [45] showed an attack on ultrasonic sensors of a vehicle using acoustic waves to impair vehicle safety. Zhang et al. [47] injected acoustic commands into a microphone using ultrasonic carriers. Bolton et al. [163] showed an acoustic attack on hard disk drives.

Resonant frequencies in pressure sensors: The resonant frequency of a pressure sensor influences its dynamic characteristics [164] and is a critical parameter in designing a pressure sensor. Designers use this frequency to design resonant pressure sensors for dynamic applications, such as [165], [166], and [167]. We are not aware of any acoustic attack on pressure sensors exploiting resonant frequencies. However, designers design pressure sensors to acquire acoustic pressure in different applications, such as for cardiac pressure [168] and sound pressure [169].

Attacks on other sensors: Barua et al. [27, 170, 171] showed a non-invasive magnetic spoofing attack on Hall sensors of solar inverters, causing a shut down in a micro-grid. Kune et al. [42] attacked analog sensors using EMIs to cause defibrillation shocks on implantable cardiac devices. Davidson et al. [44] showed how spoofing optical sensors of an unmanned aerial vehicle (UAV) can compromise complete control of its lateral movement.

While the above works address the physical-level signal injection attacks on different sensors, our work differs from them in the following ways. **First**, our attack is the first of its kind that exploits resonant frequencies of DPSs to attack the RPM and HVAC systems in an NPR facility. **Second**, we intelligently use malicious music to attack NPRs for stealthiness (i.e., a wolf in sheep's clothing). **Last**, more importantly, our attack has the potential to trigger catastrophic consequences by leaking deadly microbes from an NPR, causing losses in terms of human lives and monetary resources.

3.13 Summary

We present a non-invasive attack using malicious music on DPSs located in an NPR. We show that the NPRs have RPM and HVAC systems, which use DPSs to maintain a negative pressure inside an NPR with respect to the outside reference space. We find the resonant frequency of DPSs used in NPRs by proper experiments and show that the resonant frequencies are in the audible range. We also show that the resonant frequencies of DPSs vary within a band depending on other parameters, such as the length and diameter of the sampling tube. Therefore, we insert segments of the resonant frequency band in specific interval inside of music and end the inserted segments with their peak to maintain an average forged pressure in the DPS's transducer system. As a result, the attacker can use the malicious music to fool the DPSs used in the RPM and HVAC systems of an NPR and can turn the NPR's negative pressure into a positive pressure. This may cause an alarm, resulting in chaos in the facility and has a potential to leak deadly microbes from the facility. Our attack is strong, non-invasive, and stealthy, similar to a wolf in a sheep's clothing. The consequences of leaking deadly microbes from an NPR will be catastrophic in terms of losses in human lives and monetary resources. Therefore, our attack is impactful, and the countermeasures should be adopted to prevent any future attack like ours in an NPR.

Chapter 4

Bayesian Estimation Based .bss Imposter Attack on Industrial Control Systems

4.1 Abstract

Over the last six years, several papers used memory deduplication to trigger various security issues, such as leaking heap-address and causing bit-flip in the physical memory. The most essential requirement for successful memory deduplication is to provide identical copies of a physical page. Recent works use a brute-force approach to create identical copies of a physical page that is an inaccurate and time-consuming primitive from the attacker's perspective.

Our work begins to fill this gap by providing a domain-specific structured way to duplicate a physical page in cloud settings in the context of industrial control systems (ICSs). Here, we show a new attack primitive - *BayesImposter*, which points out that the attacker can duplicate the .bss section of the target control DLL file of cloud protocols using the *Bayesian estimation* technique. Our approach results in less memory (i.e., 4 KB compared to GB) and time (i.e., 13 minutes compared to hours) compared to the brute-force approach used in recent works. We point out that ICSs can be expressed as state-space models; hence, the *Bayesian estimation* is an ideal choice to be combined with memory deduplication for

a successful attack in cloud settings. To demonstrate the strength of *BayesImposter*, we create a real-world automation platform using a scaled-down automated high-bay warehouse and industrial-grade SIMATIC S7-1500 PLC from Siemens as a target ICS. We demonstrate that *BayesImposter* can predictively inject false commands into the PLC that can cause possible equipment damage with machine failure in the target ICS. Moreover, we show that *BayesImposter* is capable of adversarial control over the target ICS resulting in severe consequences, such as killing a person but making it looks like an accident. Therefore, we also provide countermeasures to prevent the attack. The findings in this chapter have been published in [172].

4.2 Introduction

Historically, Industrial Control Systems (ICSs) follow the ANSI/ISA 95 model [173], where *disconnected* computer systems and *isolated* sensor frameworks were used to screen various operations and tasks in lower *levels* of the *automation pyramid* [174]. As we enter the fourth industrial revolution [175] (Industry 4.0), the ANSI/ISA95 model is going under different transformations. These transformations include the vertically/horizontally *interconnected* and *decentralized* ICSs in all levels of the *automation pyramid* for flexible monitoring and control. The decentralization of ICSs in Industry 4.0 adds fuel to movement to the Industrial Internet of Things (IIoT) trend, where *cloud servers* and *virtualization* [176] play an important role by providing easy-to-access automation platforms.

In Industry 4.0, Infrastructure-as-a-Service (IaaS) enables Programmable Logic Controllers (PLCs) to connect with clouds [177]. Moreover, to support multiple PLCs and supervisory platforms, today's ICSs use multiple Virtual Private Servers (VPSs) in a single cloud platform [178]. The cloud server has memory deduplication feature enabled [179], which is a *widespread optimizing* feature present in today's cloud servers to support virtualization. In this typical ICS platform, the user sends control programming and supervisory commands from VPSs

using cloud protocols (i.e., MQTT, AMQP) to PLCs [180]. The cloud protocol's software stack has a specific DLL file, which transports these commands and is located in the server computer. We call this specific DLL file as *target control DLL* file.

In this paper, at first, we show that *the .bss section* of the target control DLL file of cloud protocols transports the critical control commands from VPSs to PLCs (i.e., lower level of the automation pyramid). Next, after identifying the target control DLL file, we introduce the *Bayesian estimation* by which an attacker can recreate or fake the memory page of the .bss section of the target control DLL file. We name the fake .bss section[1]as the *.bss imposter* and denote the attack model by *BayesImposter*.

The intuition behind *BayesImposter* is that as ICSs can be expressed as state-space models [181], our *BayesImposter* exploits the *Bayesian estimation* technique to accurately predict the current state of the industrial controller. As control commands are directly related to the current states of the industrial controller, after estimating the states, the attacker can also estimate the control commands from the estimated states. As the .bss section contains the control commands, hence, the attacker can successfully recreate the .bss section using the estimated control commands. We show that our proposed *Bayesian estimation* results in less memory and attack time to recreate the page of the *.bss imposter* compared to the brute force approach demonstrated in recent works [182–185].

After recreating the fake .bss section, *BayesImposter* uses the underlying memory deduplication feature enabled in the cloud to merge the page of the fake .bss section with the legitimate .bss section. In this way, the attacker can locate the memory address of the fake .bss section in the host machine and can use a malicious *co-located VPS* to trigger a bit-flip in the page of the .bss section using the Rowhammer bug [182–185] of the host machine. As the .bss section contains the control commands, this paper shows that a bit flip in this section may

[1]In this paper, *the .bss section* means the .bss section of the target control DLL file of cloud protocols; unless otherwise mentioned.

cause corruption or even change the actual command. This method can be termed as *false command injection*. The injected false commands propagate from VPSs to the PLCs and may cause an unplanned behavior with catastrophic machine failure in the target ICS. It is worthwhile to mention here that, as *BayesImposter* has more control over the recreation of a fake .bss section, our attack is capable of *adversarial control* over the target ICS from a *co-located VPS* on the same cloud. To the best of our knowledge, *BayesImposter* is the first work that successfully merges the idea of *Bayesian estimation* of the state-space models of ICSs with the memory deduplication and the Rowhammer bug in cloud settings in the context of ICSs.

Technical Contributions: Our contributions are:

• We are the first to point out how the .bss section of the target control DLL file of cloud protocols can be exploited by using memory deduplication in modern ICSs.

• We are the first to introduce Bayesian estimation to recreate the .bss section. Our attack requires less memory and time compared to the brute force approach used in recent works [182–185].

• We create a real-world scaled-down factory model of a practical ICS, which has an automated high-bay warehouse from *fischertechnik* [186]. We use an industrial-grade PLC with a part# SIMATIC S7-1500 [187] from Siemens to create the automation platform and connect the PLC to clouds using industry-standard cloud protocols.

• We evaluate *BayesImposter* in our factory model considering five variants of industry-standard cloud protocols and show the adversarial control to generalize our attack model in cloud settings. The demonstration of our work is shown in the following link: https://sites.google.com/view/bayesmem/home.

4.3 Background

4.3.1 Connecting PLCs with clouds

IIoT enables PLCs to upload the acquired data directly to clouds [188]. PLCs are connected to clouds normally in two ways: using an adapter or directly using a standard protocol. Standard cloud protocols, such as MQTT and AMQP support *bidirectional* and *event-based* data transmission between PLCs and upper managements. The upper management can modify control functions of PLCs in run-time by flashing new *control programs* to PLCs from clouds.

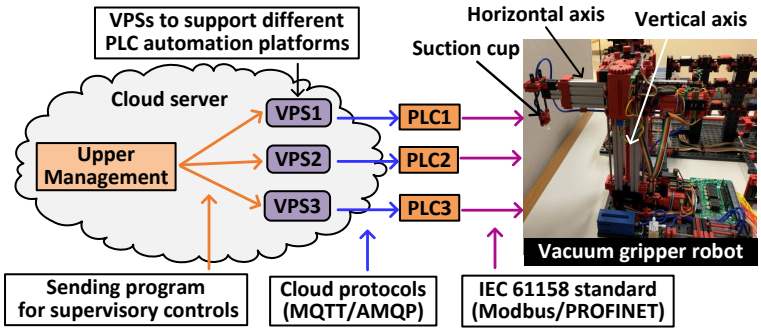


Figure 4.1: Different components of an ICS in cloud settings.

4.3.2 Programs for supervisory controls

The IEC 61131 programming standard [189] is used for control programming of PLCs. Control programs can be broadly divided into three categories: (i) programs for basic functions, (ii) programs for supervisory controls, and (iii) programs for critical time-constraint functions (e.g., security and real-time response, etc.). Traditionally, all these three categories of control programs were implemented in PLCs in industrial premises. However, with the new trend in Industry 4.0, nowadays, only the programs for critical time-constraint functions are implemented in PLCs. Programs for basic functions and supervisory controls are not implemented in PLCs; rather, they are implemented in clouds or in web-server. For example, basic functions and supervisory control programs are outsourced as web services to a cloud or to a

server for class C33 PLC controller [180]. *This gives more flexibility to upper managements as they can change programs remotely in run-time to tackle abruptly changing situations.*

4.3.3 Use of VPSs with PLCs

ICSs are becoming more complex in Industry 4.0. ICSs often need to support multiple automation platforms that may conflict with each other. Moreover, multiple PLC controllers and supervisory platforms may need multiple software packages that may require multiple operating systems. Also, introducing web servers and clouds to ICSs increases the necessity of using multiple private servers. As using multiple separate physical machines to support multiple automation platforms or operating systems or private servers is one of the available solutions, industries evidently use VPSs to reduce the number of required physical machines to reduce cost [190]. Moreover, modern cloud platforms offer cheap access to VPSs by sharing a single server among multiple operating systems on a single server machine using virtualization software [191].

4.3.4 A motivational example of an ICS

A motivational example is shown in Fig. 4.1 where we consider an automated high-bay warehouse as our example ICS. It has a vacuum gripper robot, which stores objects in the storage rack of the warehouse using a suction cup and moves along the horizontal and vertical axis. We elaborate more on this in Section 4.8.1 while demonstrating our attack model. Here, multiple PLCs having different platforms are supported by a cloud using multiple VPSs. Upper management located in the cloud send programs for supervisory controls from VPSs to PLCs using cloud protocols (i.e., MQTT/AMQP). PLCs communicate with the underlying sensors and controllers using IEC 61158 standard protocols (e.g., Modbus, PROFINET, etc.). Given this background, an attacker can perturb the supervisory control commands (i.e., false command injection) in our example ICS and remotely hamper its

normal operation using our attack model - *BayesImposter*.

4.3.5 Memory deduplication

Memory deduplication is a process that merges identical pages in the physical memory into one page to reduce redundant pages having similar contents. It is a widely used feature in cloud servers allowing multiple VPSs to run on less allocated memory in a single physical machine. The amount of redundant pages can be as high as 86% [192] and memory deduplication can save up to 50% of the allocated memory in the cloud server [193]. This feature is available in Windows 8.1, Windows Server 2016, 2019, and 2022 and Linux distribution. Windows Servers have it as Data Deduplication [194] and Linux distributions have it as Kernel Samepage Merging (KSM), which is implemented in Kernel-based Virtual Machine (KVM).

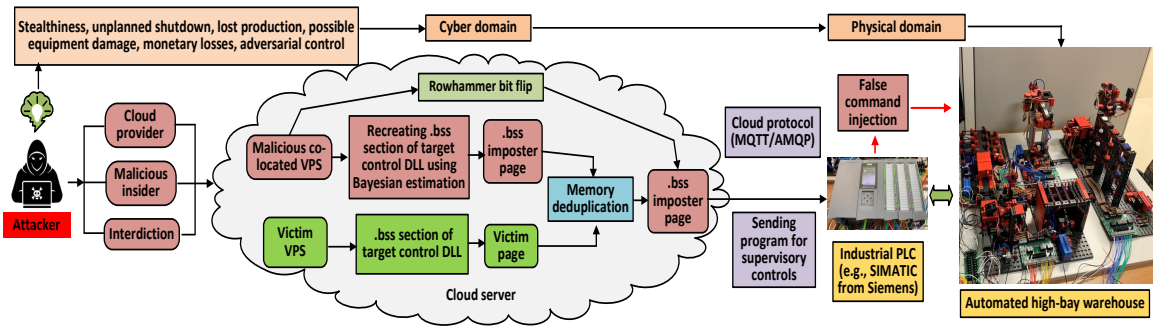


Figure 4.2: Different components of our attack model - *BayesImposter* on industrial control systems in cloud settings.

4.4 Attack model

Fig. 4.2 shows the attack model - *BayesImposter* in cloud settings. The essential components of *BayesImposter* are described below.

(i) Target system: We consider an infrastructure [195] where PLCs are connected with a cloud for maintenance and control programming, and multiple Virtual Machines (VMs)

acting as VPSs are located in the same cloud to support multiple automation platforms. As multiple VPSs in the same cloud share the same hardware, an attacker can exploit the shared hardware from a co-located VPS.

(ii) Attacker's capabilities: Let us consider a scenario where a user gives commands from his proprietary VPS to a PLC to do control programming and supervisory controls.

• **.bss imposter:** A few specific DLL files (i.e., target control DLL) of the cloud protocols transport these commands from VPS to PLCs. These DLL files are organized into different *sections*. Each section can be writable or read-only and can encapsulate executable (i.e., code) or non-executable (i.e., data) information. The section, which encapsulates uninitialized data, is known as *.bss* section. The .bss section of the target control DLL contains control programming and supervisory control specific information/data, which are mostly *boolean* type coming from the user as commands. This .bss section is page-aligned in virtual memory as well as in physical memory. Let us denote this as *victim page*. If an attacker can recreate the *victim page*, the attacker can use this *recreated victim page* (a.k.a., *.bss imposter page*) to trigger memory deduplication.

• **Bottleneck:** To recreate the *victim page*, the attacker needs to guess all the initialization values of uninitialized variables of the .bss section. As there could be hundreds of control variables present in the .bss section, this is **almost impossible** for the attacker to successfully guess the *victim page* and recreate it following the brute force approach adopted in recent works [182–185]. The brute force approach was successful in [182–185] because they only guessed a specific 32-bit data to recreate a *victim page*. To guess hundreds of variables in the .bss section, the brute force approach could require hundreds of hours. Moreover, the attacker may need to spray the physical memory with terabyte amount of recreated pages to initiate a successful attack in the brute-force approach.

• **Solution:** Thankfully this challenge can be handled by using *BayesImposter*. The intu-

ition behind *BayesImposter* is that if an attacker knows the state-space model of the ICS, the attacker can estimate the boolean and non-boolean control commands because the control commands are directly correlated with the current states of an ICS. As the .bss section transports the control commands, the estimation of the control commands helps the attacker to successfully guess the control variables present in the .bss section leading to a successful recreation of the *victim page* (i.e., *.bss imposter page*).

• **Memory deduplication + Rowhammer:** After recreating the .bss imposter page using our *BayesImposter*, the attacker can initiate memory deduplication to merge the *victim page* with the attacker's provided *.bss imposter page*. In this way, the attacker maps the *victim page* in his address space to initiate the Rowhammer attack on the *.bss imposter page* from his address space. It can flip bits in the *.bss imposter page* and change values of control commands.

(iii) Outcomes of the attack: As the .bss section contains important data dedicated to control programming and supervisory controls, the bit flips in the .bss section may lead to potential failure in ICSs. It can cause an unplanned shutdown, possible equipment damage, catastrophic machine failure, monetary losses, or even can kill a person but making it looks like an accident in the target ICS.

(iv) Attacker's access level: Our attack requires the deployment of a malicious co-located VPS in the cloud where the victim VPS resides. As public clouds are not common in ICSs, the clouds in ICSs can be either private or hybrid. The access needed to private or hybrid clouds can be possible in at least three scenarios.

In the first scenario, the attack can be originated from the cloud provider targeting the VPS of cloud users [196]. As cloud providers provide software, platform, and infrastructure as service [197], they have physical access to target clouds where the victim VPS resides.

In the second scenario, a malicious insider [198, 199], which can be a disgruntled employee,

can use his insider knowledge of the system to deploy the malicious co-located VPS. A similar incident is found in the literature where a disgruntled ex-employee of an ICS posted a note in a hacker journal indicating that his insider knowledge of the system could be used to shut down that ICS [68].

The third scenario is interdiction, which has been rumored to be used in the past [70–72] and has been recently proven to be practically feasible [74]. In this scenario, during interdiction, a competitor can intercept the installation of VPS in clouds while providing service and may deploy the malicious VPS.

(v) Stealthy attack: The authorities may not be aware of the co-located malicious VPS and would possibly not detect the source of our attack. In this sense, our attack is stealthy and can alter the normal behavior of PLCs in ICSs while remaining unidentified.

(vi) Attacker's cost: Most of these specific DLLs are available as open-source, and very few are proprietary. To acquire the open-source DLL files, the attacker has a zero cost. To acquire the DLL files of the proprietary cloud protocols, the attacker just needs to buy a basic commercial license that may cost a minimum of $100 [200]. Moreover, most proprietary cloud protocols have a free evaluation for few days, and the attacker can also use this free evaluation period to access the .bss section of the target control DLL.

4.5 .bss section of target control DLL

To recreate the .bss imposter page, the attacker first needs to find the target control DLL file of cloud protocols (i.e., MQTT, AMQP) that transports the control commands from the VPS to PLCs.

4.5.1 Target control DLL file

Mostly, the name of the target control DLL file depends upon the cloud protocol's implementation variants. For example, the name of a popular implementation of MQTT cloud protocol is Mosquitto, and the target control DLL file for this variant to access by the attacker is mosquitto.dll. We do an exhaustive search and tabulate five popular variants of MQTT and their target control DLL files in Table 4.1. The same approach is equally applicable to other cloud protocols. The DLL files are located in the parent directory of the installation folder in the cloud.

Table 4.1: Target control DLL file of cloud protocol variants

Sl.	Cloud protocol variants	Target control DLL
1	EMQ X Broker [201]	erlexec.dll
2	Mosquitto [202]	mosquitto.dll
3	MQTT-C [203]	mqtt_pal.dll
4	eMQTT5 [204]	MQTT_client.dll
5	wolfMQTT [205]	MqttMessage.dll

4.5.2 Format of target control DLL files

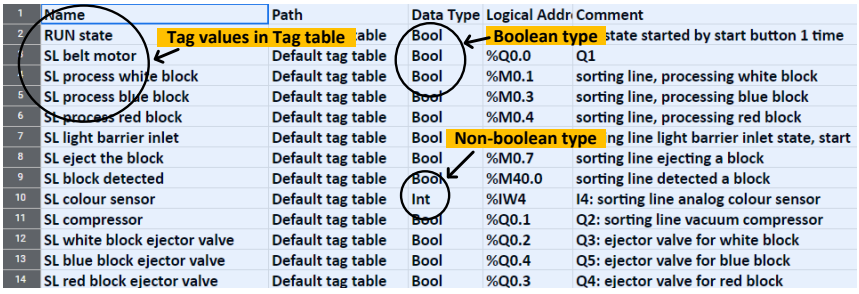
In 64-bit Windows, DLL files follow Portable Executable 32+ (PE32+) format. In high level, PE32+ has a number of *headers* and *sections* (Fig. 4.4). The header consists of DOS header, PE header, optional header, section headers, and data directories. These headers have *Image base Address* and relative virtual address (RVA) of every section that tells the dynamic linker how to map every section of the DLL file into physical memory. There are different sections placed after headers in DLL. Among different sections in DLLs, we want to mention four sections, namely .rdata, .data, .text, and .bss sections. The .rdata section contains string literals, the .data section contains global/static initialized variables, the .text section contains machine code of the program, whereas the .bss section contains zero-initialized variables. It is important to note that all these sections are *page-aligned* [206]. This means that these sections must begin on a multiple of a page size in both virtual and physical memory. These

sections of DLL files are mapped to pages in physical memory after the *base-relocation* [206]. The base-relocation is randomized, and the *ASLR technique* is used to map these sections to pages in physical memory at *load time* by the operating system.

4.5.3 Reasons for choosing the .bss section

The intention of the attacker is to find a section in the DLL file that has less entropy, which leads to a successful guess of the section. As the .rdata, the .data, and the .text sections consist of different unknown data and addresses, the pages in physical memory corresponding to these three sections have higher entropy. Hence, the estimation of these pages by the attacker requires large memory and time [183] that is not computationally feasible.

On the other hand, we examine that the .bss section of a target control DLL file of cloud protocols (i.e., MQTT, AMQP) is responsible for transporting control programming and supervisory control-related data, which are static except a new control command is issued. The .bss section contains different uninitialized global/static variables. They are also known as *tag values* and are organized in a tag table. The tag table is typically placed in the .bss section.



	Name	Path	Data Type	Logical Addr	Comment
2	RUN state	able	Bool		state started by start button 1 time
3	SL belt motor	Default tag table	Bool	%Q0.0	Q1
4	SL process white block	Default tag table	Bool	%M0.1	sorting line, processing white block
5	SL process blue block	Default tag table	Bool	%M0.3	sorting line, processing blue block
6	SL process red block	Default tag table	Bool	%M0.4	sorting line, processing red block
7	SL light barrier inlet	Default tag table	Bool		ng line light barrier inlet state, start
8	SL eject the block	Default tag table	Bool	%M0.7	sorting line ejecting a block
9	SL block detected	Default tag table	Bool	%M40.0	sorting line detected a block
10	SL colour sensor	Default tag table	Int	%IW4	I4: sorting line analog colour sensor
11	SL compressor	Default tag table	Bool	%Q0.1	Q2: sorting line vacuum compressor
12	SL white block ejector valve	Default tag table	Bool	%Q0.2	Q3: ejector valve for white block
13	SL blue block ejector valve	Default tag table	Bool	%Q0.4	Q5: ejector valve for blue block
14	SL red block ejector valve	Default tag table	Bool	%Q0.3	Q4: ejector valve for red block

Figure 4.3: Tag values in tag table of the TIA portal.

An example of the tag values: We use a real-world testbed of an automated high-bay warehouse from *fischertechnik*. The warehouse is connected with a SIMATIC S7-1500 PLC from Siemens. The PLC communicates with the cloud using a TIA portal [207] through the

MQTT cloud protocol Mosquitto. A snippet of tag values in the tag table sent from the TIA portal to the SIMATIC PLC are shown in Fig. 4.3. A complete list of the tag values is provided in the following link: https://sites.google.com/view/bayesmem/home.

If we analyze the tag values in tag tables (Fig. 4.3), we can observe that tag values correspond to particular states of the target ICS, e.g., the position of a vacuum gripper robot in the warehouse. Most of the tag values are boolean, and very few of them are other data types. The initialization of tag values to either *0 or 1* or non-boolean values in .bss section depends on states of the target ICS and increases entropy. Therefore, it provides a challenge to the attacker to successfully recreate the .bss section. Thankfully, this challenge can be handled by using the Bayesian estimation of specific command data in the .bss section. This process is discussed in the next section.

4.6 Bayesian estimation of .bss section

We first mathematically model ICSs using the Bayesian estimation and then use the model to recreate the .bss imposter page.

Proposition 1- State-space model of an ICS: An ICS is dynamic in nature and can be expressed as a discrete-time state-space model [181]. Therefore, a control system in ICS can be expressed by a state vector x_k, which is a parameter of interest, and a measurement vector y_k, which is the measurement for x_k at discrete-time index k (see Fig. 4.4). The terms x_k and y_k can be expressed as:

$$x_k = f_{k-1}(x_{k-1}, q_{k-1}) = p(x_k|x_{k-1}) \tag{4.1}$$

$$y_k = h_k(x_k, r_k) = p(y_k|x_k) \tag{4.2}$$

where q_{k-1} and r_k are state noise and measurement noise vector respectively, and they are mutually exclusive. Please note that both x_k and y_k are stochastic processes, and Eqn. 4.1 implies that current state x_k at time index k depends on only the previous state x_{k-1} at time index $k-1$ (i.e., Markov process). We implement the state space model of ICS in lines 2-3 of our *BayesImposter* algorithm 3.

Source of the data to create the state-space model: To create the state-space model and to estimate x_k and y_k, the main challenge for the attacker is to gather the previous states, $x_{1:k-1}$ and previous measurements, $y_{1:k-1}$. *The attacker can gather $x_{1:k-1}$ and $y_{1:k-1}$ from OPC tags, historian data, specific PLC block information, or network traffic [199]. Moreover, as mentioned in Section 4.4, the cloud provider, or a malicious insider, or an interdiction method can make it possible to get $x_{1:k-1}$ and $y_{1:k-1}$ from these sources.* The attacker can use $x_{1:k-1}$ and $y_{1:k-1}$ to create a probabilistic graphical model - Bayes net, which is a directed acyclic graph describing how a joint density can be factorized. The Bayes net also illustrates conditional dependencies among all the states in the ICS (Fig. 4.4).

The tag values located in the .bss section are directly related to the current states (x_k) and measurements (y_k). Therefore, *BayesImposter* has the following two parts:

Part 1. Estimation of the current states (x_k) and measurements (y_k) of the state-space model.

Part 2. Estimation of tag values from the estimated x_k and y_k.

4.6.1 Estimation of states and measurements

At first, we define the univariate and multivariate ICS to provide background on the design space of the state-space model of ICSs.

Definition 1 (Univariate ICS). We define an univariate ICS as where each state x_k has only a single measurement quantity y_k at any time step k.

Definition 2 (Multivariate ICS). We define a multivariate ICS as where each state x_k has multiple (i.e., n number) measurement quantities, $[y_k^1, y_k^2, ..., y_k^n]$ at any time step k.

Practically speaking, an ICS is a mixture of univariate and multivariate state-space models. Therefore, the main challenge for the attacker is to satisfactorily estimate the current state x_k and measurement y_k for both univariate and multivariate ICSs. To handle this challenge, we bring Propositions 2 and 3 to estimate x_k and y_k for a univariate ICS and Propositions 4 and 5 for a multivariate ICS.

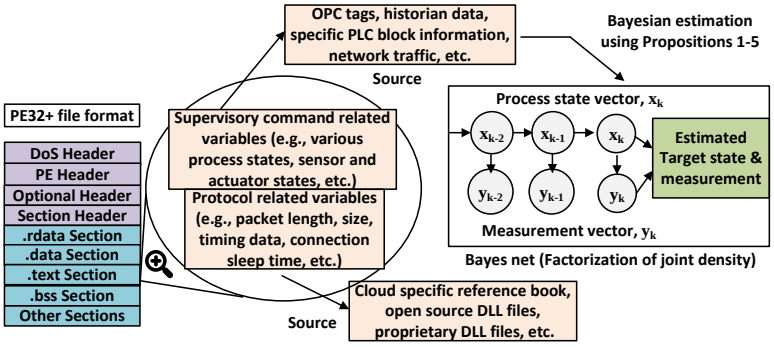


Figure 4.4: An overview of duplicating the .bss section of the target control DLL file.

Proposition 2: *BayesImposter* can predict the current state x_k at time k if the attacker has information only on the previous state x_{k-1} and previous measurements $y_{1:k-1}$, by using the Chapman-Kolmogorov equation. Here, $y_{1:k-1}$ consist of all previous measurement data $[y_1\, y_2 \ldots y_{k-1}]$ up-to time $k-1$.

Explanation of Proposition 2: Let us give an example to clear this concept. Let us

denote the *states* of a suction cup of the vacuum gripper robot in our example warehouse as x_k at time k. Let us consider the suction cup can be in one of two states, $x_k \epsilon \{\text{ON, OFF}\}$. The activation of the suction cup in each state depends on the position of the horizontal and vertical axis of the vacuum gripper robot (see Fig. 4.1). The position measurement can be expressed by y_k at time k. If the attacker knows previous state x_{k-1} of the suction cup and previous position measurements $y_{1:k-1}$, then the attacker can use these data to accurately estimate the current state x_k at time k by using Eqn. 4.3 (i.e., Chapman-Kolmogorov equation). The L.H.S of Eqn. 4.3, $p(x_k|y_{1:k-1})$, is a conditional estimation of current state x_k, while previous measurements $y_{1:k-1}$ are given. The R.H.S of Eqn. 4.3 depicts that $p(x_k|y_{1:k-1})$ is a function of previous state, x_{k-1}, that is an indication of Markov process. The Proposition 2 is implemented in lines 6-7 of our *BayesImposter* algorithm 3.

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \tag{4.3}$$

An example: The name of a specific tag value in the .bss section of the mosquitto.dll is `suctionstate`, which corresponds to the state information $x_k \epsilon \{\text{ON, OFF}\}$ of the suction cup of our example automated high-bay warehouse. After estimating the state x_k using Eqn. 4.3, the attacker can initialize the tag value to 0 or 1 of the variable `suctionstate` in the .bss section. If the .bss section contains multiple uninitialized tag values originating in the VPS, the attacker can use a similar technique to successfully estimate all uninitialized tag values and can recreate the .bss section.

Proposition 3: *BayesImposter* can predict the current measurement y_k if the attacker has information on current state x_k.

Explanation of Proposition 3: It is important to note that along with state information x_k, the .bss section transports current measurement y_k from VPSs to PLCs. The importance

of sending measurement information y_k from VPSs to PLCs is explained below.

An example: In the automated high-bay warehouse, a solenoid is present in the suction cup of the vacuum gripper robot that is turned on/off if the position of the horizontal and vertical axis is above/below a threshold position. Let us denote this threshold position by S_θ. If the threshold position is required to be changed by the upper management located in the cloud, the VPS can send a new threshold position S_k^θ to overwrite the previous value S_{k-1}^θ. The new threshold position S_k^θ is equivalent to the current measurement y_k, which depends on the current state x_k of the suction cup. Therefore, the current measurement, $y_k = S_k^\theta$, can be calculated using the Naive Bayes estimation equation as below:

$$p(y_k = S_k^\theta | x_k) = \frac{p(x_k | y_k = S_k^\theta) \times p(y_k = S_k^\theta)}{\sum_{y_k} p(y_k) p(x_k | y_k)} \tag{4.4}$$

Here, the likelihood term, $p(x_k | y_k = S_k^\theta)$, is calculated from the frequency distribution of the measurement y_k for the state x_k. The frequency distribution is calculated from the OPC tags and the historian data (Fig. 4.4). The prior probability, $p(y_k = S_k^\theta)$, is the probability that the parameter takes on a particular value S_k^θ, prior to taking into account any new information (i.e., current state x_k). If the probability of the estimation, $p(y_k = S_k^\theta | x_k)$, is below a cut-off value (K_c), *BayesImposter* discards that estimation and picks another $y_k = S_k^\theta$ to test in Eqn. 4.4. By this way, the attacker can use *BayesImposter* to estimate any measurement quantity y_k at time step k. It is noteworthy that if the current state x_k is unknown, *BayesImposter* can use the Proposition 2 to calculate the current state x_k first, and then use the Proposition 3 to calculate $p(y_k | x_k)$ using Eqn. 4.4. The Proposition 3 is implemented in lines 9-17 of our proposed *BayesImposter* algorithm 3.

Proposition 4: If multiple (i.e., n) measurement quantities, [y_k^1, y_k^2, y_k^3,..., y_k^n], at a time step k, jointly contribute to estimate any state x_k, *BayesImposter* uses the joint probability

of multiple measurement quantities, $p(y_k^1 \cap y_k^2 \cap y_k^3 \cap ..... \cap y_k^n)$, in Eqn. 4.3.

Explanation of Proposition 4: Let us assume that each state x_k in a multivariate ICS has n number of measurements at every time step. For example, at state x_1, the ICS has $y_1^1, y_1^2, y_1^3, ..., y_1^n$ measurement values; at state x_2, the ICS has $y_2^1, y_2^2, y_2^3, ..., y_2^n$ measurement values and so forth. Let us denote the joint probability of n number of measurement values at state x_k by $Y_k = p(y_k^1 \cap y_k^2 \cap y_k^3 \cap ..... \cap y_k^n)$. Eqn. 4.3 is modified in the following way to accommodate the joint probability of measurement values.

$$p(x_k|Y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|Y_{1:k-1})dx_{k-1} \tag{4.5}$$

where joint probability of measurement values from time step 1 to $k-1$ is denoted by $Y_{1:k-1}$. The Proposition 4 is implemented in lines 20-22 of our proposed *BayesImposter* algorithm 3.

An example: From the explanation of the Proposition 2, we know that the suction cup can have any one of the following two states: $\{ON, OFF\}$, depending upon the position of the horizontal and vertical axis of the vacuum gripper robot. In multivariate ICS, instead of having a single position value for a particular state, the horizontal and vertical axis could have multiple position values within a range. For example, a position within 0 cm to 10 cm of the horizontal axis could trigger the state to ON from OFF. If there are n measurement values within the position range of 0 cm to 10 cm, *BayesImposter* uses Eqn. 4.5 to estimate the next state x_k.

Proposition 5: If multiple (i.e., n) measurement quantities, $[y_k^1, y_k^2, y_k^3, ..., y_k^n]$, at a time step k, present in a multivariate ICS, *BayesImposter* finds y_k that gives the highest probability in Eqn. 4.4.

Algorithm 3: BayesImposter Algorithm.

Input: Previous measurements, $y_{1:k-1}$ and states $x_{1:k-1}$ up to k-1
Output: Current measurements, y_k and states, x_k at k step

```
for k ← 1 to k-1 do // Proposition 1 for state-space model
    Collect y_{1:k-1} and x_{1:k-1} information of ICS
    Create state-space model: x_k = p(x_k|x_{k-1}) & y_k = p(y_k|x_k)
    if ICS is univariate then
        for Each unknown x_k do // Proposition 2 for x_k
            Find p(x_k|y_{1:k-1}) for every x_k
            Select x_k having the highest p(x_k|y_{1:k-1})
        end
        for Each unknown y_k do // Proposition 3 for y_k
            if x_k is known then
                Find p(y_k|x_k) for every x_k
                if p(y_k|x_k) > cut-off K_c then
                    Select the y_k as the estimation
                end
                else
                    Discard the estimated y_k
                end
            end
            else
                Find x_k first using Proposition 2
                Then use Proposition 3
            end
        end
    end
    if ICS is multivariate then
        for Each unknown x_k do // Proposition 4 for x_k
            Find joint probability Y_k = p(y_k^1 ∩ y_k^2 ∩ ..... ∩ y_k^n)
            Find p(x_k|Y_{1:k-1}) for every x_k
            Select x_k having the highest p(x_k|Y_{1:k-1})
        end
        for Each unknown y_k do // Proposition 5 for y_k
            if x_k is known then // max_{∀y_k} function
                Find p(y_k^1|x_k) for y_k ϵ {y_k^1, y_k^2, .., y_k^n}
                max ← p(y_k^1|x_k)
                for Every y_k ϵ {y_k^2, y_k^3, .., y_k^n} do
                    Find p(y_k|x_k)
                    if p(y_k|x_k) > max then
                        max ← p(y_k|x_k)
                    end
                end
                Select max as the y_k for given x_k
            end
            else
                Find x_k first using Proposition 2
                Then use Proposition 5
            end
        end
    end
end
```

Explanation of Proposition 5: The Proposition 5 is an extension of the Proposition 3 for multiple number of measurement values $[y_k^1, y_k^2, y_k^3, ..., y_k^n]$, at a current state x_k. To estimate a measurement value from multiple measurement values, *BayesImposter* plugs in most frequent values from the distribution of measurement values $[y_k^1, y_k^2, y_k^3, ..., y_k^n]$ in Eqn. 4.4 with an intention to maximize the left hand side of Eqn. 4.4. For example, if the threshold position in the explanation of Proposition 3 has multiple values $S_k^{\theta 1}$, $S_k^{\theta 2}$,...,$S_k^{\theta n}$ for current state x_k, we can write Eqn. 4.4 as below.

$$\max_{\forall y_k} \{p(y_k|x_k)\} = \max_{\forall y_k} \{\frac{p(x_k|y_k) \times p(y_k)}{\sum_{y_k} p(y_k)p(x_k|y_k)}\} \tag{4.6}$$

where $y_k \epsilon \{S_k^{\theta 1}, S_k^{\theta 2}, ..., S_k^{\theta n}\}$. The $\max_{\forall y_k}$ is the function that maximizes $p(y_k|x_k)$ for all y_k that is implemented using an iterative approach in lines 24-34 of the proposed *BayesImposter* algorithm 3.

4.6.2 Tag values from the estimated x_k and y_k

It is mentioned earlier in section 4.5 that the .bss section contains different uninitialized global/static tag variables. They can be broadly divided into two categories, namely the control programming or command related variables and protocol related variables (Fig. 4.4).

Estimation of control commands from x_k ***and*** y_k***:*** After estimating x_k and y_k, the next challenge is to look for the corresponding control commands from the estimated x_k and y_k. It can be done in two ways. *Firstly*, most control commands are the direct values of x_k and y_k that are already estimated by *BayesImposter*. For example, from the Proposition 2, the threshold position S_k^{θ} is equal to the estimated measurement y_k in the .bss section. *Secondly*, rest of the control commands are estimated from OPC tags and specific PLC information (Fig. 4.4) using the estimated x_k and y_k. For example, the value of `suctionstate`

$\epsilon\{ON, OFF\}$ corresponding to 0 or 1 can be found from specific PLC information (see Section 4.6.3).

Estimation of protocol related variables: The protocol-related variables are specific to cloud protocols and hence, are fixed and initialized at the load time of the control DLL file. The attacker can get the list of all the protocol-related variable names and their values from the reference book of a specific cloud protocol. As mentioned in Section 4.4, most of the target control DLLs are available as open-source, and very few are proprietary, which are accessible by a basic commercial license (cost less than $100 [200]).

4.6.3 Entropy in the .bss section

The size of the specific control variable used in the .bss section can be a maximum of 64 bits in a 64-bit machine. Therefore, we have an entropy of 2^{64} possible values. For example, the tag variable `suctionstate` ideally could have 2^{64} values. But, in real-world implementation, the control variables are problem-specific and they have very few key values, which are also problem specific. Therefore, as mentioned in Proposition 2, the state variable, `suctionstate`, has two possible key values: {ON, OFF}. So, the entropy of the `suctionstate` is not 2^{64}; instead, the entropy is only two. Moreover, these key values are declared in the header files of the program codes, and programmers, as a good practice, generally use user-defined data types, such as *Enumeration (enum)* type to declare these key values. The use of enum data type by the programmer makes the declared control variable (e.g., `suctionstate`, etc.) more predictable. For example, after careful examination of control-related application codes that are running on top of cloud protocols, we find the following code snippet that supports our observation:

```
enum statepool {0,1};

enum statepool suctionstate;
```

This indicates that the values of ON/OFF is 0 or 1. In this way, the attacker can specifically know the tag values in the .bss section to recreate the .bss imposter page.

4.7 Memory Deduplication+Rowhammer

So far, we have discussed how the attacker can recreate the .bss imposter page using *Bayes-Imposter*. Now, we discuss how the attacker uses the memory deduplication + Rowhammer bug to trigger a bit flip in the recreated .bss imposter page to corrupt control commands. As recent works [182–185] have already provided details on the memory deduplication + Rowhammer bug, we will not repeat the same details here. Instead, we provide advantages of our approach over [182–185]. Let us briefly discuss the memory deduplication + Rowhammer first.

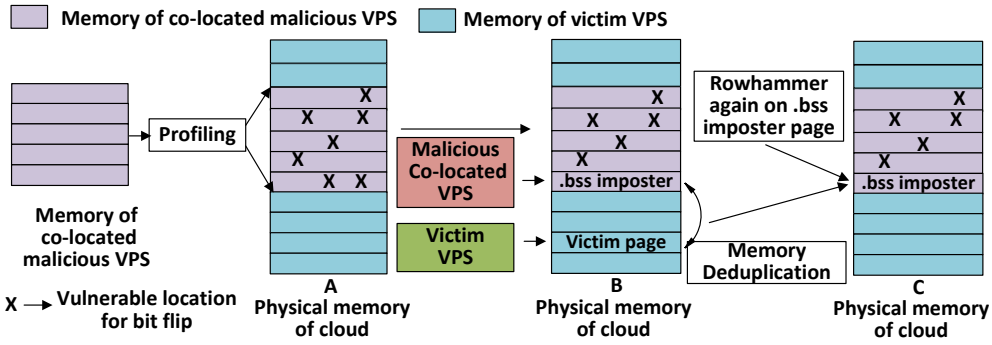


Figure 4.5: (A) Profiling the memory of cloud. (B) Placing *.bss imposter page* in the vulnerable location. (C) After memory deduplication, *victim page* is backed by the *.bss imposter page* and the Rowhammer causes bit flips in the *.bss imposter page*.

Brief overview: Memory deduplication merges identical pages located in the physical memory into one page. Rowhammer [208] is a widespread vulnerability in recent DRAM devices in which repeatedly accessing a row can cause bit flips in adjacent rows.

Memory deduplication thread (i.e., KSM) running in the host cloud hypervisor (i.e., KVM in Linux) maintains stable/unstable trees in a red-black tree format to keep track of the pages having identical contents in memory. If the *.bss imposter page* arrives first in the memory

provided from the co-located malicious VPS, the node of the red-black tree will be updated first with the *.bss imposter page*. Therefore, if the *victim page* comes later from the victim VPS, the *victim page* is merged with the *.bss imposter page*, and the *victim page* shares the same memory location of the *.bss imposter page*. In this way, the attacker can control the memory location of the *victim page* and can trigger a Rowhammer on that page.

The first step to initiate Rowhammer is to find the *aggressor/victim* addresses in the physical memory of the running system. This step is named as **profiling**. The *aggressor* addresses are the memory locations within the process's virtual address space that are hammered, and the *victim* addresses are the locations where the bit flips occur (Fig. 4.5(A)). From the profiling step, the attacker knows the aggressor rows for the vulnerable memory locations. After placing the *.bss imposter page* in one of the vulnerable locations, the attacker hammers again on the aggressor rows (Fig. 4.5(C)). This results in bit-flips in the *.bss imposter page* that in effect changes the control commands in the .bss section of the target control DLL.

4.7.1 Advantages of BayesImposter

No first precedence and two copies of target pages

To ensure that the *.bss imposter page* arrives first in the memory, the attacker's VPS should start first before the victim VPS. This is known as the first precedence. Recent works [182–185] use this technique along with creating two copies of target pages to place the *.bss imposter page* in the red-black tree before the target *victim page*. These techniques require more control over the victim VPS and may not be feasible in practical ICSs. For example, the attacker may not know when the victim VPS is started.

Thanks to the Bayesian estimation of the *victim page*. Referring to Section 4.6, if the attacker can predict the current states (x_k) and measurements (y_k), this means that he actually can predict the *victim page* before time k. As the attacker has the predicted *victim page*, the attacker can provide this predicted *victim page* to the memory deduplication thread at any

time. Hence, the attacker does not need to start his VPS before the victim or does not need to create two copies of the target pages in our attack model. This makes our attack model more practical and reliable in the context of ICSs.

BayesImposter provides simpler profiling step

Recent works [182–185] activate the *large pages* [209] in VPS to exploit the double-sided Rowhammering. However, *large pages* may not be explicitly turned on in the victim VPS. Therefore, double-sided Rowhammering may not be feasible in the context of ICSs [210]. Therefore, *BayesImposter* uses the random address selection approach for profiling the bit-flippable memory locations.

In this approach, *BayesImposter* allocated a 1 GB block of memory using a large array filled with doubles. A value of $1.79769313486231 \times 10^{308}$ is stored as double that gives 1 in memory locations. Next, the attacker randomly picks virtual aggressor addresses from each page of this large memory block and reads 2×10^6 times. Then the attacker moves to the next page and repeats the same steps. As the attacker can know the number of memory banks of the running system from his VPS, he can calculate his chance of hammering addresses in the same bank. For example, in our experimental setup, the machine has 2 Dual Inline Memory Modules (DIMMs) and 8 banks per DIMM. Therefore, the machine has 16 banks, and the attacker has a 1/16 chance to hit aggressor rows in the same bank. Moreover, the attacker hammers 4 aggressor rows in the same iteration that increases the chance of having successful Rowhammering.

4.8 Attack model evaluation

4.8.1 Automated high-bay warehouse testbed

We prepare a testbed to evaluate *BayesImposter* on a practical ICS. We choose a scaled-down model of an automated high-bay warehouse (AHBW) from *fischertechnik* connected with a vacuum gripper robot (VGR), multiprocessing oven (MPO), and sorting line (SL). The process begins first in MPO with a workpiece placed in the oven feeder. The processed workpiece from the MPO is then sent to SL using a conveyor belt. The SL sorts the workpiece depending upon color and stores it in the storage location. Next, the VGR uses its suction cup to hold the workpiece and transports it from the storage location to the pre-loading zone of the rack feeder of the AHBW. Then the rack feeder stores the workpiece in the warehouse. A video demonstration of the factory system is given here: https://sites.google.com/view/bayesmem/home.

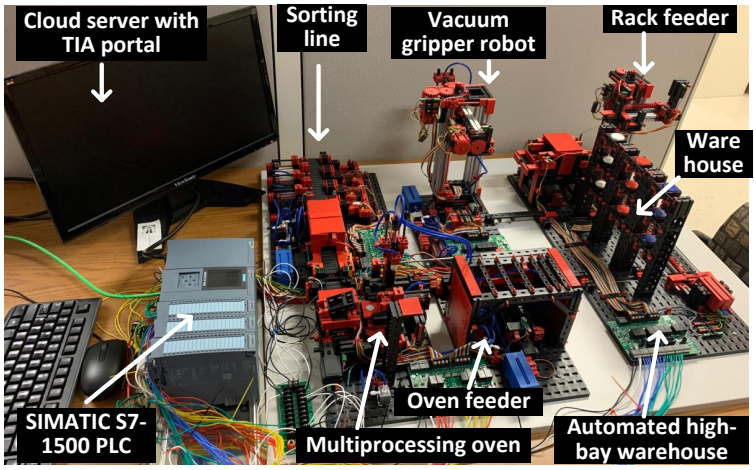


Figure 4.6: A small scale real-world testbed of automated high-bay warehouse to evaluate *BayesImposter*.

The AHBW is connected with a SIMATIC S7-1500 PLC from Siemens using 32 input/output ports and 8 analog input ports. The PLC communicates with the cloud using a TIA portal through the MQTT cloud protocol Mosquitto. The cloud server runs on Intel CPU i7-6900K

with 8 cores and 64GB of DDR3 RAM. We use Ubuntu Server 14.04.2 LTS x86_64 as the cloud server, which has a Kernel-based Virtual Machine (KVM). Memory deduplication is implemented as Kernel Samepage Merging (KSM) in KVM. The KVM is kept at its default configuration. The parameters for KSM are also kept at their default settings. *All VPSs run with Windows 10* [211] and have 2 GB of main memory. *The idea of BayesImposter is equally applicable to the Linux VPSs with .so file [183] of cloud protocols.* The victim VPS is using MQTT to communicate with the PLC using TIA portal. The testbed is shown in Fig. 4.6.

4.8.2 Estimation accuracy of *BayesImposter*

A practical ICS could have hundreds of states (x_k) and measurement values (y_k). Let us mathematically formulate this first.

Proposition 6: If an ICS has M state variables (x_k) and each state variable has N probable states, N^M combinations are possible among state variables and probable states. Similarly, If an ICS has P measurement variables (y_k) and each measurement variable has Q probable values, Q^P combinations are possible among measurement variables and probable values.

After counting, we find that our testbed - automated high-bay warehouse has $M = 420, N = 3, P = 160, Q = 4$. We find that the estimation accuracy for next states or next measurements using Propositions 1-5 of our *BayesImposter* algorithm is ∼91%. It means that *BayesImposter* can estimate the next state or measurement variables within $1/0.91 = 1.09$ attempt.

Table 4.2: Estimation accuracy of *BayesImposter*.

Estimating state variables x_k	Estimating measurement variables y_k
90.2%	91.47%

4.8.3 Recreating the .bss imposter page

The automated high-bay warehouse testbed has $M = 420$ state variables (x_k) in total, and each state has an average of $N = 3$ probable states. The brute-force approach gives $3^{420} \approx 2.4 \times 10^{200}$ combinations according to the Proposition 6. Moreover, this ICS in hand has also $P = 160$ measurement variables (y_k) in total, and each variable has an average of $Q = 4$ probable values. The brute-force approach gives $4^{160} \approx 2.13 \times 10^{96}$ combinations. In combined, there are $2.4 \times 10^{200} + 2.13 \times 10^{96} = 2.4 \times 10^{200}$ combinations are possible for the ICS in hand. For a 4KB page size, this may require $(4 \times 2.4 \times 10^{200})$ KB $= 9.6 \times 10^{194}$ GB of guessed pages. In other words, the attacker may need to spray 9.6×10^{194} GB pages in the physical memory for successful memory deduplication that is not possible in terms of time and memory. It is not possible to accommodate 9.6×10^{194} GB pages in one attempt of the attack, and the attacker may require thousands of attempts to spray the memory with the guessed pages. In contrast, as *BayesImposter* has an estimation accuracy of ∼91% (see Section 4.8.2), it does not require to guess N^M or Q^P combinations; instead, it can guess states and measurement variables in $1/0.91 = 1.09$ attempt. Therefore, most of the time, *BayesImposter* requires only one or two pages (because of ∼91% accuracy) of size 4KB to spray in the physical memory.

The victim VPS in our example ICS has a 2 GB main memory, and it takes ∼13 minutes to scan all the pages of main memory in a single attempt (see Section 4.8.7). And, out of 2 GB of memory, we can spray 1.2 GB with the guessed pages at each attempt (i.e., remaining 0.8 GB for operating systems and other applications). Therefore, brute force requires $(9.6 \times 10^{194})/1.2 = 8 \times 10^{194}$ attempts, whereas *BayesImposter* requires only a 1.09 attempt. As each attempt takes ∼13 minutes, *BayesImposter* requires only ∼13 minutes compared to $9.6 \times 10^{194} \times 13$ min. $= 2 \times 10^{194}$ hours of brute force approach which is not feasible. This reduction of attempts also reduces the attack time (see Section 4.8.7). As the attack time for *BayesImposter* is significantly low compared to a brute force approach,

BayesImposter gives more control over the ICS from the attacker's perspective. Table 4.3 shows the memory and time requirements for brute-force and *BayesImposter* approaches.

Table 4.3: Attack time of *BayesImposter*

BayesImposter		Brute force	
Guessed page	Time	Guessed page	Time
4KB or 8KB	13 min.	9.6×10^{194} GB	2×10^{194} Hr.

4.8.4 Attacking the vacuum gripper robot (VGR)

As mentioned in Section 4.8.1, the VGR uses its suction cup to transport the workpiece from the SL to the rack feeder of the AHBW. The solenoid present in the suction cup is turned on/off if the position of the horizontal and vertical axis of the VGR is above or below a threshold position. The threshold position is a measurement value (i.e., y_k) and can be estimated by *BayesImposter*. The correct value of the threshold position where the suction cup is turned off (release the workpiece) is 2 cm. The estimated value of the threshold position is also calculated as 2 cm using *BayesImposter* at a particular state (i.e., moving from SL to AHBW). After the successful estimation of the threshold position with all other tag values of the *victim page* using the same *BayesImposter*, the attacker can recreate the .bss imposter page. Now, the attacker initiates the memory deduplication + Rowhammer attack and arbitrarily causes a bit-flip in the .bss imposter page. A demonstration of the attack is shown in Fig. 4.7, which indicates the location of the occurred bit-flip in the victim row. *(0 0 1 7 3c97 0)* means address of channel 0, dimm 0, rank 1, bank 7, row 3c97, column 0 in DRAM with a row-offset 0743, which has a byte value *f7* after the bit-flip; however, byte expected according to fill pattern is *ff (i.e., all erased)*. The victim byte *f7* is the upper byte of the threshold position being corrupted that changes the 2 cm threshold position to 2050 cm. This causes an out-of-range value for the VGR resulting in a wrong drop-off location of the workpiece other than the rack-feeder. This may result in possible equipment damage or even can kill a person if the attacker drops the workpiece on a target person. A video

demonstration of this attack is given here: https://sites.google.com/view/bayesmem/home

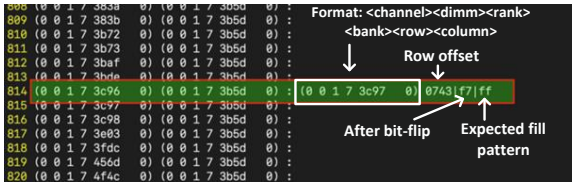


Figure 4.7: Bit-flip in the .bss imposter page.

4.8.5 Adversarial control using BayesImposter

As the attacker knows the physical location of a tag value in the tag table of the .bss imposter page, he can target a particular tag value and initiate an adversarial control over that tag value. For example, the attacker can cause a bit-flip of `suctionstate` from $1 \rightarrow 0$ and can adversarially drop the workpiece from the suction cup when it is not supposed to drop the workpiece (Fig. 4.8). This may result in possible equipment damage or even can kill a person if the attacker drops the workpiece on a target person. This adversarial control makes *BayesImposter* stronger compared to [182–185].

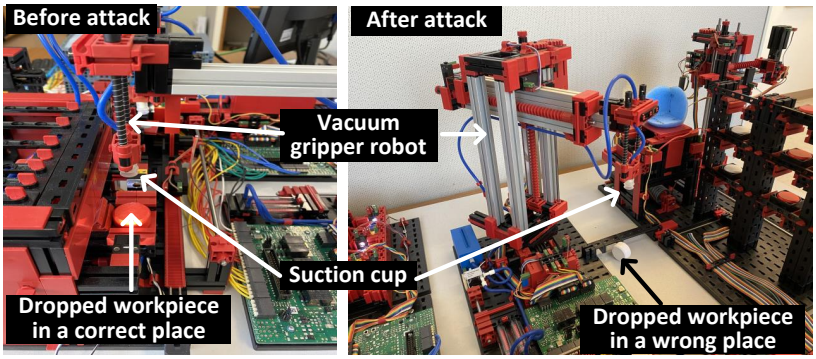


Figure 4.8: Dropping workpiece using adversarial control.

4.8.6 Profiling time in our testbed

Fig. 4.9 evaluates the profiling time (see Section 4.7) for different number of VPSs in the cloud. *BayesImposter* takes ∼51.45 seconds to complete single-sided Rowhammer for each

target row. We searched for vulnerable locations for the Rowhammer in the memory space, and Fig. 4.9 shows that to get ∼20000 vulnerable locations, ∼100 hours are required. With the increase of VPSs, this profiling time increases due to more memory pressure in the system memory. Fig. 4.9 shows the profiling time for 1, 3, and 6 VPSs in the same cloud.

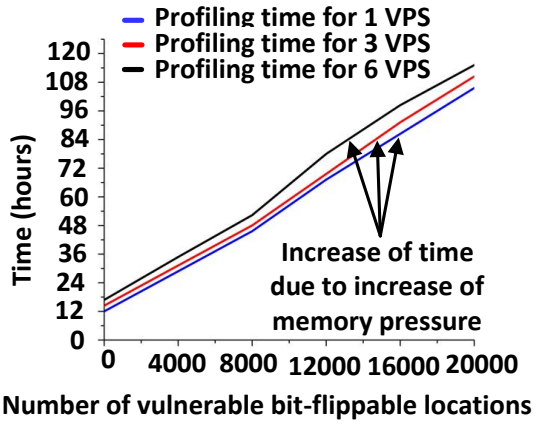


Figure 4.9: Profiling time for different number of VPSs.

4.8.7 Attack time

Here, we define attack time as how much time it takes to cause a bit flip in the .bss section. Attack time is the summation of the memory deduplication time and the Rowhammer implementation time. The exact time required for memory deduplication can be calculated using the timing side-channel [184]. However, roughly, the maximum time for memory deduplication is the time needed to scan all the memory of the co-located VPSs in the cloud. Here, for simplicity, we assume that deduplication happens within this maximum time frame, and hence, we consider this maximum time as the memory deduplication time. The memory deduplication time depends upon the parameters `pages_to_scan` and `sleep _millisec`. In default configuration, `pages_to_scan` = 100 and `sleep _millisec` = 20. Therefore, Linux/KSM can scan 1000 pages/second, which results in a total scan time of almost 5 minutes per 1GB of main memory [212]. As the victim VPS has a main memory of 2 GB, it should take approximately 10 minutes to scan all the pages in the main memory of a VPS. In

our testbed, the memory deduplication takes approx. 13 minutes, and the Rowhammering process takes approx. 51.45 seconds to complete a single-sided Rowhammer for each target row. Therefore, after summing up these two figures, the total attack time is approximately 13 minutes and 52 seconds for 1 target VPS.

Fig. 4.10 shows the memory deduplication time for five variants of MQTT cloud protocol for 1, 3, and 6 VPSs. This figure indicates that all five variants of the cloud protocol give almost equal deduplication time. As the addition of a VPS increases the scannable memory locations, the deduplication time increases with the number of co-located VPS in the cloud. The Rowhammer implementation time for a target row is almost the same for all five protocol variants.

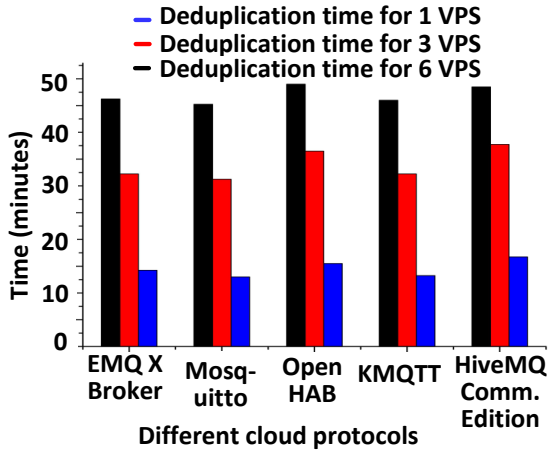


Figure 4.10: Deduplication time for different protocols.

4.8.8 Evaluation for different cloud protocols

As our attack model does not require any software bug present in the implementation of cloud protocols, state-of-the-art variants of cloud protocols should be vulnerable to our attack model. To support this claim, we implement a total of five variants of the MQTT protocol in our testbed and find that all are equally vulnerable, which proves the generalization of our attack model in ICSs.

4.9 Defense

The following mitigations should be adopted against *BayesImposter*.

Increasing entropy in the .bss section: To prevent the attack, we increase entropy in the .bss section. This is done using a random variable as a signature in the .bss section. The attacker requires a significant amount of memory and time to break this signature variable [183] as this variable is not a part of the state variable. This approach is also effective against a malicious insider.

Table 4.4: Cloud protocol variants vulnerable to *BayesImposter*

Sl.	Cloud protocol variants	Vulnerability
1	EMQ X Broker [201]	✓
2	Mosquitto [202]	✓
3	MQTT-C [203]	✓
4	eMQTT5 [204]	✓
5	wolfMQTT [205]	✓

Securing cloud server from the malicious VPS: Any unauthorized cloud provider or personnel, or visitor should not access the cloud server without the presence of authorized personnel. Periodic screening by an authorized person needs to be carried out to look for any unauthorized co-hosted VPS. Any unnecessary or suspicious co-located VPS should be considered as a security breach and should be immediately contained in the cloud.

Turning off the KSM: To prevent memory deduplication, KSM can be turned permanently off. KSM is off by default in recent Linux kernel [213]. However, the KSM service, which is included in the `qemu-kvm` package, is turned on by the KVM host in the cloud setting. We turn off the KSM using the `ksm/ksmtuned` services in the KVM host. However, turning off the KSM may increase memory usage in clouds. Therefore, it is not favorable where memory workloads are high in cloud settings [214].

Preventing Rowhammer in DRAM: The next way to prevent *BayesImposter* is to

prevent the Rowhammer in DRAM. While the built-in error-correcting codes (ECCs) can prevent single bit-flip in 64-bit words [215], it may not be enough where the Rowhammer causes multiple bit-flips [216, 217]. While only modern AMD Ryzen processors support ECC RAM in consumer hardware, Intel restricts its support to server CPUs [218]. One method to prevent Rowhammer is to increase (e.g., double) the refresh rate in DRAM chips [219]. This can reduce the probability of multiple bit-flips in DRAM, but causes more energy consumption and more overhead in the memory [208, 220]. Another method is to probabilistically open adjacent or non-adjacent rows, whenever a row is opened or closed [221]. An introduction of a redundant array of independent memory (i.e., RAIM) [222], and ANVIL [223] in the server hardware can make the Rowhammer attack infeasible. Moreover, replacing older chips with DDR4 having Target Row Refresh (TRR) capability can prevent single-sided and multi-sided Rowhammer attack on cloud networks [224]. However, [225] shows that DDR4 can also be compromised using TRR-aware attacks.

4.10 Related Work

Attacks on ICSs: The attacks on ICSs can be broadly classified as attacks on physical hardware (e.g., PLCs, control modules, etc.), attacks on communication networks, and attacks on sensing side.

Abbasi et al. [226] demonstrated an attack on PLCs by exploiting pin control operations of certain input/output pins resulting in abnormal hardware interrupt in PLCs. Garcia et al. [227] presented a malware-PLC rootkit that can attack PLCs using the physics of the underlying systems. Bolshev et al. [228] showed an attack on the physical layer (i.e., analog-to-digital converter), resulting in false data injection into PLCs. Spenneberg et al. [229] developed a worm - PLC Blaster, that independently searches any network for S7-1200v3 devices and attacks them when the protective mechanisms are switched off. *Compared to our attack model, these attacks on PLCs lack the presence of adversarial control over PLCs*

and do not provide any means of stealthiness with respect to the monitoring entity.

Klick et al. [230] showed that internet-facing controllers act as an SNMP scanner or SOCKS proxy, and their protocols can be misused by an adversary to inject false codes into PLCs, which are not directly connected to the internet. Basnight et al. [231] presented an attack on firmware exploiting communication protocols of PLCs. Beresford et al. [232] discovered vulnerabilities in Siemens S7 series communication protocol and showed a replay attack on ICSs. *Compared to these attacks, our attack model does not need any vulnerabilities in the communication protocol and does work without any presence of software bugs at any level of the system.*

Barua et al. [27, 100, 156, 171, 233], Liu et al. [234], and McLaughlin et al. [235] showed *false data injection* attack on different sensing nodes of ICSs leading to abnormal behaviour of the underlying system. *Compared to these attacks, our attack model is capable of false command injection from a remote location with adversarial control in ICSs.*

Attacks using memory deduplication and/or Rowhammer: Bosman et al. [184] demonstrated memory deduplication based exploitation vector on Windows using Microsoft Edge. Barresi et al. [183] exploited the memory deduplication in a virtualized environment to break ASLR of Windows and Linux. This attack uses brute force to duplicate the target page in the memory. Razavi et al. [182] provided Flip Fleng Shui (FFS) to break cryptosystems using both the memory deduplication and Rowhammer. **There are fundamental differences between our work and [182–184].** ***First****, our attack model exploited the .bss section of cloud protocols that is more impactful and realistic in ICSs.* ***Second****, our attack uses the Bayesian estimation to duplicate the target page compared to the brute force approach in [182–184]. This results in significantly less memory usage (i.e., in KB compared to GB) and time (i.e., in minutes compared to hours) to duplicate the target page. This makes our attack model more feasible.* ***Third,*** *our attack model demonstrates adversarial control over the target ICS that is absent in [182–184].*

Seaborn et al. [210] exploited CPU caches to read directly from DRAM using the Rowhammer bug. Gruss et al. [236] used cache eviction sets and Transparent Huge Pages (THP) for a successful double-sided Rowhammer. Tatar et al. [237] used Rowhammer attacks over the network to cause bit-flips using Remote DMA (RDMA). *Compared to these works, our work uses memory deduplication to skip the knowledge of physical memory location and uses single-sided Rowhammer on the target cloud memory. Moreover, our attack does not require any RDMA to happen that makes our attack more flexible in the context of ICSs.*

4.11 Summary

We present an attack model-*BayesImposter* that can hamper the availability and integrity of an ICS in cloud settings. We are the first to point out how the .bss section of the target control DLL file of cloud protocols is vulnerable in ICS. *BayesImposter* exploits the memory deduplication feature of the cloud that merges the attacker's provided .bss imposter page with the victim page. To create the .bss imposter page, *BayesImposter* uses a new technique that involves the *Bayesian estimation*, which results in less memory and time compared to recent works [182–184]. We show that as ICSs can be expressed as state-space models; hence, the *Bayesian estimation* is an ideal choice to be combined with the memory deduplication in cloud settings. We prepare a scaled-down model of an automated high-bay warehouse using SIMATIC PLC from Siemens and demonstrate our attack model on this practical testbed. We show that our attack model is effective on different variants of cloud protocols, and does not need any vulnerabilities in the cloud protocol, and works without any presence of software bug in any level of the system that proves a generalization of our attack model. We show that *BayesImposter* is capable of *adversarial control* that can cause severe consequences through *system demage*. Therefore, our attack is impactful, and the countermeasures should be adopted to prevent any future attack like ours in ICSs.

Chapter 5

HALC: A Real-time In-sensor Defense against the Magnetic Spoofing Attack on Hall Sensors

5.1 Abstract

Several papers have been published over the last ten years to provide a defense against intentional spoofing to sensors. However, these defenses would only work against those spoofing signals, which have a separate frequency from the original signal being measured. These defenses would *not* work if the spoofing attack signal (i) has a frequency equal to the frequency of original signals, (ii) has zero frequency, and (iii) is *strong* enough to drive the sensor output close to its saturation region. More specifically, these defenses are not designed for a *magnetic* spoofing attack on *passive Hall sensors*.

Our work begins to fill this gap by providing a defense against the magnetic spoofing attack on passive Hall sensors. Our proposed defense HALC can detect and contain all types of strong and weak magnetic spoofing, such as constant, sinusoidal, and pulsating magnetic fields, in real-time. HALC works up to ∼9000 G of external magnetic spoofing within a frequency range of 0 - 150 kHz, whereas existing defenses work only when the spoofing signals have a separate frequency from the original signal being measured. HALC utilizes the analog and

digital cores to achieve a constant computational complexity O(1). Moreover, it is low-power (∼1.9 mW), low-cost (∼$12), and can be implemented in the sensor hardware. We have tested HALC on ten different industry-used Hall sensors from four different manufacturers to prove its efficacy and found that the correlation coefficient between the signals before and after the attack is greater than 0.91 in every test case. Moreover, we demonstrate its efficacy in two practical systems: a grid-tied solar inverter and a rotation-per-minute measurement system. We find through experiments that HALC is a robust real-time defense against a magnetic spoofing attack on passive Hall sensors. The findings in this chapter have been published in [157].

5.2 Introduction

Recent decades have observed the proliferation of smart sensors in embedded and cyber-physical systems (ECPSs). One widely used sensor is the Hall sensor, which can output analog voltage proportional to the magnetic field it senses in the environment. Due to the continuous development in Hall sensing technology, nowadays, the Hall sensor has excellent accuracy, high efficiency, and good linearity, and their markets are growing rapidly [238–244]. Despite this growth, they are still not secured, and recently, it has been proved that an attacker can compromise its integrity by injecting fake external magnetic fields [48, 245], causing an intentional spoofing and denial-of-service (DoS) in ECPSs. Therefore, a robust defense for Hall sensors is much needed to protect them from intentional magnetic spoofing attacks by an attacker.

The output voltage of the Hall sensor is linear to input magnetic fields [246]. Therefore, broadly speaking, the external magnetic field injected by an attacker can introduce two types of errors in the Hall sensor's output: the attacker can inject *strong magnetic field*, which can change the sensor's output on a large scale (i.e., volt range) and drive the output from its linear region to close to its saturation region, or can spoof with *weak magnetic fields*,

which can change the sensor output in millivolt scale only in the linear region. We refer to the above definition of *strong* and *weak* magnetic fields when we mention these two terms in this paper. Moreover, Hall sensors are of two types: active and passive Hall sensors. As passive sensors [247] are naive devices, they blindly send signals to the upper level without proper authentication. Therefore, the security of passive Hall sensors is always challenging.

The state-of-the-art defenses [42, 47, 51, 248–250] target specific sensors other than passive Hall sensors, such as MEMS microphones, accelerometers, gyroscopes. However, these defenses have the following limitations: (i) They cannot contain strong spoofing signals, which can change the sensor output in volt scale, driving the output close to its saturation region. (ii) They cannot contain such spoofing signals, which have a frequency other than the resonant frequency of the target sensors. (iii) They don't work against DC/constant spoofing signals, which have a zero frequency. (iv) They cannot contain a spoofing signal if it has the same frequency as the original input signal being measured. Moreover, these defenses can handle spoofing signals having different modalities *other* than magnetic fields, such as acoustics, ultrasounds. Therefore, these defenses cannot be used for a *passive Hall sensor* [1]as it use magnetic fields.

Therefore, we propose ***HALC***[2]*: **Hal**l Spoofing **C**ontainer*, to provide a robust real-time defense against magnetic spoofing on passive Hall sensors by handling the above limitations that exist in the recent works [42, 47, 51, 248–250]. HALC can *detect and contain* all types of *weak and strong* magnetic spoofing, such as constant, sinusoidal, and pulsating fields up to ∼9000 G and can prevent both *intentional spoofing and denial-of-service* of the system.

The core idea behind HALC is that it can separate the injected fake spoofing signal from the original signal using two different cores - analog and digital core. The analog core removes the fake AC (i.e., time-dependent) magnetic fields using inexpensive fast-order filters irrespective

[1]In this paper, Hall sensors mean unipolar/bipolar, open-loop/closed-loop ***passive*** Hall sensors, unless stated otherwise.

[2]Pronounced as Hulk, who is a mighty superhero in Marvel Comics.

of their frequencies, and the digital core removes the fake DC (i.e., constant) fields using a DC feedback signal keeping the original signal intact. The analog core is implemented in such a way that it introduces two parallel paths to process inputs enabling faster signal processing. *The digital core runs a low-power algorithm with O(1) complexity that can even prevent attack signals having the same frequency/amplitude as the original input signals.* HALC is low-power and can be implemented in the sensor hardware domain. Therefore, we name this solution as *in-sensor defense* that is cheap and does not hamper the existing data-processing speed of connected systems. To the best of our knowledge, HALC is a robust real-time and in-sensor defense against the strong and weak magnetic spoofing attack on Hall sensors. We believe that the defense demonstrated here can be applied to a broad array of sensors beyond Hall sensors, including accelerometers and more.

Contributions: Our main technical contributions are:

1. We design HALC - a low-cost (∼$12) and low-power (∼1.9 mW) defense that can *detect and contain* the strong and weak magnetic spoofing in hard real-time with O(1) computational complexity.

2. We show the effectiveness of HALC through over 150 experiments on ten different Hall sensors from four different manufacturers. We experiment with different types, namely unipolar, bipolar, open-loop, closed-loop, and differential sensors to prove its efficacy.

3. We prove the efficacy of HALC in two critical systems: a grid-tied inverter in smart grids and a rotation-per-minute (RPM) system in industrial control systems (ICSs). The demonstration of HALC is shown in the following link: https://sites.google.com/view/hallspoofingcontainer/home

5.3 Background

5.3.1 Hall in-sensor components

The basic components of Hall sensors are shown in Fig. 5.1 (Left). A Hall sensor has a Hall element (i.e., p-type semiconductor), which generates a Hall voltage (V_{Hall}) proportional to an input magnetic field, B. A DC voltage bias is applied across the Hall element to energize it. The generated V_{Hall} is given as input to a differential amplifier with closed-loop feedback and a self-calibration block to reduce the measurement error. *It is clear from this discussion that state-of-the-art Hall sensors are still lacking hardware in the sensor domain to contain injected fake magnetic fields.*

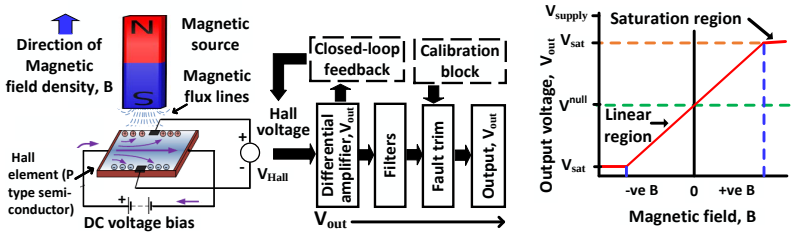


Figure 5.1: (Left) Hall *in-sensor* components of a typical Hall sensor. (Right) The transfer function of a typical Hall sensor.

Transfer function: The term V_{Hall} can be +ve or -ve because input magnetic field B can be +ve or -ve (i.e., north/south pole). Therefore, the output of the differential amplifier, denoted by V_{out}, can go either +ve or -ve from the *null-voltage* position. The null-voltage is denoted by V^{null}, which is the position of the V_{out} with no input magnetic field (i.e., $B = 0$). Therefore, the transfer function of a typical Hall sensor can be expressed as:

$$V_{out} = (K \times B) + V^{null} \tag{5.1}$$

where K is a coefficient. The graphical representation of Eqn. 5.1, which is shown in Fig. 5.1 (Right), indicates that V_{out} linearly varies with the input magnetic field B. As mentioned

earlier, the existing defenses [42, 47, 51, 248–250] work against weak magnetic spoofing, which can vary the output in its linear region, but don't work against strong magnetic spoofing, which can change V_{out} in volt scale, driving close to the saturation voltage, V_{sat}.

5.3.2 Passive and active Hall sensor

A passive Hall sensor can simply detect magnetic fields coming from the environment, whereas an active Hall sensor [251] transmits a signal first and gathers data after the reflection of that transmitted signal from a target. PyCRA [252] works only with the active sensor but *does not* work with the passive one. State-of-the-art passive Hall sensors are largely blind that relay signals to the upper level without considering the signal integrity. *Therefore, our proposed defense targets passive hall sensors.*

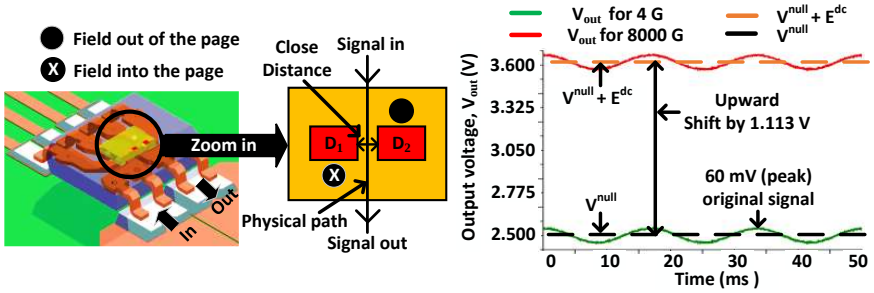


Figure 5.2: (Left) A differential Hall sensor. (Right) A differential Hall sensor may not work against a strong field.

5.3.3 Differential Hall sensor

The differential Hall sensor is the state-of-the-art sensor, which can reject common-mode spoofing signal [249]. It is an *in-sensor* defense. As our defense is also an in-sensor, the differential Hall sensor's limitations are important to understand the novelty of our work.

A differential Hall sensor has two [249] Hall elements, D_1 and D_2, placed *close* to each other (Fig. 5.2 (Left)). Let us assume D_1 sees magnetic field B_1, and D_2 sees magnetic field B_2. Therefore, the transfer function of a differential Hall sensor is:

$$V_{out} = K \times (B_1 - B_2) + V^{null} \tag{5.2}$$

where K is a proportionality coefficient. Let us assume an attacker injects an external magnetic field, B_{atk}. As D_1 and D_2 are placed close to each other, they may see the same magnetic field, B_{atk}. As a result, after the injection of B_{atk}, Eqn. 5.2 is changed as follows:

$$\begin{aligned} V_{out} &= K \times \{(B_1 + B_{atk}) - (B_2 + B_{atk})\} + V^{null} \\ &= K \times (B_1 - B_2) + V^{null} \end{aligned} \tag{5.3}$$

The B_{atk} can only be nullified in Eqn. 5.3 if and only if D_1 and D_2 can see the same (i.e., common-mode) B_{atk}. However, practically speaking, there is always a small physical distance between D_1 and D_2 as a physical signal path is present between D_1 and D_2. Therefore, they may not see the same B_{atk}. As a result, B_{atk} may not be exactly nullified in Eqn. 5.3. The mismatch gets worse if the injected magnetic field is strong. At a strong magnetic field, the magnetic reluctance of the material present in the tiny distance between D_1 and D_2 gets increased. The increase of reluctance increases the magnetic field gradient between Hall elements D_1 and D_2.

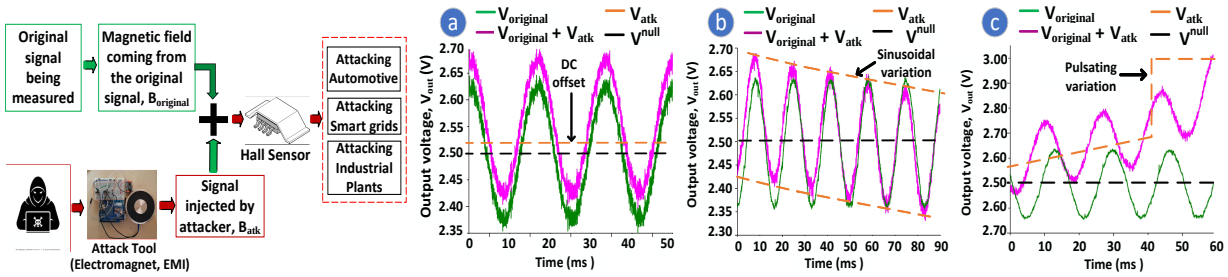


Figure 5.3: (Left) Noninvasive magnetic spoofing attack on Hall sensors. (Right) (a) The constant B_{atk} adds a DC offset. (b) The sinusoidal B_{atk} modulates the $V_{original}$ sinusoidally. (c) The square pulsating B_{atk} creates a pulsating variation in $V_{original}$.

To prove this claim, an experiment is carried out on a differential Hall sensor (Part# ACS724) by injecting a weak 4G external magnetic field, and a strong 8000G magnetic field using an electromagnet and a permanent magnet (part# H33 [253], respectively from 1 cm distance. The ACS724 typically has $V^{null} = 2.5$ V. Fig. 5.2 (Right) shows that 4 G shifts the V_{out} by 0.8 μV, which is negligible, whereas 8000 G adds a large DC offset, denoted by E^{dc}, with V^{null}. This shifts the V_{out} by 1.113 V upward causing a 44.52% change in V^{null}. This can corrupt the sensor data resulting in a DoS attack on the connected systems. It proves that industry-used Hall sensors are still vulnerable to strong magnetic spoofing.

5.4 Attack Model

The components of the attack model against which HALC works are explained below and also shown in Fig. 5.3 (Left).

a. Attacker's capability: The attacker can be a disgruntled employee or a guest, who may not get a long time to modify the target Hall sensor like a lunch-time attack [254]. The attacker just needs brief one-time access to *noninvasively* spoof Hall sensors from a *close* distance using external magnetic fields. The attacker is not allowed to physically alter any components of Hall sensors.

b. Attacker's strength: The attacker can inject any type of magnetic field. Here, we consider constant, sinusoidal, and square pulsating fields to cover the whole attack surface because all other patterns can be derived from these three basic fields (i.e., Fourier transformation [255]). Moreover, the attacker can use different intensities of magnetic fields inside or outside of the normal sensing range of the sensor. Let's denote the original magnetic field being measured by $B_{original}$ and magnetic fields injected by the attacker by B_{atk} (Fig. 5.3 (Left)). The term B_{atk} can be modeled as follows:

$$B_{atk} = \begin{cases} M; & \text{constant field,} \\ B_m \sin \omega t; & \text{sinusoidal field,} \\ sgn(B_m \sin \omega t); & \text{square pulsating field.} \end{cases} \tag{5.4}$$

where M is a constant, ω is the angular frequency and B_m is the magnitude of the injected magnetic field, and sgn is the signum function. Eqn. 5.1 can be written after an attack as:

$$\begin{aligned} V_{out} &= \{(K \times B_{original}) + V^{null}\} + (K \times B_{atk}) \\ &= V_{original} + V_{atk} \end{aligned} \tag{5.5}$$

Eqn. 5.5 shows that Hall sensor's output V_{out}, after an attack, has two components: an original component, $V_{original}$, coming from the $B_{original}$ and an attack component, V_{atk}, coming from the injected B_{atk}. An ideal defense should filter out the attack component V_{atk} originating from any type of attack magnetic field B_{atk}.

Demonstration: A demonstration of injecting a constant, sinusoidal, and square pulsating malicious magnetic fields B_{atk} into a Hall sensor is shown in Fig. 5.3 (Right). Before injecting the B_{atk}, the Hall sensor is giving a sinusoidal voltage $V_{original}$ (green line) at its output. A constant 300 G malicious B_{atk} adds a DC offset, shifting the $V_{original}$ by 0.02 V. A 2 Hz sinusoidal and pulsating 300 G malicious B_{atk} modulates the $V_{original}$ in a sinusoidal and pulsating fashion, respectively.

c. Attack tool and cost: The attacker can use a cheap electromagnet and an Arduino with pulse-width modulation to generate above distinct types of B_{atk}. For example, a simple electromagnet, such as Uxcell [256] can generate sufficient magnetic fields (i.e., $\sim$ 8000 G) for a strong magnetic spoofing attack, and it is cheap ($\sim$\$37) and easily available on eBay/Amazon.

d. Ineffective Shield: The Hall sensor may or may not be secured inside of a shield [257] depending on its application. In the presence of a shield, the injected B_{atk} is strong enough to penetrate a shield.

e. Target system: Hall sensors are used in many safety-critical applications, such as power grid monitoring, motor speed monitoring, proximity sensing in industrial plants, and braking in automotive. Therefore, the consequences of attacking Hall sensors can be catastrophic. For example, injecting fake magnetic fields into Hall sensors located in a micro-grid may cause a denial-of-service (DoS) attack on the power system [245]. Another notable attack happens on automotive systems where an attacker may cause a brake failure by spoofing Hall sensors located in anti-lock-braking systems (ABSs) of a vehicle [48]. The consequences of these attacks on Hall sensors are significant in terms of loss of human life and monetary resources. Moreover, an inaccurate Hall effect reading would cause immediate damage in the absence of a fail-safe, like an electric motor running at higher RPM than it is mechanically designed for. This may cause a complete shut-down of the compromised system. *Therefore, a defense (i.e., like HALC) is critical in the Hall sensor domain to prevent these catastrophic consequences.*

5.5 Hall Spoofing Container (HALC)

In this section, we provide details on the design process of HALC by answering the following three questions.

Q1. How can HALC contain all types, such as DC/constant, sinusoidal, and pulsating attack magnetic fields?

Q2. How can HALC contain a strong magnetic spoofing attack?

Q3. How can HALC remove the injected fake magnetic field B_{atk} from the original magnetic field $B_{original}$ even if the frequencies of B_{atk} and $B_{original}$ are same?

We start by mathematically modeling the attack at first.

Attack modeling: A Hall sensor can measure AC (i.e., time-dependent) and DC (i.e., constant) magnetic fields. Therefore, the AC and DC magnetic fields will have a proportional AC and DC voltage components in the sensor output V_{out} in Eqn. 5.5. Let us define the AC and DC voltage components coming from the AC and DC components of original input magnetic field $B_{original}$ by $V(t)$ and V^{dc}, respectively. Therefore, we can write the original component, $V_{original} = V(t) + V^{dc} + V^{null}$ in Eqn. 5.5.

Let us assume that the attacker can cause a DC error voltage E^c by injecting a constant magnetic field, a sinusoidal error voltage $E(t)$ by injecting sinusoidal magnetic fields, and a square error voltage $E^s(t)$ by injecting square magnetic fields. *Here, we consider an extreme scenario when the attacker injects all three patterns at the same time.* Therefore, the attack component in the output voltage of the compromised Hall sensor can be written as, V_{atk} $= E^c + E(t) + E^s(t)$. Moreover, Fourier analysis [258] of the square error voltage $E^s(t)$ shows that it has a DC portion E^s and a low and high frequency portion $\delta_l(t)$ and $\delta_h(t)$, respectively (i.e., $E^s(t) = E^s + \delta_l(t) + \delta_h(t)$). Therefore, the output, V_{out}, of the compromised Hall sensor during an attack can be written from Eqn. 5.5 as:

$$\begin{aligned} V_{out} &= V_{original} + V_{atk} \\ &= (V(t) + V^{dc} + V^{null}) + (E^c + E(t) + E^s(t)) \\ &= V_{original} + (E^c + E(t) + E^s + \delta_l(t) + \delta_h(t)) \end{aligned} \tag{5.6}$$

From Eqn. 5.6, it is apparent that V_{out} under attack has two components, namely *AC (i.e., time-dependent)* component, $V(t) + E(t) + \delta_h(t) + \delta_l(t)$, and *DC (i.e. constant)* component, $V^{dc} + V^{null} + E^c + E^s$. Please note that inside of the AC component, the AC attack component is $E(t) + \delta_h(t) + \delta_l(t)$, and inside of the DC component, the DC attack

component is $E^c + E^s$.

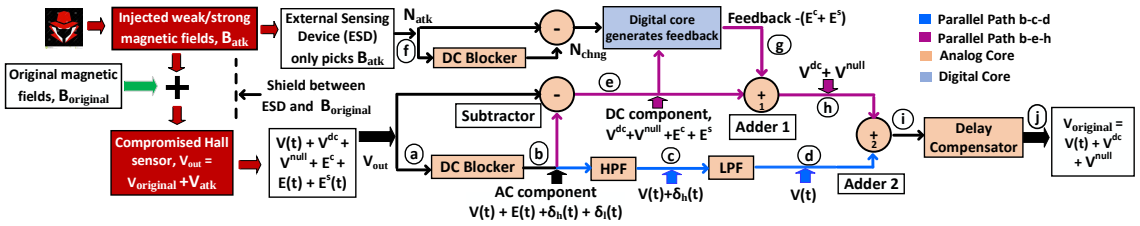


Figure 5.4: Basic blocks of the Hall Spoofing Container (HALC).

Parallelism: In an extreme scenario, if the attacker injects AC and DC attack components simultaneously, a proper defense should contain these attack components in real-time without hampering the existing speed of the connected systems. However, there is no defense exists that can contain the AC and DC attack components inside of existing sensors. Moreover, a naive solution, which may *sequentially* handle the AC and DC attack components, may make the defense slow, hampering the real-time requirement of the defense. To solve this problem, our proposed defense HALC introduces two different cores - analog and digital cores, to *parallelly* handle the AC and DC attack components in the following ways:

(i) Analog core: The analog core removes the high and low frequency AC attack components, $E(t) + \delta_h(t) + \delta_l(t)$, from the V_{out} using different filtering techniques.

(ii) Digital core: The digital core removes the DC attack components, $E^c + E^s$, from the V_{out} using a novel algorithm.

Fig. 5.4 shows all blocks of the two cores, and Fig. 5.5 shows the details of each block of the two cores. Fig. 5.4 shows two paths- paths b-c-d and b-e-h - that host the two cores. The parallel handling of the AC and DC attack components by two separate cores in two different paths makes HALC faster than the sequential handling of each attack component. We mathematically discuss each core in the following sections with implementation details.

5.5.1 Analog Core

At first, the analog core needs to separate the AC and DC components from V_{out} to parallelly process them in two different paths - paths b-c-d and b-e-h. To separate the AC and DC components from V_{out}, the analog core uses two blocks - DC blocker and subtractor, which are discussed below.

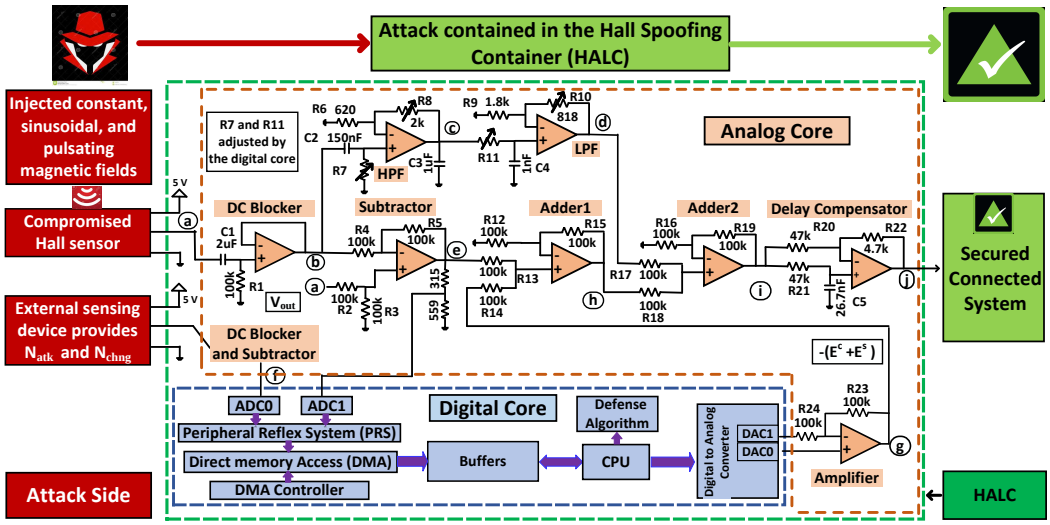


Figure 5.5: Implementation details of the analog and digital cores of the proposed Hall Spoofing Container (HALC).

DC Blocker

The DC blocker blocks the DC component (i.e., $V^{dc} + V^{null} + E^c + E^s$) of V_{out} and outputs only the AC component (i.e., $V(t) + E(t) + \delta_h(t) + \delta_l(t)$) at node ⓑ to path b-c-d (Fig. 5.4). It uses a first-order high pass filter (Fig. 5.5) with R1 = 100 kΩ and C1 = 2 μF having a cut-off frequency, $f_c = 1/(2\pi R1C1) = 0.8$ Hz (i.e., it only blocks the DC signal).

Subtractor

The subtractor subtracts the AC component of V_{out} from V_{out} and outputs only the DC component of V_{out} (i.e., $V^{dc} + V^{null} + E^c + E^s$) at node ⓔ to path b-e-h (Fig. 5.4). The

subtractor is implemented using a differential amplifier with a transfer function, V_+ - V_-, when R2=R3=R4=R5 (Fig. 5.5). Here, V_+ is the +ve input and V_- is -ve input of the amplifier.

Next, after separation, the AC component of V_{out} are processed by the high-pass and low-pass filters, and the DC component of V_{out} is processed by the digital core (see Section 5.5.2).

High-Pass Filter (HPF) & Low-Pass Filter (LPF)

A first-order active HPF and a LPF are used to filter out the low-frequency (i.e., $E(t)+\delta_l(t)$) and high-frequency (i.e., $\delta_h(t)$) *attack components* from V_{out}, respectively, by keeping the original AC component $V(t)$ intact. The digital core, which works as an adjunct to the analog core (Section 5.5.2), can control the cut-off frequencies (i.e., f_c) of the HPF and LPF to filter out *only* low/high frequency *attack components*. In our implementation, the rheostats R7 and R11 (Fig. 5.5) are used to vary the cut-off frequencies of the HPF and LPF within 0 - 150 kHz (see Sections 5.5.3 and 5.8.1 for more details). Please note that the gain of the DC blocker, LPF, and HPF is $\sim$1 (i.e., unity) and phase-shift is linear (i.e., constant) for all frequencies over f_c. *Therefore, path b-c-d does not add any non-linearity and unstability to the original AC component $V(t)$ in our design.*

Delay Compensator

The signal, V_{out} travels from node ⓐ to node ⓘ through different blocks. These blocks have capacitors and resistors with different values that introduce different *phase delays*. As a result, the signal at node ⓘ is a phase-delayed version of the signal from node ⓐ. For example, a 2.34 ms *leading* phase delay is present between node ⓐ and node ⓘ of our HALC. This could cause a 2.34 ms delay while taking a time-critical decision by the connected system. To compensate for the phase delay, a delay compensator is placed after node ⓘ. The delay compensator is an all-pass filter with a voltage gain, $A_v = 1$ at all frequencies

and can create a specific phase shift. A *lagging* phase shift of 50.63° is implemented in our design that is equivalent to 2.34 ms of *lagging* delay. *As a result, the 2.34 ms of leading delay at node ① is compensated to* ***zero*** *(See Fig. 5.8).* This ensures that HALC does not create any timing predictability issue to connected systems and preserves the hard real-time requirement of the overall system.

In summary, the analog core separates the AC component from the V_{out} and then contain the AC attack component (i.e., $E(t) + \delta_h(t) + \delta_l(t)$) by keeping the original AC component $V(t)$ intact.

5.5.2 Digital Core

The digital core uses a novel algorithm to generate a feedback signal -($E^c + E^s$), equal but opposite polarity to the injected DC attack component $E^c + E^s$. The digital core uses the generated feedback signal to nullify the injected DC attack component while keeping the original DC signal V^{dc} intact (see Eqn. 5.6). Moreover, it controls the cut-off frequencies of HPF/LPF of the analog core to remove AC attack components while keeping the original AC component $V(t)$ intact. To accomplish these two tasks, the digital core takes input from an external sensing device that is explained below.

External sensing device (ESD): As a Hall sensor under attack is a naive device, it cannot alone differentiate the original input magnetic fields ($B_{original}$) from the attacker's provided magnetic fields (B_{atk}). Therefore, the digital core uses an *external sensing device (ESD)*, which helps the compromised Hall sensor by *only* sensing the *presence* of the injected attack fields, B_{atk} (see Fig. 5.4 and 5.5). The ESD could be an external coil or another Hall sensor, which should be placed *side by side* with the compromised Hall sensor. *As the ESD can only sense the injected attack fields B_{atk}, the attacker cannot confuse the defense using multiple magnetic sources.*

Shield between ESD and $B_{original}$: *The next question is how to ensure that the ESD only picks the injected attack fields (B_{atk}), not the original fields ($B_{original}$).* Let's consider two scenarios. Firstly, suppose the original field is internal, such as for voltage/current Hall sensors (sensors 1-4, and 9-10 in Table 5.1). In that case the ESD only picks the injected external attack fields, not the original fields. Secondly, if the original fields and injected attack fields both are external to a sensor (sensors 5-8 in Table 5.1), there is a chance that the ESD can pick both the original and injected external fields. To prevent this happens, a shield is used between the ESD and the source of original magnetic fields, so that the ESD cannot pick up the original fields but only can pick up the injected external fields. As the direction of the original fields is known to a designer, he can safely place a simple shield to prevent the original fields from going into the ESD (see Fig. 5.4 for the shield between ESD and $B_{original}$).

A question may arise if the shield can be bypassed by the attacker. Please note that the use of the shield is not to prevent attackers from influencing the target Hall sensor. However, the use of the shield is to prevent the original magnetic fields ($B_{original}$) from going into the ESD so that the ESD can only pick the injected attack fields (B_{atk}), not the original magnetic fields. Therefore, bypassing the shield with the B_{atk} by an attacker will not impact the defense because the ESD still can pick up the injected attack fields B_{atk}.

How the ESD is different from the recent works: Although the ESD is placed close to the compromised Hall sensor, there should always be a physical distance between the ESD and the compromised Hall sensor. Because of this physical distance, the ESD is unable to measure the exact *amplitude* of the external attack fields injected into the compromised Hall sensor. This is why we can not use the signal from the ESD to simply subtract the injected attack signals from the original signals to recover the original signal. Therefore, HALC uses the ESD differently compared to its use in the adaptive filtering technique found in recent work [42].

The ESD only provides the following two pieces of information to the digital core (see Fig. 5.4): (i) the attack notification signal, N_{atk}, which is only activated when the ESD senses the external attack field B_{atk}, and (ii) the notification signal, N_{chng}, when the ESD senses that the injected DC attack signal / component, $E^c + E^s$ changes. The N_{atk} and N_{chng} both do not consider any *absolute amplitude* of the attack signal, instead just only consider the *change/difference* in attack signal. Next, we discuss how the N_{atk} and N_{chng} are used by the digital core to generate the feedback signal -(E^c + E^s) to nullify the injected DC attack signal (E^c + E^s).

Removing injected DC attack signal $E^c + E^s$: The digital core runs a novel algorithm 4 in a central processing unit (CPU) to remove the injected DC attack signal $E^c + E^s$. Let us summarize the algorithm first before introducing its technical implementation. When the ESD gives an attack notification signal (i.e., N_{atk}) that an attack happens at time t, the algorithm subtracts the DC component (see Eqn. 5.6) of *original signal* at time t from the previous DC component of *original signal* at time $t-1$ (i.e., data before the attack). The difference between the DC components during the attack and before the attack gives the amount of injected DC attack signal $E^c + E^s$ after the attack. The algorithm tracks this difference all the time and generates -($E^c + E^s$) to nullify the injected DC attack signal $E^c + E^s$. If the injected DC attack signal changes during an attack, the algorithm 4 can also track it from the previously calculated difference. It is noteworthy that algorithm 4 also tracks when the DC component of the original signal changes without any attack. This helps to correctly retrieve the original signal with and without attack. In summary, the continuous tracking of the DC component of the original signal before, after, and during the attack gives information of the injected DC attack signal, and this information is used to retrieve the DC component of original signal. *This idea and its implementation are absent in recent works [42, 47, 51, 248–250] that exist in the literature.* Next, we discuss the implementation (see Fig. 5.4 and 5.5) of algorithm 4 in detail.

ADC0 and ADC1

Two analog-to-digital converters - ADC0 and ADC1 provide data to the CPU (Fig. 5.5). ADC0 is connected with the ESD and provides the two information coming from the ESD, namely, notification signals N_{atk} and N_{chng} to the defense algorithm 1 running in CPU. Parallelly, ADC1 also provides the DC component (i.e., $V^{dc} + V^{null} + E^c + E^s$) of the V_{out} to algorithm 1 from node ⓔ. To reduce the power consumption, both ADCs use a low sampling frequency (35 kHz) at normal operating conditions (i.e., no attack), but start using a high sampling frequency (900 kHz) when an attack happens.

Peripheral Reflex System (PRS) and Direct Memory Access (DMA)

To satisfy real-time requirement and reduce energy consumption, the workload of the CPU is shared with a peripheral reflex system (PRS) and direct memory access (DMA). The PRS and DMA handle the workload related to data movement from ADCs to CPU, whereas the CPU handles the workload related to running algorithm 4 and providing feedback signals to the analog core.

Central Processing Unit (CPU)

The CPU runs the defense algorithm 4 and provides a feedback signal to nullify the DC attack signal (i.e., $E^c + E^s$) that is explained below.

Line 1-10: The CPU always checks the data coming from the ESD for the attack notification signal N_{atk} using the ADC0. Let's assume an attack happens at time t. Before any attack (at $t-1$ time), there is no presence of external spoofing magnetic fields. Therefore, the output of the ESD is zero, which indicates no attack happens (i.e., $N_{atk} = NO$). Moreover, when no attack happens, the data from ADC1 at $t-1$ is simply equal to $V^{dc}(t-1)+V^{null}(t-1)$ because no DC attack signals are present (i.e., $E^c + E^s = 0$) at node ⓔ. As no DC attack

Algorithm 4: Proposed Defense Algorithm.

Input: Data from ADC0 and ADC1
Output: Feedback signal at node ⓖ to nullify the $E^c(t) + E^s(t)$

$t \leftarrow$ *attack happens*
Setup ADC0, and ADC1 $\leftarrow$ (12 bits, sampling freq. = 35kHz)
$V^{dc}(t-1) + V^{null}(t-1) \leftarrow$ *ADC1(t-1)*
for $t \leftarrow 1$ ***to*** ∞ **do**
  $N_{atk} \leftarrow$ *ADC0(t-1)*
  if $N_{atk} = NO$ **then**
    $V^{dc}(t-1) + V^{null}(t-1) \leftarrow$ *ADC1(t-1)*
    func_Notifies_system (no_attack_happens)
    ADC0, ADC1 $\leftarrow$ sampling frequency 35 kHz
    Output = *no feedback signal (i.e., 0V at node ⓖ)*
  end
  else
    func_Notifies_system (attack_happens)
    ADC0, ADC1 $\leftarrow$ sampling frequency 900 kHz
    $V^{dc}(t) + V^{null}(t) + E^c(t) + E^s(t) \leftarrow$ *ADC1(t)*
    $V^{dc}(t) + V^{null}(t) = V^{dc}(t-1) + V^{null}(t-1)$
    $E^c(t) + E^s(t) \leftarrow$ *ADC1(t)* $-V^{dc}(t) - V^{null}(t)$
    if $E^c(t) + E^s(t) > 0$ **then**
      Output = *feedback signal -($E^c(t) + E^s(t)$) at node ⓖ to nullify the $E^c(t) + E^s(t)$*
    else
      Output = *feedback signal +($E^c(t) + E^s(t)$) at node ⓖ to nullify the -($E^c(t) + E^s(t)$)*
    if *The data from ADC1 changes after t at t+n time* **then**
      $N_{chng} \leftarrow$ *ADC0*(t+n)
      if $N_{chng} = YES$ **then**
        $V^{dc}(t+n) + V^{null}(t+n) = V^{dc}(t) + V^{null}(t)$
        $E^c(t+n) + E^s(t+n) \leftarrow$ *ADC1(t+n)* $-V^{dc}(t+n) - V^{null}(t+n)$
        if $E^c(t+n) + E^s(t+n) > 0$ **then**
          Output = *feedback signal -($E^c(t+n) + E^s(t+n)$) at node ⓖ to nullify the $E^c(t+n) + E^s(t+n)$*
        else
          Output = *feedback signal +($E^c(t+n) + E^s(t+n)$) at node ⓖ to nullify the -($E^c(t+n) + E^s(t+n)$)*
      end
      else
        $E^c(t+n) + E^s(t+n) \leftarrow E^c(t) + E^s(t)$
        $V^{dc}(t) + V^{null}(t) =$ *ADC1(t+n)*$+E^c(t+n) + E^s(t+n)$
        if $E^c(t+n) + E^s(t+n) > 0$ **then**
          Output = *feedback signal -($E^c(t+n) + E^s(t+n)$) at node ⓖ to nullify the $E^c(t+n) + E^s(t+n)$*
        else
          Output = *feedback signal +($E^c(t+n) + E^s(t+n)$) at node ⓖ to nullify the -($E^c(t+n) + E^s(t+n)$)*
      end
      $V^{dc}(t-1) + V^{null}(t-1) = V^{dc}(t) + V^{null}(t)$
    end
  end
end

signals are present, the CPU does not need to nullify the DC attack signals $E^c + E^s$. That is why the CPU provides a NULL signal to digital-to-analog converters (DACs), and the DACs provide no feedback (0 V) at node ⓖ.

Line 11-16: However, when the attacker injects magnetic fields at time t, the ESD senses this injection that generates an attack notification signal, $N_{atk} = YES$. The ADC0 and ADC1 increase the sampling frequency from 35 kHz to 900 kHz to capture tiny changes of injected signals. During attack at time t, the data from ADC1 is equal to $V^{dc}(t) + V^{null}(t) + E^c(t) + E^s(t)$. As the DC component of the $V_{original}$ does not change, $V^{dc}(t) + V^{null}(t)$ at time t is equal to the previous value of $V^{dc}(t-1) + V^{null}(t-1)$ at time $t-1$. As $V^{dc}(t-1) + V^{null}(t-1)$ is known, the injected DC attack signal $E^c(t) + E^s(t)$ can be calculated as shown in line 16.

Line 17-20: After calculating the value of the injected DC attack signal $E^c(t) + E^s(t)$, the DACs (Fig. 5.5) generate a analog signal which is equal to the $E^c(t) + E^s(t)$. If the injected DC attack signal $E^c(t) + E^s(t)$ is positive, the amplifier in Fig. 5.5 is configured as inverting amplifier with a gain of -1 and outputs a feedback signal -$(E^c(t) + E^s(t))$ at node ⓖ with the help of DACs. If $E^c(t) + E^s(t)$ is non-positive, the amplifier is configured as non-inverting amplifier with a gain of +1 and outputs a feedback signal $+(E^c(t) + E^s(t))$ at node ⓖ with the help of DACs. The adder1 adds signals at node ⓖ with signals at node ⓔ and nullifies the injected DC attack signal $E^c(t) + E^s(t)$ from the V_{out} (see Fig. 5.4).

Line 21-29: After an attack happens at time t, the DC component of V_{out} sampled by ADC1 may change anytime after time t. Let us assume the data from ADC1 changes at time $t + n$ where $n\epsilon\{1, 2, 3, .., \infty\}$. The change can happen under *two scenarios*: either the attacker changes the DC attack signal $(E^c + E^s)$, or the DC component $(V^{dc} + V^{null})$ of the $V_{original}$ may change naturally. Under the *first scenario*, when the attacker changes the DC attack signal at time $t + n$, the ESD outputs a notification signal $N_{chng} = YES$, which is extracted from the ADC0 at $t+n$. As the DC component of the $V_{original}$ do not change under the first scenario, the previously saved DC component $(V^{dc}(t) + V^{null}(t))$ of the $V_{original}$ at

time t must be equal to the most recent DC component $(V^{dc}(t+n) + V^{null}(t+n))$ of the $V_{original}$ at time $t+n$. Therefore, the injected DC attack signal $(E^c(t+n)+E^s(t+n))$ can be calculated using the data from ADC1 at time $t+n$ shown in line 25. The $E^c(t+n)+E^s(t+n)$ can be similarly used to generate feedback signals explained in line 17-20.

Line 30-37: Under the *second scenario*, when the DC component $(V^{dc} + V^{null})$ of the $V_{original}$ changes naturally at time $t+n$, the ESD outputs a notification signal $N_{chng} = NO$, which is extracted from the ADC0 at $t+n$. As the injected DC attack signal does not change under the second scenario, the previously saved DC attack signal $(E^c(t) + E^s(t))$ at time t must be equal to the most recent DC attack signal $(E^c(t+n)+E^s(t+n))$ at time $t+n$. The calculated $E^c(t+n) + E^s(t+n)$ is similarly utilized to generate feedback signals, which is explained in line 17-20. The DC component $(V^{dc}(t) + V^{null}(t))$ of the $V_{original}$ at time t are updated in line 32 that is used in line 37 to update $V^{dc}(t-1) + V^{null}(t-1)$. The updated $V^{dc}(t-1) + V^{null}(t-1)$ will be used in the next iteration at line 15.

In lines 21-29, two scenarios are considered, change due to attack and change naturally. A question might arise what will happen if a persistent attack coincides with a natural change. The answer lies in the execution time of lines 21-23. Let us denote the time required to execute lines 21-23 as p. Therefore, if the time difference between change due to attack and change naturally is greater than p, HALC can successfully detect both changes. For example, the time required to execute lines 21-23 is $\sim$3 μs for our prototype. The time difference can be reduced to a lower value using a faster CPU resulting in a more robust defense against the error.

5.5.3 Controlling HPF & LPF of the analog core

The digital core decides the appropriate cut-off frequencies of the HPF and LPF after sensing the frequency of the injected attack magnetic fields (B_{atk}) using the *ESD*. If the injected attack magnetic field has a single frequency (i.e., single tone), the digital core configures

the HPF and LPF in such a way that the HPF and LPF jointly act as a *band-stop* filter, which stops the injected single tone attack signals $E(t) + \delta_h(t) + \delta_l(t)$. If the injected attack magnetic field has multiple frequencies (i.e., multiple tones), the digital core configures the HPF and LPF in such a way that the HPF and LPF jointly act as a *band-pass* filter, which only passes the original input signal ($V_{original}$), removing the injected attack signals behind. In this way, with the help of the digital core, the HPF and LPF jointly eliminate the AC attack components ($E(t) + \delta_h(t) + \delta_l(t)$) of the injected V_{atk} from the V_{out} by keeping the $V_{original}$ intact.

5.5.4 Removing equal frequency attack signals

A concern may arise what will happen if the amplitude and frequency of the injected V_{atk} are same as the $V_{original}$. To handle this concern, a Hall sensor should be used in the differential configuration [249]. Referring to Section 5.3.3, let us assume two Hall elements D_1 and D_2 are placed close to each other in a differential configuration. During an attack, let us assume the two Hall elements D_1 and D_2 sense $B_{original1}$, B_{atk1} and $B_{original2}$, B_{atk2}, respectively, while measuring an original signal $B_{original}$. As $V_{original} \propto B_{original}$ and $V_{atk} \propto B_{atk}$, we can write the transfer function of the differential sensor during an attack from Eqn. 5.3 as,

$$\begin{aligned} V_{out} &= k[B_{original1} + B_{atk1} - B_{original2} - B_{atk2}] + V^{null} \\ &= V_{original1} + V_{atk1} - V_{original2} - V_{atk2} + V^{null} \\ &= (V_{original1} - V_{original2}) + V^{null} + (V_{atk1} - V_{atk2}) \\ &= 2V_{original1} + V^{null} + E^c \end{aligned} \tag{5.7}$$

where $V_{original1} \approx$ - $V_{original2}$ (i.e., differential input). The B_{atk1} and B_{atk2} both have the same frequency because they are coming from the same attack signal. But they have different amplitudes because there is a small gap present between the two Hall elements D_1 and

D_2 (refer to Section 5.3.3 for explanation). Therefore, the V_{atk1} and V_{atk2} have the *same* frequency, but they have *different* amplitudes (i.e., $B_{atk1} \neq B_{atk2}$). Therefore, the ($V_{atk1} - V_{atk2}$) results in a *constant* error E^c, which acts as a DC attack signal. Therefore, the defense algorithm 4 can remove the DC attack signal E^c from V_{out} in the same way that is already described in Section 43.

5.5.5 Novelty of HALC

The novelty of HALC compared to recent work [42, 47, 51, 248–250] is discussed here by answering the three questions from Section 5.5.

Q1. How can HALC contain all types, such as DC/constant, sinusoidal, and pulsating attack magnetic fields?

Answer: The Eqn. 5.6 models the constant, sinusoidal, and pulsating attack++ magnetic fields using the AC attack component $E(t) + \delta_h(t) + \delta_l(t)$ and DC attack component $E^c + E^s$. The AC attack component is contained by HPF and LPF of the analog core with the help of the digital core (see Sections 5.5.1 and 5.5.3). The DC attack component is contained by the digital core with a feedback signal $E^c + E^s$ (see Section 5.5.2). In this way, HALC can contain constant, sinusoidal, and pulsating injected attack magnetic fields.

Q2. How can HALC contain strong magnetic spoofing attack?

Answer: Fig. 5.1 (Right) indicates that the attacker can spoof the output of the Hall sensor within its linear region close to its saturation voltage using a strong magnetic field. Please note that the digital core of HALC can generate the DC feedback signal $E^c + E^s$ within the entire linear region of the Hall sensor close to the supply voltage (i.e., greater than the saturation voltage) to nullify the injected DC attack component. Moreover, the analog core can filter out the AC attack component within the entire operating region of the HPF and LPF. As the operating region of the HPF and LPF is greater than the linear region

of the Hall sensor (i.e., the supply voltage of HPF/LPF is greater than the Hall sensor), the analog core can also contain the AC attack components within the entire linear region of the hall sensor close to the saturation voltage. In this way, HALC can contain strong magnetic spoofing attack.

Q3. How can HALC remove the injected fake magnetic field B_{atk} from the original magnetic field $B_{original}$ even if the frequencies of B_{atk} and $B_{original}$ are same?

Answer: The answer is already given in Section 5.5.4.

Another important point to note is that HALC only nullifies injected attack component V_{atk} in the sensor output V_{out} by keeping the original component $V_{original}$ intact. It is possible that original signals may contain anomalous data. HALC does not alter any anomalous data present in original signals as HALC only works on the injected attack component. The ESD is used to differentiate between original and attack components (refer to Section 5.5.2).

5.6 Performance Analysis

5.6.1 A prototype of the proposed HALC

A prototype of the proposed HALC is implemented in a lab setup as a proof-of-concept and is shown in Fig. 5.6 (Left). The DC blocker, subtractor, adder1, adder2, delay compensator, HPF, and LPF of the analog core are implemented using a low-power op-amp (part # TL084CN from Texas Ins.). The TL084CN has a JFET input stage that provides high slew rates, low input bias, and low offset currents. The values of discrete resistors and capacitors of the analog core are shown in Fig. 5.5. The digital core is implemented in an EFM-32 Giant Gecko board from Silicon Labs [259] that has Cortex M-3 based 32-bit CPU with PRS, ADCs, DACs, and DMA. It has an ultra low-power CPU with a 48 MHz clock.

5.6.2 Testbed

Different tools used in the testbed are shown in Fig. 5.6 (Right). We use a Hall sensor (part #ACS718) as the external sensing device (Fig. 5.6 (Left)). We test 10 different Hall sensors (see Table 5.1) of all types, such as unipolar, bipolar, open-loop, and closed-loop Hall sensors, from four different manufacturers in the testbed. As these sensors require different types of inputs (S_{in}), we use different sources to supply input signals to these Hall sensors. We use a variable AC/DC power supply to supply current/voltage as original input signals to the Hall sensors with serial no. 1-4, and 9-10 of Table 5.1. We use a permanent magnet [253] to supply magnetic fields as input signals to the Hall sensors with serial no. 5-8 of Table 5.1. We use an electromagnet [256] with an Arduino Uno as an attack tool to generate constant, sinusoidal, and pulsating attack fields using a pulse-width-modulation (PWM) technique. In addition, we use a function generator connected with a monopole antenna [260], which is also used as an attack tool, to radiate high and low frequency EMI signals. We can change the power and frequency of the electromagnet and EMI to generate weak and strong magnetic fields with different frequencies within a range of 0 - 9000 G in our testbed. The demonstration of the testbed is shown in the following link: https://sites.google.com/view/hallspoofingcontainer/home

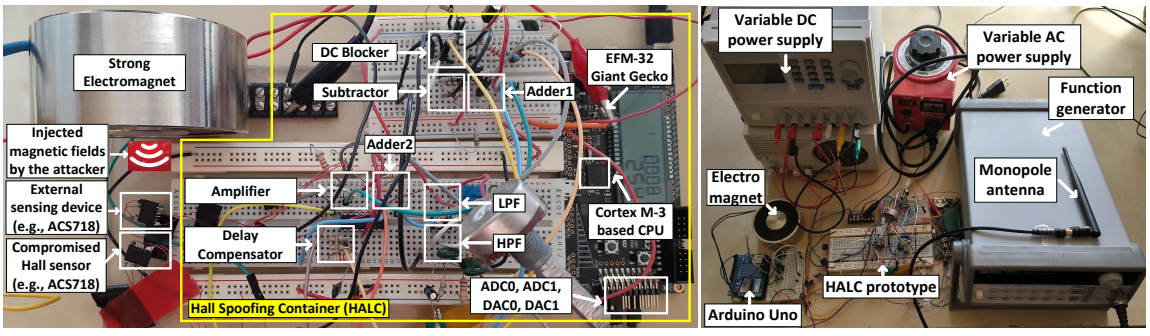


Figure 5.6: (Left) A prototype of our proposed HALC implemented in the lab. (Right) Different types of tools used in the testbed.

5.6.3 Justification of HALC

Now, we justify how HALC can contain the injected magnetic fields by analyzing signals at all of its nodes. We arbitrarily choose ACS718 from Table 5.1 as the target Hall sensor and connect it to HALC to analyze signals at all of its nodes. A 3 A peak-to-peak AC current of 60 Hz and a 0.5 A DC current are given as input signals (S_{in}) to the target sensor. Before any attack, the Hall sensor outputs the $V_{original}$ at node ⓐ (Fig. 5.7 (i)). A $V(t) = 300$ mV peak-to-peak, $V^{null} = 2.5$ V, and $V^{dc} = 50$ mV are present in the $V_{original}$ before any attack. An electromagnet with a magnetic field density of 5600 G is used to inject constant (E^c), sinusoidal ($E(t)$) and pulsating ($E^s(t)$) external magnetic fields from 1 cm distance. We use 2 Hz as the frequency of injected $E(t)$ and $E^s(t)$ as an example. Fig 5.7 (ii) shows that the output of the Hall sensor at node ⓐ is shifted close to its saturation voltage (4.7 V) after the attack. The injection of the AC attack signal, $E(t) + E^s(t)$ distorts the $V_{original}$, and the injection of the DC attack signal, $E^c + E^s$ shifts the $V^{null} + V^{dc}$ of the $V_{original}$ from 2.55 V to 4.56 V. The DC blocker blocks only the DC components, $V^{dc} + V^{null} + E^c + E^s$ and outputs only the AC components, $V(t) + E(t) + \delta_h(t) + \delta_l(t)$ at node ⓑ (Fig. 5.7 (iii)).

The signals from node ⓑ propagate forward using two paths - path b-c-d and path b-e-h. Let us discuss the path b-c-d first. The HPF filters out the injected low-frequency error, $E(t) + \delta_l(t)$ and outputs $V(t) + \delta_h(t)$ at node ⓒ (Fig. 5.7 (iv)). The LPF filters out the injected high-frequency errors ($\delta_h(t)$) and outputs the AC component of the original input signal $V(t)$ at node ⓓ (Fig. 5.7 (v)).

Now, we discuss the path b-e-h. The subtractor outputs the overall DC components, $V^{dc} + V^{null} + E^c + E^s$, at node ⓔ (Fig. 5.7 (vi)). The value of $V^{dc} + V^{null} + E^c + E^s$ is 4.56 V. As the $V^{null} + V^{dc}$ of the original input signal is shifted from 2.55 V to 4.56 V, a DC error ($E^c + E^s$) of 2.01 V is injected by the attacker. Therefore, our proposed defense algorithm running in the digital core gives a feedback signal ($-E^c - E^s$) of -2.01 V at node ⓖ (Fig.

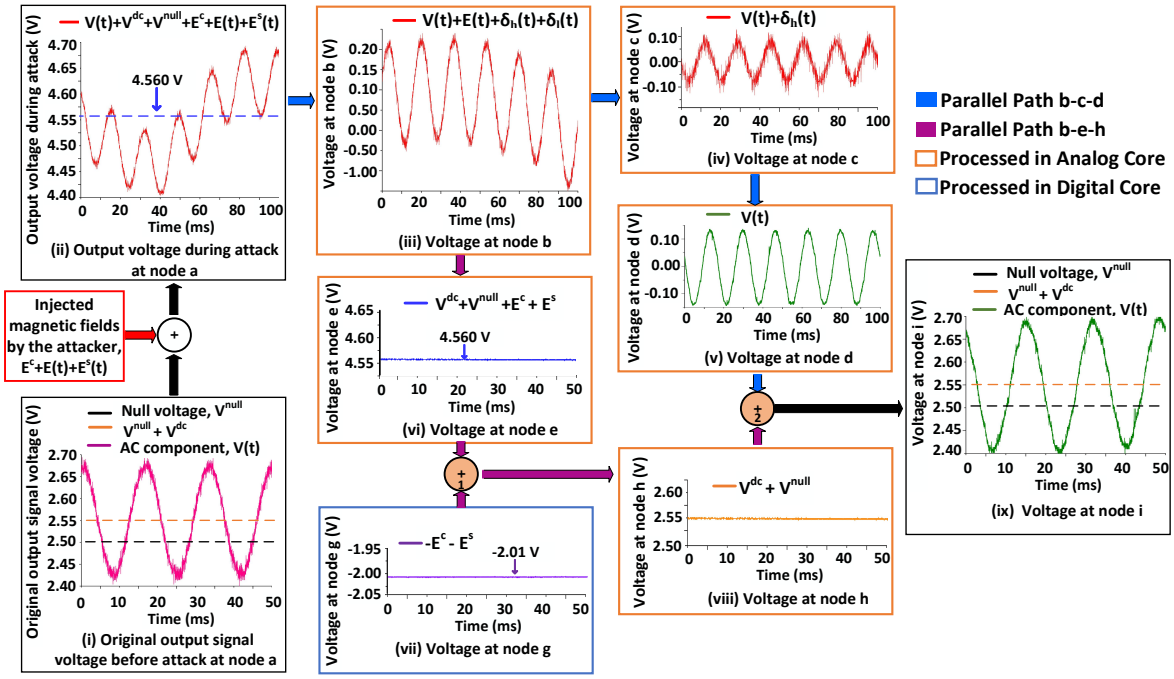


Figure 5.7: Signal analysis at all nodes of HALC. The signal at node ⓘ is a phase-delayed form of the input signal at node ⓐ.

5.7 (vii)). The adder1 adds signals from node ⓔ and node ⓖ, and outputs only $V^{dc} + V^{null}$ with a value of 2.55 V at node ⓗ (see Fig. 5.7 (viii)).

The adder2 adds signals from nodes ⓓ and ⓗ and outputs a delayed version of the $V_{original}$ at node ⓘ (Fig. 5.7 (ix)). A 2.34 ms of leading delay is present between signals at node ⓐ and node ⓘ (Fig. 5.8 (i)). A delay compensator compensates for the delay and outputs the $V_{original}$ at node ⓙ (Fig. 5.8 (ii)).

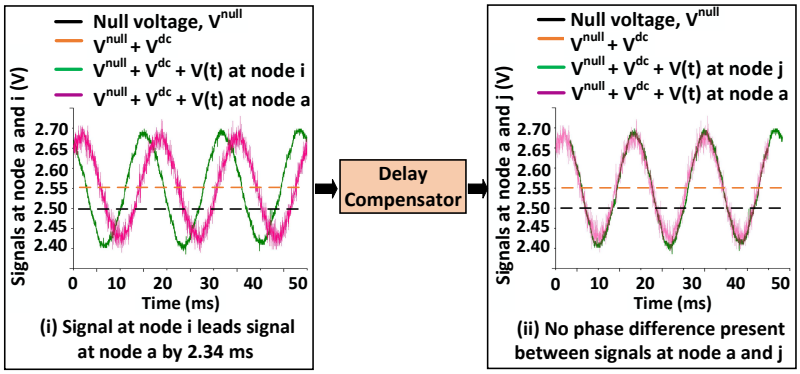


Figure 5.8: The delay between nodes ⓐ and ⓙ is compensated.

Performance metric: If we can prove that the output of the target Hall sensor before an

attack is *same* to the output of the target Hall sensor after an attack with HALC in a *point-by-point* fashion, we can claim that HALC is effective to prevent the spoofing attack. To quantify the similarity between signals before and after an attack, we calculate *correlation coefficient (C)* [261] between signals of node ⓐ (i.e., before an attack) and node ⓙ (i.e., after an attack). If the correlation coefficient is close to unity, it indicates that the output of the target Hall sensor before and after an attack is same in a *point-by-point* fashion.

The value of C is 0.93 for this case, that is very close to unity (i.e., due to the presence of noise, C is slightly less than unity). This indicates that the signal at node ⓙ (i.e., after an attack) is *same* as the original signal at node ⓐ (i.e., before an attack) in a point-by-point fashion. This proves that HALC can separate V_{atk} from $V_{original}$ and successfully contain the spoofing attack inside of it.

5.6.4 Varying the amplitude of the input signals

We vary the amplitude of the input signals (S_{in}) to 10 different Hall sensors within their entire input range (Table 5.1(a)). We keep the frequency of the S_{in} fixed at 15/60 Hz (Table 5.1(b)). We calculate C for every different amplitudes and do an average of C for every sensor. The avg. of C is *greater than 0.93* when HALC is used compared to 0.2 when HALC is not used (Table 5.1(c)). This indicates that HALC works within the entire input range of every Hall sensor.

5.6.5 Varying the frequency of the input signals

We vary the frequency of input signals (S_{in}) to 10 different Hall sensors within their entire input frequency range (Table 5.1(d)). We keep the amplitude of the S_{in} fixed at 1 A/100 G/110 V (Table 5.1(e)). We calculate C for every different frequency and do an average of C for every sensor. The avg. of C is *greater than 0.93* for every sensor when HALC is used compared to 0.2 when HALC is not used (see Table 5.1(f)). This indicates that HALC

Table 5.1: Testing Hall sensors with HALC for different amplitudes and frequencies of input signals, and with a MuMetal shield.

Sl.	Manufacturers	Part #	Polarity / Loop	Different amplitudes (a)	Freq. (b)	Avg. C (c)	Diff. freq. (d)	Amp. (e)	Avg. C (f)	C (0.4 in thick) (g)	C (0.9 in thick) (h)	Avg. C (i)
1	Allegro	ACS718 [262]	Bipolar / Open	1A, 5A, 10A, 15A, 20A	60 Hz	0.93	0 - 40 kHz	1 A	0.93	0.43	0.55	0.95
2	Allegro	ACS710 [263]	Bipolar / Open	2A, 4A, 6A, 8A, 10A	60 Hz	0.93	0 - 120 kHz	1 A	0.93	0.39	0.47	0.94
3	Allegro	ACS715 [264]	Unipolar / Open	1A, 5A, 10A, 15A, 20A	60 Hz	0.94	0 - 80 kHz	1 A	0.93	0.43	0.51	0.93
4	Allegro	ACS724 [265]	Unipolar / Open	2A, 4A, 6A, 8A, 10A	60 Hz	0.97	0 - 120 kHz	1 A	0.97	0.49	0.56	0.95
5	Honeyw	SS49 / SS19 [266]	Bipolar / Open	100G, 200G, 300G, 400G, 500G	15 Hz	0.94	0 - 30 Hz	100 G	0.93	0.36	0.46	0.96
6	Honeyw	SS39ET [267]	Bipolar / Open	100G, 200G, 300G, 400G, 500G	15 Hz	0.94	0 - 40 Hz	100 G	0.94	0.39	0.49	0.95
7	Honeyw	SS494B [268]	Bipolar / Open	100G, 200G, 300G, 400G, 500G	15 Hz	0.94	0 - 30 Hz	100 G	0.94	0.48	0.56	0.94
8	Texas Ins.	DRV5053 [269]	Bipolar / Open	100G, 200G, 300G, 400G, 500G	15 Hz	0.94	0 - 20 Hz	100 G	0.94	0.54	0.59	0.95
9	LEM	LTSR6-NP [270]	Bipolar / Closed	1A, 2A, 3A, 4A, 5A	60 Hz	0.96	0 - 100 kHz	1 A	0.96	0.33	0.43	0.96
10	LEM	LV 25 P [271]	Bipolar / Closed	30V, 50V, 70V, 90V, 110V	60 Hz	0.96	0 - 25 kHz	110 V	0.96	0.37	0.51	0.94

works within the entire input frequency range of every Hall sensor.

5.6.6 Varying the magnetic field density of B_{atk}

In Sections 5.6.4 and 5.6.5, we keep the magnetic field density (i.e., ∼5600 G) and distance (i.e., 1 cm) of the source of B_{atk} (i.e., electromagnet) fixed. In this section, we vary the magnetic field density of the source of B_{atk} from a fixed distance (1 cm) and keep the frequency and amplitude of the input signals (S_{in}) fixed at 60 Hz/15Hz and 1 A/100 G/110 V, respectively. We vary the magnetic field density from 0 G to 9000 G at frequency zero and calculate C for every case for 10 different Hall sensors. The C is *less than 0.2* before HALC is used. However, the C is *greater than 0.93* for every sensor (Fig. 5.9 (Left)) when HALC is used. This proves that HALC can satisfactorily contain *both the weak and strong (i.e., 0 - ∼9000 G)* magnetic fields injected by the attacker.

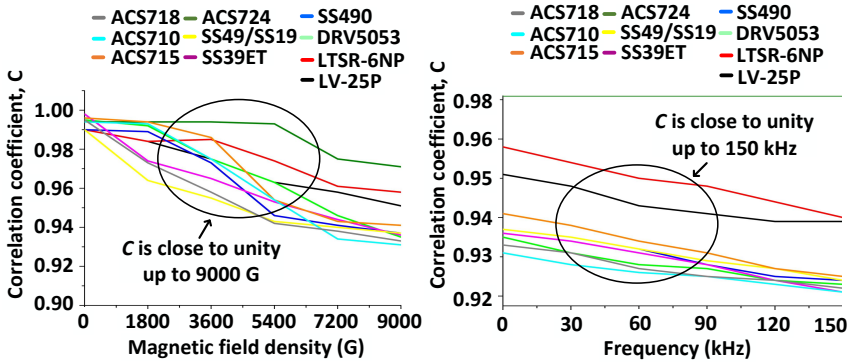


Figure 5.9: (Left) C with varying the magnetic field density of the B_{atk}. (Right) C with varying the frequency of the B_{atk}.

5.6.7 Varying the frequency of the B_{atk}

We keep the magnetic field density of B_{atk} at 300 G and very the frequency of the B_{atk} within 0 to 150 kHz. We keep the frequency and amplitude of the input signals (S_{in}) fixed at 60 Hz/15Hz and 1 A/100 G/110 V, respectively. The C is *within 0.3 to 0.71* before HALC is used. However, the C is *greater than 0.92* for every sensor (Fig. 5.9 (Right)) when HALC is used. This proves that HALC can satisfactorily contain both the low and high frequency magnetic spoofing within 0 - 150 kHz. It is important to note that the range 0 - 150 kHz

covers the entire input frequency range (see Table 5.1 (d)) supported by 10 different Hall sensors from 4 different manufacturers. The range also includes the same frequency as the input signals.

5.6.8 Varying the distance of the attack tool

We vary the distance of the attack tool (i.e., electromagnet, EMI) from the Hall sensor. We use a magnetic field density of 9000 G for B_{atk} and keep the frequency and amplitude of the input signals (S_{in}) fixed at 60 Hz/15Hz and 1 A/100 G/110 V, respectively. We vary the distance from 0 cm (very close) to 7 cm with an increment of 1 cm and calculate C for every case for all Hall sensors listed in Table 5.1. The value of C is greater than 0.91 for every case (Fig. 5.10 (Left)) when HALC is used. This proves that HALC can contain a magnetic spoofing attack from a very close distance.

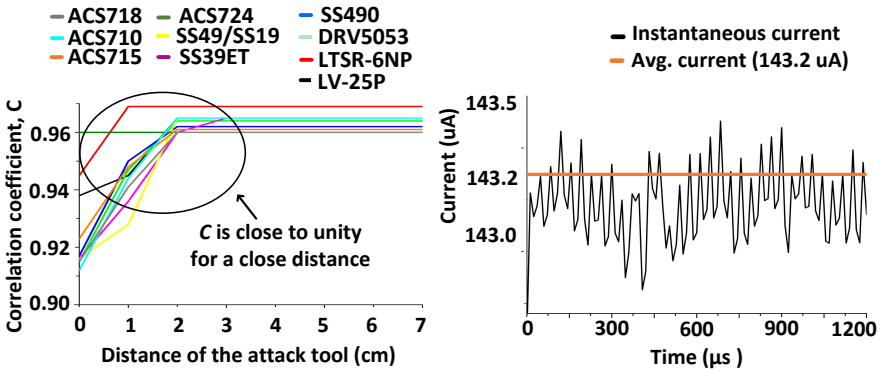


Figure 5.10: (Left) C with distance variation of the attack tool. (Right) The average and instantaneous current of digital core.

5.6.9 Comparing HALC with a shield

We compare HALC with a MuMetal shield [272], which is a foremost industry-used specialized material for magnetic shielding. We keep the source of $B_{atk} = 9000$ G 1 cm away outside of the shield and keep Hall sensors 1 cm away inside of the shield. We vary the thickness of the shield and find that even a 0.9 inch thick shield cannot prevent the $B_{atk} = 9000$ G

(i.e., low C in Table 5.1 (g), (h)). Because strong magnetic fields can saturate the MuMetal and diminish its shielding property, making it vulnerable to strong fields [79]. Next, we use HALC without the shield and find that C is close to unity (Table 5.1 (i)). This proves the efficacy of HALC over a shield.

5.6.10 Timing analysis of the analog core

The analog core is typically implemented by using a high-speed op-amp (Section 5.6.1) with very high slew rate, low rise-time, and high bandwidth. Therefore, the delay associated with the DC blocker, subtractor, adder1, and adder2 is typically less than 20 μs. The path b-c-d of the analog core comprises HPF and LPF. They introduce a delay in the form of phase shifts at nodes ⓒ, and ⓓ. The HPF creates a leading phase shift of +72.43°, and the LPF creates a lagging phase shift of -21.68°. The total phase shift occurs in path b-c-d is +72.43° + (-21.68°) = +50.74° leading. The +50.74° phase shift is equivalent to 2.36 ms of delay between signals at node ⓐ and node ⓓ. *This 2.34 ms of delay is compensated to* ***zero*** *by using a delay compensator (see Section 5.5.1). This preserves the hard real-time requirement of the overall system.*

5.6.11 Constant computational complexity

We implement the necessary filters in the analog core using first-order circuits. If these filters were implemented in the digital core using *higher-order* FIR or IIR filters, the CPU would require higher-order operations with high computational complexity. *HALC utilizes the analog and digital cores in such a way that the CPU does not need to handle higher-order arithmetic operations.* Instead, it handles first-order tasks that ensure a constant computational complexity of O(1). Moreover, the complexity of the defense algorithm 4 does not grow with the input data, and it remains constant independent of the different input signals/magnetic fields.

5.6.12 Timing analysis of the digital core

Broadly speaking, the digital core of HALC handles the following four tasks: (i) It samples signals using ADCs, (ii) It transfers sampled data to internal variables using DMAs, (iii) It processes the sampled signals by using an algorithm 4, and (iv) It generates DC feedback signals ($-E^s$-E^c) at node ⓖ using DACs. In this section, we calculate the time required to execute each of these tasks by considering the clock cycles required for each of these tasks. Four different clocks are used for the ADCs, DMAs, CPU, and DACs in the digital core. The frequencies of these clocks and the execution time required for each task are tabulated in Table 5.2.

Table 5.2: Timing analysis of the digital core

Task #	Clock name	Clock freq.	Min. time	Max. time
Task 1	ADC clock	11 MHz	16 μs	16 μs
Task 2	DMA clock	48 MHz	19 μs	19 μs
Task 3	CPU clock	48 MHz	31 μs	43 μs
Task 4	DAC clock	500 kHz	27 μs	27 μs
			93 μs (total)	105 μs (total)

The minimum and maximum execution time of tasks 1, 2, and 4 are constant as they don't involve the CPU. Task 3 involves the CPU and requires a minimum execution time of 31 μs and a maximum execution time of 43 μs. The CPU requires minimum and maximum execution time when a minimum and maximum number of cache miss occurs, respectively. The digital core requires a maximum of 105 μs or a minimum of 93 μs in total to generate the DC feedback signals -(E^s+E^c) to contain the DC attack component.

5.6.13 Attack containment in hard real-time

It is guaranteed that the digital core will provide feedback signals within a maximum of 105 μs of delay after signal changes at node ⓔ. The digital core executes the four tasks sequentially, and there is no task-scheduling involved in the process. Therefore, the delay associated with the digital core is always deterministic. Moreover, the digital core typically handles the low-frequency DC signals, which vary less slowly than the introduced delay/latency by the

digital core. Therefore, a 105 μs of delay is negligible compared to the rate of signal change in path b-e-h. *In addition, the phase-shift introduced by the analog core is taken care of by the delay compensator.* Therefore, the attack is contained in hard real-time.

5.6.14 Low-power HALC

The digital core consumes 0.5 mW and 0.3 mW average power when an attack happens and does not happen, respectively. When there is no attack, the digital core runs in energy-saving mode. The power is measured using an energy profiler app of the Simplicity Studio IDE [273]. The average and instantaneous current are shown in Fig. 5.10 (Right). The spike of the instantaneous current occurs during the ADC conversion. Moreover, the analog core consumes 1.4 mW of average power with or without an attack. Therefore, the total power consumed by HALC is ∼1.7-1.9 mW, which is compatible with power ∼ 10 mW [274] consumed by the Hall sensor itself.

5.6.15 Low-cost HALC and easy to integrate

HALC uses a cheap (∼$2) Hall sensor as the ESD. The total cost of our prototype is ∼$12, which is comparable with the sensor cost (∼$2 - $70). *However, as ∼$12 is the cost of the prototype, the actual cost will be much less in mass level production using SoC fabrication.* HALC can be connected with the target Hall sensor in a plug-&-play manner after fabricating HALC in a chip.

5.7 Evaluation of HALC

We evaluate HALC in two practical systems: a grid-tied solar inverter and a rotation-per-minute (RPM) system.

5.7.1 Grid-tied solar inverter

Grid-tied solar inverters are typically used as central inverters in solar/industrial plants or shopping malls. They widely use Hall sensors to measure AC and DC current. A 140 Watt inverter from Texas Ins. [275], which is a miniature version of a practical inverter, is used in the testbed to evaluate HALC. This inverter has a Hall effect current sensor with a part # ACS712ELCTR-20A-T with a *magnetic shield* around it. At first, we use our attack tool to inject constant, sinusoidal, and pulsating magnetic fields with a magnetic field density of 7000 G into the Hall sensor from a 1 cm distance. This drives the Hall sensor close to saturation and forces the inverter to shut down, causing a denial-of-service (DoS) attack even with a shield. Next, we connect HALC with the Hall sensor and repeat the same experiment (Fig. 5.11). At this time, nothing happens to the inverter, and it continues working without any disruption.

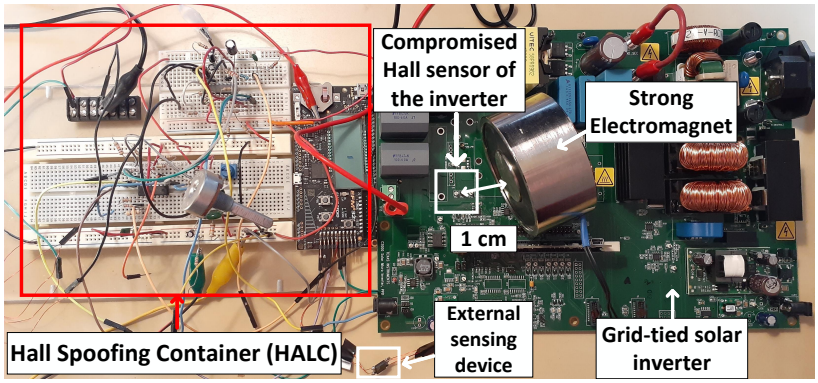


Figure 5.11: HALC can prevent the magnetic spoofing attack on the grid-tied solar inverter.

5.7.2 Rotation-per-minute (RPM) system

The RPM system is used in ICSs to measure the rotational speed of any rotating structure, such as a motor shaft, wheel. We use a motor shaft in our testbed with a Hall sensor having part # SS490. A small permanent magnet (part # HE510-ND) is mounted on the motor shaft. When the motor shaft rotates, the permanent magnet also rotates. The Hall sensor can sense the change of magnetic fields coming from the motor shaft (i.e., permanent

magnet) and use this information to count motor shaft rotations. At first, we provide a 100 RPM speed to the motor shaft. Then we inject magnetic fields with a magnetic field density of 5000 G from a 1 cm distance into the Hall sensor. As a result, the Hall sensor cannot measure the number of rotations correctly. Next, we connect HALC with the Hall sensor and repeat the same experiment. Now, the Hall sensor starts measuring the RPM correctly without any error (Fig. 5.12).

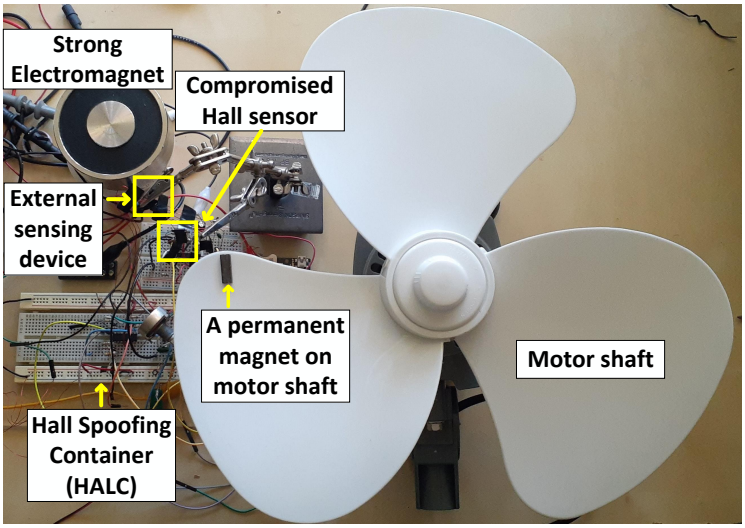


Figure 5.12: HALC is connected with the Hall sensor of the RPM system to prevent magnetic spoofing.

We find that if the C is < 0.8, a DoS attack happens in both the solar inverter and the RPM system. As HALC can keep the C close to unity, it can prevent the DoS and spoofing attack on Hall sensors.

5.8 Limitations

There are a few limitations of HALC. These limitations exist because of the limitations of the practical hardware.

5.8.1 Non-zero settling time of rheostat

The digital rheostats R7 and R11 used in the design have a non-zero settling time. We use MCP4252 [276] to implement rheostats R7 and R11 in our prototype (see Section 5.5.1). MCP4252 has an SPI interface that supports a 10 MHz clock. The total time required to calculate R7 and R11 and write these values to the MCP4252 chip using a 10 MHz SPI port is ∼3.5 μs. The time required to settle down the wiper of the digital rheostat is ∼240 μs. Therefore, the total settling time of the rheostat is 240 + 3.5 = 243.5 μs in our prototype. If the attacker changes the injected magnetic fields within 243 μs, the timeliness of the defense will not be guaranteed. The settling time of the rheostat results from its parasitic capacitance. Therefore, the settling time can be reduced from 243 μs to a lower value using a rheostat having lower parasitic capacitance, which can be achieved using JFETs instead of traditional MOSFETs in a rheostat.

5.8.2 Upper limit magnetic field density of B_{atk}

HALC can work up to a magnetic field density of ∼9000 G. The upper limit ∼9000 G originates from the amplifier in Fig. 5.5, which cannot provide the feedback signal -($E^c + E^s$) more than the supply voltage (i.e., 5 V). By increasing the supply voltage from 5 V to a higher value, the upper limit of the B_{atk} can be increased.

5.8.3 Upper limit frequency of B_{atk}

The prototype of HALC can prevent a B_{atk} with frequencies of 0 Hz to 150 kHz. The upper limit of 150 kHz can be increased beyond 150 kHz by increasing the maximum upper limits of rheostats R7 and R11. To increase the maximum upper limits of rheostats, multiple digital rheostats can be connected in series. However, it may increase the settling time of the rheostats.

5.8.4 Multiple co-located Hall sensors

In the present state of the design, if multiple sensors are co-located, HALC should be used with each co-located sensor separately.

5.9 Related work and Limitations

To the best of our knowledge, no state-of-the-art work provides a defense against a strong magnetic spoofing attack on Hall sensors. However, few related works exist for other sensors that can not be applicable to Hall sensors for the following reasons.

Trippel et al. [51] proposed randomized and 180^0 out-of-phase sampling to nullify acoustic spoofing signals injected into MEMS accelerometers. Randomized sampling samples at random times and 180^0 out-of-phase sampling takes 2 samples with 180^0 out of phase within the resonant frequency period to nullify the spoofing signals. These defenses will fail in two scenarios: (i) When the spoofing signal has the same frequency as the original signal being measured because randomized sampling will nullify both the spoofing and original signals. (ii) When the spoofing signal is a DC/constant signal because randomized sampling cannot filter out a DC signal.

Cheng et al. [248] and Alexander [249] from Allegro Microsystems proposed differential Hall sensors to nullify common-mode spoofing signals. This technique would work for weak magnetic spoofing but does not work against strong magnetic spoofing. The reasons behind this limitation are already explained in Section 5.3.3 in detail.

Kune et al. [42] proposed adaptive filtering to estimate the spoofing attack signal first and then subtract the estimated attack signal from the original signal to clean up the original signal. This technique will fail in two scenarios: (i) Because of the physical distance between the adaptive filter and the compromised Hall sensor, the adaptive filter cannot measure

the exact amplitude of the external attack fields. This is why we can not simply subtract the estimated attack signals from the original signals to recover the original signal. (ii) This technique uses higher order FIR filters for adaptive filtering that is computationally expensive and may hamper the real-time requirement of the defense (refer to Section 5.6.11 for details).

Zhang et al. [47] used a Support Vector Machine (SVM), and Roy et al. [250] proposed a non-linearity tracing classifier to contain the inaudible voice commands injected into MEMS microphones in ultrasonic range. These defenses have following limitations: (i) They will work only for spoofing signals located in ultrasonic frequency band ($>$ 20kHz), which has a clear separation from the audible voice signals ($<$ 20 kHz). As the spoofing signal may share the same band as the original signal in Hall sensors, these defenses don't work for Hall sensors. (ii) They will not work for DC spoofing signals.

The works in [42, 51, 248, 249] are sensor-level and [47, 250] are system-level defenses. There are other system-level defenses. Shoukry et al. [252] proposed PyCRA that only can detect an attack but cannot prevent it. Cardenas et al. [277] and Urbina et al. [278] incorporated the knowledge of the physical system under control to detect an attack on ICSs. But their approaches cannot contain the attack. Again, Shoukry et al. [279] proposed to reconstruct the state to recover from a sensor spoofing attack using the satisfiability modulo theory (SMT) that can not be implemented in the *in-sensor* hardware.

Table 5.3: Summary of the strength of HALC.

Strength	Values
values of injected B_{atk}	up to $\sim$ 9000 G
frequencies of injected B_{atk}	0 - 150 kHz
proximity of the attack tool	$<$ 1 cm
power consumption	$\sim$ 1.7 - 1.9 mW
cost	$\sim$ \$12
latency	93 μs - 105 μs
constant, sinusoidal, pulsating B_{atk}	✓
spoofing signal having same frequency as original signal	✓
Works within entire input signal (S_{in}) range	✓

Moreover, machine learning techniques and other system-level defenses require complex

computations to converge for attack detection and recovery, requiring powerful hardware resources. Therefore, they are not suitable for low-power real-time sensor systems with constrained resources. *In addition, they may not work against a time-varying magnetic spoofing as a time-varying signal may create oscillations between two safe states of the controller, and they are incapable of handling these oscillations in real-time.*

HALC is novel in the sense that it can *detect and contain* a strong magnetic spoofing up to ∼9000 G of any type, such as constant/DC, sinusoidal, and pulsating magnetic fields, in real-time and can keep the connected system running during the attack. A summary of the strength of HALC is given in Table 5.3.

5.10 Summary

We have presented HALC, a defense against a weak and strong magnetic spoofing attack on Hall sensors. HALC can not only detect but also contain the weak and strong magnetic spoofing of different types, such as constant, sinusoidal, and pulsating fields, in hard real-time. HALC utilizes the analog and digital cores to achieve a constant computational complexity O(1) and keep the existing data processing speed of the connected system undisturbed. We have done extensive analysis of HALC on 10 different Hall sensors from 4 different manufacturers and proved its efficacy against the magnetic spoofing attack. We have demonstrated that our proposed defense is low-power and low-cost and can be implemented in the sensor hardware domain. Moreover, we have evaluated the effectiveness of HALC in two practical systems. Our results from these experiments prove that HALC can accurately and reliably detect and mitigate the magnetic spoofing attack in hard real-time. To the best of our knowledge, HALC is the first of its kind that can provide defense against a weak/strong magnetic spoofing on the Hall sensor. Finally, we believe that HALC has the potential to be adopted for other passive sensors in general to protect them from a spoofing attack.

Chapter 6

PreMSat: Preventing Magnetic Saturation Attack on Hall Sensors

6.1 Abstract

Spoofing a passive Hall sensor with fake magnetic fields can inject false data into the downstream of connected systems. Several works have tried to provide a defense against the intentional spoofing to different sensors over the last six years. However, they either only work on active sensors or against externally injected unwanted weak signals (e.g., EMIs, acoustics, ultrasound, etc.), which can only spoof sensor output in its *linear* region. However, they *do not* work against a strong magnetic spoofing attack that can drive the passive Hall sensor output in its *saturation* region. We name this as the saturation attack. In the saturation region, the output gets flattened, and no information can be retrieved, resulting in a denial-of-service attack on the sensor.

Our work begins to fill this gap by providing a defense named PreMSat against the saturation attack on passive Hall sensors. The core idea behind PreMSat is that it can generate an internal magnetic field having the *same* strength but in *opposite polarity* to external magnetic fields injected by an attacker. Therefore, the generated internal magnetic field by PreMSat can nullify the injected external field while preventing: (i) intentional spoofing in the sensor's *linear region*, and (ii) saturation attack in the *saturation region*.

PreMSat integrates a low-resistance magnetic path to collect the injected external magnetic fields and utilizes a finely tuned PID controller to nullify the external fields in real-time. PreMSat can prevent the magnetic saturation attack having a strength up to ∼4200 A-t within a frequency range of 0 Hz–30 kHz with low cost (∼$14), whereas the existing works cannot prevent saturation attacks with any strength. Moreover, it works against saturation attacks originating from *any type*, such as constant, sinusoidal, and pulsating magnetic fields. We did over 300 experiments on ten different industry-used Hall sensors from four different manufacturers to prove the efficacy of PreMSat and found that the correlation coefficient between the signals before the attack and after the attack is greater than 0.94 in every test case. Moreover, we create a prototype of PreMSat and evaluate its performance in a practical system — a grid-tied solar inverter. We find that PreMSat can satisfactorily prevent the saturation attack on passive Hall sensors in real-time. The findings in this chapter have been published in [156].

6.2 Introduction

A Hall sensor can measure magnetic fields from the surrounding environment and generates a proportional voltage at its output [246]. Hall sensors are pervasive in many safety-critical systems, ranging from industrial controllers to power systems, computers to home automation, and automobiles to aircraft [159, 242–244, 280–282]. Over the last three decades, Hall sensors have been technically improved in terms of stability, accuracy, and linearity [283]; however, to the best of our knowledge, designers still do not consider security as one of the important requirements while designing hall sensors. The vulnerability of Hall sensors has recently been exposed by few works [48, 245]. In these works, the attacker uses an external magnetic field to spoof Hall sensors located in a solar inverter and anti-lock braking system, resulting in a denial-of-service (DoS) attack on the connected power grids and automotive systems, respectively.

A Hall sensor has a Hall element [284], which outputs a voltage proportional to the sensed magnetic fields to a differential amplifier. The input-output characteristic of a differential amplifier is linear. If the output voltage from the Hall element is small, the differential amplifier typically works in its linear region. However, if the output voltage from the Hall element is large, the differential amplifier cannot work in its linear region and is driven to its saturation region [285]. In the saturation region, the input-output characteristic gets flattened; hence, no information can be recovered, causing a DoS attack on the Hall sensor. An attacker can use this knowledge to drive the differential amplifier to its saturation region by using a strong external magnetic field. We name this attack as the *saturation attack*. Please note that here, *sensor saturation* does not refer to *magnetic saturation* [286]. Moreover, Hall sensors are broadly two types: active and passive. Passive Hall sensors are naive devices; they send signals to the upper level without checking the integrity of the signals that makes them vulnerable to external fake magnetic fields.

Recent works [42, 47, 51, 157, 248–250] may prevent spoofing a sensor in its linear region to some extent. However, to the best of our knowledge, no work in literature can prevent a *saturation attack* on *passive* Hall sensors. Therefore, we provide a defense for passive Hall sensors against a saturation attack. We name it as PreMSat: Preventing Magnetic Saturation, which can prevent a saturation attack on passive Hall sensors[1]in real-time and also prevent spoofing in the linear region.

The core idea behind PreMSat is that it can generate an internal magnetic field having the *same* strength but in *opposite polarity* to the external magnetic field injected by an attacker. As a result, the internal magnetic fields generated by PreMSat can nullify the externally injected magnetic fields with two consequences: (i) it prevents magnetic spoofing in the linear region, and (ii) it prevents the saturation attack. Please note that only a portion of injected magnetic fields may contribute to the saturation attack on Hall sensors. Therefore,

[1]Hall sensors mean unipolar, bipolar, open/closed-loop ***passive*** Hall sensors, unless stated otherwise.

PreMSat introduces the following three techniques: (i) PreMSat provides a low-resistance magnetic path, made with ferrite core, to collect the contributing portion of the externally injected fields, (ii) PreMSat provides a *secondary sensor*, mounted in the ferrite core, to measure the strength and polarity of the contributing external field, and (iii) PreMSat uses a *proportional-integral-derivative (PID) controller* and a *primary coil* to generate an internal magnetic field equal to the contributing external field to nullify it. The PID controller is well-tuned so that it takes a settling time of 23 μs to generate the stable internal magnetic field. The low settling time of the PID controller fulfils the real-time requirement of PreMSat. We demonstrate the efficacy of PreMSat on a grid-tied inverter proving its real-time effectiveness against the saturation attack on practical systems. We present a prototype of PreMSat that nullifies external fields with a strength up to $\sim$ 4200 A-t. It seems that a strong attacker may overcome the defense prototype with a field higher than 4200 A-t. However, the strength of the prototype theoretically can be increased to any higher limit using stronger hardware that may prevent a stronger attacker.

Contributions: Our main technical contributions in this paper are listed below:

1. We propose PreMSat that can protect a passive Hall sensor against: (i) spoofing attacks on linear regions and (ii) saturation attacks on saturation regions. It works against any type, such as constant, sinusoidal, and pulsating magnetic fields, in real-time.

2. We create a prototype of PreMSat and show its effectiveness through experiments on ten different Hall sensors from four different manufacturers. We consider different types, namely unipolar, bipolar, open-loop, and closed-loop Hall sensors to prove that PreMSat is a general defense technique against the saturation attack on passive Hall sensors.

3. We evaluate the efficacy of PreMSat on a real-world practical system — a grid-tied inverter and demonstrate that PreMSat prevents the DoS attack on a practical system.

Demonstration: The demonstration of the proposed defense is shown in the following link:

https://sites.google.com/view/preventingmagneticsaturation/home

6.3 Preliminaries

6.3.1 The physics of the Hall sensor

The physics of a typical Hall sensor is shown in Fig. 6.1 (left). The Hall sensor [287] has a Hall element, which is a p-type semiconductor [288]. Let us denote the thickness of the Hall element by d. A DC voltage bias is applied across the Hall element that causes a bias current, I_{Bias} flowing through the Hall element along the +X axis. Let us assume a magnetic field/flux density, B is present along the +Z axis. The magnetic field, B exerts a Lorentz force, F [289] on electrons and holes of the Hall element that deflects them to either side of the Hall element along the +Y axis [290]. As electrons and holes move sidewards along the +Y axis, a voltage is generated between two sides of the Hall element along the +Y axis. The voltage is known as Hall voltage, V_H and is expressed as:

$$V_H = k(\frac{I_{Bias}}{d} \times B) \tag{6.1}$$

where k is the Hall coefficient. Typically I_{Bias}, d and k are held constant; therefore, V_H is proportional to the magnetic field density B. In this way, a Hall sensor can sense a magnetic field B and convert it to a useful electrical signal V_H.

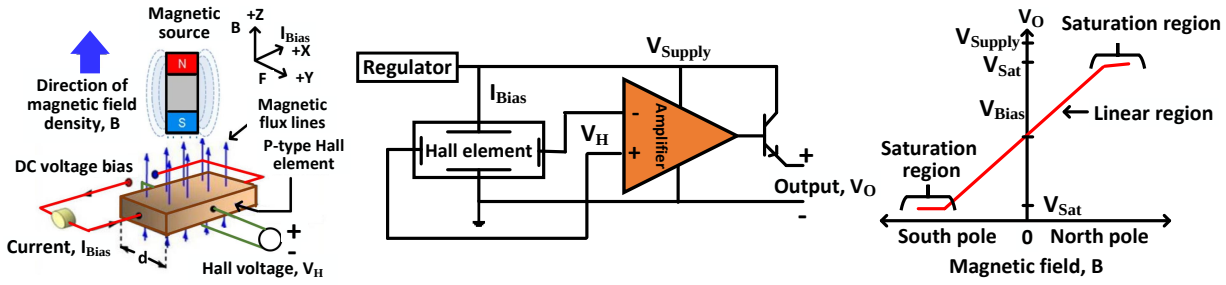


Figure 6.1: (left) The physics of a typical Hall sensor. (middle) Hall sensor electronics. (right) The linear and saturation regions of a typical Hall sensor.

6.3.2 Hall sensor electronics

Hall sensors have other electronics in addition to a Hall element that is shown in Fig. 6.1 (middle). The output of the Hall element is given to a signal conditioning block, which has a differential amplifier. A differential amplifier amplifies the output voltage of the Hall element (i.e., V_H) and also removes the common-mode noises from V_H. Common-mode noises are unwanted signals that are present at +ve and -ve input leads of the differential amplifier with respect to analog ground. Moreover, a voltage regulator is used to provide a stable bias current I_{Bias} to the Hall element. The stable I_{Bias} keeps the output V_H proportional to the input magnetic field B in Eqn. 6.1. It is clear from this discussion that Hall sensors don't have dedicated hardware to prevent a spoofing attack on them.

6.3.3 Linear and saturation regions of a Hall sensor

The sensed magnetic field B in Eqn. 6.1 can be either +ve or -ve depending upon its polarity (i.e., north/south pole). Therefore, the differential amplifier's output, denoted as V_O in Fig. 6.1 (middle), can go either +ve or -ve, thus requiring two (i.e., both +ve and -ve) power supplies. To avoid using two power supplies, a fixed bias voltage, V_{Bias} is added to the differential amplifier. Therefore, a +ve/-ve magnetic field B can drive the V_O to upper/lower position from the V_{Bias} and $V_O = V_{Bias}$ when B is zero. The term V_O works in the linear region, and the V_O cannot exceed the limit imposed by the power supply. In fact, the V_O will begin to flatten before the power supply limits are reached. This flattened region is known as the saturation region, denoted by V_{Sat}, which is illustrated in Fig. 6.1 (right). *Please note that the exact value of input field B cannot be recovered while the differential amplifier's output V_O is in the saturation region.* Moreover, saturation occurs in the differential amplifier, not in the Hall element. Therefore, a strong spoofing magnetic field can drive the Hall sensor to saturation without damaging the Hall element.

A naive approach to prevent a saturation attack is to increase the saturation voltage of a differential amplifier. However, this is not a complete solution because the attacker can still spoof a Hall sensor in its linear region. We discuss the advantages of PreMSat over increasing the saturation voltage of a differential amplifier in Sections 6.5.3 and 6.6.10. In addition, the detection of the saturation attack can be done by checking the output V_O stuck to the +ve/-ve limits; however, this will not help to recover information from the saturation region and cannot prevent spoofing in the linear region of the amplifier.

6.3.4 Active and passive Hall sensor

An active Hall sensor [251] can measure signals transmitted by the sensor that were reflected, refracted, or scattered by the physical environment. A passive Hall sensor [247] can only measure natural emissions coming from the physical environment. PyCRA [252] works only for active sensors but not for passive Hall sensors. Therefore, we aim to provide a defense against the saturation attack on passive Hall sensors.

6.3.5 Proportional-integral-derivative (PID) controller

A PID controller [291] is a closed-loop control system that generates a feedback signal to minimize an error. It continuously calculates an error, $e(t) = r(t)$ - $u(t)$, as the difference between a desired setpoint $r(t)$ and a feedback signal $u(t)$. It continuously updates $u(t)$ to minimize the error $e(t)$ so that $u(t)$ achieves a value closer to desired setpoint $r(t)$. It uses proportional, integral and derivative operations on $e(t)$ following Eqn. 6.2.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\,dt + K_d \frac{de(t)}{d(t)} \tag{6.2}$$

where K_p, K_i, and K_d are proportional, integral, and derivative gain, respectively. The values of K_p, K_i, and K_d should be tuned optimally so that the PID controller remains

stable with a minimum overshoot of $u(t)$. Usually, tuning takes place in the s-domain for a continuous-time PID controller or in the z-domain for a discrete-time PID controller. The s-domain is used to solve the continuous-time differential equation, whereas the z-domain is used to solve a discrete-time equation with Z-transformation [292].

6.4 Saturation attack model and its consequences

The important four components of the saturation attack model are explained below:

1. Assumptions on attackers: The attacker can be a disgruntled employee or a guest, who is not allowed to modify the target Hall sensor like a lunch-time attack [254].

2. Attacker's goals: The attacker only uses high power magnetic energy from a distance to *noninvasively* spoof and inject malicious signals into the Hall sensor to drive it to its saturation region. Therefore, the attack can be seen as a *noninvasive physical attack*.

3. Attack tool and cost: The attacker can use an electromagnet [256] to generate strong fields for a saturation attack. The electromagnet can be controlled with a MOSFET [293] and an Arduino [294] using pulse-width modulation (PWM) technique to generate different types, such as constant, sinusoidal, pulsating magnetic fields, with different frequencies. The total cost of attack tools is $<$ \$60, which will not be increased for higher strength or frequency of the magnetic fields. By changing the duty cycle and frequency of the PWM signal, it is possible to increase or decrease the strength or frequency of the magnetic fields at the same cost. Moreover, the attack tools are easily available on Amazon/Digikey. Therefore, the saturation attack is realistic for a strong attacker.

4. Sensor shield: A sensor shield may or may not be present around a Hall sensor. The saturation attack is strong enough to drive the Hall sensor to its saturation region even in the presence of a shield. We compare PreMSat with a shield in Section 6.6.9 in detail.

Consequences of the saturation attack: As Hall sensors are critical parts of safety-critical systems (i.e., autonomous vehicles, smart grids, etc.), the consequences of a saturation attack on a target Hall sensor can be catastrophic. A similar incident is found in the literature where an attacker injects fake magnetic fields into Hall current sensors located in a solar inverter and drives the hall sensor to its saturation. As a result, the solar inverter shuts down itself because the saturated Hall sensor cannot provide correct values from the micro-grid, causing a blackout in the micro-grid [245]. Another incident demonstrates a *disruptive attack* on a Hall sensor located in an anti-lock braking system (ABS) of a vehicle, resulting in a possible brake failure [48]. An example of a saturation attack other than on a Hall sensor is demonstrated by Shin et al. [46] on lidars used in an autonomous vehicle. The outcome of this attack is the loss of control of the vehicle. Park et al. [43] saturate a drop sensor of a medical infusion pump using an IR laser. This attack makes the drop sensor insensitive to any fluid drops. *All these examples indicate that saturation attacks can cause a DoS attack on critical sensors and have catastrophic consequences in terms of loss of human life and monetary resources.* Therefore, a defense (i.e., like PreMSat) is necessary against a saturation attack on sensors.

6.5 The defense scheme - PreMSat

The core idea behind PreMSat is that it can generate an internal magnetic field having the *same* strength but in *opposite polarity* to the externally injected magnetic fields. *As a result, the internal magnetic fields can nullify the external magnetic fields.* Before designing PreMSat, it is required to discuss few important concepts related to electromagnetism that will be conceptualized in PreMSat.

6.5.1 Contributing direction of the magnetic fields on Hall sensors

The Hall element in the Hall sensor is not sensitive to all directions of a magnetic field. Rather, the Hall element is sensitive to a particular magnetic field direction that actually

contributes to the generation of the Hall voltage V_H. We bring Proposition 1 below to state the contributing direction of magnetic fields on Hall sensors.

Proposition 1: The Hall element located in the Hall sensor is sensitive to only the vertical component of the magnetic fields that is perpendicular to the bias current I_{Bias}.

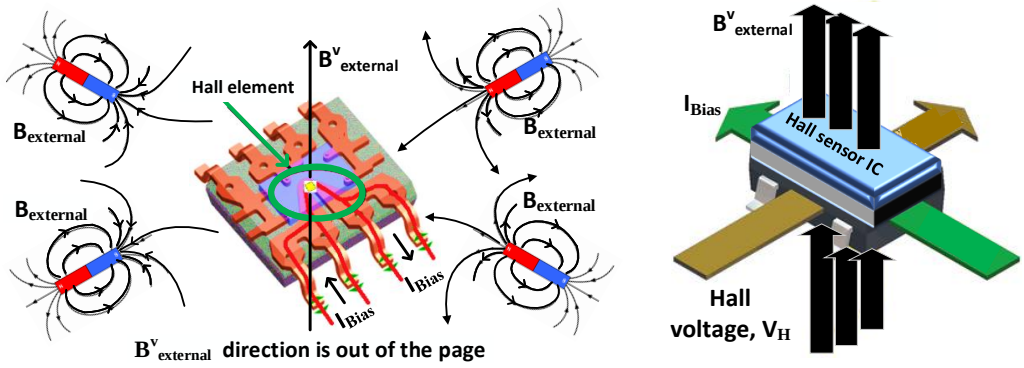


Figure 6.2: For multiple sources of $B_{external}$, the vector summation of vertical components of $B_{external}$, which is perpendicular to I_{Bias}, only contributes to the Hall voltage, V_H.

Explanation of Proposition 1: According to Lorentz Force [289], the Hall voltage V_H in Eqn. 6.1 is only sensitive to magnetic fields B, which is perpendicular to bias current I_{Bias}. This phenomena is also illustrated in Fig. 6.1 (left), where magnetic fields B is in +Z axis and the bias current I_{Bias} is flowing along +X axis. Therefore, the Hall element is only sensitive to the vertical component of magnetic fields that is perpendicular to I_{Bias}.

Terminology: Let us denote the external magnetic fields injected by the attacker by $B_{external}$. If the attacker uses multiple magnetic sources to generate $B_{external}$, the vector summation of all the vertical components of the $B_{external}$ will contribute to the Hall voltage, V_H. Let us denote the magnitude of the summation of all vertical components of $B_{external}$ perpendicular to I_{Bias} by $B^v_{external}$. Fig. 6.2 depicts the presence of the $B^v_{external}$ in the case of multiple magnetic sources. As $B^v_{external}$ only contributes to the V_H, PreMSat should need to generate an internal magnetic field having the same magnitude of $B^v_{external}$ in opposite polarity to nullify the $B^v_{external}$. Let us denote the magnitude of the internal magnetic field

generated by PreMSat by $B_{internal}$, where the $B_{internal}$ should be equal to the $B^{v}_{external}$ in opposite polarity to nullify the $B^{v}_{external}$.

6.5.2 Internal magneto-motive force (MMF) generated by PreMSat

The attacker needs a *magnetic source* (i.e., electromagnet, electromagnetic interference - EMI, etc.) to generate external magnetic fields $B_{external}$ to drive the target Hall sensor to its saturation region. The strength of the magnetic source is quantified by magneto-motive force (MMF) [289]. For defense, PreMSat needs to use an internal magnetic source that can generate the exact MMF to provide an internal field $B_{internal}$ to nullify the $B^{v}_{external}$. Let us denote the internal MMF generated by PreMSat by $MMF_{internal}$.

Primary coil: PreMSat implements a circular *ferrite core* [295] with a coil winded in spiral direction to generate the $MMF_{internal}$. As the ferrite core has circular shape, it can also be called by a *toroid*. The term *toroid* is used interchangeably with *ferrite core* in this paper. Let us denote the winding coil, which generates the $MMF_{internal}$, by the *primary coil*. The construction of the toroid with the primary coil is shown in Fig. 6.3. The $MMF_{internal}$ generated by the primary coil is expressed in Eqn. 6.3.

$$MMF_{internal} = N_{primary} I_{primary} \tag{6.3}$$

where $N_{primary}$ is the total number of turns in the primary coil and $I_{primary}$ is the current flowing through the primary coil. The $MMF_{internal}$ generates the internal magnetic field $B_{internal}$, which can be expressed as follows for a toroid:

$$B_{internal} = \frac{\mu_r \mu_o N_{primary} I_{primary}}{2\pi r} = \frac{\mu_r \mu_o MMF_{internal}}{2\pi r} \tag{6.4}$$

where μ_o is the magnetic permeability of air, μ_r is the relative permeability of a ferrite core, and r is the radius of a toroid. The generated $B_{internal}$ should be equal to the $B^v_{external}$ but in opposite polarity to nullify the $B^v_{external}$. This will be discussed in the next section.

6.5.3 Primary coil nullifies the $B^v_{external}$

As discussed earlier, the primary coil generates a $B_{internal}$, which is equal to the $B^v_{external}$ but in opposite polarity to nullify the $B^v_{external}$. PreMSat generates the $B_{internal}$ by addressing the following two important questions:

Q1. How can PreMSat generate $B_{internal}$ having *equal magnitude* to the $B^v_{external}$?

Q2. How can PreMSat align the $B_{internal}$ in *opposite direction* to nullify the $B^v_{external}$?

These two questions are addressed below in Sections 6.5.3, 6.5.3, and 25.

Generating the $B_{internal}$ having equal magnitude to the $B^v_{external}$

At first, PreMSat needs a methodology to sense the *magnitude and direction* of the $B^v_{external}$ correctly to generate a correct $B_{internal}$. The steps to accomplish this is explained below.

■ ***1. Introducing a secondary sensor:*** As a Hall sensor under attack is a naive device, it cannot alone differentiate between the natural input magnetic fields and the attacker's provided external magnetic fields $B^v_{external}$. Let us denote the natural *input* magnetic field by B_{input} that actually needs to be measured by the Hall sensor. To differentiate the B_{input} from the $B^v_{external}$, PreMSat uses a *secondary sensor* placed in the toroid. Please note that the secondary sensor is used only to sense the external magnetic field $B^v_{external}$. The secondary sensor is placed close to the target Hall sensor so that it can sense the external magnetic fields injected into the target Hall sensor (see Fig. 6.3 and 6.4). The secondary sensor can be implemented using either a Hall sensor or a magnetic coil.

The next question is how the secondary sensor actually differentiates the natural input magnetic fields B_{input} from the externally injected magnetic fields $B^{v}_{external}$. Let us answer the above question by considering the following two scenarios.

First scenario: When the natural input field B_{input} is internal, the secondary sensor only senses the injected external magnetic field $B^{v}_{external}$ (Fig. 6.3). This happens for voltage/current Hall sensors, where the natural input magnetic field is generated internally from an internal voltage/current signal inside of the Hall sensor (sensors 1-6 in Table 6.2).

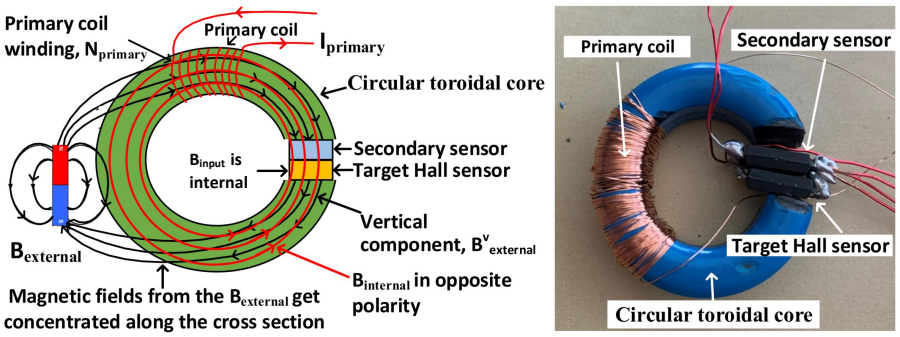


Figure 6.3: (left) The toroid hosts the target Hall sensor and the secondary sensor and provides a magnetic path to collect the injected $B_{external}$. Here, the natural input magnetic field B_{input} is internal. (right) The implementation of the toroid.

Second scenario: When the natural input field B_{input} and the injected external field $B^{v}_{external}$ both are external, there is a chance that the secondary sensor can sense both the natural and injected external fields (sensors 7-10 in Table 6.2). To prevent this from happening, a shield between the secondary sensor and the source of natural input field is used in PreMSat. This concept is illustrated in Fig. 6.4. We use a shield having six segments (i.e., i - vi in Fig. 6.4) made of a ferromagnetic material in such a way that it guides the external natural input field B_{input} not to go to the secondary sensor but only to go to the target Hall sensor. The segment (i) prevents the B_{input} to induce in the circular toroid. The segment (ii) guides the B_{input} to penetrate through the target Hall sensor for being measured. The segment (iii) provides a path to close the loop of the B_{input}. And the segments (iv), (v) and (vi) will be needed if the source of $B_{external}$ is placed at the same side of the source of

B_{input}. Because in this scenario, the $B_{external}$ may influence the segments (i) and (iii) and may bypass the secondary sensor. To prevent this from happening, the segments (iv), (v), and (vi) are used to guide the $B_{external}$ to go to the circular toroid core without influencing the segments (i), (iii) and the B_{input}.

Please note that the construction of the shield may vary for different requirements and locations of the B_{input} depending on its different use-cases. It is possible that more or fewer segments may be needed other than the above six segments. For example, the segment (vi) can be safely omitted if the attacker cannot access this side to place the source of $B_{external}$. The sizes of the shield's segments are not large compared to the toroid. Therefore, the structure shown in Fig. 6.4 (right) will work in most applications, such as proximity sensing, and throttle angle sensing. However, few applications where moving parts are involved, such as brushless motors, may find it difficult to install the segments.

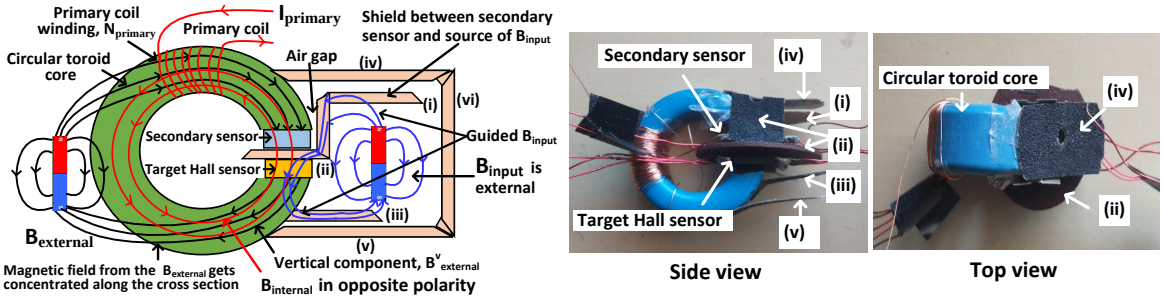


Figure 6.4: (left) The toroid hosts the target Hall sensor and the secondary sensor and provides a magnetic path to collect the injected $B_{external}$. Here, the natural input magnetic field B_{input} is external. (right) Side and top views of the implemented toroid.

As the direction of the natural field B_{input} is known to the designer, he can always design a shield to prevent the natural field from going into the secondary sensor. Moreover, as the strength of the natural field is known to the designer, he can use a shield with proper ferromagnetic material to ensure that the natural field cannot penetrate the shield.

A further question may arise if the attacker can bypass the shield. Please note that the use of the shield is not to prevent attackers from influencing the target Hall sensor. However, the use of the shield is to prevent the input magnetic fields (B_{input}) from going into the

secondary sensor. Therefore, bypassing the shield with the $B_{external}$ by an attacker will not impact the defense because the secondary sensor and target Hall sensor can still sense the injected $B_{external}$. Moreover, if a strong injected $B_{external}$ penetrates the shield, that would not be a problem. Because in that case, the secondary sensor will still sense the injected $B_{external}$ by the attacker and won't sense the natural input field B_{input}.

■ ***2. Sensing*** $B^v_{external}$ ***from the*** $B_{external}$: PreMSat uses a magnetic path to collect the vertical components $B^v_{external}$ from the $B_{external}$. The circular ferrite core in PreMSat provides that magnetic path. We bring Proposition 2 below to explain this concept.

Proposition 2: As the ferrite core in PreMSat has very low magnetic resistance compared to the air, practically speaking, most of the magnetic fields from $B_{external}$ will get concentrated along the cross-section of the ferrite core [296–298].

Explanation of Proposition 2: The way how the circular ferrite core provides a magnetic path to collect the vertical components $B^v_{external}$ is shown in Fig. 6.3 and 6.4. When single/multiple sources of $B_{external}$ are present near the target Hall sensor, the $B_{external}$ needs to overcome the air gap present between the target Hall sensor and the source of $B_{external}$. As air has a very low magnetic permeability (e.g., $4\pi 10^{-7}$ Wb/A-t.m), the air gap present between the target Hall sensor and the $B_{external}$ works as a magnetic path having very high resistance. Therefore, the magnetic field lines coming from the $B_{external}$ change their normal path and try to find a new path having a low magnetic resistance. The circular ferrite core provides the very low resistive magnetic path to the $B_{external}$. In numbers, the relative magnetic permeability of ferrites can vary between 1150 to 25000 [299]. In other words, the magnetic resistance of the ferrite core is 1150 - 25000 times less than air. As the ferrite core has very low magnetic resistance compared to air, practically speaking, most of the external magnetic fields from the $B_{external}$ get concentrated along the cross-section of the ferrite core, hence influencing the field pattern of the $B_{external}$.

Vertical projection of $B_{external}$ onto the Hall sensor: As the $B_{external}$ is concentrated along the cross-section of the ferrite core, if we could place the target Hall sensor in the cross-section of the ferrite core, the $B_{external}$ will be projected onto the target Hall sensor vertically. The reason behind this is that as the ferrite core has a circular shape, the concentrated fields $B_{external}$ along the circular core will be vertical to any *plane* placed in the cross-section of the circular core. The idea is illustrated in Fig. 6.3 and 6.4. A small gap is created to place the target Hall sensor in the cross-section of the ferrite core. Therefore, the concentrated $B_{external}$ will act as the $B^v_{external}$ to the target Hall sensor as the target Hall sensor is placed in the cross-section of the circular ferrite core.

The secondary sensor is also placed together with the target Hall sensor in the gap of the circular ferrite core. This is illustrated in Fig. 6.3 and 6.4. As the secondary sensor is placed together with the target Hall sensor, the same $B^v_{external}$ passes through the secondary sensor. Therefore, the secondary sensor sees the same amount of $B^v_{external}$, similar to the target Hall sensor. In this way, the secondary sensor placed in the ferrite core can sense the $B^v_{external}$ injected by the attacker.

■ ***3. Generating a voltage proportional to the $B^v_{external}$ by the secondary sensor***: PreMSat uses a Hall sensor as the secondary sensor for simplicity. A magnetic coil could also be used as the secondary sensor. As a Hall sensor is used as a secondary sensor, after sensing the $B^v_{external}$, the secondary sensor generates a Hall voltage following Eqn. 6.1. Let us denote the generated Hall voltage in the secondary sensor by $V_{secondary}$.

Types of $B^v_{external}$: We consider a strong attacker who can use constant, sinusoidal, and pulsating fields for a saturation attack because all other patterns can be derived from these three basic fields (i.e., Fourier transformation [255]). Therefore, we discuss how $V_{secondary}$ changes for the constant, sinusoidal, and pulsating magnetic fields. This information on $V_{secondary}$ is required to design algorithm 5, which can prevent the saturation attack generating from any type of $B^v_{external}$. Let us define the constant, sinusoidal and pulsating magnetic fields

mathematically in Eqn. 6.5.

$$B^{v}_{external} = \begin{cases} C; & \text{constant field,} \\ B_{amplitude} \sin \omega t; & \text{sinusoidal field,} \\ B_{amplitude}\{sgn(\sin \omega t)\}; & \text{square pulsating field.} \end{cases} \tag{6.5}$$

where C is a constant, ω is the angular frequency and $B_{amplitude}$ is the magnitude of the injected magnetic field, and sgn is the signum function. If we use $B^{v}_{external}$ from Eqn. 6.5 in Eqn. 6.1, we can calculate the $V_{secondary}$, which is graphically illustrated in Fig. 6.5.

The $V_{secondary}$ is proportional to the $B^{v}_{external}$: Eqn. 6.1 shows that the term V_H is proportional to the magnetic fields B present in the +Z direction. Therefore, the secondary sensor also generates the $V_{secondary}$, which is proportional to the vertical components of the externally injected magnetic fields, previously denoted by $B^{v}_{external}$. Hence, the $V_{secondary}$ has the shape and frequency equal to $B^{v}_{external}$ that is illustrated in Fig. 6.5.

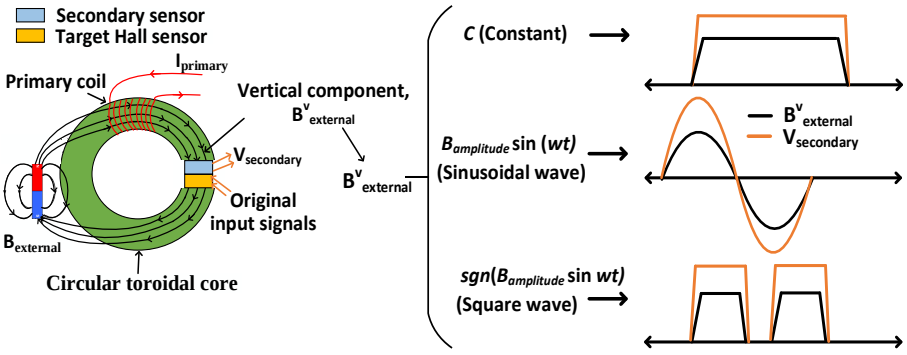


Figure 6.5: The $B^{v}_{external}$ can have constant, sinusoidal, or pulsating shapes. The generated voltage in the secondary sensor, $V_{secondary}$, has the same shape as the $B^{v}_{external}$.

■ ***4. Back calculating $B^{v}_{external}$ from the $V_{secondary}$:*** PreMSat needs to back calculate the magnitude of the $B^{v}_{external}$ to use it in a defense algorithm 5 for generating the $B_{internal}$. It is evident from Eqn. 6.1 that if I_{Bias}, d, and V_H are known, B can be calculated. As the secondary sensor provides the $V_{secondary}$, it is possible to calculate the $B^{v}_{external}$ from the $V_{secondary}$ using Eqn. 6.6. The Eqn. 6.6 is derived by adjusting the terms of Eqn. 6.1.

$$B^{v}_{external} = K(\frac{d \times V_{secondary}}{I_{Bias}}) = K_c \times V_{secondary} \tag{6.6}$$

where K_c is the *sensitivity* of a Hall sensor that includes all constant terms for simplification. The term K_c is provided by the manufacturer of the Hall sensor in its datasheet.

In the next section, we discuss how the different blocks of PreMSat uses the $V_{secondary}$ to generate the internal magnetic fields $B_{internal}$ to nullify the $B^{v}_{external}$.

Blocks of PreMSat

In this section, we discuss all the blocks and algorithms used in PreMSat (see Fig. 6.6).

1. Circular ferrite core: PreMSat uses a circular ferrite core to host the primary coil and secondary sensor (see Sections 6.5.2 and 6.5.3 for details).

2. Differential amplifier: The differential amplifier takes the $V_{secondary}$ as its input and removes the common-mode noises from it (see Section 6.3.2 for common-mode noise). The differential amplifier is implemented using an operational amplifier shown in Fig. 6.6. It has four resistors R1, R2, R3, and R4. When resistors R1 = R2 and R3 = R4, the output of the differential amplifier, denoted by $V^{diff}_{secondary}$, can be simplified to Eqn. 6.7.

$$V^{diff}_{secondary} = \frac{R_3}{R_1} V_{secondary} \tag{6.7}$$

The ratio R_3/R_1 in Eqn. 6.7 is set to 1 in PreMSat. Therefore, the differential amplifier only rejects the common-mode noises from the $V_{secondary}$ with a gain 1.

3. Analog-to-digital converter (ADC): The ADC samples the $V^{diff}_{secondary}$, digitizes it,

and provides the digitized value to the algorithm 5 running on a processor. To reduce the power consumption, the ADC is configured at a low sampling frequency (900 kHz) at normal operating conditions (i.e., when no attack happens). But the ADC uses a high sampling frequency when an attack happens (i.e., when there is a presence of $B^{v}_{external}$).

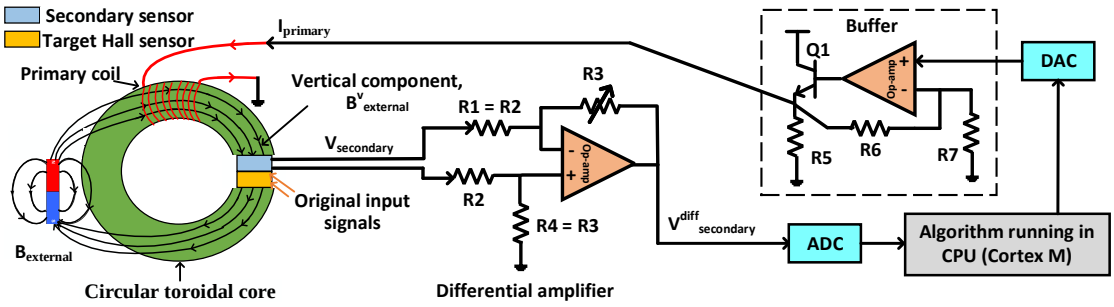


Figure 6.6: The different blocks of PreMSat.

4. PID controller: The output $V^{diff}_{secondary}$ from the ADC is given to a PID controller to generate a proper $B_{internal}$. The algorithm for the PID controller is designed in such a way that the generation of $B_{internal}$ should be fast enough so that it can nullify the $B^{v}_{external}$ in real-time. To meet the real-time requirement of PreMSat, the PID controller (see Section 6.3.5) is implemented in z-domain/discrete-time domain. There are three reasons behind implementing the PID controller in the z-domain instead of the s-domain/continuous-time domain. *First*, the z-domain takes ADC's sampling time in consideration that makes the PID controller more stable in the z-domain compared to the s-domain. *Second*, the PID controller in z-domain is highly deterministic. *Third*, most importantly, the PID controller in the z-domain has a much faster response time than the s-domain implementation. These properties are critical for real-time defense against the saturation attack.

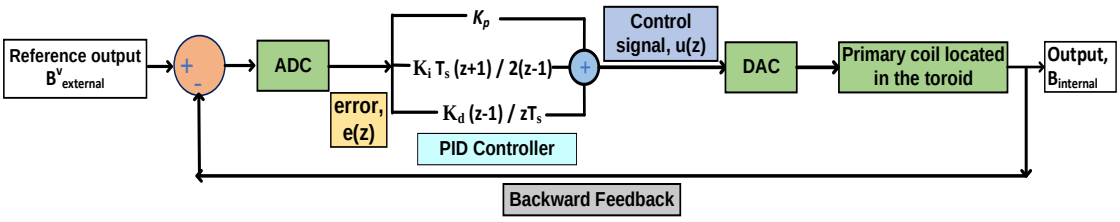


Figure 6.7: The PID controller tries to minimize the error between $B_{internal}$ and $B^{v}_{external}$.

The functional diagram of the PID controller is shown in Fig. 6.7. The variable $e(z)$

represents the error, which is the difference between the desired output $B^v_{external}$ and the actual output $B_{internal}$. Here, the $B^v_{external}$ is defined as the desired output because the PID controller should generate the $B_{internal}$ equal to the $B^v_{external}$. The $B^v_{external}$ is also known as the reference output. The error signal $e(z)$ is fed to the PID controller, and the controller computes both the derivative and the integral of this error signal.

The control signal $u(z)$ is fed to the primary coil, and the new output $B_{internal}$ is obtained. To obtain a continuous-time signal $B_{internal}$ from a discrete-time signal $u(z)$, a digital-to-analog converter (DAC) is used before the primary coil. The new output $B_{internal}$ is then fed back and compared to the reference $B^v_{external}$ to find the new error signal $e(z)$. The controller takes this new error and computes an update of the control signal $u(z)$ again. This process continues until the error $e(z)$ settles to a minimum value.

The transfer function of the PID controller in the z-domain is expressed in Eqn. 6.8.

$$\begin{aligned} \frac{u(z)}{e(z)} &= K_p + K_i\frac{T_s(z+1)}{2(z-1)} + K_d\frac{z-1}{zT_s} \\ => u(z) &= z^{-1}u(z) + ae(z) + bz^{-1}e(z) + cz^{-2}e(z) \end{aligned} \tag{6.8}$$

where $a = K_p + K_i\frac{T_s}{2} + \frac{K_d}{T_s}$, $b = -K_p + K_i\frac{T_s}{2} - \frac{2K_d}{T_s}$, $c = \frac{K_d}{T_s}$, and T_s is the sampling period of the ADC. Eqn. 6.8 can be expressed as a difference equation shown in Eqn. 6.9.

$$u(k) = u(k-1) + ae(k) + be(k-1) + ce(k-2) \tag{6.9}$$

where $u(k)$ and $e(k)$ are discrete-time domain equivalent of z-domain terms $u(z)$ and $e(z)$, respectively. Eqn. 6.9 is a recursive equation and has a second-order infinite-impulse-response (IIR) filter format. Therefore, the PID controller, used in PreMSat, is a second-order IIR filter that requires less memory space and computational time compared to the finite-impulse-

response (FIR) filters. This supports the idea that PreMSat provides real-time defense against the saturation attack on Hall sensors.

Please note that the secondary sensor only measures $B^{v}_{external}$ before generating $B_{internal}$. When PreMSat generates $B_{internal}$, the secondary sensor correlates more with the error $e(z)$ than $B^{v}_{external}$. Therefore, the PID controller minimizes $e(z)$ between $B^{v}_{external}$ and $B_{internal}$. Once $e(z)$ is close to zero, the secondary sensor starts to measure $B^{v}_{external}$ again.

■ ***Parameters of the PID controller:*** As the PID controller is a critical component of the real-time machine of PreMSat, few parameters that control the real-time properties of the PID controller are discussed here. These parameters are rise time, overshoot, settling time, and steady-state error. The values of K_p, K_i, K_d are tuned using MATLAB for a sampling frequency of 900 kHz to result in the lowest rise time, overshoot, settling time, and steady-state error. The values of these parameters are tabulated in Table 6.1.

Table 6.1: Parameters of the PID controller used in PreMSat

Response	Rise time	Overshoot	Settling time	Steady-state error
$K_p = 350$; $K_i = 300$; $K_d = 50$	8 μs	$< 1\%$	23 μs	$< 1\%$

Table 6.1 indicates that the settling time is 23 μs. In other words, it takes 23 μs to generate the $B_{internal}$ equal to the $B^{v}_{external}$ with less than 1% steady-state error. The less than 1% steady-state error is negligible compared to the large values of the $B^{v}_{external}$ required for the saturation attack (see Section 6.8.3).

■ ***Prevents strong or weak multiple signal shapes at the same time:*** The PID controller minimizes the error $e(z)$ while generating the $B_{internal}$ equal to the $B^{v}_{external}$, irrespective of the *strength* and *shapes* of the injected $B^{v}_{external}$. Even when multiple shapes, such as constant, sinusoidal, and pulsating fields, are injected at the same time, the vector summation of these fields will have a vertical component $B^{v}_{external}$ influencing the target Hall sensor (see Section 6.5.1, Fig. 6.2 and 6.5). The PID controller will nullify this $B^{v}_{external}$ in exactly the same way using the $B_{internal}$.

Moreover, a weak $B^{v}_{external}$, which can spoof the differential amplifier in its *linear region* (see Fig. 6.1), can also be nullified by the $B_{internal}$. Because a weak injected $B^{v}_{external}$ will also be picked up by the ferrite core, and PreMSat can nullify it using the $B_{internal}$.

In addition, PreMSat can nullify a injected $B^{v}_{external}$ even if the $B^{v}_{external}$ has the *same frequency* as the natural input signal B_{input}. Because the generated $B_{internal}$ by PreMSat can nullify the $B^{v}_{external}$ irrespective of its frequency, which is equal to the B_{input} or not.

5. Algorithm: The Algorithm 5, which handles the PID controller and controls the generation process of $B_{internal}$, is explained below.

Line 1-4: The ADC is configured initially to a low sampling frequency of 35 kHZ to ensure low power consumption by PreMSat. The ADC samples the $V^{diff}_{secondary}$ and algorithm 5 continuously tracks the $V^{diff}_{secondary}$ to check whether any attack happens.

Line 5-8: As $V^{diff}_{secondary}$ is coming from the secondary sensor, any change of $V^{diff}_{secondary}$ from a reference voltage indicates the presence of the $B^{v}_{external}$. The ADC changes its sampling frequency (i.e., $1/T_s$) to a higher value (i.e., 900 kHz) to provide the optimum a, b, and c in Eqns. 6.8 and 6.9. Then the $B^{v}_{external}$ is calculated using Eqn. 6.6 and the calculated $B^{v}_{external}$ is used to calculate the term $e(z)$.

Line 9-18: The PID controller is implemented using the difference equation from Eqn. 6.9. The PID controller generates $u(k)$, which is the discrete-time representation of $u(z)$, and converts the term $u(k)$ to an equivalent analog signal $I_{primary}$ using a DAC. The $I_{primary}$ is used to generate $B_{internal}$ using Eqns. 6.3, and 6.4. The error signal $e(z)$ is calculated and this process repeats until the term $e(z)$ settles within the 1% of the reference $B^{v}_{external}$. If the $e(z)$ does not settle down to 1% of $B^{v}_{external}$ within a certain time x, there is a possibility that $B_{internal}$ is not strong enough to nullify the $B^{v}_{external}$. This may cause the $V^{diff}_{secondary}$ to stuck in +ve/-ve saturation voltage. If this happens, PreMSat notifies the authority to fail-safe the system. The value of x is user defined. We use $x = 50$ μs.

Line 19-20: If no attack happens, the algorithm does not generate any $B_{internal}$ and keeps the Hall sensor running as it is.

Algorithm 5: Algorithm running on PreMSat.

Input: Data from ADC: $V^{diff}_{secondary}$
Output: Current signals to the primary coil: $I_{primary}$
1 Setup ADC ← (12 bits, sampling freq. = 35 kHz)
2 $B_{internal} \leftarrow 0$
3 **for** $t \leftarrow 1$ ***to*** ∞ **do**
4 Track $V^{diff}_{secondary}$
5 **if** $V^{diff}_{secondary}$ *changes* **then**
6 Setup ADC ← (12 bits, sampling freq. = 900 kHz)
7 Calculate $B^{v}_{external}$ from $V^{diff}_{secondary}$ using Eqn. 6.6
8 Calculate $e(z) \leftarrow B^{v}_{external}$ - $B_{internal}$
9 **for** *Continue until e(z) is within 1% of the* $B^{v}_{external}$ **do**
10 Generate $u(z)$ and $u(k)$ using Eqns. 6.8, and 6.9
11 Convert $u(k)$ to $I_{primary}$, where $I_{primary}$ is an analog version of $u(k)$, using DAC
12 Generate $B_{internal}$ from $I_{primary}$ using Eqns. 6.3, and 6.4
13 Calculate $e(z) \leftarrow B^{v}_{external}$ - $B_{internal}$
14 **if** $e(z)$ *does not settle down to 1% of the* $B^{v}_{external}$ *within x time* **then**
15 Possibility that $B_{internal}$ cannot nullify the $B^{v}_{external}$
16 Possibility that $V^{diff}_{secondary}$ is stuck in +ve/-ve saturation voltage
17 Notify authority to fail-safe the system and break from the loops
18 **end**
19 **end**
20 **Output** = $I_{primary}$
21 **end**
22 **else**
23 Do not generate $B_{internal}$ from $I_{primary}$ using Eqns. 6.3, and 6.4
24 **end**
25 **end**

6. Buffer: A digital-to-analog converter (DAC) converts the digital signal $u(k)$, which is the output of the PID controller, to an analog signal $I_{primary}$. As the DAC does not have the capability to provide high values of $I_{primary}$ to the primary coil, a buffer is used after the DAC to support high current to the primary coil (see Section 6.6.1). The primary coil, next, generates the $B_{internal}$ that is already explained in Section 6.5.2.

■ ***Security of PreMSat itself:*** An important question may arise what will happen if the attacker attacks the different components of the defense itself, such as the secondary sensor and differential amplifier. As the secondary sensor is placed in the ferrite core, the generated $B_{internal}$ will also nullify the injected $B^{v}_{external}$ to the secondary sensor in the same

way it prevents the saturation attack on the target Hall sensor. Therefore, the differential amplifier connected with the secondary sensor will not be saturated.

Generating the $B_{internal}$ in opposite direction to the $B^{v}_{external}$

As the $B^{v}_{external}$ is concentrated along the cross-section of the toroid, the $B_{internal}$ should also be provided along the same cross-section but in the opposite direction to nullify the $B^{v}_{external}$. *To provide the $B_{internal}$ in opposite polarity, the primary coil is connected in reverse polarity with the buffer chip.* Therefore, the PID controller does not need to spend any extra time to make the polarity of the $B_{internal}$ reverse to nullify the $B^{v}_{external}$.

6.6 Evaluation of PreMSat

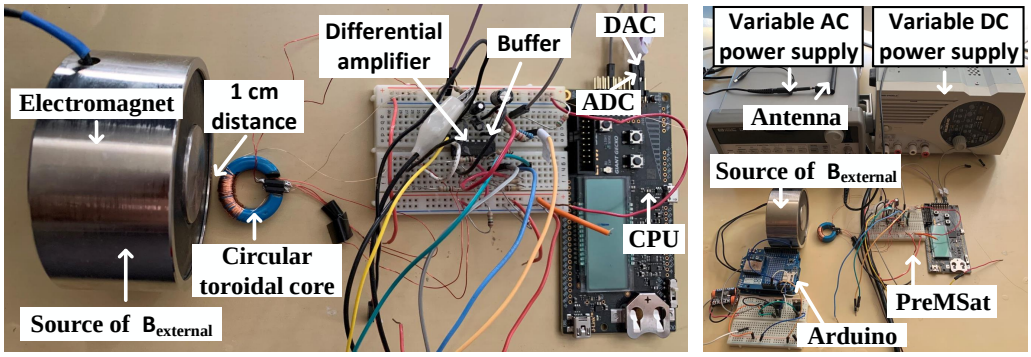


Figure 6.8: (left) The prototype. (right) The different instruments used in the testbed.

6.6.1 A prototype

A prototype of the proposed PreMSat is implemented using different discrete components, which is shown in Fig. 6.8 (left). A Hall sensor (part #ACS718) is used as the secondary sensor. The differential amplifier uses a low-power op-amp (i.e., part # TL084CN) with a high slew rate, low input bias and offset currents with a rise time of 0.05 μs and a unity-gain bandwidth of 3 MHz. The buffer uses an op-amp (part # TL084CN) in voltage-follower configuration with a high-power transistor $Q1$ (part # TO-220 [300]) connected at op-amp's output (see Fig. 6.6). The CPU of PreMSat is an EFM-32 Giant Gecko development board

from Silicon Labs [259] having a Cortex M-3 based 32-bit CPU with built-in ADCs and DACs. The EFM-32 has an ultra-low-power CPU with a 48 MHz clock. A low-cost soft ferrite, such as Mn-Zn ferrite is used as the material of the circular toroidal core [301]. Mn-Zn ferrite [302] has a high relative permeability (∼25000), and can support high frequency and low eddy current loss. Therefore, Mn-Zn ferrite can provide a low-resistive magnetic path to collect the externally injected field $B_{external}$ for PreMSat.

6.6.2 Testbed

We test ten different Hall sensors (Table 6.2 (a)) of all types, such as open/close loop, bipolar/unipolar sensors from four different manufacturers. As different Hall sensors measure different types of input signals, we use different sources to supply input signals to these different Hall sensors. We use a variable AC and DC source to supply current/voltage as original input signals to Hall sensors with serial no. 1-6 and use a magnet [253] to supply magnetic fields as input signals to Hall sensors with serial no. 7-10 in Table 6.2. The external fields $B_{external}$ are generated in two ways: an electromagnet (uxcell [256]) with a MOSFET (part #STP4NK80Z [293]) connected with an Arduino is used to generate constant, sinusoidal, and pulsating fields, and a function generator connected with a monopole antenna [260] is used to radiate high and low frequency electromagnetic interference (EMI) signals to attack Hall sensors. The testbed is shown in Fig. 6.8 (right).

6.6.3 PreMSat prevents the saturation attack

Here, we justify how PreMSat prevents the saturation attack on Hall sensors. We randomly pick ACS710KLATR-10BB from Table 6.2 as the target Hall sensor. A 7.5 A peak-to-peak AC current of 60 Hz frequency is given as an input signal to ACS710KLATR-10BB. Before any injection of external magnetic fields, the output of the target Hall sensor is shown in Fig. 6.9 (i), which shows an undistorted sinusoidal signal. An electromagnet with an MMF of $\sim$ 3600 A-t is used to inject different types of external magnetic fields $B_{external}$, such as

constant, sinusoidal, and square pulsating fields, to the target Hall sensor from 1 cm. We use 2 Hz as the frequency of injected sinusoidal and square pulsating fields as an example. Fig. 6.9 (ii) shows that the output of the target Hall sensor is driven to its saturation voltage (4.8 V) after the saturation attack resulting in a flattened output signal. As the output signal is flattened, any critical information cannot be recovered from the output signal in its saturation region. After integrating PreMSat with the target Hall sensor, the external magnetic fields $B_{external}$ cannot drive the output of the target Hall sensor to its saturation region. We can see from Fig. 6.9 (iii) that the output of the target Hall sensor remains unperturbed during the saturation attack.

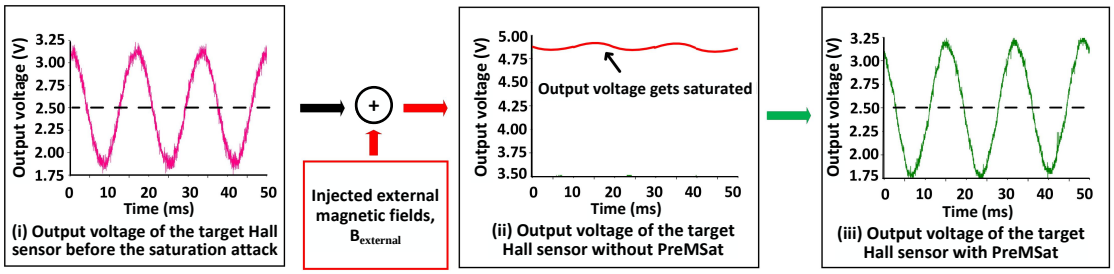


Figure 6.9: (i) The output signal of the target Hall sensor before the saturation attack. (ii) The output signal of the target Hall sensor gets saturated if PreMSat is not used. (iii) The output signal of the target Hall sensor does not change if PreMSat is used.

Performance metric: If we can prove that the output voltage of the target Hall sensor before the saturation attack is similar to the output voltage of the target Hall sensor after the saturation attack *with* PreMSat, we can claim that PreMSat is effective to prevent the saturation attack. To quantify the similarity, we calculate the *correlation coefficient (C)* [261] between signals in Fig. 6.9 (i) (i.e., before the saturation attack) and Fig. 6.9 (iii) (i.e., after the saturation attack with PreMSat). The value of correlation coefficient (C) is 0.97 for this case that is very close to unity. This indicates that the signal in Fig. 6.9 (i) (i.e., before the saturation attack) is *statistically the same* as the signal in Fig. 6.9 (iii) (i.e., after the saturation attack with PreMSat) in a point-by-point fashion. This proves that PreMSat can successfully prevent the saturation attack on a Hall sensor.

6.6.4 Testing PreMSat for different amplitudes of input signals

Table 6.2 (c) shows the average correlation coefficient C for different amplitude of input signals to ten different Hall sensors for a $B_{external}$ having an MMF of 3000 A-t. We vary the amplitude of the input signals within the entire input range (Table 6.2 (b)) of Hall sensors and calculate C for every input value and do an average of C for every sensor. The average of C is *greater* than 0.94 for every sensor when PreMSat is used (Table 6.2 (c)) compared to 0.1 when PreMSat is not used. This indicates that PreMSat works within the entire input range of every Hall sensor. We use 60 Hz as the frequency of input signals to Hall sensors with serial 1-6 and 10 Hz to Hall sensors with serial 7-10.

Table 6.2: Testing different Hall sensors in testbed for different amplitudes of input signals.

Sl.	Manufac. (a)	Part # (a)	Polarity/Loop (a)	Amplitude of input signal (b)	Avg. C (c)
1	Allegro	ACS718MATR-20B[262]	Bipolar/Open	1A, 5A, 10A, 15A, 20A	0.94
2	Allegro	ACS710KLATR-10BB [263]	Bipolar/Open	2A, 4A, 6A, 8A, 10A	0.95
3	Allegro	ACS715ELCTR-20A [264]	Unipolar/Open	1A, 5A, 10A, 15A, 20A	0.95
4	Allegro	ACS724LLCTR-10AU [265]	Unipolar/Open	2A, 4A, 6A, 8A, 10A	0.96
5	LEM	LTSR 6-NP [270]	Bipolar/Closed	1A, 2A, 3A, 4A, 5A	0.95
6	LEM	LV 25 P [271]	Bipolar/Closed	30V, 50V, 70V, 90V, 110V	0.96
7	Texas Ins	DRV5053OA [269]	Bipolar/Open	100G,200G,300G,400G,500G	0.97
8	Honeywell	SS49/SS19 [266]	Bipolar/Open	100G,200G,300G,400G,500G	0.97
9	Honeywell	SS39ET [267]	Bipolar/Open	100G,200G,300G,400G,500G	0.96
10	Honeywell	SS494B [268]	Bipolar/Open	100G,200G,300G,400G,500G	0.96

6.6.5 Testing PreMSat for different frequencies of input signals

Section 6.6.4 shows the performance of PreMSat for different amplitudes of the input signals. In this section, we vary the frequency of the input signals to different Hall sensors within their entire input range (Table 6.3 (a)) and calculate the correlation coefficient (C) for every case. We keep the amplitude of input signals fixed at 1 A/100 G/110 V. We find that the average value of C is greater than 0.94 for every sensor when PreMSat is used compared to 0.1 when PreMSat is not used (see Table 6.3 (b)). This indicates that PreMSat works within the entire input frequency range of every Hall sensor.

6.6.6 Testing PreMSat for different strength of injected $B_{external}$

At first, we find the strength of the external magnetic fields $B_{external}$ required to drive the Hall sensors to their saturation region (i.e., saturation attack) experimentally in our testbed. It is already mentioned in Section 6.5.2 that the strength of the magnetic field is quantified by the magneto-motive force (MMF). At first, we vary the MMF of the $B_{external}$ using an electromagnet and find that an MMF $>$ 3600 A-t can cause the saturation attack from 1 cm distance for all of the ten different Hall sensors. If the distance is $<$ 1 cm, an MMF less than 3600 A-t is required for the saturation attack.

Table 6.3: Testing different Hall sensors for different frequencies of input signals and different strengths of injected $B_{external}$.

Sl.	Part #	Frequency range of input signal (a)	Avg. C (b)	Strength of $B_{external}$ (c)	Avg. C (d)
1	ACS718MATR-20B	0 Hz–40 kHz	0.94	0 A-t–4200 A-t	0.95
2	ACS710KLATR-10BB	0 Hz–120 kHz	0.94	0 A-t–4200 A-t	0.94
3	ACS715ELCTR-20A	0 Hz–80 kHz	0.96	0 A-t–4200 A-t	0.97
4	ACS724LLCTR-10AU	0 Hz–120 kHz	0.96	0 A-t–4200 A-t	0.95
5	LTSR 6-NP	0 Hz–100 kHz	0.94	0 A-t–4200 A-t	0.94
6	LV 25 P	0 Hz–25 kHz	0.95	0 A-t–4200 A-t	0.95
7	DRV5053OA	0 Hz–20 Hz	0.96	0 A-t–4200 A-t	0.96
8	SS49/SS19	0 Hz–30 Hz	0.97	0 A-t–4200 A-t	0.97
9	SS39ET	0 Hz–40 Hz	0.95	0 A-t–4200 A-t	0.96
10	SS494B	0 Hz–30 Hz	0.96	0 A-t–4200 A-t	0.94

To test PreMSat, we vary the MMF from 0 A-t to 4200 A-t (i.e., $\sim$ 1.2x of 3600 A-t) at frequency zero with a step size of 200 A-t (see Table 6.3 (c)) and calculate C for every case for ten different Hall sensors. We do a total of $\sim$200 experiments in our testbed and find that the average value of C is greater than 0.94 for every sensor when PreMSat is used compared to 0.1 when PreMSat is not used (see Table 6.3 (d)). This proves that the prototype PreMSat can prevent the external magnetic fields $B_{external}$ having an MMF within 0 - 4200 A-t. *This indicates that PreMSat can prevent a weak MMF that can cause spoofing in the linear region as well as a strong MMF that can cause a saturation attack.*

6.6.7 Testing PreMSat for different frequencies of injected $B_{external}$

In Section 6.6.6, we vary the MMF of the $B_{external}$ from 0 A-t to 4200 A-t by keeping the frequency of the $B_{external}$ at zero. In this section, we vary the frequency of the $B_{external}$. As mentioned in Section 6.6.2, we use an electromagnet and a function generator connected with a mono-pole antenna to radiate high and low frequency $B_{external}$. We vary the frequency of the $B_{external}$ from 0 Hz to 30 kHz with a step size of 1 kHz (see Table 6.4 (a)) and calculate C for every case for ten different Hall sensors. We do an average of C for every Hall sensor in our testbed and find that the average value of C is greater than 0.94 for every sensor when PreMSat is used compared to 0.1 when PreMSat is not used (see Table 6.4 (b)). *This proves that the prototype PreMSat can prevent both low and high frequency external magnetic spoofing within a range of 0–30 kHz.*

6.6.8 Testing PreMSat for different distances of the magnetic source

In Sections 6.6.4, 6.6.5, 6.6.6, and 6.6.7, we place the source of $B_{external}$ 1 cm away from the target Hall sensor. In this section, we vary the distance of the magnetic-source (i.e., $B_{external}$) from the Hall sensor. We use an MMF of ∼3600 A-t for the $B_{external}$ and keep the frequency and amplitude of the input signals fixed at 60 Hz/10 Hz and 1 A/100 G/110 V, respectively. We vary the distance from 0 cm (very close) to 7 cm with an increment of 1 cm (Table 6.4 (c)) and calculate the average of C for every Hall sensor. The average value of C is greater than 0.94 for every case when PreMSat is used compared to 0.1 when PreMSat is not used (Table 6.4 (d)). This proves that PreMSat can prevent the saturation attack from a very close distance.

6.6.9 Comparing PreMSat with a ferromagnetic shield

We compare PreMSat with a ferromagnetic shield to prove PreMSat's effectiveness over a shield. There are specialized materials for magnetic shielding. The foremost of these is

MuMetal [272], which has high magnetic permeability and is used in industry. We use a strong electromagnet as a source of $B^{v}_{external}$ with an MMF of $\sim$ 3600 A-t. We use a box made of MuMetal as a shield and enclose the target Hall sensors with it. We keep the source of MMF 1 cm away outside of the shield and keep the Hall sensor 1 cm away inside of the shield. We vary the thickness of the shield and measure C for every thickness. We find that even an 1 inch thick shield cannot prevent a strong MMF of 3600 A-t (i.e, low value of C in Table 6.5 (c)). The reason behind this is that at strong magnetic fields, MuMetal gets saturated [272]. In saturation, the shielding property of the MuMetal is diminished [79], and sensors become vulnerable to external magnetic fields. Next, we only use PreMSat without a shield and find that PreMSat can maintain C close to unity (see Table 6.5 (d)). This proves the efficacy of PreMSat over a shield.

Table 6.4: Testing different Hall sensors for different frequencies and distances of $B_{external}$.

Sl.	Part #	Different frequencies of $B_{external}$ (a)	Avg. C (b)	Different distances of $B_{external}$ (c)	Avg. C (d)
1	ACS718MATR-20B	0 Hz - 30 kHz	0.94	0 cm - 7 cm	0.94
2	ACS710KLATR-10BB	0 Hz - 30 kHz	0.95	0 cm - 7 cm	0.97
3	ACS715ELCTR-20A	0 Hz - 30 kHz	0.96	0 cm - 7 cm	0.95
4	ACS724LLCTR-10AU	0 Hz - 30 kHz	0.97	0 cm - 7 cm	0.96
5	LTSR 6-NP	0 Hz - 30 kHz	0.94	0 cm - 7 cm	0.97
6	LV 25 P	0 Hz - 30 kHz	0.96	0 cm - 7 cm	0.94
7	DRV5053OA	0 Hz - 30 kHz	0.96	0 cm - 7 cm	0.95
8	SS49/SS19	0 Hz - 30 kHz	0.97	0 cm - 7 cm	0.96
9	SS39ET	0 Hz - 30 kHz	0.94	0 cm - 7 cm	0.94
10	SS494B	0 Hz - 30 kHz	0.96	0 cm - 7 cm	0.95

6.6.10 Comparing PreMSat with a high supply voltage

It may appear that increasing the supply voltage, denoted by V_{Supply}, of the differential amplifier located in a Hall sensor (see Fig. 6.1 and Section 6.3.3) may prevent the saturation attack because increasing the V_{Supply} will also increase the saturation voltage of a differential amplifier. To verify this claim, we vary the V_{Supply} of 10 Hall sensors within their acceptable ranges (see Table 6.5 (e)) and measure the maximum MMF, up to which every sensor can tolerate within their supply voltage ranges. We find that increasing the V_{Supply} may increase

the maximum MMF up to which Hall sensors can tolerate before going to saturation (see Table 6.5 (f)). *However, the maximum MMF, up to which they can tolerate, is still much smaller compared to PreMSat's capability of preventing an MMF of* $\sim$ *4200 A-t.* Please note that the output of DRV5053OA gets saturated at $\sim$ 2 V irrespective of the V_{Supply} variation within 2.5–38 V (see [269]). Therefore, DRV5053OA's maximum MMF spans within a small range of 1200 - 1400 A-t.

Table 6.5: Comparing PreMSat with a ferromagnetic shield and a high supply voltage.

Sl.	Part #	C (0.3 in. thick) (a)	C (0.5 in. thick) (b)	C (1 in. thick) (c)	C (PreMSat only)(d)	V_{Supply} range (e)	Max. MMF range (f)
1	ACS718...	0.20	0.26	0.37	0.94	4.5-5.5 V	1300-1900 A-t
2	ACS710...	0.14	0.21	0.31	0.96	3-5.5 V	1000-2200 A-t
3	ACS715...	0.19	0.27	0.36	0.97	4.5-5.5 V	1200-2000 A-t
4	ACS724...	0.27	0.35	0.39	0.95	4.5-5.5 V	1500-2700 A-t
5	LTSR 6-NP	0.28	0.39	0.41	0.97	4.7-5V	1700-1900 A-t
6	LV 25 P	0.13	0.28	0.38	0.94	12-15 V	1600-2000 A-t
7	DRV5053OA	0.33	0.39	0.42	0.96	2.5-38 V	1200-1400 A-t
8	SS49/SS19	0.10	0.21	0.32	0.96	4-10 V	1100-3600 A-t
9	SS39ET	0.15	0.29	0.35	0.95	2.7-6.5 V	1300-2800 A-t
10	SS494B	0.25	0.32	0.37	0.94	4.5-10.5V	1400-3400 A-t

6.6.11 Real-time defense against the saturation attack

Table 6.6: Timing analysis of PreMSat.

Task name	Block name	Clock freq.	Time
Remove common-mode noise	Differential amplifier	NA	0.25 μs
Sample the $V_{secondary}^{diff}$	ADC	11 MHz	1.2 μs
Generate the $B_{internal}$ (PID controller)	CPU	48 MHz	23 μs
Convert the $u(k)$ to $I_{primary}$	DAC	500 kHz	4 μs
provide the $B_{internal}$ in opposite polarity	Buffer	NA	0.34 μs
			28.79 μs (total)

Broadly speaking, PreMSat spends most of its time executing the following five tasks: (i) to remove common mode noise by the differential amplifier, (ii) to sample the $V_{secondary}^{diff}$ by the ADC, (iii) to generate the $B_{internal}$ and settle it (i.e., PID controller), (iv) to convert the $u(k)$ to $I_{primary}$ by the DAC, and (v) to provide the $B_{internal}$ in opposite polarity. In Table

6.6, we provide the amount of time required to execute each of these tasks along with the name of the block responsible for each task.

From Table 6.6, it is important to note that PreMSat can provide the $B_{internal}$ within 28.79 μs. This execution time is deterministic, and no additional latency/delay is involved in this process. Therefore, PreMSat can prevent the saturation attack within 28.79 μs that can be termed as a real-time defense against the saturation attack.

6.6.12 Feasible structure, and maintenance

To integrate the Hall and secondary sensors in a toroid, a small gap needs to be created in the cross-section of a toroid. *Industries are already using a similar structure where creating a small gap and winding a primary coil is similar to creating a transformer (Fig. 5-22 in [303]). Therefore, the structure is feasible in today's technology.* Moreover, regular maintenance is sufficient as PreMSat does not have parts that may be easily damaged.

6.6.13 Cost

The total cost of our prototype is ∼$14, comparable with the sensor cost (∼$2–$70). *However, the actual cost will be much less than ∼$14 in mass level production.*

6.6.14 Power consumption

The CPU runs at a low power, ADCs work at a low sampling frequency (i.e., 35 kHz), and the buffer, primary coil, and ferrite core consume low power when no attack happens. The overall power consumption when no attack happens is ∼ 5 mW. However, when an attack happens, the CPU and ADCs start working with high frequencies, and the primary coil generates fields $B_{internal}$. The primary coil is the main source of power consumption during an attack as it needs to generate a counter MMF. We use Mn-Zn soft-ferrite as the toroid, which has high relative permeability and magnetization. Therefore, the primary coil can

generate strong counter MMF (i.e., 0–4200 A-t), consuming power within 0–3 W. Moreover, Mn-Zn soft-ferrite has low resistance resulting in a low eddy-current loss. The overall power consumption when an attack happens is $\sim$ 5 mW–3 W.

6.7 Demonstration of preventing the saturation attack

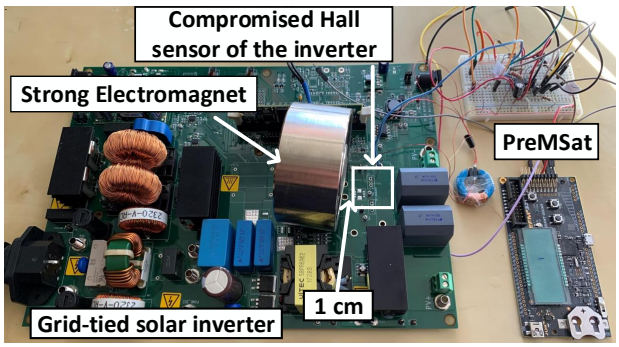


Figure 6.10: PreMSat prevents the saturation attack on the grid-tied solar inverter.

In this section, we demonstrate PreMSat's capability on a practical system — a grid-tied solar inverter. Grid-tied solar inverters are critical components in smart grids and are typically used as a power source in solar plants. Solar inverters have Hall sensors, which are typically used to measure AC and DC current or voltage [85]. Therefore, an attacker can target Hall sensors located in grid-tied inverters and inject external magnetic fields to drive Hall sensors to their saturation regions. This type of attack can shut down the inverter, and for a weak grid scenario, it can also cause a blackout in the region. To demonstrate that PreMSat can prevent the saturation attack, we use a 140 Watt inverter from Texas Instruments [275] in the testbed. This inverter has a Hall effect current sensor with a part # ACS712ELCTR-20A-T. At first, we inject constant, sinusoidal, and pulsating magnetic fields into the inverter with an MMF = 3600 A-t from a 1 cm distance. This causes a saturation attack on the Hall sensor located inside of the inverter. As a result, the inverter shuts down itself, causing a DoS attack on the inverter. To evaluate PreMSat, we integrate PreMSat with the Hall sensor and repeat the same experiment (Fig. 6.10). We notice that the inverter continues working

without any shutdown at this time. This proves that PreMSat can prevent the saturation attack on a practical system.

6.8 Limitations of PreMSat

6.8.1 Power consumption and usability of PreMSat

PreMSat consumes 0 - 3 W power while generating a strong internal field. The power consumption may be small for some applications, such as solar inverters and automotives, but may be substantial for few applications, such as proximity detection and position sensing. Moreover, the toroid has a 1.3 cm outer radius which is accommodable if designers would plan ahead to provide the space so that it would not affect the common use cases. However, few applications, such as Hall sensors in brush-less motors, may not fit the toroid. Moreover, the current prototype cannot be directly applicable to multiple axis Hall sensors. However, the idea of generating an internal magnetic field to nullify the external field would be applicable to a multiple-axis sensor with a change in the ferrite core's structure.

6.8.2 Non-zero settling time of the PID controller

It is already described in Section 6.5.3 that the PID controller has a non-zero settling time (i.e., 23 μs), which is also the main contributing factor to the total time (see Table 6.6) required to generate the $B_{internal}$. Therefore, if the attacker changes the injected magnetic fields $B_{external}$ within 23 μs, the timeliness of the defense will not be guaranteed. We have already finely tuned the values of K_p, K_i, K_d to obtain the lowest possible rise-time and settling time for the PID controller.

6.8.3 Non-zero steady-state error of the PID controller

The PID controller is tuned in such a way to have the lowest amount of steady-state error (i.e., <1%) possible for the problem at hand. In spite of the fine-tuning, the PID controller

has a non-zero steady-state error, which may add error to the $B_{internal}$ while nullifying the $B^{v}_{external}$. However, <1% error is negligible compared to the large values of the $B^{v}_{external}$ required for the saturation attack. For example, 3600 A-t is required for the saturation attack from a 1 cm distance, and 1% of 3600 A-t is only 36 A-t, which results in a negligible noise at the output of the Hall sensor.

6.8.4 Upper limit strength of the injected $B_{external}$

Our prototype can prevent an external magnetic field $B_{external}$ up to an MMF of 4200 A-t. The reason behind this is that our prototype cannot generate a $B_{internal}$ having an MMF more than 4200 A-t. The upper limit 4200 A-t is limited by the amount of power that the buffer can provide. The idea is supported by Eqn. 6.4, which says $B_{internal}$ depends on the $I_{primary}$. The $I_{primary}$ is provided by the buffer to the primary coil. The buffer used in the prototype has its maximum capacity that can support a $I_{primary}$, which can generate an MMF up to 4200 A-t. However, the limit can be theoretically increased from 4200 A-t to any higher value using a stronger buffer, causing a trade-off between cost and strength.

6.8.5 Upper limit frequency of the injected $B_{external}$

Our prototype can prevent the $B_{external}$ up to a frequency of ∼30 kHz. The upper limit 30 kHz results from the total time 28.79 μs required to generate the $B_{internal}$ (see Table 6.6). The reciprocal of 28.79 μs is 1/ 28.79 μs = ∼35 kHz. The prototype supports up to ∼30 kHz instead of 35 kHz because an additional time is spent to overcome the parasitic inductance/capacitance present in the primary coil. Note that the total time of 28.79 μs is obtained for our prototype using a clock frequency of 48 MHz. This time can be reduced further using a faster CPU having a clock frequency higher than 48 MHz.

6.9 Related work

To the best of our knowledge, no state-of-the-art work can prevent a saturation attack on Hall sensors. However, there is related work exists for other sensors that cannot be used to prevent a saturation attack on Hall sensors for the following reasons.

Barua et al. [157] proposed an in-sensor defense for Hall sensors. However, it does not work for a saturation attack. Trippel et al. [51] proposed randomized and 180^0 out-of-phase sampling to provide defenses against an acoustic signal injection into MEMS accelerometers. They sample at random times with 180^0 out-of-phase within the resonant frequency period to nullify the spoofing signals. They are not suitable for saturation attacks because: (i) They can only filter out a forged signal, which has a frequency equal to the resonant frequency of the MEMS sensor. Therefore, they do not work other than a specific resonant frequency, for example, any attack frequency. (ii) They do not work against a DC/constant forged signal because randomized sampling cannot filter out a DC signal. (iii) They do not work when the sensor output is flattened.

Cheng et al. [248] and Alexander [249] from Allegro Microsys. used differential sensing by using two sensing elements to cancel out common-mode attack signals. However, it does not prevent a sensor output from getting flattened during a saturation attack.

Kune et al. [42] used an adaptive filter to mitigate EMIs in microphones. An adaptive filter estimates EMIs first and then subtracts the estimated EMIs from the original signal to recover the original signal. This technique cannot estimate any attack signal if the sensor output is flattened because of the saturation attack.

Zhang et al. [47] used a Support Vector Machine (SVM), and Roy et al. [250] used a non-linearity tracing classifier to filter inaudible ultrasonic voice commands from MEMS microphones. They have the following limitations: (i) They will work only for spoofing

signals located in ultrasonic frequency band (> 20kHz), which has a clear separation from the audible voice signals (< 20 kHz). As the spoofing signal may share the same band as the original signal in Hall sensors, these defenses don't work for Hall sensors. (ii) They don't work if the sensor output is flattened because of the saturation attack.

Shoukry et al. [252] proposed PyCRA to detect spoofing attempts by turning off the active sensor's transmitter at random instants such that the attacker cannot react to the sudden changes. However, PyCRA only works for active sensors; it is not applicable for passive sensors. Moreover, PyCRA only detects intentional spoofing but cannot prevent it.

Table 6.7: Comparing PreMSat with other defenses.

Properties	Recent works [42, 47, 51, 248–250]	PreMSat
Saturation attack	✗	✓
Spoofing in linear region	few work for a specific resonant frequency or a frequency other than the natural signal's frequency	works for any frequency within 0 - 30 kHz
0 A-t $\leq$ MMF $\leq$ 4200 A-t	✗	✓
Constant/DC, sinusoidal, and square magnetic spoofing	✗	✓
0 $\leq$ frequency $\leq$ 30 kHz	✗	✓
External signal has the same frequency as the natural input signal	✗	✓
Power consumption	mW range	5mW - 3 W
Overhead	extra parts for adaptive filter, etc.	ferrite core

Wang et al. [304] designed a state graph-based approach to detect state corruption due to intentional spoofing. Again, Shoukry et al. [279] used the satisfiability modulo theory (SMT) to recover from corrupted states. The main drawback of the above-mentioned state recovery techniques as a defense is that they do not work against time-varying spoofing signals, which may create oscillations between corrupted and recovered states of the system controller. The oscillations between corrupted and recovered states may eventually compromise the integrity and availability [28, 30] of the system under attack. Moreover, they cannot prevent saturation attacks on any sensor.

In contrast, PreMSat uses a PID controller to generate an internal magnetic field to nullify the injected external field. The PID controller can nullify the injected external field even

if the injected external field (i) is constant, sinusoidal or square magnetic fields, (ii) has zero/DC frequency, (iii) has the same frequency as the natural signal being measured, and (iv) can cause a saturation attack. Moreover, PreMSat can work against $\sim$ 4200 A-t and within 0–30 kHz. However, PreMSat achieves these advantages with high power and physical overhead compared to recent works (see Table 6.7 for a summary).

6.10 Summary

PreMSat is the first of its kind in literature and industry that can prevent the saturation attack satisfactorily on passive Hall sensors. PreMSat can prevent the saturation attack originating from different types, such as constant, sinusoidal, and pulsating magnetic fields, in hard real-time. Moreover, PreMSat can also prevent weak magnetic spoofing in the linear region of the differential amplifier. PreMSat integrates a low resistive magnetic path to collect the external magnetic fields injected by the attacker and utilizes a finely tuned PID controller to nullify the external fields. The PID controller is tuned in such a way that it has minimum settling time and steady-state error. This helps to keep the existing data processing speed of the connected system undisturbed. We have presented a prototype of PreMSat, which can nullify external fields up to $\sim$ 4200 A-t. We have done an extensive analysis of PreMSat through more than 300 experiments on ten different Hall sensors from four different manufacturers and proved its efficacy against the saturation attack. However, PreMSat has high power cost and overhead that might not be suitable for all applications. Moreover, we have demonstrated the efficacy of PreMSat on a practical system — a grid-tied solar inverter. The demonstration proves that PreMSat can prevent the DoS attack on a practical system by nullifying the saturation attack on a Hall sensor. *Finally, we believe that the necessity of developing a similar defense like ours is going to be increased in the near future for other sensors when sensors will pervade our lives.*

Chapter 7

Magnetic Spectrum Hopping for Securing Voltage and Current Magnetic Sensors

7.1 Abstract

Voltage and current magnetic sensors (VCMSs) are pervasive in safety-critical systems. They use a magnetic field as a transduction medium to sense the input signal. Therefore, if an attacker manipulates the magnetic transduction medium of this sensor by using an intentional EMI or external magnetic fields, no amount of security mechanism after the fact can help. Fortunately, our work provides a defense against this form of physical attack. The core idea of our defense is to shift the frequency spectrum of the magnetic field, which is used as the transduction medium of the sensor, to another spectrum unknown to an attacker. In addition, the frequency spectrum, which carries the magnetic field in the transduction medium, is varied in a pseudo-random fashion so that the attacker will not be able to track it to inject any EMI into it. Even a sweeping attacker, who can vary the EMI's frequency, cannot bypass our defense because of *the check and select approach* of our defense. As the magnetic field's spectrum in the transduction medium of the sensor hops in a different spectrum, the defense is named as Magnetic Spectrum Hopping (MagHop). While prior works fail to prevent an EMI, which has the same frequency as the input signal, MagHop

is equipped to handle this limitation of the prior works. Moreover, a low-power, real-time coherent prototype of MagHop is designed that is evaluated with a real-world application: a grid-tied inverter. Finally, we thoroughly evaluate MagHop on ten different sensors from six different manufacturers to prove its robustness against the EMI or external magnetic field injection attack on VCMSs. The findings in this chapter have been published in [305].

7.2 Introduction

A voltage or current signal is the most common signal in critical systems. Almost all analog signals from different modalities, such as electrical energy, acoustic, and vibration, are converted into voltage or current signals for further processing. Therefore, voltage and current sensors are abundant in safety-critical systems, ranging from computers to industrial controllers and automobiles to aircraft [242, 282, 306].

Among different voltage and current sensors present in the industry, Faraday's law and Hall effect based voltage and current sensors [61] are the widely used ones because of their galvanic isolation compared to the resistive drop/divider approach. Both sensors use a proportional magnetic field to sense the voltage and current signal and output a scaled-down signal. As these sensors use magnetic energy as a transduction medium, they are named voltage-current *magnetic* sensors (VCMSs) in our paper. Though researchers devote much of their efforts to improving their performance, their security is still neglected to date. And prior works [42, 48, 307–309] show that they are still not secured against attack signals, such as EMIs and magnetic fields. *The E-field and B-field of an EMI induces noise like voltage in VCMSs, and a pure magnetic field can perturb the magnetic transduction medium of VCMSs.*

Note that prior works [42, 47, 51, 161, 310] provide filtering and sampling-based defenses against unwanted attack signals. The main drawbacks of them are: *(i) they don't contain the injected EMIs/magnetic attack signals having the same frequency as the legitimate input*

voltage or current signal being measured, (ii) they cannot prevent an attacker, who can sweep the frequency of the injected EMIs, and (iii) they can't separate the injected magnetic field from the actual magnetic field, which is used as the transduction medium of VCMSs.

Therefore, this paper proposes a novel defense to solve the above limitations. The core idea of the defense is that it shifts the frequency spectrum of the magnetic field, which is the transduction medium of VCMSs, to a different spectrum. The spectrum is varied in a pseudo-random fashion, so the attacker cannot inject EMIs into the unknown frequency spectrum. As the frequency spectrum of the magnetic field hops from one frequency to another within the sensor bandwidth, the defense is named as Magnetic Spectrum Hopping (MagHop).

As the magnetic field's frequency spectrum in the transduction medium of VCMSs is shifted to an unknown frequency, the proportional input signal and the corresponding output signal of the sensor are also shifted to the same spectrum, which is also unknown to the attacker. The pseudo-random variation of the spectrum is only known to VCMSs. Therefore, an attacker, who uses EMIs to inject E-field or B-field into VCMSs, cannot interfere with an unknown frequency spectrum of the magnetic field in the transduction stage. Moreover, an attacker, who also targets the conductors connected with the input and output of VCMSs, cannot inject any EMIs into the input and output signal, because the frequency spectrum of the input and output signal is also shifted to an unknown spectrum. Even a strong attacker, who can sweep the frequency of the EMIs, cannot interfere with the frequency spectrum of the magnetic field in the transduction stage. The reason behind this is that the defense always checks whether the spectrum is attacked by the EMIs before switching to that spectrum. Last but not least, the defense syncs up all the fragmented pseudo-random frequency spectrum at the output, so that the signal being measured is always coherent. *Hence, the defense never hampers the real-time behavior of the sensor.* We believe that our idea and implementation details will be beneficial to building the next generation of *secured*

VCMSs having robust immunity to both EMIs and magnetic fields.

Contributions: Our main technical contributions are:

1. We introduce a methodology to pseudo-randomly vary the frequency spectrum of the magnetic field used as the transduction medium of voltage and current magnetic sensors.

2. We show the effectiveness of MagHop against the injected EMIs or magnetic fields through experiments on ten different VCMSs from six different manufacturers. We experiment with both types, Faraday's law and Hall effect based VCMSs.

3. We do a low-power implementation of MagHop on an FPGA and Cortex-M processor and prove its real-time efficacy on a practical grid-tied solar inverter system.

7.3 Background

7.3.1 Voltage & current magnetic sensor (VCMS)

According to Ampere's law [286] of electromagnetism, a current signal has magnetic fields *associated* with it. VCMSs use the *associated magnetic field* to measure the voltage and current. Broadly speaking, the associated magnetic fields are used in two different techniques in VCMSs. The first one is related to Faraday's law and the second one is related to the Hall effect. These two techniques are briefly explained below.

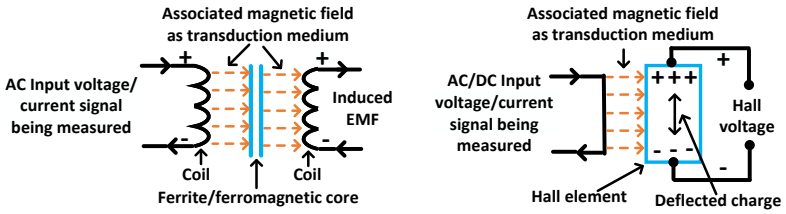


Figure 7.1: (Left) Faraday's law and (Right) Hall effect based VCMS.

Faraday's law based VCMSs: A time-varying (i.e., AC) voltage or current signal has a time-varying magnetic field associated with it. Faraday's law of induction [289] states that

a time-varying magnetic field can induce a proportional time-varying electromotive force (EMF) in a coil. Therefore, Faraday's law based VCMSs use the induced EMF to measure the time-varying voltage or current. A current transformer (CT), potential transformer (PT), audio transformer, and Rogowski coil are examples of this type (see Fig. 7.1 (Left)). This type uses a *ferrite/ferromagnetic* core [311] to host the coil, where the EMF is induced. As the ferrite core has high bandwidth (i.e., $\sim$ kHz), these sensors can measure high frequency signals. The high bandwidth of the ferrite/ferromagnetic core enables *magnetic spectrum hopping* technique in our defense.

Hall effect based VCMSs: A Hall effect based VCMS has a Hall element (i.e., p-type semiconductor) (see Fig. 7.1 (Right)). When the Hall element is placed in the *magnetic field associated with a voltage or current signal*, the moving charge present inside of the Hall element gets deflected across it by obeying the Lorentz law [312]. This deflection across the Hall element generates a voltage known as Hall voltage, which is proportional to the magnetic fields associated with voltage or current signals. Either a constant or a time-varying associated magnetic field can deflect the moving charge of the Hall element. Therefore, Hall effect based VCMSs can measure both AC and DC signals. Similar to the ferrite/ferromagnetic core, the Hall element has high bandwidth (i.e., $\sim$ kHz) that enables *magnetic spectrum hopping* technique in our defense.

7.3.2 Importance and security consequences

VCMSs have good linearity, high accuracy, and faster response with galvanic isolation and are abundant in safety-critical systems. However, they are still not secured because these sensors cannot differentiate between the original associated magnetic field and the fake magnetic field injected by an attacker in the form of an EMI. The injected fake signal can be propagated to connected systems, resulting in a denial-of-service (DoS) attack on the system. A similar incident is found in the literature where an opportunistic attacker injects fake magnetic

fields to current magnetic sensors in a micro-grid, causing a blackout in the power system [245]. Therefore, a robust defense is much needed to make VCMSs secure against intentional EMI/magnetic field injection.

7.4 Threat Model

We first explain the following four components of the threat model (see Fig. 7.2) against which our defense works.

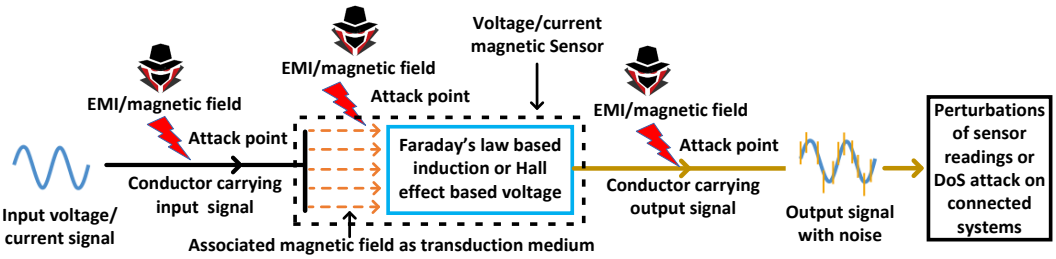


Figure 7.2: The threat model for the proposed defense.

1. Attacker's target: The attacker uses electromagnetic energy from a distance to *noninvasively* inject malicious signals into VCMSs. In this way, the attacker can inject false data into VCMSs that can eventually propagate to connected systems, resulting in an erroneous state or DoS attack on the system. The attacker may not get a long time to *modify or observe* the target VCMS like a lunch-time attack [254] or is not allowed to physically alter any parts of VCMSs. Attackers can target two attack points: (i) magnetic transduction medium of VCMSs and (ii) connected conductors which behave as antennas.

2. Attack signal's bandwidth: *The attacker can use EMIs with single or multiple tones to inject false data into VCMSs.* Moreover, the injected EMI can have the same bandwidth as the original signal, making it difficult to differentiate between injected EMIs and original signals. The attacker can vary the injected EMI's frequency in different ways. For example, a *static attacker* can inject EMIs with static frequency for a long time. A *sweeping attacker* can vary the EMI's frequency in a random or particular order. A *responsive attacker* at

first can sense the frequency of the ongoing voltage or current signal and next use the same frequency EMI to attack the sensors.

3. Attack tool: The attacker can use an electromagnet to inject only B-field or an antenna connected with an oscillating signal to inject both E-field and B-field into VCMSs.

4. Penetrating the sensor shield: VCMSs may or may not be placed inside a shield [257] depending on their applications. In the presence of a shield, the injected EMI/magnetic field should be strong enough to penetrate the shield first.

7.5 Modeling and Evaluating the Attack

Here, we mathematically model the consequences of our attack model and evaluate it through experiments.

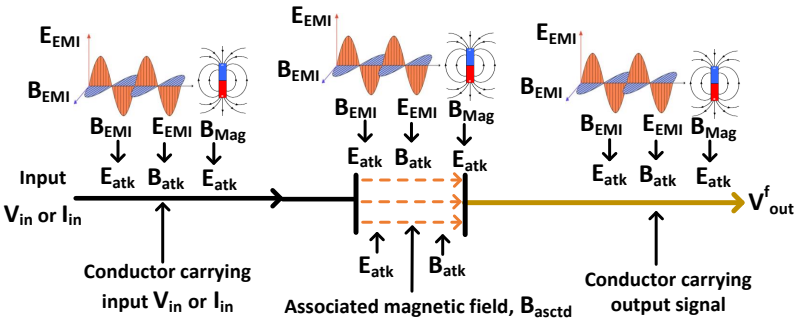


Figure 7.3: The surrounding electromagnetic field of the conductor and the associated magnetic field are perturbed by the injected EMI/magnetic field.

7.5.1 Mathematical modeling

Let us denote the input voltage and current signal being measured by the VCMS as V_{in} and I_{in}, respectively. The magnetic field density associated with the V_{in} or I_{in} is denoted by B_{asctd}. Ampere's law states that $B_{asctd} \propto \{V_{in}$ or $I_{in}\}$ (see Eqn. 7.1). The associated field B_{asctd} is induced in a ferrite/ferromagnetic core (i.e., Faraday's law) or sensed by the

Hall element (i.e., Hall effect), resulting in a sensor output voltage. Let us denote the sensor output voltage *before an attack* by V_{out}, which is modeled by Eqn. 7.2.

$$B_{asctd} = k_1 V_{in} \text{ or } k_1 I_{in}; \quad \text{Ampere's law.} \tag{7.1}$$

$$V_{out} = \begin{cases} k_2 \frac{\delta B_{asctd}}{\delta t} = k_6\{V_{in} \text{ or } I_{in}\}; & \text{Faraday's law,} \\ k_3 B_{asctd} = k_6\{V_{in} \text{ or } I_{in}\}; & \text{Hall effect.} \end{cases} \tag{7.2}$$

where k_1, k_2, k_3, and k_6 are proportionality constants and depend on the properties of ferrite core and Hall element.

Fig. 7.3 illustrates that the B_{asctd} is the only medium for information transfer from the input stage (i.e., V_{in} or I_{in}) to the output stage (i.e., V_{out}) of VCMSs and there is no authentication or encryption in this magnetic medium. Therefore, an attacker can simply inject an external magnetic field into the magnetic medium to perturb the input signal V_{in} or I_{in}.

The attacker can use an EMI or electromagnet as the *attack-source*. Let us denote the electric and magnetic fields in EMIs by E_{EMI} and B_{EMI}, respectively. Let us denote the magnetic field from an electromagnet by B_{Mag}. The terms E_{EMI} and B_{EMI} from an EMI are always time-varying. The field B_{Mag} from an electromagnet, can be *static or time-varying* depending upon how the power is given to the electromagnet.

From Maxwell's equations [286], the attack electric field E_{EMI} generates a magnetic field B_{atk} (see Eqn. 7.3), and the attack magnetic field B_{EMI} or B_{Mag} generate an electric field E_{atk} (see Eqn. 7.4). The generated B_{atk} and E_{atk} are added to Eqn. 7.2, resulting in a false output voltage V_{out}^{f} (see Eqn. 7.5).

$$\Delta \times B_{atk} = \quad k_4 \frac{\delta E_{EMI}}{\delta t}; \quad \text{Maxwell's eqn.} \tag{7.3}$$

$$\Delta \times E_{atk} = -k_5 (\frac{\delta B_{EMI}}{\delta t} or \frac{\delta B_{Mag}}{\delta t}); \quad \text{Maxwell's eqn.} \tag{7.4}$$

$$V_{out}^{f} = \begin{cases} k_2 \frac{\delta (B_{asctd} + B_{atk})}{\delta t} - E_{atk}\, \delta s; \text{ Faraday's law,} \\ k_3 (B_{asctd} + B_{atk}) - E_{atk}\, \delta s; \text{ Hall effect.} \end{cases} \tag{7.5}$$

where k_4 and k_5 are proportionality constants, and δs is the direction along which δE_{atk} changes. The R.H.S of Eqn. 7.5 indicates that the fake output voltage V_{out}^{f} from the target sensor has the cumulative effect of the injected EMI or magnetic field by the attacker.

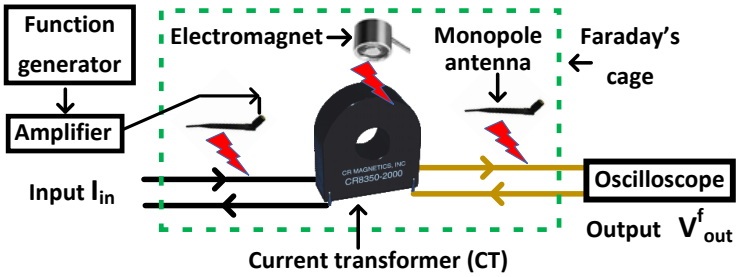


Figure 7.4: Experimental setup for the attack model evaluation.

7.5.2 Evaluating the attack model

We evaluate the attack model from Eqn. 7.5 with the experimental setup shown in Fig. 7.4. A Faraday's law based CT (part# CR8348-2000 [313]) is used as the target sensor. An off-the-shelf electromagnet (part # Grove [314]) having a strength of 1000 Gauss is used to inject a magnetic field into the sensor from a 1 cm distance. Moreover, an antenna is used to inject 1 kHz EMI into the conductor connected to the sensor from a 1 cm distance. The antenna has a 3 dB gain and 1 W input power from an amplifier and signal generator. The

experimental setup is placed inside a Faraday's cage [315] to avoid external noise.

Results: A 60 Hz and 150 mV peak signal is given as input to the CT (see Fig. 7.5). The 1 kHz EMI signal injects noise-like perturbations into the sensor corrupting its measurement. The 1000 Gauss static magnetic field from the electromagnet adds a DC offset to the sensor's output. This shifts the V_{out} by 100 mV upward. The fake output voltage after an attack (i.e., V_{out}^{f}) has the accumulated impacts of B_{EMI}, E_{EMI}, and B_{Mag} on the CT, supporting our attack model.

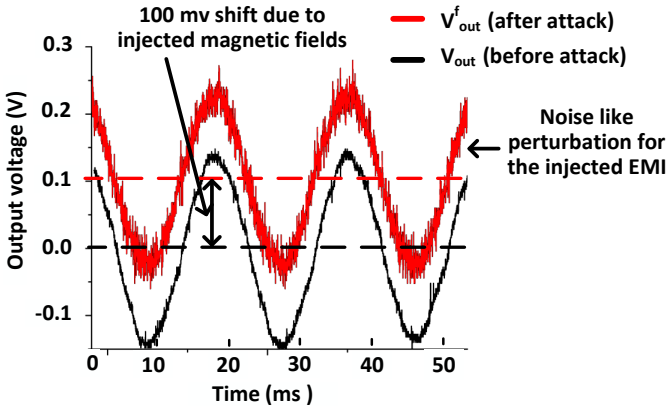


Figure 7.5: EMI/magnetic fields injected into the CT and connected conductor.

7.6 Motivation and Defense Outline

7.6.1 Motivation

Let us first denote the input voltage or current signal (i.e., V_{in} *or* I_{in}) being measured has a bandwidth BW_{in}. If an injected EMI has a bandwidth BW_{atk}, Eqns. 7.3 and 7.4 indicate that the electric and magnetic field attack components E_{atk} and B_{atk} also have the same bandwidth BW_{atk}.

A *naive* defense could be to use adaptive or other different filters [42, 47, 51, 161, 310] to remove the attack bandwidth BW_{atk}. This strategy fails in the following two scenarios:

- First, if the attack frequency BW_{atk} overlaps with the input signal's frequency BW_{in}, a

filter-based defense may filter out BW_{in} while filtering BW_{atk}, resulting in a distortion.

• Second, if a sweeping/responsive attacker sweeps the frequency of the injected EMI, the filter-based defenses may not be able to track the *sweeping* frequency. Therefore, they may not be successful against a sweeping/responsive attacker.

In the next section, we discuss why our proposed defense is strong enough to solve the above two major limitations of the recent work [42, 47, 51, 161, 310].

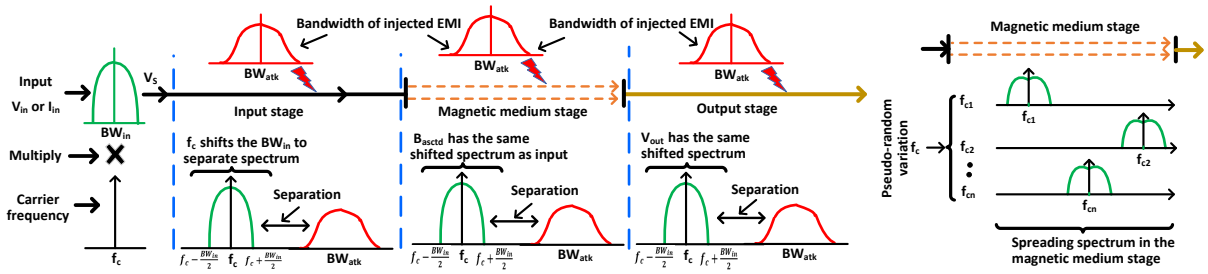


Figure 7.6: (Left) The bandwidth BW_{in} of an input signal is shifted to a separate spectrum so that it does not interfere with the BW_{atk} of injected EMIs. (Right) The pseudo-random hopping of B_{asctd} causes a spread spectrum in the magnetic medium stage.

7.6.2 Defense outline

To ease the explanation of the defense, we divide the pipeline of VCMS into the following three stages:

(i) **Input stage:** where the voltage or current signal (V_{in} or I_{in}) is given as input to measure the signal.

(ii) **Magnetic medium stage:** it is the transduction stage where input signal V_{in} or I_{in} is transferred to the output stage via an associated magnetic field B_{asctd}.

(iii) **Output stage:** where a scaled down output voltage V_{out} is generated proportional to the input V_{in} or I_{in}.

The attack model in Eqn. 7.5 holds for the injected EMI/magnetic fields in all stages of the

sensor.

• **Attack modalities:** Attack components in Eqn. 7.5 have two different modalities: B_{atk} is the attack magnetic field and E_{atk} is the attack electric field. Our proposed defense adopts the following strategies to work against the E_{atk} and B_{atk} that are also illustrated in Fig. 7.6.

Defense against the attack electric field E_{atk}

To separate the E_{atk} from the input V_{in} or I_{in} signal being measured, our defense MagHop uses an unknown frequency to separate the input V_{in} or I_{in} signal from the E_{atk}. The unknown frequency is defined as the carrier frequency, f_c. When a carrier frequency carries a voltage or current signal, the bandwidth BW_{in} of the input signal is shifted to the carrier frequency f_c. If we consider a voltage or current signal, V_{in} or $I_{in} = A_{in}cos(2\pi BW_{in}t)$ with bandwidth BW_{in} and a carrier signal $c(t) = A_c cos(2\pi f_c t)$, the frequency shifting process is expressed by multiplication as follows:

$$\begin{aligned} V_s &= A_{in}cos(2\pi BW_{in}t) \times A_c cos(2\pi f_c t) \\ &= A_{in} + A_c\{2\pi(f_c + BW_{in})t\} \end{aligned} \tag{7.6}$$

where A_{in} and A_c are the amplitudes of the voltage or current signal being measured and carrier signal, respectively, and V_s is the signal after the frequency shift to $f_c + BW_{in}$.

The shifting process shifts the BW_{in} in such a way that $f_c \pm BW_{in}/2$ does not co-inside with BW_{atk}. Therefore, after frequency shifting, a filter can separate the attack frequency BW_{atk} from the input V_{in} or I_{in} signal. In this way, the E_{atk} can be removed from the input voltage or current signal.

Defense against the B_{atk}

Eqn. 7.1 implies that if V_{in} or I_{in} has a bandwidth BW_{in}, the B_{asctd} should also have the same bandwidth BW_{in}. Therefore, when a carrier frequency shifts the BW_{in} to $f_c \pm BW_{in}/2$, the frequency spectrum of the B_{asctd} is also shifted to $f_c \pm BW_{in}/2$. Because of this shifting, the attack magnetic field B_{atk} cannot interfere with the B_{asctd}. The frequency shifting of the B_{asctd} takes place in the *magnetic medium stage* of the sensor (Fig. 7.6 (Left)).

Mathematical intuition

Eqn. 7.5 can be written after an injection of EMI/magnetic fields to VCMSs as:

$$V_{out}^{f} = \begin{cases} V_{out} + k_2 \frac{B_{atk}}{\delta t} - E_{atk}\,\delta s; \text{ Faraday's law,} \\ V_{out} + k_3 B_{atk} - E_{atk}\,\delta s; \text{ Hall effect.} \end{cases} \tag{7.7}$$

where V_{out} is the output voltage of VCMSs before an attack. MagHop uses a carrier frequency f_c to shift the frequency spectrum of V_{out}. Therefore, the V_{out} will have a different spectrum than the attack signal (i.e., E_{atk} and B_{atk}). Therefore, a filter can separate the V_{out} from the attack signals.

Choice of the carrier frequency

Please note that the success of MagHop relies on how we choose the carrier frequency f_c. After shifting the spectrum of the input voltage or current signal and its associated magnetic fields, we must ensure that it does not overlap with the bandwidth BW_{atk} of the injected EMIs. This technique has a few pitfalls.

Pitfall 1 - Sweeping and responsive attacker: One solution could be, at first, we need to calculate the bandwidth BW_{atk} and use BW_{atk} to calculate the correct carrier frequency

f_c. This strategy could work against a static attacker, who keeps the bandwidth BW_{atk} static. However, it may not work against a sweeping/responsive attacker because the BW_{atk} must be calculated whenever the sweeping attacker changes it. The calculation of BW_{atk} requires Fast Fourier transformation (FFT) [316], which is computationally expensive, taking a finite amount of computation time. Therefore, if the sweeping/responsive attacker sweeps the bandwidth BW_{atk}, within the BW_{atk} computation time, the defense may not work.

Pitfall 2 - Hampering real-time sensor measurement: In addition, while waiting for the FFT computation, the input voltage or current signal cannot be transferred from the *input stage* to the *output stage* of VCMSs due to the lack of a correct carrier frequency f_c. Therefore, the sensor needs to wait, and this wait time may hamper the real-time measurement.

Solution: Pitfalls 1 and 2 imply that MagHop should avoid measuring the BW_{atk} of the injected EMI/magnetic fields to avoid FFT calculation. Therefore, we propose to pseudo-randomly vary the carrier frequency f_c to avoid these pitfalls.

Eqn. 7.6 indicates that if the carrier frequency varies in a pseudo-random fashion, the frequency spectrum BW_{in} of the input signal V_{in} or I_{in} also varies in the same pseudo-random fashion. Therefore, a static attacker cannot interfere with the input signal's frequency spectrum because it is not static anymore. Moreover, a sweeping/responsive attacker cannot also interfere because he/she does not know the pseudo-random sequence of the carrier frequency variation.

The probability of overlapping with the bandwidth BW_{atk} of injected EMIs/magnetic fields depends upon the number of frequency channels among which the carrier frequency hops from one another. For example, if there is n carrier frequencies: $\{f_{c1}, f_{c2}, .., f_{cn}\}$, the probability of overlapping is $1/n$. Therefore, for a large n, the probability of overlapping with the bandwidth BW_{atk} is reduced. In our design, we have used $n = 255$ channels, among which

f_c can hop in a pseudo-random fashion. Therefore, it has only a $1/255 = 0.39\%$ chance to overlap with the bandwidth BW_{atk} of injected EMIs.

Pseudo-random variation of f_c is not enough: Though the pseudo-random variation of the f_c gives a very low (i.e., 0.39%) chance of overlapping, however, the attacker still has this low chance to perturb the input signal's frequency. Specifically, a sweeping attacker, who can sweep the EMI's bandwidth BW_{atk}, may be lucky enough to overlap with the input signal's frequency spectrum after several attempts.

Solution - Check and select approach: MagHop ensures that the overlapping with the input signal's frequency does not happen even after several attempts in the following way. After selecting a carrier frequency f_c from n members, $\{f_{c1}, f_{c2}, ., f_{cn}\}$, MagHop first checks whether f_c has any interference with injected EMIs. If there is no interference, only then that carrier frequency will be selected. If there is an interference, that carrier frequency is skipped, and a new frequency is selected pseudo-randomly from n members. A check circuit, present in the sensor pipeline's output stage, is used to tune on the carrier frequency f_c to sense the presence of interference (see Section 7.7.4 for details).

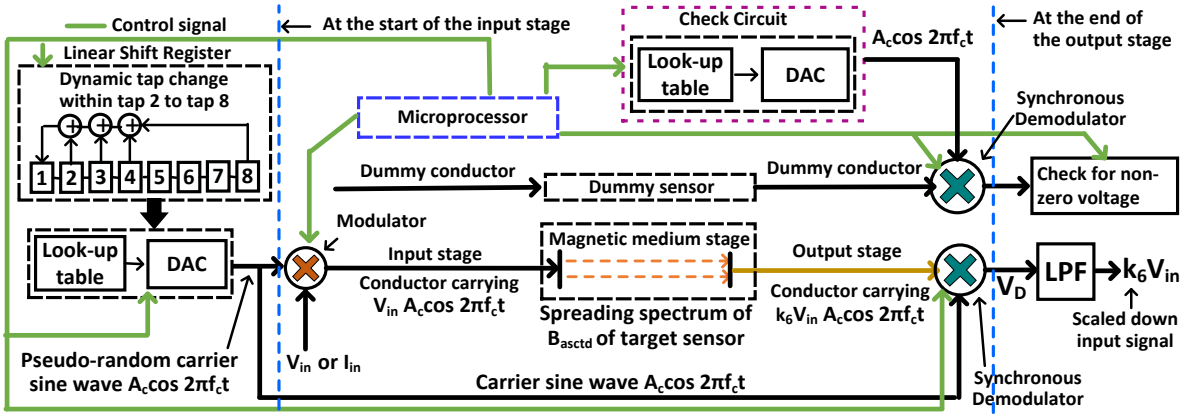


Figure 7.7: The implementation details of the proposed defense MagHop. There are 4 taps and the dynamic tap change happens within tap 2 to tap 8.

Magnetic Spectrum Hopping (MagHop)

Here, we explain why we name the defense as magnetic spectrum hopping (MagHop). When the carrier frequency f_c is varied or *hopped* within a set of n members, the hopping of f_c also results in periodic shifting of input signal's (i.e., V_{in} or I_{in}) frequency spectrum. As the associated magnetic field B_{asctd} is proportional to V_{in} or i_{in}, the spectrum of B_{asctd} also hops within the same set of n members. As the B_{asctd} is the magnetic transduction medium, this frequency hopping technique is termed as magnetic spectrum hopping, shortly MagHop (see Fig. 7.6 (Right)). Because of the frequency hopping in the magnetic medium stage, the proportional sensor output voltage V_{out} has the same spread spectrum in the output stage of the sensor pipeline. Therefore, the V_{out} is also immune to injected EMI/magnetic fields in the output stage conductor.

7.7 Implementation of MagHop

The blocks used in MagHop are shown in Fig. 7.7. MagHop has two blocks in the input stage of the sensor pipeline: (i) Pseudo-random frequency generator and (ii) Modulator.

7.7.1 Pseudo-Random Frequency Generator (PRFG)

The PRFG generates a carrier frequency f_c, which hops within a set $S = \{f_{c1}, f_{c2}, f_{c3}, f_{c4}, ..., f_{cn}\}$ of n members, in a pseudo-random fashion. It has the following steps.

Maximal sequence pseudo-random code: An 8-bit linear feedback shift register (LFSR) with 4 tap positions is used to generate a maximal sequence pseudo-random code. A maximal code would be the ideal secured code [317] since it has the lowest possible auto-correlation and is easy to generate. The 8-bit LFSR can generate 2^8 - 1 = 255 (a value of zero is not possible) pseudo-codes. A pseudo-code corresponds to a carrier frequency. Therefore, n = 255 carrier frequencies are present in set S. A $n = 255$ carrier frequencies seemed reasonable

because large values of n may reduce the separation between two adjacent carrier frequencies, resulting in a reliability issue. For example, if the separation between two adjacent carrier frequencies, say separation between f_{c1} and f_{c2} is low, there is a chance that both f_{c1} and f_{c2} can be within the same bandwidth BW_{atk} of the injected EMIs. As MagHop has the *check and select* approach, in this case, neither f_{c1} nor f_{c2} will be selected as the carrier frequency.

Code security: There are 12 different combinations of 4 tap positions that can generate maximal sequences in an 8-bit LFSR [318]. As we always keep one tap at position 8 and we don't use position 1 for tapping, there are only 8 possible tap combinations that can generate 255 maximal sequences. These 8 possible combinations of 4 tap positions are: (8,4,3,2), (8,5,3,2), (8,6,3,2), (8,6,5,2), (8,6,5,3), (8,6,5,4), (8,7,3,2), and (8,7,5,3). After every 255 cycles, the tap positions are dynamically changed to a random set of tap positions (i.e., (8,4,3,2) to (8,6,5,4)) so that attackers may not track the codes.

Largest carrier frequency: The generated carrier frequencies should support the sensor bandwidth. Therefore, the largest carrier frequency f_{cn} in the set S should always be less than the sensor bandwidth, denoted by BW_S, and the relationship can be expressed by Eqn. 7.8.

$$BW_S \geq f_{cn} + \frac{1}{2}BW_{in} \tag{7.8}$$

Look-up table: A faster approach to generate carrier frequencies is to take values from a look-up table. The look-up table must have values, at a minimum, twice the number of possible carrier frequencies because of the *Nyquist criteria*. Since there are $n = 255$ possible carrier frequencies, there must be a minimum of 510 values in the table. For an improved quality of the generated waveform, a total of $v = 2048$ values are used in the look-up table in our design. A digital-to-analog (DAC) converter takes $v = 2048$ values from the look-up

table and generates the carrier sine wave. If the DAC takes values from the look-up table at a rate of f_{DAC} Hz, the minimum frequency of the generated sine wave is $\approx f_{DAC}$ / v. To change the frequency, DAC can skip some values from the look-up table. For example, if every other value is taken instead of every value from the table, the frequency gets doubled. If m is the output of the pseudo-random code generator (i.e., 1 to 255), then the carrier frequency can be calculated by Eqn. 7.9.

$$f_c = f_{DAC} \times \frac{m}{v} \tag{7.9}$$

If m is not a factor of v, the generated wave is not sinusoidal. Interpolation [319] is used to solve this issue.

7.7.2 Modulator

The carrier wave $A_c cos(2\pi f_c t)$ from the PRFG is multiplied (see Fig. 7.7) by the input V_{in} or I_{in} (a proportional V_{in} is generated from I_{in}) by a modulator as below.

$$V_M = V_{in} \times A_c cos(2\pi f_c t) \tag{7.10}$$

The modulation takes place at the start of the input stage of the sensor pipeline. The carrier frequencies, which are *greater* than the bandwidth BW_{in} of the input V_{in} or I_{in}, are chosen for the modulation. As the $B_{asctd} \propto V_{in}$ or I_{in}, the frequency spectrum of the B_{asctd} is also spread in the *magnetic medium stage* and a scaled-down proportional voltage V_{out} is generated at senor's output, which can be written as,

$$V_{out} = k_6 \times V_{in} \times A_c cos(2\pi f_c t) \tag{7.11}$$

where k_6 is the scaling factor. *Because of the shifted frequency spectrum in the input and output stages of the sensor, the EMIs induced in the connecting conductors in the input and output stages cannot perturb the signal.*

7.7.3 Synchronous demodulator

A synchronous demodulator recovers the input V_{in} or I_{in} from the shifted spectrum *at the end of the output stage*. It multiplies the output V_{out} with the same carrier signal as:

$$\begin{aligned} V_D &= k_6 \times A_c cos(2\pi f_c t) \times V_{in} \times A_c cos(2\pi f_c t + \phi) \\ &= k_6(A_c^2/2)cos\phi \times V_{in} + k_6(A_c^2/2)cos(4\pi f_c t + \phi) \times V_{in} \end{aligned} \tag{7.12}$$

where ϕ is the phase difference present between the carrier signal from the modulator and the demodulator. A low-pass filter (LPF) can simply filter out the high frequency part $k_6(A_c^2/2)cos(4\pi f_c t + \phi) \times V_{in}$ from Eqn. 7.12 and gives output only the low frequency part $k_6(A_c^2/2)cos\phi \times V_{in}$.

As the same carrier signal generated in the PRFG is given to the modulator and demodulator, the phase difference ϕ is close to zero and constant. Therefore, the term $k_6 A_c^2$ /2 cosϕ is also constant in Eqn. 7.12, and a simple amplifier gives the correct $k_6 V_{in}$, which is a scaled-down version of V_{in}, at its output. The LPF does the amplification.

7.7.4 Check circuit

The *check and select* approach checks if the carrier frequency interferes with the EMI's bandwidth BW_{atk}. A check circuit, made with a look-up table and a DAC, executes the *check and select* approach. The check circuit is connected with a dummy conductor and a dummy sensor, which have the same physical properties as the input signal carrying conductor and target magnetic sensor, respectively. The dummy sensor and conductor are placed close to the input signal carrying conductor and target sensor. Therefore, they can sense the same EMIs injected into the target sensor/conductor and provide this signal to a synchronous demodulator (Fig. 7.7).

The check circuit generates the same f_c *before* the modulator uses the f_c to modulate the input V_{in} or I_{in} (see Section 7.7.5), and provides the carrier signal to a synchronous demodulator. The check circuit also varies the carrier frequency within $f_c - BW_{in}/2$ to $f_c + BW_{in}/2$. If the EMI's bandwidth BW_{atk} interferes within $f_c - BW_{in}/2$ to $f_c + BW_{in}/2$, Eqn. 7.12 indicates that a non-zero voltage will be generated at the output of the synchronous demodulator. The presence of a non-zero voltage indicates that the EMI interferes with the carrier frequency. Therefore, the current carrier frequency is discarded, and a new carrier frequency will be chosen next.

7.7.5 Coherency, real-time measurement, and overhead

An important question is how MagHop keeps a real-time and coherent measurement of the voltage or current signals. As different carrier frequencies carry the bandwidth BW_{in} of the input V_{in} or I_{in}, the signal gets fragmented. The reassembly of the fragmented signal after the demodulation is challenging because the coherency among fragments should be maintained. Here, a critical parameter is how long MagHop takes to generate a carrier signal. Let us denote it by generation time, t_{gen}. Another important parameter is how long a carrier frequency operates before hopping to another carrier frequency. Let's denote it by

operating time, t_{op} (Fig. 7.8).

It takes one clock cycle to generate a pseudo-random code by the LFSR. It takes another 3 clock cycles to calculate how the look-up table will be sampled by DAC to generate the carrier wave. Therefore, the carrier wave is generated after $t_{gen} = 4$ clock cycles. The operating time t_{op} is chosen by the designer. It is kept short so that the attacker cannot anticipate the carrier frequency. We choose t_{op} within 0.1 ms - 2 ms.

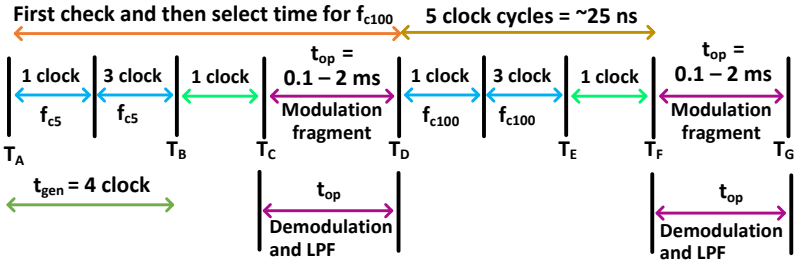


Figure 7.8: Timing information of MagHop for keeping real-time coherency.

Let's say, within time $T_A \rightarrow T_B$ (4 clocks), a pseudo-random carrier frequency, say f_{c5} is generated. The time $T_B \rightarrow T_C$ (1 clock) is used to prepare for modulation. The carrier frequency f_{c5} operates for $t_{op} = T_C \rightarrow T_D$, where the modulation takes place. During $T_C \rightarrow T_D$, parallelly, a demodulation takes place in the synchronous demodulator. Again, within $T_D \rightarrow T_F$ (5 clocks), a new carrier frequency, say f_{c100} is prepared, and within $T_F \rightarrow T_G$ both the modulation and demodulation take place by f_{c100}. The check circuit finishes the interference checking for carrier frequency f_{c100} within $T_A \rightarrow T_D$ before the f_{c100} is generated. Therefore, no extra time is used for the *check and select* approach. It is apparent that there is a 5 clock cycles gap (i.e., $T_D \rightarrow T_F$) between one modulation fragment $T_C \rightarrow T_D$ to the next modulation fragment $T_F \rightarrow T_G$. As the PRFG has a high-speed clock with a period 5ns, the 5 clock cycles is only 25 ns. As the bandwidth of VCMSs is typically 0 - 200 kHz (i.e., $\sim$ 5 μs), the 25 ns is 200x times smaller than the smallest rate of change of the input voltage or current signal. Therefore, a 5-clock delay does not hamper the coherency and real-time behavior of any of the existing sensors. Moreover, as the check circuit works ahead of the next carrier frequency being generated, the *check and select* approach does not overload

the defense. In addition, The LPF parallelly works within the demodulation time $T_C \rightarrow T_D$. Therefore, the LPF does not hamper the coherency of VCMSs.

Justification: To justify the coherency, a voltage sensor LV25P is connected with MagHop. A 10 V and 100 Hz signal is given as an input V_{in} to the sensor. The bandwidth 0-25 kHz of the LV25P is divided into 255 carrier frequencies. As the input signal is 100 Hz, the carrier frequencies are selected within 100 Hz $< f_c <$ 25 kHz in a pseudo-random fashion. Fig. 7.9 shows four such carrier frequencies. which are selected pseudo-randomly to modulate the input V_{in}. Each carrier frequency modulates the V_{in} for t_{op} before switching to the next carrier frequency. A simultaneous demodulation takes place within each t_{op}. Fig. 7.9 indicates that the output signal is coherent and real-time after the demodulation.

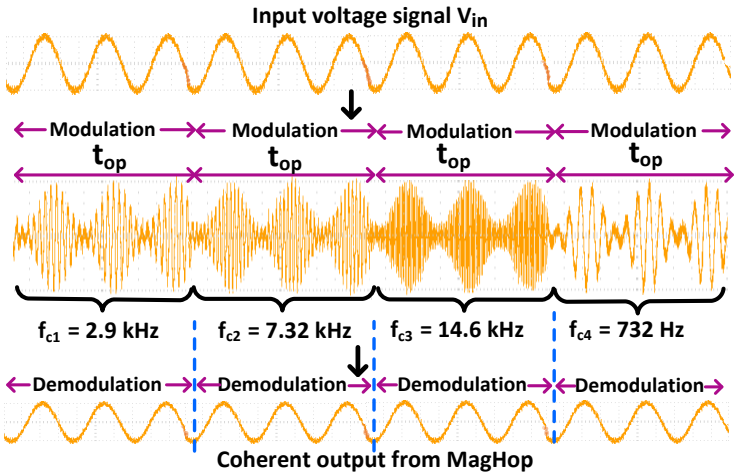


Figure 7.9: The output of MagHop is coherent.

7.7.6 Defense algorithm and control signals

The algorithm 6, which binds together all the components of MagHop, runs on a microprocessor. The microprocessor generates control signals to the PRFG to start and stop the shift register and sampling from the look-up table. The control signal also syncs the operation of the check circuit with the PRFG and controls their execution order following Fig. 7.8. Another control signal controls the start and stop of the operating time, t_{op}. Lines 1-14 in algorithm 6 are self-explanatory and already discussed in detail in previous sections.

7.7.7 Security of the defense itself

A question may arise what will happen if the attacker attacks the defense itself? The defense does not have any magnetic medium stage; therefore, it could not be directly impacted by the attack. However, in an extreme case, the attacker can cause a random bit flip in the LFSR/look-up table using EMI. It may result in a new carrier frequency. However, this will not create any problem as the carrier frequency is still unknown to the attacker. As the *same* carrier frequency is used in the modulator and synchronous demodulator, any change/perturbation in this will not hamper the normal processing of the defense.

Algorithm 6: Defense algorithm.

Input: Voltage/current signal V_{in} or I_{in} being measured
Output: Scaled down voltage: $k_6 V_{in}$

```
POINT A:
Generate pseudo-random code m & configure DAC to sample look-up table
Sample the look-up table and generate the carrier wave, f_c
Check circuit checks if f_c has interference with BW_atk
while f_c has interference do
    Generate the next pseudo-random code m
    Generate another carrier wave, f_c
    Check circuit checks if f_c has interference with BW_atk
end
Start the modulator
Shift the bandwidth BW_in of the V_in or I_in to f_c ± BW_in/2
Continue transmission for one operating time, t_op
Parallel demodulation by the synchronous demodulator
The LPF outputs k_6 V_in
If operating time, t_op is over, JUMP to POINT A and iterate over
```

7.7.8 A prototype

A prototype of MagHop is shown in Fig. 7.10 (Left). The LFSR and the look-up table are implemented on a Zynq-7000 SoC with a 200 MHz clock on a Zedboard [320]. The inbuilt SPI flash of the Zedboard is utilized to hold the look-up table. As mentioned earlier, the size of the look-up table is $v = 2048$. A *Perl* script is used to automate the process of making this look-up table which came with the System Verilog code. An 8-bit DAC (part# ADV7125V

[321]) is used to sample the $v = 2048$ data from the look-up table. As the DAC is 8 bits, the memory needed to store the look-up table is $2048 \times 8 = 16384$ bits. The DAC accesses the look-up table with a rate of $f_{DAC} = 1.5$ MHz. For $m = 1$ to 255, Eqn. 7.9 indicates that the carrier frequency will be between 732 Hz and 186 kHz. By increasing the f_{DAC}, we can generate greater than 186 kHz carrier wave. If a carrier frequency is higher than the sensor's bandwidth, that frequency will not be used to modulate.

The modulator and the synchronous demodulator are implemented [322] using a high frequency power MOSFET (part# R6012JNX C7G) with a modulation index < 1. The check circuit uses a MOSFET of the same type as the demodulator and the same Zedboard for the look-up table. A second-order LPF is implemented using a low-power op-amp (part # TL084CN). The defense algorithm runs on an ARM Cortex-M3 (part #EFM32GG990F1024 [323]) with a 48 MHz clock.

Table 7.1: Evaluation of MagHop. Here, H = Hall effect; F = Faraday's law; Curr = Current; Vol = Voltage; D = Differential

Manuf.	Part #	Type/Modality /Loop	EMI power/freq. (a)	Avg. R (fixed interval) (b)	EMI power/freq. (c)	Avg. R (rand. interval) (d)
Allegro	ACS710 [263]	H/Curr./Open	0 - 10 W / 0 - 120 kHz	0.99	10 W / 0 - 120 kHz	0.99
Allegro	ACS724 [265]	D/H/Curr./Open	0 - 10 W / 0 - 120 kHz	0.98	10 W / 0 - 120 kHz	0.98
Honeywell	CSNS300M [324]	F/Curr./Closed	0 - 10 W / 0 - 150 kHz	0.97	10 W / 0 - 150 kHz	0.98
Acu AMP	CTF-5RL [325]	F/Curr./Open	0 - 10 W / 50 - 400 Hz	0.97	10 W / 50 - 400 Hz	0.99
CR Mag.	CR8410 [326]	F/Curr./Open	0 - 10W / 50 - 50 kHz	0.99	10 W / 50 - 50 kHz	0.99
CR Mag.	CR8320 [313]	F/Curr./Open	0-10W / 50 - 50 kHz	0.98	10 W / 50 - 50 kHz	0.97
LEM	LTSR 6-NP [270]	H/Curr./Closed	0 - 10 W / 0 - 100 kHz	0.98	10 W / 0 - 100 kHz	0.98
LEM	LV 25 P [271]	H/Vol./Closed	0 - 10 W / 0 - 25 kHz	0.99	10 W / 0 - 25 kHz	0.99
Triad Mag.	MET-28-T [327]	F/Vol./Open	0 - 10 W / 300 - 100 kHz	0.98	10 W / 300 - 100 kHz	0.98
Triad Mag.	MET-42-T [328]	F/Vol./Open	0 - 10 W / 300 - 100 kHz	0.97	10 W / 300 - 100 kHz	0.98

7.8 Evaluation of the defense MagHop

7.8.1 Testbed

A testbed (see Fig. 7.10 (Right)) is used to evaluate MagHop. Ten different VCMSs of all types, such as open-loop, closed-loop, and differential sensors from six different manufacturers, are used to evaluate MagHop (see Table 7.1). The defense is tested against two sources: (i) a pseudo-random frequency generator, which is implemented in the Zedboard with logic circuits, connected with signal amplifiers (0-200 kHz) and a monopole antenna is used as a source of EMI, and (ii) a Grove electromagnet [314] is used as the source of static magnetic fields. *The monopole antenna and the electromagnet are placed within 1 cm of the conductors and also near the sensors.* Moreover, variable DC and AC power supplies are used to provide input voltage or current signals to sensors.

Performance metric: If the sensor output before the attack is *similar* to the sensor output after the attack with MagHop, we can claim that MagHop works. The similarity between the sensor output before and after the attack is quantified by calculating the correlation coefficient [329], R between both signals. If the correlation coefficient is $\sim$1, the output voltage before the attack and after the attack with MagHop are similar, indicating the effectiveness of MagHop under an attack.

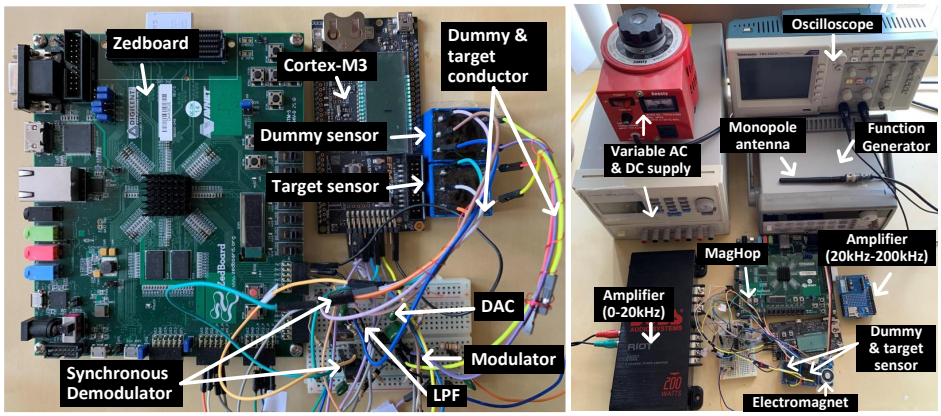


Figure 7.10: (Left) The prototype. (Right) The testbed.

7.8.2 Evaluating sweeping & responsive attacker

Varying EMI power/frequency in fixed interval

To evaluate the efficacy of MagHop, at first, we vary the power of the injected EMI from 0 to 10 W with a 0.1 W increment. For every power increment, the frequency of EMI is varied within the *entire bandwidth* of the sensor with a 100 Hz increment and with a fixed 2 ms interval. For example, say for LV25P sensor, we use a 0.1 W and 100 Hz EMI at the beginning. After 2 ms, we increase the EMI frequency to 200 Hz. In this fashion, we vary the EMI frequency within the entire sensor bandwidth (25 kHz). Next, we repeat the same process for a 0.2 W EMI, and so forth. We calculate the correlation coefficient R for every combination of the power and frequency of the EMI and do an average of R for every sensor. The experimental data is logged and analyzed, and the average R is calculated using a *Python* script. The average R for every sensor in hand is less than 0.7 before MagHop is used compared to close to $\sim$1 after MagHop is used (see Table 7.1(a, b)). This indicates that MagHop works against a sweeping or responsive attacker, who can vary the frequency and power of the EMI signal within a *fixed* time interval.

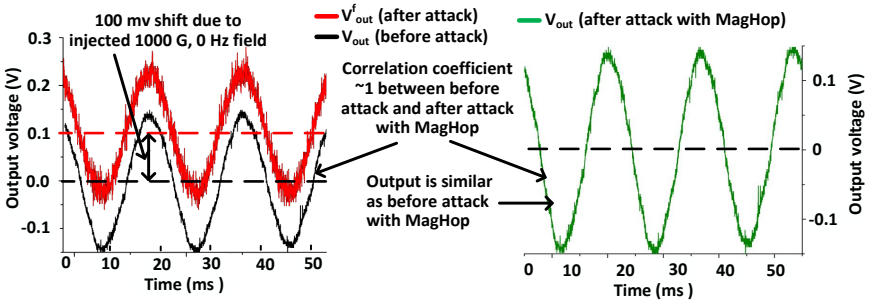


Figure 7.11: Justification of MagHop when an attack happens.

Justification: Fig. 7.11 shows a justification for using MagHop during an attack as an example. We consider a strong attacker who can inject 3W and 1 kHz EMI, and 0 Hz and 100 G static magnetic field together into a target VCMS. The 1 kHz EMI signal causes noise-like perturbations, and 1000 G static magnetic fields add a DC offset to the sensor's output. Now, after using MagHop, all the injected attack fields are contained. Therefore, the

sensor output during an attack with MagHop is exactly similar to before the attack, with a correlation coefficient of 0.99. This justifies that MagHop can provide an unperturbed output signal during an attack.

Varying EMI frequency in random interval

We keep the EMI power fixed at 10 W but randomly vary its frequency in a *random* time interval within 0.1 s to 3 s. For example, a 10 W EMI has a 100 Hz frequency at the beginning. After a random time interval, we increase the EMI frequency to a random value and repeat the same process for the entire sensor bandwidth. We calculate the R for every reading and do an average of R for every sensor. The average R for every sensor in hand is $<$ 0.5 before MagHop is used compared to $\sim$1 after MagHop is used (see Table 7.1(c, d)). This indicates that MagHop works against a sweeping/responsive attacker, who can randomly vary the frequency in a *random* interval.

Varying operating time t_{op} in incremental interval

In Sections 7.8.2 to 7.8.2, we keep the t_{op} fixed at 0.3 ms. Here, we vary the t_{op} within 0.1 ms to 4 ms with an increment of 0.1 ms. For each t_{op}, we keep the EMI power fixed at 10 W and vary the EMI frequency with a 100 Hz interval with an *increment* of 0.1 ms time interval. We see that when the defense has $t_{op} > 2.2$ ms, the average of R is dropped below $\sim$0.94. The reason behind the drop of R is that the probability of having interference gets increased for a long modulation fragment (i.e., large t_{op}). Therefore, the t_{op} should be kept short (i.e., 0.1 - 2 ms) to be effective against a sweeping attacker. In addition, if the $t_{op} < 0.1$ ms, MagHop faces reliability issues because of the fast switching between small modulation fragments. A $t_{op} < 0.1$ ms can be achieved by increasing the hardware speed with a trade-off in the cost.

Varying the EMI's bandwidth BW_{atk}

A question may arise what will happen if the attacker jams the entire sensor bandwidth BW_S using an EMI, which has the same bandwidth BW_{atk} equal to BW_S. If this happens, MagHop will not work as the carrier frequency f_c will not find any unoccupied channel within BW_S. For this case, MagHop will do a *fail-safe shutdown* and notify the system about the possible reason.

However, if the attacker partially jams the sensor bandwidth BW_S, MagHop still works, and its performance varies depending upon what percentage of sensor bandwidth is jammed. Because, if BW_S is partially jammed, MagHop should hop multiple carrier frequencies to find an unjammed channel and check circuit kicks in before every hopping. As the check circuit requires a certain time (i.e., 115 ns) to check a carrier frequency whether its jammed or not, if the check circuit needs to do multiple frequency checks to find an unjammed bandwidth, the latency before every operating time t_{op} increases, hampering the real-time measurement of sensors. Fig. 7.12 (Left) shows how latency is related to the percentage of sensor bandwidth BW_S jammed by the attacker. It shows that latency is less impacted until 37% of the jammed BW_S. However, latency keeps increasing exponentially after 37% until a fail-safe shutdown occurs at 100%.

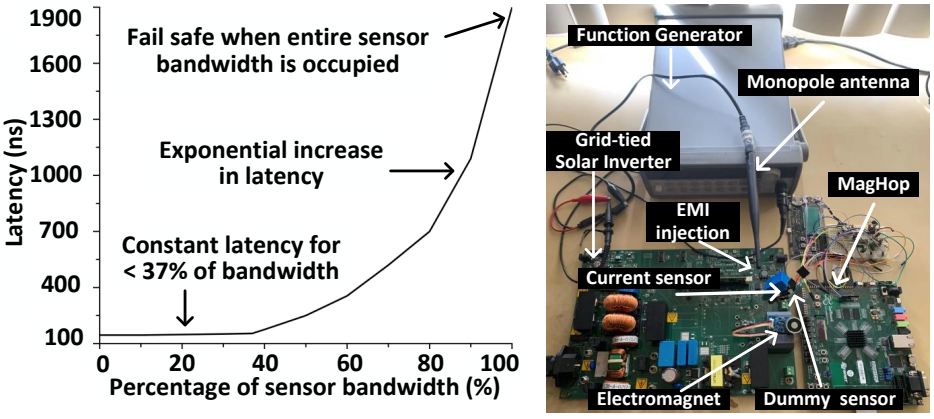


Figure 7.12: (Left) Latency for different attack bandwidth. (Right) Evaluating MagHop on a practical system: a grid-tied solar inverter.

7.8.3 Varying the frequency of the input signal V_{in} or I_{in}

In Sections 7.8.2 to 7.8.2, we keep V_{in} or I_{in} fixed at 1 A/110 V with 60 Hz. Here, we vary the frequency of V_{in} or I_{in} within the *entire sensor bandwidth* while keeping $t_{op} = 0.3$ ms. We also vary the 10 W EMI's frequency with a 1 ms interval. The avg. R for every sensor is less than 0.6 before MagHop is used compared to $\sim$1 after MagHop is used, indicating that MagHop does not hamper the sensor bandwidth.

7.8.4 Varying the magnetic field strength

To test the strength of MagHop, we vary the magnetic field density of injected magnetic fields upto 12000 G with a 100 G interval using an electromagnet. Please note that an 8000 G can penetrate a ferromagnetic shield [79, 80]. Therefore, the 12000 G is indeed a high amount for a shield to defend. Without MagHop, the high magnetic flux of the injected field gives an average $R = 0.2$. After using MagHop, the average R is $\sim$1. It indicates that MagHop also works against a strong magnetic field, which can even penetrate a shield.

7.8.5 Low cost, low-power and easy to integrate

The shift registers and look-up tables are implemented in an FPGA with power gating [330] for low-power design. Low-power discrete ICs are used for modulators, demodulators, LPF, and DAC for rapid prototyping. The total power consumed by the prototype is 1.5 mW, which is compatible with $\sim$10 mW [274] consumed by VCMSs itself. MagHop can be connected with the target VCMS in a plug-&-play manner.

As we use a Zedboard for rapid prototyping, the size and cost of the prototype are high. However, the shift register and look-up table can be implemented in discrete cheap registers instead of a Zedboard. For this, we estimate the total cost is $<$ \$20 for bulk orders, excluding the dummy sensor. A complete SoC design would reduce the size and cost even more.

7.9 Evaluating on a Practical System

The effectiveness of MagHop is evaluated with a practical system: a grid-tied solar inverter that uses VCMSs to calculate power. An attacker can target these VCMSs and inject EMIs/magnetic fields to mislead the inverter's controller with wrong data, leading to a wrong operating state (e.g., disconnecting the grid-tied inverter from the power grid).

We integrate MagHop with a scaled-down 140 W grid-tied solar inverter [275] kit (part# = TMDSOLARUINVKIT) from Texas Instruments (Fig. 7.12 (Right)). Before the EMI attack, the inverter generates 94 W. This inverter has a Hall effect current sensor with a part # ACS712ELCTR-20A-T. Now, we inject a 100 G magnetic field from an electromagnet into this current sensor from a 1 cm distance. The power reading goes up to 129 W because of the perturbation in the sensor reading. Next, we connect MagHop with the Hall effect current sensor and repeat the same experiment. At this time, the power is not changed from 94 W. This proves the efficacy of MagHop against a magnetic field injection into a VCMS.

7.10 Limitations

Skin effect: As MagHop uses high-frequency carrier waves, the skin effect [331] can occur in the conductor present in the input/output stage of the sensor. However, as the conductor length is small, the resistance increase from the skin effect is negligible and does not impact the overall measurement.

High-speed hardware: To maintain coherency among different fragments, the microprocessor should be comparatively faster than the input voltage or current signal; otherwise, a time lag will be introduced while shifting the carrier frequency. Empirically, the microprocessor clock should be at least 100× faster than the sensor's bandwidth. Moreover, the time required for carrier frequency generation and switching to a new carrier frequency also

should be low. Therefore, hardware speed is a critical requirement. We use a 200 MHz clock for the FPGA and a 48 MHz clock for the microprocessor in our prototype.

Jamming the entire sensor bandwidth: As mentioned before, if the entire sensor bandwidth is jammed, MagHop will fail safely. However, if there is a free spectrum to use, MagHop will *slip though* using the free spectrum.

Exploitation by a strategic attacker: There are 12 different combinations of 4 tap positions that can generate maximal sequences in an 8-bit LFSR [318]. As we keep one tap at position 8 and we have not used position 1 for tapping, there are 8 possible tap combinations that can generate *8 sets* of LFSR sequences (see Section 7.7). A strategic attacker who can observe the frequency for a long time (i.e., contradictory to the threat model) can theoretically calculate all 8 possible next-states for the current state and jam those 8 frequencies. However, practically speaking, this could be *complicated* as the attacker needs to know the timing information, such as t_{gen} and t_{op} of MagHop, and needs to be very fast to be always one step ahead of the defense; otherwise, the *check circuit* can sense the channel as jammed, and MagHop will find another unjammed channel to slip through using the free spectrum. If this process continues to happen, a situation similar to Section 7.8.2 will happen, and MagHop will do a fail-safe shutdown notifying the system about the possible reason.

7.11 Related Work

EMI shielding: Bora et al. [332] and Merizgui et al. [333] proposed new shielding material to prevent EMIs. The main limitation of the shielding approach is that it may work against time-varying EMIs, but not against a static magnetic field. Moreover, an attacker can saturate the shield using a strong magnetic field to diminish its shielding property [79].

Filtering: Zhang et al. [310] provided ways for only EMI detection but did not provide any

defense against it.

Kune et al. [42] proposed an adaptive filtering technique to estimate the attack signal first and then subtract it from the input signal to filter out attack signals from analog acoustic sensors. This technique may fail for the following two reasons: (i) Because of the physical distance between the adaptive filter and the compromised sensor, the adaptive filter cannot measure the exact amplitude of the external attack fields injected into the sensor. (ii) This will also fail if the attack signals and the original signals have identical frequencies.

Zhang et al. [47] proposed low-pass filters to filter out the injected ultrasound from baseband voice commands. This approach only worked because the ultrasonic signal and the baseband voice signal have two different spectra, which are separable by filters. This approach will also fail if the attack signal and the original signals have identical frequencies.

Trippel et al. [51] proposed randomized and 180^0 out-of-phase sampling to prevent acoustic injection on inertial sensors. They take two samples with 180^0 out-of-phase from input signals at random intervals to cancel attack signals. They will fail for the following two scenarios: (i) If the attack and original input signals have identical frequencies, they will filter out original signals while filtering the attack signals. (ii) They cannot work against static magnetic fields as randomized sampling cannot filter out a DC signal.

Tu et al. [161] proposed a transduction shield (TS) to estimate attack signals first and then subtract attack signals from the original one. The main limitation is that it assumes the TS and the target sensor are identical and TS sees the same attack signals as the target sensor. However, there is always a mismatch and physical distance between the TS and the sensor. Therefore, the attack signals cannot be exactly nullified. The defenses proposed by Barua et al. [156], [157] have upper limits for frequency and power of EMIs up to which these defenses could work, whereas MagHop does not have these limits.

State machine: The state machine based defenses by Cardenas et al. [28, 277], Urbina et

al. [278], and Shoukry et al. [279] do not directly prevent EMI injection. Instead, they use state information to recover the controller from an attack.

MagHop is novel in the sense that it encrypts the information within the magnetic medium stage using pseudo-random channel. Therefore, there is no practical limit to the attack signal's strength up to which MagHop can tolerate. MagHop can handle attack signals of any strength as long as the entire sensor bandwidth is not jammed. Moreover, MagHop can contain attack signals having the same bandwidth as the input signal being measured. A comparison between MagHop and recent works is provided below:

Table 7.2: Comparison between MagHop and recent work.

Comparison	Recent works [42, 47, 51, 161, 310]	MagHop
strength of injected B_{atk}	support up to a limit	no limit if free bandwidth exists
frequencies of injected B_{atk}	does not support entire sensor bandwidth	support entire sensor bandwidth
injected B_{atk} power	low power ($\sim 4W$)	no theoretical limit
power consumption	unknown	$\sim$1.5 mW

7.12 Summary

We present MagHop to design a secure VCMS against an intentional EMI/magnetic field. MagHop shifts the frequency spectrum of the transduction medium to another spectrum, which is unknown to an attacker. Therefore, the attacker cannot inject any perturbations in the form of an EMI/magnetic field into the magnetic medium stage. Even a strong sweeping/responsive attacker cannot interfere with the magnetic transduction stage because of the *check and select* approach of MagHop. We implement a low-power design of MagHop on an FPGA and Cortex-M processor for rapid prototyping. We thoroughly evaluate MagHop on ten different VCMSs from six different manufacturers. Our results from these experiments show a promising efficacy against intentional EMI/magnetic field injection while keeping the sensor output coherent and real-time. *As designing secure sensors is important for critical infrastructures, finally, we believe that the idea presented in this chapter will be beneficial to*

other sensor types to build the next generation of trustworthy sensors.

Chapter 8

Conclusion

This thesis presents three unconventional attack models and vulnerabilities in CPSs that originated in the cross-layers of CPSs. This thesis discusses how different cross-layer attacks happen in smart power grid systems, bio-safety labs, and industrial control systems (ICSs) and propagate from the physical domain to the cyber domain of CPSs, compromising the connected systems.

Chapter 2 of this thesis addresses the first attack model, which includes how the false data injection into a Hall sensor can compromise the solar inverter controller in a smart power grid. We have identified five attack scenarios by which the attacker can compromise the inverter and also the connected grid. Moreover, this thesis has introduced a duty-cycle variation approach for adversarial control that can alter the inverter voltage and real power noninvasively.

Chapter 3 of this thesis addresses the second attack model, which includes a non-invasive attack using malicious music on differential pressure sensors (DPSs) located in a bio-safety lab. This thesis finds the resonant frequency of DPSs used in bio-safety labs are in the audible range. Therefore, this thesis demonstrates a method to insert segments of the resonant frequency band in specific intervals inside of music and end the inserted segments with their peak to maintain an average forged pressure in the DPS's transducer system. As a result,

the attacker can use malicious music to fool the DPSs used in the RPM and HVAC systems of a bio-safety lab and can turn the negative pressure into positive pressure. This may cause an alarm, resulting in chaos in the facility, and has the potential to leak deadly microbes from the facility.

Chapter 4 of this thesis addresses the third attack model, which introduces an attack model-*BayesImposter* that can hamper the availability and integrity of an ICS in cloud settings. We are the first to point out how the .bss section of the target control DLL file of cloud protocols is vulnerable in ICS. textitBayesImposter exploits the memory deduplication feature of the cloud that merges the attacker's provided .bss imposter page with the victim page.

This thesis also presents three hardware/software co-design defenses for the unconventional vulnerabilities discussed in the first half of the thesis. This thesis discusses how cross-domain attacks involve hardware and software layers, and defenses against these vulnerabilities also demand new hardware/software co-design approaches to detect, contain and isolate vulnerabilities in CPSs.

Chapter 5 of this thesis addresses the first defense technique – HALC, against a weak and strong magnetic spoofing attack on Hall sensors. HALC can not only detect but also contain the weak and strong magnetic spoofing of different types, such as constant, sinusoidal, and pulsating fields, in hard real-time. HALC utilizes the analog and digital cores to achieve a constant computational complexity O(1) and keep the existing data processing speed of the connected system undisturbed.

Chapter 6 of this thesis addresses the second defense technique – PreMSat that can prevent the saturation attack satisfactorily on passive Hall sensors. PreMSat can prevent the saturation attack originating from different types, such as constant, sinusoidal, and pulsating magnetic fields, in hard real-time. Moreover, PreMSat can also prevent weak magnetic spoofing in the linear region of the differential amplifier. PreMSat integrates a low resistive

magnetic path to collect the external magnetic fields injected by the attacker and utilizes a finely tuned PID controller to nullify the external fields.

Chapter 7 of this thesis addresses the third defense technique – MagHop to secure voltage and current magnetic sensors (VCMSs) against an intentional EMI/magnetic field. MagHop shifts the frequency spectrum of the transduction medium to another spectrum, which is unknown to an attacker. Therefore, the attacker cannot inject any perturbations in the form of an EMI/magnetic field into the magnetic medium stage of VCMSs.

Overall, novel methods and algorithms introduced in this thesis may also be applicable to other attack and defense techniques in CPSs. As designing secure CPSs is important for critical infrastructures, finally, we believe that the idea presented in this paper will be beneficial to other systems to build the next generation of trustworthy CPSs. Moreover, the findings of this thesis may also attract researchers from other domains, such as automotive systems, robotics, medical devices, and smart city, that involves the integration of cyber and physical domains through the hardware/software cross-layers.

Appendix A

Secondary Thesis Contributions

Apart from the key thesis contributions, the findings in this thesis have also contributed to several other research works which are summarized in the following sections. Besides, while working on this thesis, the author also contributed to several other research works [334].

A.1 Hierarchical Temporal Memory based One-pass Learning for Real-Time Anomaly Detection and Simultaneous Data Prediction in Smart Grids

Smart grids are cyber-physical systems (CPSs) that are comprised of pervasive sensing, computation, and control in spatially distributed power networks. Such smart grids generate large volumes of data in real-time. Thus, the challenge and the opportunity lies in systems and algorithms that can extract useful information from the various data streams and make reliable decisions in (near) real-time.

Micro Phasor Measurement Units (μPMUs) are deployed in distribution networks of smart grids to provide rich data on voltage and current variations at a finer resolution. The operators can monitor the distribution applications in real-time, due to the high performance of μPMU technology in distribution networks. They support a wide range of control and diagnostic applications, such as real-time anomaly detection and data prediction. In this

work, we consider the specific problem of detecting anomalies and simultaneously predicting future observations in smart grids from unlabelled data that is generated by μPMUs. Anomaly detection is an important problem because failing to detect anomalies in a timely manner can affect the whole system and cause massive power failures, and prediction can help in planning and control.

The data provided by μPMUs have a few characteristics that are relevant to the problem at hand. **First**, it provides time-series data that can be observed only one at a time in the sequential order they arrive. Hence, the data are not naturally suitable for batch learning as a full dataset is not available. **Second**, the smart grid is inherently dynamic and the statistics of the generated data can change over time (*i.e., concept drift*). **Third**, the μPMU data has inherent information of the smart grid dynamics which can be leveraged to predict future observations. Hence, the problems of anomaly detection and prediction are challenging for the following reasons: (i) the algorithm should be able to learn online in *one-pass*, and in an unsupervised fashion (i.e., without human intervention); (ii) should be able to learn continuously, i.e., handle concept drifts in data, and (iii) the same algorithm should be able to perform anomaly detection and data prediction in real-time.

In this work, we introduce an architecture based on Hierarchical Temporal Memory (HTM)[335] to address the aforementioned challenges. *To the best of our knowledge, this is the first work, which demonstrates that anomaly detection and data prediction in smart grids can be performed using the same algorithm with a competitive accuracy and simultaneously in real-time, while learning from just one-pass and in an unsupervised fashion.*

A.1.1 Related Work and Contributions

A.1.2 Related work

Anomaly detection and data prediction have been studied extensively but independently in the smart grid domain. Moghaddass et al. [336] proposed a framework to detect anomalies at the customer level based on smart meter data. Zhou et al. [337] demonstrated the efficacy of *Ensemble-based* algorithm for online and robust anomaly detection in PMU data. G. Napier et al. [338] used a model-based approach to detect anomalies in the SCADA network in the smart grid. *The main limitation of these approaches is that their applicability is limited to stationary conditions and they can not perform simultaneous prediction.*

Different data-driven techniques have also been proposed for analyzing μPMU data of the smart grid. Supervised learning methods, such as decision trees [339], SVMs [340] and unsupervised learning methods, such as clustering [341] have already been proposed. Valenzuela et al. [342] used *Principal Component Analysis* (PCA) to classify power flow into regular and irregular sub-spaces to detect intrusion. Zhou et al. [343] proposed a semi-supervised approach using *kernel Principal Component Analysis* (kPCA) and *partially-hidden-structured SVM* (pSVM) to detect abnormal events in smart grids. *The drawbacks of these approaches are* that they are suitable for batch learning and can not be used for other tasks such as prediction. Brahma et al. [344] propose a dynamic real-time framework named as *Shapelets* that can accurately and speedily classify PMU data. Auto-Regressive Moving Average (ARMA) is widely used for anomaly detection and short term prediction for load forecasting [345]. Gao et al. [346] presented a dynamic state prediction method based on the Auto-Regressive (AR) Model using PMU data. Sia et al. [347] used *the Hurst exponent* to anticipate future voltage collapse using PMU signals. *The limitation of these approaches is that these models can not handle concept drifts and can not predict future observations simultaneously.* Moreover, Yang et al. [348, 349] proposed a deep PDS-ERT based learning method to realize

real-time anomaly detection; however, this method requires batch learning and cannot learn in one-pass and unsupervised fashion.

Hollingsworth et al. [350] investigated the combination of Long-Short-Term-Memory (LSTM) and Auto-Regressive Integrated Moving Average (ARIMA) to detect anomalies and simultaneously forecast energy consumption. But this method requires large datasets to train and it is not a real-time, one-pass, and unsupervised learning method. Ahmad et al.[351] demonstrated the efficacy of the HTM for real-time anomaly detection for different applications. The fundamental difference between our work and [351] is that our work demonstrates that the HTM model, which can detect real-time anomalies, can also be reused to simultaneously predict future observations. Finally, methods used by industries like Netflix's *Robust Principle Component Analysis* (RPCA) [352] and Yahoo's EGADS [353] also require batch training, and so are not applicable for the online setting that we consider. *Most importantly, none of these approaches can be used simultaneously for the prediction task.*

Contributions

Our novel technical contributions are as follows:

- We introduce an architecture for anomaly detection and simultaneous data prediction in smart grids that is inspired by neuro-cognitive mechanisms of the human called Hierarchical Temporal Memory (HTM). The HTM is a powerful framework developed by Numenta for sequence learning. At the heart of the HTM is a Cortical Learning Algorithm (CLA) that is inspired by how the human neocortex functions [335]. To the best of our knowledge, we are the first to introduce this method for applications on smart grid μPMU data.
- We demonstrate the effectiveness and applicability of the HTM approach for addressing the challenges of learning online from smart grid μPMU data for anomaly detection and simultaneous data prediction. It has been demonstrated that the HTM has the

capability for continuous and unsupervised learning from streaming data and has been proven to work well for real-time detection in other domains [351]. This justifies extending this approach to the same problems in smart grids.

- To demonstrate the effectiveness of the proposed approach, we compare the performance of the HTM with five state-of-the-art real-time anomaly detection algorithms, such as Random Cut Forest, Bayesian Change Point, Windowed Gaussian, EXPoSE, and Relative Entropy on μPMU data of the smart grid. We have used the Numenta Anomaly Benchmark (NAB) [354] to compare these anomaly detection algorithms. To the best of our knowledge, this metric is the first of its kind for smart grid data.

- For the prediction problem, we compare the performance of the HTM with a total of six state-of-the-art sequence learning and prediction algorithms for data prediction, such as online LSTMs, online LSTMs with 6000 buffer points, online LSTMs with 3000 buffer points, online LSTMs with 1000 buffer points, Time Delayed Neural network (TDNN), and Adaptive Filter. This performance comparison among all these models on μPMU data is also the first of its kind in the smart grid domain, to the best of our knowledge[1].

The remainder of this work is organized as follows. Section A.1.3 introduces the HTM model and its learning algorithm with an illustrative example of sequence learning. Section A.1.4 demonstrates the application of the HTM based learning model for anomaly detection in real-time in smart grid μPMU data. This section also explains how the HTM based method detects both temporal and spatial anomalies, learns in a continuous-online fashion and provides the comparison of the performance with five other state-of-the-art real-time anomaly detection algorithms. Section A.1.5 demonstrates that the same HTM from the previous section can be reused for multi-step prediction and compares the performance with

[1]The source code of this work is available in the following link: `https://github.com/unknown-commits/HTM_upmudata_anomaly_detection_prediction`

six other state-of-the-art sequence prediction algorithms. Finally, limitations and conclusions are drawn in Section A.1.6 and Section 3.13, respectively.

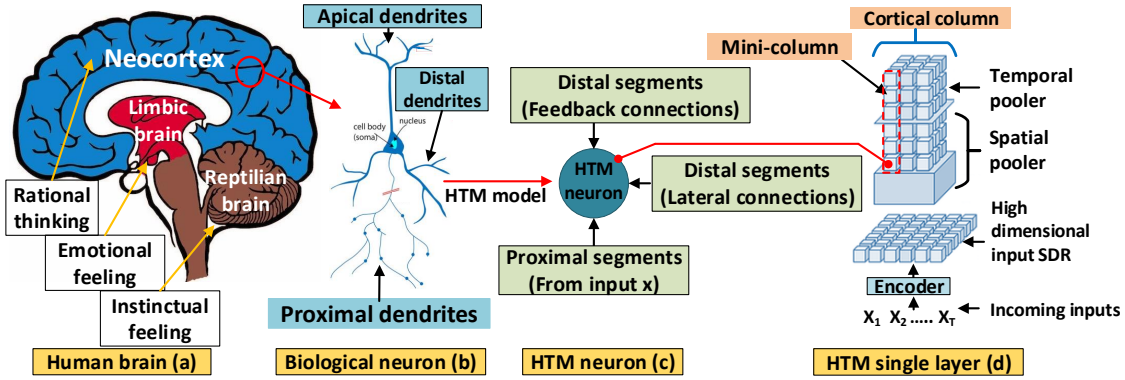


Figure A.1: Neocortical architecture of the Hierarchical Temporal Memory.

A.1.3 HTM Architecture and Learning Algorithm

In this section, we briefly discuss the structure of an HTM model. We then discuss the activation and the learning algorithm and illustrate the temporal representations learnt by an HTM through an example.

Hierarchical Temporal Memory (HTM)

Human neocortex and HTM neuron

The human brain (Fig.A.1(a)) has primarily three parts: neocortex, limbic part, and reptilian part. The limbic part handles the emotional feeling, the reptilian part supports survival instincts and the neocortex is responsible for human learning, cognition, and perception. The pyramidal neuron cell (Fig.A.1(b)) is the unit element of the neocortex and all the neurons in the neocortex are similar [355]. The functionality of this biological neuron is replicated in the HTM neuron cell [335]. This work uses cell/neuron/HTM neuron interchangeably from this point onwards. The neuron model of the HTM is depicted in Fig. A.1(c). A neuron is connected to three types of dendrite segments, namely (i) proximal dendrite segment

which receives inputs from the neurons of the lower layer, (ii) distal dendrite segment which comprises of synaptic connections with the other neurons in the layer above, and (iii) distal dendrite segment which comprises of the lateral connections with other cells in the same layer. The neurons are stacked one above the other to form a mini-column and a row of mini-columns form a single layer of the HTM. In our case, we include only a single layer of the HTM (Fig.A.1(d)). Therefore, the feedback connections from the layer above do not exist in the neuron model of ours. The lateral connections of the distal dendrites enable the HTM to learn the temporal relations in the data stream. This is the most critical aspect of the HTM that we leverage for detection and prediction. Each HTM neuron can be in three possible states similar to biological neuron, namely (i) inactive, (ii) predictive state, and (iii) active state. The default state of a neuron is inactive.

Encoder

We denote the sequence of incoming data by $(X_1, X_2....X_T)$, where $X_i \in \mathbb{R}^m, i \in \{1, 2, ..., T\}$, i denotes the time index and m is the dimension of the signal. The encoder converts the signal at each time instant i into a sparse distributed representation (SDR), which is a high dimensional binary representation of size n (Fig. A.1). SDRs are sparse representations where only a few bits are active for any input. In HTM this is typically set to 2% of the total size of the SDR that gives a good accuracy with a sparse representation of the incoming data. As a result, only fewer active bits overlap across different inputs. This is in contrast to dense representations, where many active bits can overlap.

Spatial Pooler

The next step is the spatial pooler. The spatial pooler computes a second SDR, which is the activation state of the mini-columns, of the same size as the input SDR. Each mini-column of an HTM is connected to a subset of the bits of the input SDR through synaptic connections

collectively called as proximal dendrite segment. Typically, each neuron of the mini-columns is connected to a large fraction of the bits of the input SDR (50%). Initially, the bits are randomly selected and could be fixed for the rest of the time. All the neurons in a single mini-column share the same proximal synapses. Each synapse has a permanence value, which determines whether a connection is existent or not. This value can be incremented or decremented to create new synaptic connections or remove existing synaptic connections. A synaptic connection is said to be active if the input to the connection is one. The mini-columns are rank-ordered based on the number of active connections. The mini-columns that are activated are a certain number of columns from the top of this ordered list. The activation state of the mini-columns is the output of the spatial pooler, and thus the output of the spatial pooler is also an SDR.

Temporal Pooler

The output SDRs from the spatial pooler are given as inputs to the temporal pooler. The temporal pooler consists of multiple mini-columns, and each mini-column has a fixed number of HTM neurons stacked upon one another. Multiple mini-columns are stacked side by side to form a cortical column (Fig. A.1). Each neuron of the mini-columns can comprise a minimum of two and sometimes up to a dozen distal dendrite segments. Each distal dendrite segment has synaptic connections that originate from multiple cells of the neighboring mini-columns in the same layer. The synaptic connections capture the temporal relations and constitute the temporal memory of an HTM. If there is an active synaptic connection between two cells of different mini-columns in the same layer, then a temporal relation exists between those cells. An active synaptic connection means that the cell from where the synaptic connection is originated is active. If the sum of the active synapses in any of the dendrite segments exceeds a certain threshold, then the cell enters the predictive state. The predictive state of a cell also provides the temporal context for the activation decision in the next time step.

The synaptic connections that present among different cells of the nearby mini-columns have weights. While learning, the weights of the inter-column synaptic connections are adjusted depending on the activation state of the cells in the predictive state in the next time step. If the predictive state in the current time step and the activation state in the next time step overlap, then it is taken to indicate that the temporal relation represented by the active synapses in the previous time step is correct. In such a case, weights of the active synaptic connections that correctly identified the predictive state are strengthened, and those that incorrectly identified or failed to identify are weakened. This is a Hebbian type learning and allows the HTM to learn a higher-order temporal representation of the sequential data, which can be used for prediction and detect anomalies.

Activation and Learning

Activation

Let's denote the current activation state of cells in a particular layer at time t by A^t (a $M \times N$ matrix where M is the number of cells per mini-column and N is the number of mini-columns in the layer), where $a^t_{i,j}$ is the i, jth element of A^t and denotes the activation state of cell i in column j. Let's denote a distal dendrite segment by d. Let the weight of the synapses of the dth segment of the ith cell of jth column be $D^d_{i,j}$. We note that only the weights of the synapses which are above a certain threshold are considered to be valid as a synaptic connection. The matrix of the established connection weights is denoted by $\tilde{D}^d_{i,j}$. The entries corresponding to the weights below the threshold are set to be zero in $\tilde{D}^d_{i,j}$.

Denote the predictive state of a neuron (i, j) by $\pi_{i,j}$. The neuron (i, j) is in a predictive state provided the sum of active synapses of at least one of the distal segments exceeds a certain threshold of θ_d. Thus, $\pi_{i,j}$ is given by,

$$\pi_{i,j}^{t} = \begin{cases} 1; & \text{if } \exists_d \ ||\tilde{D}_{i,j}^{d} \circ A^t||_1 > \theta_d, \\ 0; & \text{otherwise.} \end{cases} \tag{A.1}$$

where $\circ$ denotes the element-wise multiplication operation. Finally, only the cells of the mini-columns that were in the predictive states at time $t-1$ are activated. The activated cell is the cell of the active mini-column that was in the predictive state. The other cells in the mini-column are inhibited. This inhibition accounts for the specific temporal context as determined by the predictive states of the neurons in the mini-column. If none of the cells of an active mini-column are in a predictive state, then all the cells are activated. Let's denote the set of activated columns by the spatial pooler by C_a^t. Then the activation of a neuron (i, j) is given by,

$$a_{i,j}^{t} = \begin{cases} 1; & j \in C_a^t \text{ and } \pi_{i,j}^{t-1} = 1, \\ 1; & j \in C_a^t \text{ and } \sum_i \pi_{i,j}^{t-1} = 0, \\ 0; & \text{otherwise.} \end{cases} \tag{A.2}$$

Learning

The learning is a Hebbian type learning [356]. The learning algorithm only updates the weights of the synaptic connections of the cells that were in the predictive state or became active. If the predictive state of a neuron at the previous time step overlaps the activation state of a neuron at the current time step, then it is taken to indicate that the temporal relations captured by the synaptic connections are correct. The learning algorithm reinforces the temporal relations represented by the active synaptic connections. This results in the correct prediction. The learning algorithm also reduces the strength of the temporal relation represented by the inactive synaptic connections that failed to predict. Hence, the weights of

the active synaptic connections and the weights of the inactive synaptic connections of a cell are increased and decreased respectively. Formally, the weights $D^d_{i,j}$ of the distal segment d of cell (i, j) that became active at time t are changed by the adaptation rule given by,

$$\Delta D^d_{i,j} = r^+ \hat{D}^d_{i,j} \circ A^{t-1} - r^- \hat{D}^d_{i,j} \circ (1 - A^{t-1}) \tag{A.3}$$

where r^+ and r^- are the increase and decrease rates of the permanence values of the synaptic connections. The synaptic permanence increment and decrement rates (i.e., r^+ and r^-) are set to 0.1. The matrix $\hat{D}^d_{i,j}$ is the matrix of weights with positive entries,

$$\hat{D}^d_{i,j} = \begin{cases} 1; & \text{if } D^d_{i,j} > 0, \\ 0; & \text{otherwise.} \end{cases} \tag{A.4}$$

If a column becomes active and no neuron in the column was predicted to become active in the previous time step, then the neuron with the most activated segments is picked and updated as above.

If the neuron that was in the predictive state does not become active, then it indicates that the active lateral connections that resulted in the predictive state represent an incorrect temporal relation. So the active lateral segments of the cells that were in the predictive state but remained inactive are decreased at a rate of r^-_f. The synaptic permanence decrement for predicted inactive segments, r^-_f is set to 0.01 such that $r^-_f \ll r^-$. Formally, the weights of these active segments are decreased by the adaptation rule given by,

$$\Delta D^d_{i,j} = -r^-_f \hat{D}^d_{i,j} \text{ where } a^t_{i,j} = 0 \text{ and } ||\tilde{D}^d \circ A^{t-1}||_1 > \theta_d \tag{A.5}$$

In summary, the learning algorithm described here learns a temporal representation of the sequential data by adjusting the weights of the lateral connections between the cells. The adjustments of the weights are based on the correctness of the temporal relation represented by the synaptic connections.

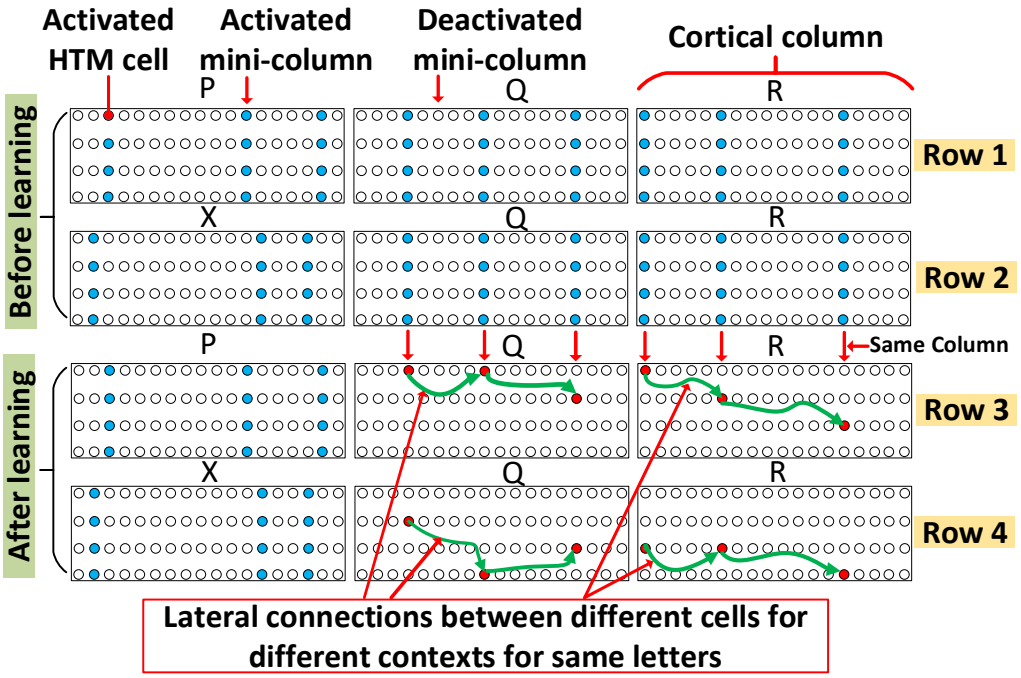


Figure A.2: Illustrative example of learning two different sequences: 'PQR' and 'XQR'.

Time complexity of the HTM

Here we discuss the time complexity of the three operations: encoder, spatial pooler, and temporal pooler.

The encoder converts the input data into a sparse vector. The encoder used in our model is a scalar encoder (refer to Table A.1). A bit is turned on in a scalar encoder if the value of the input falls within the window of the values the bit is associated with. Essentially, this operation entails checking which window the input value falls within. This requires time complexity of $O(W)$, where W is the number of windows the range of values are represented by. The maximum number of windows is limited by the size of the encoder. Therefore, the

overall time complexity of the encoder is $O(n)$ where n is the size of the encoder.

The spatial pooler converts the sparse vector computed by the encoder into a second sparse vector as outlined earlier. Each bit of the output of the spatial pooler is connected to as many as 50% of the bits of the input sparse vector. Let's denote the size of the sparsity by w (i.e., $w < 2\%$), which denotes the number of bits that will be active among the n bits of a vector and is typically a small number. First, the spatial pooler computes the number of proximal synaptic connections that are active for each bit in the output of the spatial pooler. For each bit, this is an $O(w)$ operation because the pooler only needs to check which of the active bits of the input SDR are connected to this particular bit in the output of the pooler. Therefore, the operation of computing the number of active connections for each of the n bits is in total $O(nw)$. Next, the spatial pooler orders the bits in descending order of the number of active connections and the top k bits are chosen. This is an $O(n \log n)$ operation. Therefore, the overall time complexity of the spatial pooler is $O(n \log n) + O(nw) \approx O(n \log n)$ (since $w \ll n$). Also, the final activation step described by Eqn. A.2 is just $O(wM)$, because this step is the computation to decide which of the M neurons of each of the w active mini-columns are to be activated.

The temporal pooler computes the predictive state of the neuron cells in the mini-columns as given by Eqn. A.1. This operation computes $\tilde{D}^d_{i,j} \circ A^t$ for each of the d segments of a neuron cell. For each segment the computation $\tilde{D}^d_{i,j} \circ A^t$ is an $O(w)$ operation. Therefore, when repeated for the d segments, it is an $O(dw) \approx O(w)$ operation. The predictive state is set based on this computation as described in Eqn. A.1. This is repeated for every neuron; therefore, the time complexity of the temporal pooler is $O(nMw) \approx O(n)$.

An illustrative example

In this section, we discuss an example that illustrates how the HTM learns to represent multiple temporal sequences even if there are overlaps between the sequences. The example

is shown in Fig. A.2. An HTM response for two sequences 'PQR' and 'XQR' is shown in Fig. A.2. Time advances from left to right in all the rows. Each ∘ in Fig. A.2 represents an HTM cell and these cells (in this case 4 cells) are stacked one on top of another to form a mini-column. Each region of the HTM contains 18 mini-columns, called as cortical column. The top two rows depict the cell firings before learning the temporal relations, and the bottom two rows depict the cell firings after learning the weights of the lateral connections.

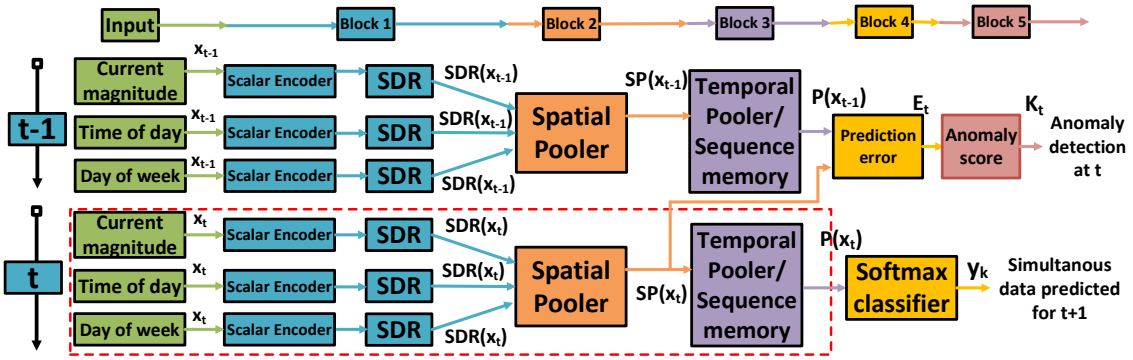


Figure A.3: HTM model for anomaly detection and simultaneous data prediction.

The response of the HTM before learning is shown in the top two rows. Before learning (the top two rows), the lateral synaptic connections between the neighboring cells (i.e., green connection in Fig. A.2) have not been learned yet, and so none of the cells are driven to be in the predictive state. As a result, all the cells (i.e., blue circles for activation in Fig. A.2) in an active mini-column are activated. We also note that only a very small fraction of the mini-columns are activated for every input.

After the lateral weights are learned (the bottom two rows), we observe that the same mini-column is invoked for the same letter but only one cell is activated in a mini-column this time. Though the cells activated are of the same mini-columns, the positions of the cells active in the same mini-columns are different. For example, the positions of the activations in the mini-columns that correspond to 'Q', shown in row-3, is completely different from the positions of the activations in row-4 for 'Q'. This is because the temporal context is different for the two scenarios depicted in row-3 and row-4 where in row-3 'Q' follows 'P' and in row-4

'Q' follows 'X'. For the same reason, the positions of the activations for 'R' are different in row-3 and row-4. Though 'R' follows 'Q' in both cases, the starting letters in the sequence are different ('P' and 'X') in the two cases. Therefore, the sparse encoding of 'R' not only depends on 'Q' but also on 'P' or 'X'. This clearly shows that the HTM learns higher-order overlapping temporal sequences.

It follows that if a temporal sequence is different but contains a letter that overlaps with another sequence, then a different cell of the mini-column corresponding to the common letter is activated for the two sequences. Thus, the different lateral connections responsible for activation for the different cells are strengthened over time. This allows the HTM to learn different temporal sequences even when there are overlaps between the sequences. *We emphasize that the column structure with multiple cells is the key structural aspect that allows the HTM to learn multiple temporal sequences.*

Source of μPMU dataset

To demonstrate anomaly detection and simultaneous multi-step prediction, we use *two open-source*, and real-time current magnitude datasets collected from the μPMU sensors installed at the Lawrence Berkeley National Laboratory's (LBNL) distribution grid [357]. This is the first μPMU network installed on a real electrical grid for research purposes, and the datasets collected from this network are the only available open-source datasets on real-time μPMU data, to the best of our knowledge [358]. We randomly pick two different datasets from the available LBNL's datasets to test the HTM. The first dataset (i.e., dataset 1) is collected from the *a6 bus 1* located in a 7.2 kV grid, and the second dataset (i.e., dataset 2) is collected from the low side of a 1500 kVA delta/wye transformer with a 480V/208V rating. The part name of the μPMU used to collect the two datasets is *PQube3*, which can output at 120 Hz frequency. Since data is collected in millisecond resolution and almost all practical events happen at a larger time scale [343], the raw data is resampled at 1-sec interval for 12 days

and 13 hours (1 Million+ data points). In the next section, we use the two datasets to show real-time anomaly detection in smart grids.

A.1.4 Demonstration of Anomaly Detection

The anomalies in a smart grid can be classified into two main categories: voltage or current magnitude variation and frequency variation. A voltage sag (a short term low voltage), a voltage or current spike (a short term high voltage or current above 110% normal value), a brownout (reduced voltage for an extended period), an overvoltage or overcurrent (an extended period of high voltage or current), etc., can be categorized as voltage or current magnitude anomalies. Frequency deviation from the normal limit can be categorized as a frequency variation. As noted earlier, μPMU sensors capture voltage or current magnitude and frequency information at ms timescales. In this section, we demonstrate that the HTM can be used to detect anomalies in μPMU data in real-time, while learning from just one-pass, and in an unsupervised fashion. We also demonstrate that learning is continuous, and by doing so, we show that the HTM model can potentially adapt to concept drifts.

Table A.1: Parameter setting

Time of day	Day of week	Current magnitude	Spatial Pooler	Temporal Pooler
Encoder type: Scalar	Encoder type: Scalar	Encoder type: Scalar	Column count, N: 2048	Column count, N: 2048
Maximum value: 60	Maximum value: 7	Maximum value: 40	Global inhibition: 1	Cells/Column: 16
Minimum value: 0	Minimum value: 0	Minimum value: 0	Seed: 1956	Max. synapses/segment: 32
Total bits, n: 600	Total bits, n: 100	Total bits, n: 109	Synaptic perm. con.: 0.5	New synapse count: 32
Total active bits, w: 29	Total active bits, w: 29	Total active bits, w: 29	Synaptic perm. act.: 0.0001	Synaptic perm. inc./dec.: 0.1

HTM implementation for anomaly detection

The architecture of the HTM model for real-time anomaly detection and simultaneous data prediction is presented in Fig. A.3. In this section, we only focus on the HTM implementation for anomaly detection, which has the following set of blocks. The first block is the scalar encoder that converts the data x_{t-1} at time step $t-1$ into a sparse distributed representation denoted by SDR(x_{t-1}) [359]. The second block is the spatial pooler [360] that transforms each SDR(x_{t-1}) matrix into a sparse binary vector representation, SP(x_{t-1}) [335] (see Section A.1.3). The final block is the temporal pooler, which generates P(x_{t-1}). The term P(x_{t-1}) is the prediction of SP(x_t) based on SP(x_{t-1}) at $t-1$, as described in Section A.1.3. The final block computes the error between the actual value, SP(x_t) and the predicted value, P(x_{t-1}) (computed at the previous time step) as given by [351],

$$E_t = 1 - \frac{SP(x_t) \circ P(x_{t-1})}{|SP(x_t)|} \tag{A.6}$$

where E_t is known as the prediction error at time-step t and $\circ$ is the dot product. $|SP(x_t)|$ is the modulus of the binary vector $SP(x_t)$. To assess how large the deviation E_t is, we compute a distribution of the deviations over a local window of time Δt. To compute this distribution, we compute the mean and standard deviation of the variables $E_{t'}$ ($t' \leq t$) over the window of time Δt that ends at the current time t. Denote the mean and standard deviation of $E_{t'}$ over this window Δt by μ_t and σ_t. Then μ_t and σ_t are given by,

$$\mu_t = \frac{\sum_{i=0}^{\Delta t-1} E_{t-i}}{\Delta t}, \tag{A.7}$$

$$\sigma_t^2 = \frac{\sum_{i=0}^{\Delta t-1}(E_{t-i} - \mu_t)^2}{\Delta t - 1}. \tag{A.8}$$

Then the anomaly score K_t is calculated using the Q function [361] that is given by,

$$K_t = 1 - Q\left(\frac{\mu_t^p - \mu_t}{\sigma_t}\right), \tag{A.9}$$

$$\text{where } \mu_t^p = \frac{\sum_{i=0}^{p-1} E_{t-i}}{p}. \tag{A.10}$$

Here the term μ_t^p is the calculated mean over a shorter time window, p. The value K_t is large if the value Q for μ_t^p is small, i.e., if the probability that a deviation is greater than μ_t^p is small. Hence, if K_t is greater than a certain threshold, the algorithm concludes that the probability of occurrence of an anomalous event is high and declares the current state to be anomalous.

Table A.1 shows the parameter values set for the HTM in the anomaly detection problem for both datasets. The same values are reused for the prediction problem in Section A.1.5 of the work. We choose this parameter setting because it was shown to work well for a wide range of datasets in other applications [351].

Capturing temporal and spatial anomalies

In smart grids, the current magnitude data from μPMU may have a significant amount of spatial and temporal fluctuations. The spatial fluctuations in voltage or current arise in real-time from sudden or unpredictable switching of large loads, sudden power outage in the same or nearby locality, or any sudden grid line faults. Temporal or contextual

fluctuations are variations that occur due to peak or off-peak hours of a day, day or night cycle, weekend or weekdays cycle, seasonal variations (temperature, humidity, etc.), etc. The two fluctuation types are of different nature. Spatial fluctuations are independent of any temporal context, whereas temporal fluctuations are dependent on specific contexts that are temporally extended. This makes the problem of detecting temporal anomalies challenging because learning higher-order sequences are harder.

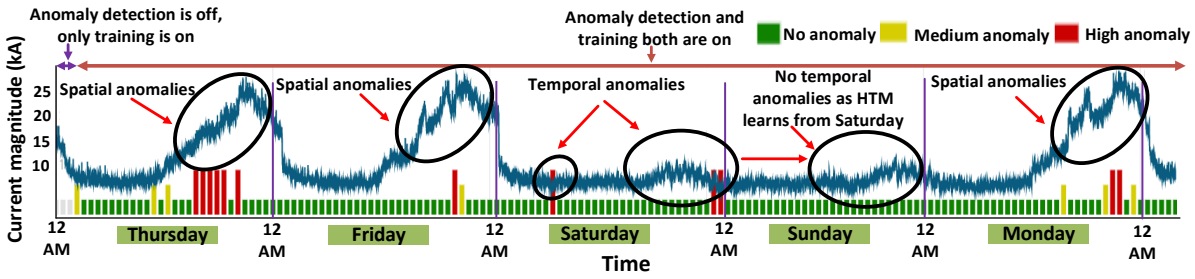


Figure A.4: Demonstration of capturing temporal and spatial anomalies by the HTM in an unsupervised fashion.

In the experiments for both datasets, we feed data into the HTM one at a time, and a *small initial amount of data* (i.e., first 1 hour 23 minutes or first 5000 data points of 1 million+ data points) is used for initial training of the temporal pooler of the HTM (the initial gray region in Fig. A.4). In this period, only the training is switched on and anomaly detection is switched off.

We demonstrate the effectiveness of the HTM in detecting spatial as well as subtle temporal anomalies present in the current magnitude of the μPMU data by using the dataset 1 in Fig. A.4. In Fig. A.4, we find that the HTM identifies six instances on *Thursday* night as anomalies (indicated by red bars). The HTM does so because the variations on Thursday night are being observed for the first time. The HTM learns this pattern after the first observation because we find that the HTM does not label most of the instances on the next occurrence of a similar pattern, which is on *Friday* night, as anomalous. Next, we observe that the HTM identifies two instances on *Saturday* night as anomalous. This is again because the pattern on Saturday night is a new pattern that has not been observed

on the previous days, i.e., on Thursday and Friday. This pattern is a temporal anomaly because it is a pattern that is typically observed on the weekends and not due to any spatial fluctuation. This suggests that the HTM identifies temporal anomalies. We also find that the HTM correctly identifies anomalies on Monday night. These are spatial anomalies because the pattern observed on Monday night shown here is a deviation from the pattern on the previous Thursday and Friday night and possibly is the result of a spatial fluctuation. Overall, the last two observations are suggestive that the HTM can identify both temporal and spatial anomalies.

Table A.2: Comparison among state-of-the-art real-time anomaly detection algorithms

Algorithms	Unsupervised	Noise immunity	Spatial anomaly	Temporal anomaly	Online learning	Non parametric[2]	Multi-step prediction	Time complexity[4]
HTM [362]	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	$O(n \log n)$
Random Cut Forest [363]	Yes	No	Yes	No	Yes	No	No	$O(D \log \frac{\|n\|}{L_1(u,v)})$
Bayesian Changepoint [364]	Yes	Yes	Yes	No	Yes	No	No	$O(n)$
Windowed Gaussian [365]	Yes	No	Yes	No	No	No	No	$O(nw^2)$
EXPoSE [366]	Yes	Yes	Yes	Yes	Yes	Yes	No [3]	$O(n)$
Relative Entropy [367]	Yes	Yes	Yes	Yes	Yes	Yes	No [3]	$O(n^\alpha), \alpha < 3$

Continuous online unsupervised learning

The challenge for the HTM is to learn the temporal patterns that are repetitious so that they are not wrongly identified as anomalies. It is clear that the weekends and the weekdays have different current magnitude patterns. In Fig. A.5, we demonstrate that the HTM does not identify any instance of the temporal pattern of the weekends on the second occurrence of the weekend as anomalous. We use the dataset 1 for this demonstration. This demonstration suggests that the HTM has learned the higher order temporal patterns that occur on the weekends on the first observation itself. This indicates that the HTM can learn in a

continuous-online fashion, which in turn is suggestive that it can account for *concept drift*. We also emphasize that the learning in HTM is without any human intervention and manual parameter tweaking, i.e., in an *unsupervised fashion*.

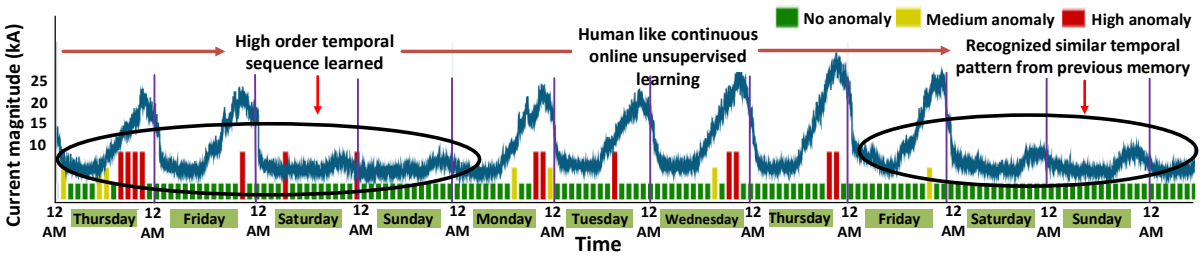


Figure A.5: Demonstration of continuous online unsupervised learning by the HTM.

Comparison with other anomaly detection algorithms

Anomaly detection algorithms can be either supervised or unsupervised. Supervised algorithms require labeled data and periodic retraining with changing conditions. The data presented here is unlabelled and so supervised learning approaches are not considered for comparison. Unsupervised algorithms can be of different types, such as simple or dynamic thresholding, complex statistical models, distant based methods, etc.

This work considers five state-of-the-art real-time, unsupervised algorithms, namely Windowed Gaussian (i.e., dynamic thresholding), Random Cut Forest (i.e., distance-based models), Bayesian Changepoint, EXPoSE, Relative Entropy (i.e., all three are complex statistical models) to compare with the HTM, and several of their properties are shown in Table A.2. This table indicates that the HTM can learn in one-pass and unsupervised fashion, detect spatial and temporal anomalies, and predict future observations in real-time. Moreover, the table also indicates that the HTM has a slightly higher time complexity compared to the other algorithms (except Random Cut Forest). The reason behind this is that the HTM learns long-term different temporal sequences even when there are overlaps between the sequences in its sparse sequential memory. The learning happens by adjusting the weights of the lateral connections between the cells. The slightly higher time complexity of the HTM

does not hamper its real-time anomaly detection capability in smart grids that is discussed in Section A.1.4.

Numenta Anomaly Benchmark (NAB)

We use the Numenta Anomaly Benchmark (NAB) [354] to compare the five algorithms with the HTM. Regular scoring methods, such as precision and recall cannot be used for scoring as they are not suitable for real-time problems. The NAB scoring mechanism has been designed based on what a good real-time detection algorithm should be able to do: *detect all anomalies in a streaming data, in real-time, with less false alarms, and in an automated fashion.*

The scoring mechanism contains three components: anomaly windows, application scoring profiles, and a scoring function. Anomaly windows are ranges of data points that surround each anomalous instances. The NAB score accounts for the differences in the importance of false positives and false negatives in applications by considering three different application dependent scoring profiles: (a) Standard, (b) Reward few false positives, and (c) Reward few false negatives.

The scoring function is such that an anomaly detected within the anomaly window is considered as true positive and given a positive score (e.g., Point 1 is given a score of +0.98 in Fig. A.6).

An anomaly detected outside the anomaly window is considered as false positive and given a negative score, which is also scaled depending on the position relative to the window. (e.g., Point 2 is given -0.95, and Point 3 is given -0.8 in Fig. A.6). The final scores for a detection d is calculated using the sigmoidal scoring function given by,

[2]Non-parametric refers to no requirement of application specific hyper-parametric tuning.

[3]Relative entropy based methods have been used for forecasting in economics [368], and methods like ExPoSE have been used for one-step prediction.

[4]Notation: n = point set; w = sparsity; D = dimension; L_1 = Manhattan distance; $(u, v) \subset n$.

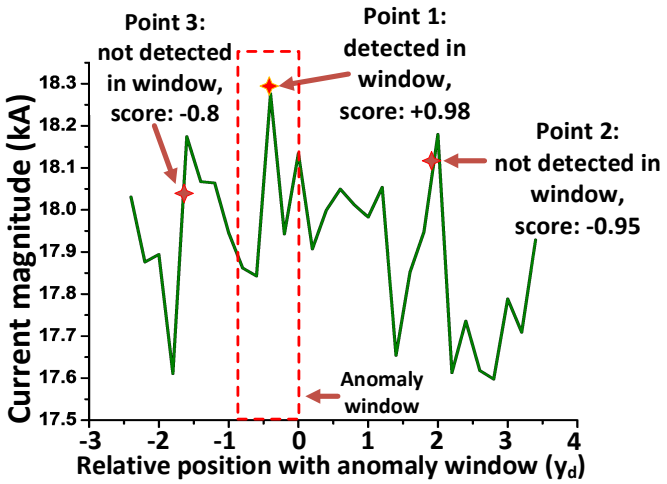


Figure A.6: NAB window and scoring process.

$$S^A(d) = (A_{TP} - A_{FP}) \left(\frac{1}{1 + e^{5y_d}} \right) - 1, \tag{A.11}$$

where A is the application profile under consideration (A $\subset$ {Standard, Reward few false positives, Reward few false negatives}), y_d is the relative position within the anomaly window, and A_{TP}, A_{FP} are weights which depend on the application profile A. Missing to detect any anomaly is considered as a false negative and is assigned a score of A_{FN}. The total score, TS^A over a dataset is calculated as,

$$TS^A = \left(\sum_{d=1}^{D} S^A(d) \right) + A_{FN} \times N_f, \tag{A.12}$$

where N_f is the total number of false negatives, and D is the total number of detected anomalies in the dataset.

Calculation of NAB score with latency time

Calculation of NAB score requires the ground truth informtion. A total of 15000 *unbiased anomaly points* have been identified and labelled as anomalies for both datasets. The calcu-

lated NAB scores for both datasets and the detection latency times for all the methods are shown in Table A.3.

Table A.3: NAB score board for both μPMU datasets

Algorithms	Latency (ms)	Standard		Reward few false positive		Reward few false negative	
		set 1	set 2	set 1	set 2	set 1	set 2
HTM	7.8	**87%**	**89%**	**83%**	**86%**	**89%**	**90%**
Random Cut Forest	19	15%	18%	12%	16%	34%	38%
Bayesian Changepoint	2.8	74%	72%	72%	70%	77%	80%
Windowed Gaussian	3.6	65%	69%	61%	65%	69%	71%
EXPoSE	2.1	67%	69%	63%	66%	71%	73%
Relative Entropy	0.5	20%	24%	21%	26%	20%	26%

Table A.3 shows that the HTM achieves a far-better score than other real-time anomaly detection algorithms in terms of detection accuracy for both datasets. However, the HTM has slightly higher detection latency (i.e., 7.8 ms) compared to other algorithms (except Random Cut Forest). The reason behind this is that the HTM has a slightly higher time complexity (see Table A.2) than others because of its sequence learning capability in the complex sequential memory. The detection latency times in Table A.3 are calculated in a 4.4 GHz Intel Core i9 processor with 16 cores and 32 GB of RAM. As the available voltage/current frequency in the smart grid is 50/60 Hz (i.e., period 20/16 ms), we want to emphasize that the detection latency of the HTM (i.e., 7.8 ms) is *low enough* to detect anomalies in smart grids in real-time. More precisely, the HTM is capable of detecting anomalies in less than half cycle time (full cycle = 20/16 ms, half cycle = 10/8 ms) in smart grids with good accuracy compared to other real-time anomaly detection algorithms. Moreover, the HTM model can be used for simultaneous *multi-step prediction* (Table A.2). Hence, we conclude that the HTM is competitive and has more capability than the other methods. This is primarily because the HTM *learns a general representation that can be used for multiple tasks* (in our case, prediction). We demonstrate this in the next section.

A.1.5 Demonstration of Multi-step Prediction

The smart grid is stochastic and dynamic in nature and predicting system behavior in real-time is critical from the point of system control. Though load forecasting [369] is a widely studied problem, in this section we focus on a specific prediction problem, namely short-term prediction (say 5 minutes ahead prediction). The real-time prediction is important from the point of view of control and planning of the grid operation. For example, it may be beneficial to predict a sudden change in power in the smart grid in real-time. This prediction can be used to respond early to compensate for this sudden change by regulating power flow to the affected part of the smart-grid. A common regulation method is to dynamically adjust the *governor* set point [370] of the generators. The set point adjustment can be made more reliable by taking into account the predictions of the transitions of the state of the grid. Here, we demonstrate that the same HTM that was trained for anomaly detection can be reused for predicting future observations in real-time.

HTM implementation for multi-step prediction

We have argued and shown that the temporal pooler of the HTM learns higher-order temporal sequences of the observed data in one-pass fashion. Therefore, the same HTM model can be *used for multi-step prediction in real-time* by adding a *softmax classifier* at the output of the temporal pooler. The softmax classifier outputs the class of the predicted state by the temporal pooler. The architecture for multi-step prediction is shown in Fig. A.3. For classification, the full range of the possible current magnitude values is divided into 22 disjoint classes (k). The classifier block is a single layer of a fully connected feed-forward network with the number of output neurons same as the number of classes. If the jth output neuron of the feed-forward network is given by a_j and the ith output of the temporal pooler is given by $P(x_t)_i$, then a_j is given by,

$$a_j = \sum_{i=1}^{n} \theta_{ij} P(x_t)_i, \tag{A.13}$$

where θ_{ij} is the weight of the connection from the ith neuron of the temporal pooler, $P(x_t)_i$, to the jth neuron of the feed-forward network, a_j, and n is the dimension of $P(x_t)_i$. The probability y_k of the predicted data falling into class k is calculated by the softmax function as follows:

$$y_k = \frac{e_k^a}{\sum_{i=1}^{k} e_i^a}. \tag{A.14}$$

The weights θ_{ij} are updated by the descent along the gradient of the least-squares error of the prediction. If the update of each weight θ_{ij} is denoted by $\Delta\theta_{ij}$, the term $\Delta\theta_{ij}$ can be expressed as follows:

$$\Delta\theta_{ij} = -\lambda(y_j - z_j)P(x_t)_i, \forall i, j, \tag{A.15}$$

where z_j is the observation and λ is the learning rate. As $P(x_t)_i$ is highly sparse, only a small portion of the weights are updated at any time and this results in faster prediction. The parameter settings of the HTM for the multi-step prediction is already listed in Table A.1.

Multi-step prediction on μPMU data using the HTM

Fig. A.7 demonstrates 5 min. ahead prediction on μPMU data for dataset 1. In this demonstration, the probabilities of all predicted data-classes are calculated using Eqn. A.14.

The value having the highest probability is chosen as the final predicted value. We note that *the prediction is in real-time.* The 'red' colored curve is the predicted curve and is shifted 5 time-steps to increase the visibility of the comparison between the actual and the predicted points. In all the circled regions of Fig. A.7, the predicted spikes ("red") follow an actual spike ("black") indicating that the HTM is able to predict the fluctuations or glitches. The predicted "red" curve may not exactly match the "black" curve point by point but we observe that the prediction by the HTM is able to capture the variations observed in the data. To prove this claim, in the next section, we quantify the HTM's performance using two Key Performance Indicators (KPIs) and compare it with other sequence prediction algorithms.

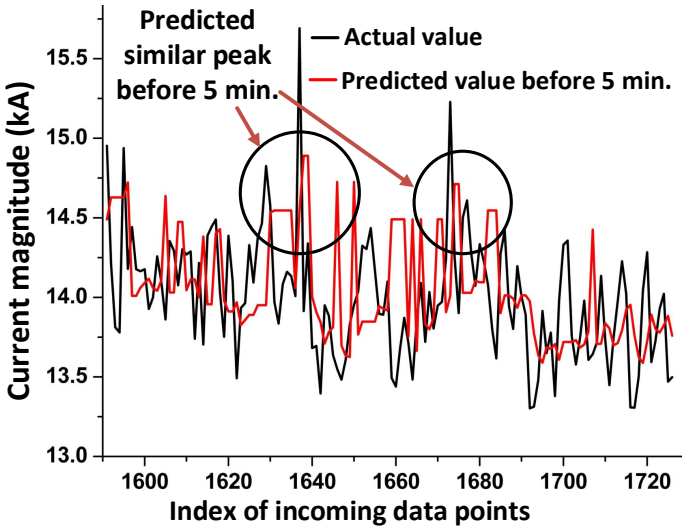


Figure A.7: Current magnitude prediction 5 min. ahead.

Comparison with other multi-step prediction algorithms

This section compares six state-of-the-art sequence prediction models, namely Adaptive Filter, Time Delayed Neural Network (TDNN) and four versions of LSTM with the HTM.

Adaptive Filter[371] is a self-learning predictive model that can adapt quickly in real-time. This model is implemented by using the LMS algorithm with a filter size of 10. TDNN [372] and LSTMs are widely used *state-of-the-art predictive models that require batches/mini-*

batches of data for training. Here, these models are made adaptive and partially online by retraining them at different time-steps using different ranges of past data points. TDNN is implemented with 100 input units, 1 hidden layer with 200 units, 1 output unit and retrained after every 336 time-steps using the last 3000 data points.

Here, we consider four versions of the LSTMs, namely LSTM-Online, LSTM-1000, LSTM-3000, and LSTM-6000. They are implemented with 3 input units, 20 LSTM units, and 1 output unit. LSTM-Online is retrained at every time-step using the last 100 data points, whereas LSTM-1000, LSTM-3000, and LSTM-6000 are retrained at every 1000th time-step using the last 1000, 3000, and 6000 data points, respectively. For retraining, different time intervals and different ranges of data points have been selected to illustrate the effectiveness of the HTM as a one-pass learner. Though the performance of LSTM is observed to improve with the increase in the data points for retraining, we emphasize that the HTM is able to achieve the same or even better performance just by learning from one-pass over the data.

The qualitative comparisons of the algorithms are given in Table A.4. The comparison is clearly suggestive that the HTM is effective in capturing long term dependency while learning from just one-pass over the data. Moreover, the HTM can quickly adapt to concept drifts without much parameter tuning compared to other methods.

Table A.4: Comparison among state-of-the-art sequence prediction algorithms.

Algorithms	One-Pass Learning	Time to Adapt	Long Term Dependency	Non Parametric
HTM [373]	**Yes**	**Short**	**Yes**	**Yes**
Adaptive Filter[374]	Yes	Short	Limited	No
TDNN [375]	No	Long	Limited	No
LSTM-Online [376]	Yes	Long	Yes	No
LSTM-1000 [376]	No	Long	Yes	No
LSTM-3000 [376]	No	Long	Yes	No
LSTM-6000 [376]	No	Long	Yes	No

To quantitatively illustrate the multi-step prediction accuracy of the HTM, we compare the HTM with other algorithms using two KPIs, namely Normalized Root Mean Square Error (NRMSE) and Negative Log-likelihood (NLL), on both datasets. NRMSE is sensitive to

outliers and low NRMSE value indicates an almost pointwise perfect fit between the true and the predicted values. The NRMSE is calculated as follows:

$$NRMSE = \sqrt{\frac{\sum_{t=1}^{N_d}(Pred_t - True_t)^2}{\sigma}} \tag{A.16}$$

where σ is the standard deviation of the actual data, N_d is the total number of data points, and $Pred_t$ is the predicted value of a true value $True_t$ at time t.

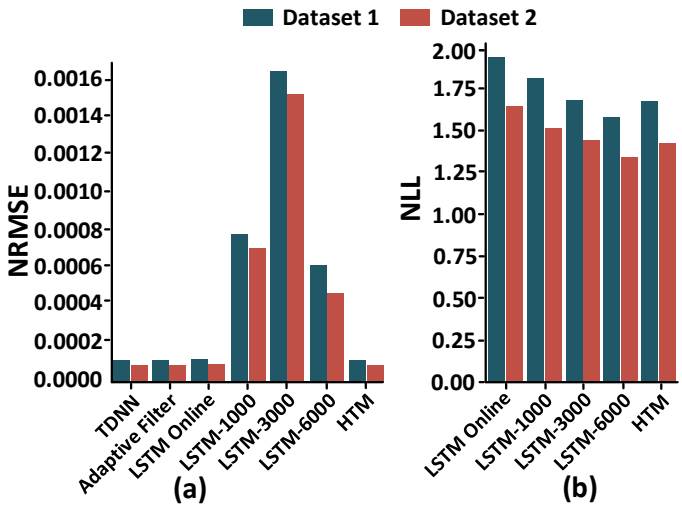


Figure A.8: Comparison of NRMSE and NLL values.

The NRMSE score only considers the point-wise prediction error. To measure the accuracy of prediction of a sequence we consider an alternate measure, the Negative Log-likelihood (NLL). The NLL score is computed using the probability function $P(y_t|y_1.....y_{t-1})$. This function captures the dependency of the prediction y_t on the predicted values at the previous time-steps, which are $y_1, ...y_{t-1}$. The NLL score is simply the negative of the log of this probability:

$$NLL = -\frac{1}{N_d}\sum_{t=1}^{N_d} log(P(y_t|y_1.....y_{t-1})), \tag{A.17}$$

where N_d is the total number of data points. Here, the lesser NLL score indicates more accurate sequence prediction.

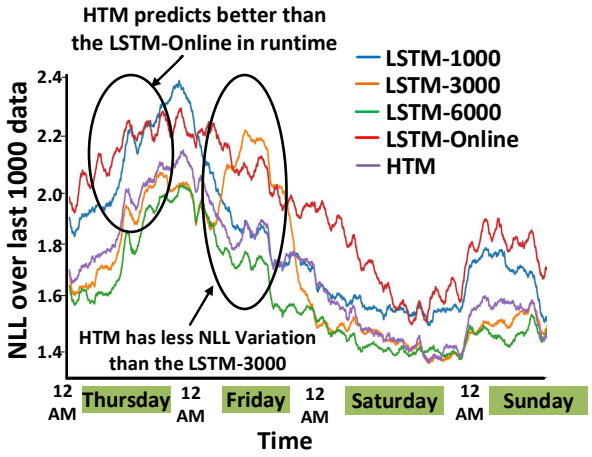


Figure A.9: Comparison of NLL value over last 1000 data points.

Fig. A.8(a) shows that HTM, TDNN, Adaptive Filter, and LSTM-Online have a similar and a lower NRMSE than LSTM-1000, LSTM-3000, LSTM-6000 models for both datasets. We attribute this to the presence of frequent outliers in the predicted data points of the LSTM-1000, LSTM-3000, LSTM-6000 models. This suggests that HTM, TDNN, LSTM-Online, and adaptive filters are more accurate. Fig. A.8(b) shows the NLL score for all the predicted data points. Adaptive filter and TDNN are not considered for the comparison because they have limited capability in capturing long term dependency (Table A.4). The NLL score of the HTM is better than the LSTM-Online, LSTM-1000 models and is similar to LSTM-3000. The NLL score of LSTM-6000 is slightly better than the HTM. We attribute this to multiple retraining with a larger set of previous data points, which is 6000 in this case. *However, the HTM is more effective because it achieves a similar score by learning from just one-pass, whereas LSTMs require multiple retraining over the same set of data points.*

Fig. A.9 shows the plot of NLL value computed over the last 1000 data points for dataset 1. It is clear from Fig. A.9 that HTM, LSTM-3000, and LSTM-6000 show a variation of NLL that over time is lower than the LSTM-Online and LSTM-1000 models. *Though the HTM has the same NLL score as LSTM-3000 (Fig. A.8(b)), it is observed to exhibit less NLL variation than LSTM-3000 overall (Fig. A.9).*

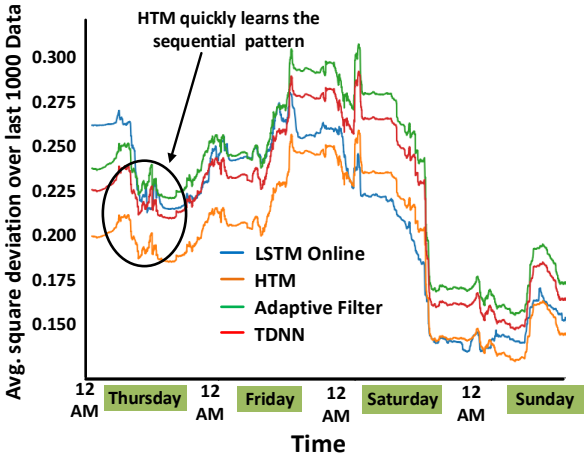


Figure A.10: Comparison of average square deviation over last 1000 data points.

Fig. A.10 shows the variation of average square deviation error where each value is the average over the last 1000 data points. It is clear from Fig. A.10 that the HTM initially quickly learns the sequential pattern and exhibits lower square deviation error than the LSTM-Online, Adaptive Filter, and TDNN. Later on, we observe that the HTM consistently exhibits similar or lower square deviation error compared to the other methods. This proves that the HTM learns faster than the LSTM-Online, Adaptive Filter, and TDNN. As LSTM-1000, LSTM-3000, and LSTM-6000 require retraining, they learn much slowly compared to the HTM; therefore, they are not considered here for comparisons.

We observe that the NLL in Fig. A.9 and the square deviation error in Fig. A.10 have higher values on Thursday and Friday (i.e., weekdays) compared to Saturday and Sunday (i.e, weekends). The reason behind this is that the current magnitude is more stochastic on the weekdays compared to the weekends because the variation of loads on the weekdays is

higher than the weekends. In other words, the loads on the weekends are more deterministic than the loads on the weekdays. We also observe that loads on Sunday are more stochastic than the loads on Saturday. Therefore, the NLL and the square deviation error on Sunday have slightly higher values than Saturday in Fig. A.9 and Fig. A.10.

A.1.6 Limitations and Future Work

The compelling feature of the proposed HTM is that it can detect anomalies and simultaneously predict future observations in real-time. Moreover, the HTM can learn in just one-pass and in an unsupervised fashion and can handle *concept drifts* efficiently. But the limitation is that the current implementation can not classify the anomalies based on the cause of the anomaly (e.g., a grid-line fault or a sudden load transient). Our future work is to extend the HTM model for real-time anomaly classification and simultaneous data prediction not only in smart grids but also in other CPSs applications [27, 73].

A.1.7 Summary

μPMU sensors sample the grid voltage/current waveform in an ultra-precise and synchronized fashion. Therefore, they can support many diagnostic applications, such as anomaly detection and simultaneous data prediction in real-time. In this chapter, we introduced a neuro-inspired architecture called the HTM and discussed its structure and cortical learning algorithm. The HTM learns a general temporal sparse representation that can be leveraged for multiple tasks. The HTM can also be trained continuously, in one-pass and in an unsupervised fashion. This makes them suitable for real-time applications. In this work, we demonstrate how the HTM can be used for anomaly detection and simultaneous multi-step prediction in real-time. To demonstrate the effectiveness of the HTM, it is compared with five state-of-the-art real-time anomaly detection algorithms and six other state-of-the-art prediction algorithms. We showed that the HTM achieves competitive scores on both these

tasks, while the other state of the art algorithms are either less accurate or can be used for one of the tasks only. *Moreover, the HTM achieves this performance by learning in just one-pass, and in an unsupervised fashion. We strongly believe that this work will serve as a motivation for research on neuro-inspired machine learning algorithms in the smart grid by demonstrating that such algorithms are more effective for real-time applications.*

Bibliography

[1] Ragunathan Rajkumar, Insup Lee, Lui Sha, and John Stankovic. Cyber-physical systems: the next computing revolution. In *Proceedings of the 47th design automation conference*, pages 731–736, 2010.

[2] Edward 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. IEEE, 2008.

[3] Edward A Lee. Cps foundations. In *Proceedings of the 47th design automation conference*, pages 737–742, 2010.

[4] Ahmadzai Ahmadi, Chantal Cherifi, Vincent Cheutet, and Yacine Ouzrout. A review of cps 5 components architecture for manufacturing based on standards. In *2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, pages 1–6. IEEE, 2017.

[5] Yihai Fang, Nazila Roofigari-Esfahan, and Chimay Anumba. A knowledge-based cyber-physical system (cps) architecture for informed decision making in construction. In *Construction Research Congress 2018*, pages 662–672, 2018.

[6] Sujit Rokka Chhetri, Sina Faezi, Arquimedes Canedo, and Mohammad Abdullah Al Faruque. Poster abstract: Thermal side-channel forensics in additive manufacturing systems. In *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, pages 1–1, 2016. doi: 10.1109/ICCPS.2016.7479115.

[7] Mohammad Abdullah Al Faruque, Sujit Rokka Chhetri, Arquimedes Canedo, and Jiang Wan. Acoustic side-channel attacks on additive manufacturing systems. In *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, pages 1–10, 2016. doi: 10.1109/ICCPS.2016.7479068.

[8] Sujit Rokka Chhetri and Mohammad Abdullah Al Faruque. Side channels of cyber-physical systems: Case study in additive manufacturing. *IEEE Design and Test*, 34 (4):18–25, 2017. doi: 10.1109/MDAT.2017.2682225.

[9] Sujit Rokka Chhetri, Sina Faezi, and Mohammad Abdullah Al Faruque. Fix the leak! an information leakage aware secured cyber-physical manufacturing system. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, pages 1408–1413, 2017. doi: 10.23919/DATE.2017.7927213.

[10] Anomadarshi Barua and Mohammad Abdullah Al Faruque. Special session: Noninvasive sensor-spoofing attacks on embedded and cyber-physical systems. In *2020 IEEE 38th International Conference on Computer Design (ICCD)*, pages 45–48, 2020. doi: 10.1109/ICCD50377.2020.00024.

[11] Públio M Lima, Lilian K Carvalho, and Marcos V Moreira. Ensuring confidentiality of cyber-physical systems using event-based cryptography. *Information Sciences*, 621: 119–135, 2023.

[12] Sujit Rokka Chhetri, Arquimedes Canedo, and Mohammad Abdullah Al Faruque. Confidentiality breach through acoustic side-channel in cyber-physical additive manufacturing systems. *ACM Trans. Cyber-Phys. Syst.*, 2(1), dec 2017. ISSN 2378-962X. doi: 10.1145/3078622. URL `https://doi.org/10.1145/3078622`.

[13] Sujit Rokka Chhetri, Sina Faezi, and Mohammad Abdullah Al Faruque. Information leakage-aware computer-aided cyber-physical manufacturing. *IEEE Transactions on Information Forensics and Security*, 13(9):2333–2344, 2018. doi: 10.1109/TIFS.2018.2818659.

[14] Shih-Yuan Yu, Arnav Vaibhav Malawade, Sujit Rokka Chhetri, and Mohammad Abdullah Al Faruque. Sabotage attack detection for additive manufacturing systems. *IEEE Access*, 8:27218–27231, 2020. doi: 10.1109/ACCESS.2020.2971947.

[15] Ahmed Didouh, Anthony Bahadir Lopez, Yassin El Hillali, Atika Rivenq, and Mohammad Abdullah Al Faruque. Eve, you shall not get access! a cyber-physical blockchain architecture for electronic toll collection security. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7, 2020. doi: 10.1109/ITSC45102.2020.9294334.

[16] Rozhin Yasaei, Felix Hernandez, and Mohammad Abdullah Al Faruque. Iot-cad: Context-aware adaptive anomaly detection in iot systems through sensor association. In *Proceedings of the 39th International Conference on Computer-Aided Design*, ICCAD '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380263. doi: 10.1145/3400302.3415672. URL `https://doi.org/10.1145/3400302.3415672`.

[17] Michael Weiß, Benjamin Weggenmann, Moritz August, and Georg Sigl. On cache timing attacks considering multi-core aspects in virtualized embedded systems. In *Trusted Systems: 6th International Conference, INTRUST 2014, Beijing, China, December 16-17, 2014, Revised Selected Papers 6*, pages 151–167. Springer, 2015.

[18] Kelvin Ly and Yier Jin. Security challenges in cps and iot: From end-node to the system. In *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 63–68. IEEE, 2016.

[19] Sina Faezi, Rozhin Yasaei, Anomadarshi Barua, and Mohammad Abdullah Al Faruque. Brain-inspired golden chip free hardware trojan detection. *IEEE Transactions on*

Information Forensics and Security, 16:2697–2708, 2021. doi: 10.1109/TIFS.2021.3062989.

[20] Rozhin Yasaei, Luke Chen, Shih-Yuan Yu, and Mohammad Abdullah Al Faruque. Hardware trojan detection using graph neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 1–1, 2022. doi: 10.1109/TCAD.2022.3178355.

[21] Yicheng Zhang, Rozhin Yasaei, Hao Chen, Zhou Li, and Mohammad Abdullah Al Faruque. Stealing neural network structure through remote fpga side-channel analysis. *IEEE Transactions on Information Forensics and Security*, 16:4377–4388, 2021. doi: 10.1109/TIFS.2021.3106169.

[22] Sina Faezi, Rozhin Yasaei, and Mohammad Abdullah Al Faruque. Htnet: Transfer learning for golden chip-free hardware trojan detection. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1484–1489, 2021. doi: 10.23919/DATE51398.2021.9474076.

[23] Sina Faezi e al. Oligo-snoop: a non-invasive side channel attack against dna synthesis machines. In *Network and Distributed Systems Security (NDSS) Symposium 2019*.

[24] Rozhin Yasaei, Sina Faezi, and Mohammad Abdullah Al Faruque. Golden reference-free hardware trojan localization using graph convolutional network. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 30(10):1401–1411, 2022. doi: 10.1109/TVLSI.2022.3191683.

[25] Shih-Yuan Yu, Rozhin Yasaei, Qingrong Zhou, Tommy Nguyen, and Mohammad Abdullah Al Faruque. Hw2vec: a graph learning tool for automating hardware security. In *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 13–23, 2021. doi: 10.1109/HOST49136.2021.9702281.

[26] Rozhin Yasaei, Shih-Yuan Yu, and Mohammad Abdullah Al Faruque. Gnn4tj: Graph neural networks for hardware trojan detection at register transfer level. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1504–1509, 2021. doi: 10.23919/DATE51398.2021.9474174.

[27] Anomadarshi Barua and Mohammad Abdullah Al Faruque. Hall Spoofing: A {Non-Invasive}{DoS} Attack on {Grid-Tied} Solar Inverter. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1273–1290, 2020.

[28] Alvaro Cardenas, Saurabh Amin, Bruno Sinopoli, Annarita Giani, Adrian Perrig, Shankar Sastry, et al. Challenges for securing cyber physical systems. In *Workshop on future directions in cyber-physical systems security*, volume 5. Citeseer, 2009.

[29] Suhail Qadir and SMK Quadri. Information availability: An insight into the most important attribute of information security. *Journal of Information Security*, 7(03): 185, 2016.

[30] Guangyu Wu et al. A survey on the security of cyber-physical systems. *Control Theory and Technology*, 14(1):2–10, 2016.

[31] Ang Chee Kiong Gary and Utomo Nugroho Prananto. Cyber security in the energy world. In *2017 Asian Conference on Energy, Power and Transportation Electrification (ACEPT)*, pages 1–5. IEEE, 2017.

[32] Blake Sobczak. Experts assess damage after first cyberattack on U.S. grid, May 6, 2019. `https://www.eenews.net/stories/1060281821`. (Accessed: 05-14-2020).

[33] Kevin Poulsen. Slammer worm crashed Ohio nuke plant net. *The Register*, 20, 2003.

[34] Nicolas Falliere, Liam O Murchu, and Eric Chien. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5(6):29, 2011.

[35] Kim Zetter. Inside the cunning, unprecedented hack of ukraine's power grid. *Wired*, 2016.

[36] Johannes Reichl, Michael Schmidthaler, and Friedrich Schneider. The value of supply security: The costs of power outages to Austrian households, firms and the public sector. *Energy Economics*, 36:256–261, 2013.

[37] Michaela D Platzer. US solar photovoltaic manufacturing: Industry trends, global competition, federal support. Library of Congress, Congressional Research Service, 2012.

[38] Phillip Brown. European Union wind and solar electricity policies: overview and considerations, 2013.

[39] Søren Lund Lorenzen, Alex Buus Nielsen, and Lorand Bede. Control of a grid connected converter during weak grid conditions. In *2016 IEEE 7th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, pages 1–6. IEEE, 2016.

[40] Robert H Lasseter and Paolo Piagi. Microgrid: A conceptual solution. In *IEEE Power Electronics Specialists Conference*, volume 6, pages 4285–4291. Citeseer, 2004.

[41] Mark Yampolskiy, Peter Horvath, Xenofon D Koutsoukos, Yuan Xue, and Janos Sztipanovits. Taxonomy for description of cross-domain attacks on CPS. In *Proceedings of the 2nd ACM international conference on High confidence networked systems*, pages 135–142. ACM, 2013.

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

[43] 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)*, 2016.

[44] Drew Davidson, Hao Wu, Rob Jellinek, Vikas Singh, and Thomas Ristenpart. Controlling UAVs with sensor input spoofing attacks. In *10th {USENIX} Workshop on Offensive Technologies ({WOOT} 16)*, 2016.

[45] Chen Yan, Wenyuan Xu, and Jianhao Liu. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *DEF CON*, 24(8):109, 2016.

[46] Hocheol Shin, Dohyun Kim, Yujin Kwon, and Yongdae Kim. Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 445–467. Springer, 2017.

[47] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 103–117, 2017.

[48] Yasser Shoukry, Paul Martin, Paulo Tabuada, and Mani Srivastava. Non-invasive spoofing attacks for anti-lock braking systems. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 55–72. Springer, 2013.

[49] 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 *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pages 881–896, 2015.

[50] Zhengbo Wang et al. Sonic gun to smart devices: Your devices lose control under ultrasound/sound. *BlackHat USA*, 2017.

[51] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin 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)*, pages 3–18. IEEE, 2017.

[52] Yazhou Tu, Zhiqiang Lin, Insup Lee, and Xiali Hei. Injected and delivered: Fabricating implicit control over actuation systems by spoofing inertial sensors. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1545–1562, 2018.

[53] Oliver Kosut, Liyan Jia, Robert J Thomas, and Lang Tong. Malicious data attacks on the smart grid. *IEEE Transactions on Smart Grid*, 2(4):645–658, 2011.

[54] Elias Bou-Harb, Claude Fachkha, Makan Pourzandi, Mourad Debbabi, and Chadi Assi. Communication security for smart grid distribution networks. *IEEE Communications Magazine*, 51(1):42–49, 2013.

[55] Yilin Mo, Tiffany Hyun-Jin Kim, Kenneth Brancik, Dona Dickinson, Heejo Lee, Adrian Perrig, and Bruno Sinopoli. Cyber–physical security of a smart grid infrastructure. *Proceedings of the IEEE*, 100(1):195–209, 2011.

[56] Arman Sargolzaei, Kang K Yen, and Mohamed N Abdelghani. Preventing time-delay switch attack on load frequency control in distributed power systems. *IEEE Transactions on Smart Grid*, 7(2):1176–1185, 2015.

[57] Amir-Hamed Mohsenian-Rad and Alberto Leon-Garcia. Distributed internet-based load altering attacks against smart power grids. *IEEE Transactions on Smart Grid*, 2 (4):667–674, 2011.

[58] Ilge Akkaya, Edward A Lee, and Patricia Derler. Model-based evaluation of GPS spoofing attacks on power grid sensors. In *2013 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, pages 1–6. IEEE, 2013.

[59] Eduard Muljadi, CP Butterfield, Brian Parsons, and Abraham Ellis. Effect of variable speed wind turbine generator on stability of a weak grid. *IEEE Transactions on Energy Conversion*, 22(1):29–36, 2007.

[60] Stephen J Chapman et al. *Electric machinery and power system fundamentals*. 2002.

[61] Harold Kirkham. Current measurement methods for the smart grid. In *2009 IEEE Power & Energy Society General Meeting*, pages 1–7. IEEE, 2009.

[62] Grid-tied Solar Micro Inverter with MPPT Schematic (Rev. A). page 4, . `http://www.ti.com/lit/df/tidr767a/tidr767a.pdf`. (Accessed: 05-12-2020).

[63] 10kW 3-Level 3-Phase Grid Tie Inverter Reference Design for Solar String Inverts (Rev. A). page 1, . `http://www.ti.com/lit/pdf/tidue53`. (Accessed: 05-12-2020).

[64] AN4070: 250 W grid connected microinverter. page 6. `https://www.st.com/content/ccc/resource/technical/document/application_note/fa/f1/fe/3d/81/1e/47/45/DM00050692.pdf/files/DM00050692.pdf/jcr:content/translations/en.DM00050692.pdf`. (Accessed: 05-12-2020).

[65] AN1444: Grid-Connected Solar Microinverter Reference Design. page 15. `http://ww1.microchip.com/downloads/en/appnotes/01444a.pdf`. (Accessed: 05-12-2020).

[66] Steve Taranovich. Teardown: The power inverter – from sunlight to power grid. `https://www.edn.com/teardown-the-power-inverter-from-sunlight-to-power-grid/`. (Accessed: 05-12-2020).

[67] Solar Inverter. `https://solarpv4u.co.nz/solar-inverters`. (Accessed: 05-12-2020).

[68] Jonathan Stidham. Can hackers turn your lights off: The vulnerability of the US power grid to electronic attack. *SANS Institute InfoSec Reading Room*, 2001.

[69] Jason Staggs. Breaking wind: Adventures in hacking wind farm control networks. *Black Hat*, 2017.

[70] J.R. Appelbaum, L. Poitras, M. Rosenbach, C. Stöcker, J. Schindler, and H. Stark. Inside TAO : documents reveal top NSA hacking unit. *Der Spiegel*, 12 2013. ISSN 0038-7452.

[71] Lonneke Van der Velden. Leaky apps and data shots: Technologies of leakage and insertion in NSA-surveillance. *Surveillance & Society*, 13(2):182–196, 2015.

[72] Bill Snyder. Snowden: The NSA planted backdoors in cisco products. *InfoWorld*, 15, 2014.

[73] Sujit Rokka Chhetri et al. Tool of spies: Leaking your ip by altering the 3d printer compiler. *IEEE Transactions on Dependable and Secure Computing*, 2019.

[74] Pawel Swierczynski, Marc Fyrbiak, Philipp Koppe, Amir Moradi, and Christof Paar. Interdiction in practice—Hardware Trojan against a high-security USB flash drive. *Journal of Cryptographic Engineering*, 7(3):199–211, 2017.

[75] Benjamin Sprecher, Rene Kleijn, and Gert Jan Kramer. Recycling potential of neodymium: the case of computer hard disk drives. *Environmental science & technology*, 48(16):9506–9513, 2014.

[76] J David Irwin. *Control in power electronics: selected problems*. Elsevier, 2002.

[77] Junjian Qi et al. Cybersecurity for distributed energy resources and smart inverters. *IET Cyber-Physical Systems: Theory & Applications*, 1(1):28–39, 2016.

[78] Vikram Kaura and Vladimir Blasko. Operation of a phase locked loop system under distorted utility conditions. *IEEE Transactions on Industry applications*, 33(1):58–63, 1997.

[79] Laurent Chiesi, Karim Haroud, John A Flanagan, and Rade S Popovic. Chopping of a weak magnetic field by a saturable magnetic shield. *Sensors and Actuators A: Physical*, 60(1-3):5–9, 1997.

[80] Charles Steinmetz. *Theory and Calculation of Electric Circuits*. The McGraw-Hill Companies, 1.00 edition, 1917. `https://books.google.com/books?id=z0IOAAAAYAAJ&pg=PA84#v=onepage&q&f=false`. (Accessed: 05-11-2020).

[81] INDUCTORS AND TRANSFORMERS. `https://www.ece.k-state.edu/people/faculty/gjohnson/files/tcchap4.pdf`. (Accessed: 05-11-2020).

[82] The Tesla Radio Conspiracy. `http://teslaradioconspiracy.blogspot.com/`. (Accessed: 05-11-2020).

[83] Loudspeaker Power Handling Vs. Efficiency. `https://sound-au.com/articles/pwr-vs-eff.htm`. (Accessed: 05-11-2020).

[84] DP Hohm and M E_ Ropp. Comparative study of maximum power point tracking algorithms. *Progress in photovoltaics: Research and Applications*, 11(1):47–62, 2003.

[85] Yanjun Shi, Lu Wang, Ren Xie, and Hui Li. Design and implementation of a 100 kW SiC filter-less PV inverter with 5 kW/kg power density and 99.2% CEC efficiency. In *2018 IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 393–398. IEEE, 2018.

[86] Frede Blaabjerg, Remus Teodorescu, Marco Liserre, and Adrian V Timbus. Overview of control and grid synchronization for distributed power generation systems. *IEEE Transactions on industrial electronics*, 53(5):1398–1409, 2006.

[87] Mihai Ciobotaru, Remus Teodorescu, and Frede Blaabjerg. Control of single-stage single-phase PV inverter. *EPE Journal*, 16(3):20–26, 2006.

[88] Yanjun Shi, Lu Wang, Ren Xie, Yuxiang Shi, and Hui Li. A 60-kW 3-kW/kg five-level T-type SiC PV inverter with 99.2% peak efficiency. *IEEE Transactions on Industrial Electronics*, 64(11):9144–9154, 2017.

[89] Enclosures for the Solar Industry. `https://fiboxusa.com/enclosures-for-solar-power/`. (Accessed: 05-11-2020).

[90] William Edwards and Scott Manson. Using protective relays for microgrid controls. In *2018 71st Annual Conference for Protective Relay Engineers (CPRE)*, pages 1–7. IEEE, 2018.

[91] Distributed Generation Photovoltaics and Energy Storage. IEEE standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces. *IEEE Std*, pages 1547–2018, 2018.

[92] James Glanz and Brad Plumer. In a High-Tech State, Blackouts Are a Low-Tech Way to Prevent Fires. `https://www.nytimes.com/2019/10/12/business/power-blackouts-california-microgrids.html` (Accessed: 05-11-2020).

[93] Amjad Ali, Wuhua Li, Rashid Hussain, Xiangning He, Barry W Williams, and Abdul Hameed Memon. Overview of current microgrid policies, incentives and barriers in the European Union, United States and China. *Sustainability*, 9(7):1146, 2017.

[94] Schatz Energy Research Center. Blue Lake Rancheria microgrid. `http://schatzcenter.org/blrmicrogrid/`. (Accessed: 05-11-2020).

[95] Magnetic Field of Current. `http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/magcur.html`. (Accessed: 05-11-2020).

[96] Takashi Sato, Toshio Yamada, and Masami Kobayashi. Magnetic shielding material, September 3 1991. US Patent 5,045,637.

[97] Warren R Osborn and Bryan P Dunford. Protective container for readable cards, January 16 2007. US Patent 7,163,152.

[98] `https://www.allegromicro.com/~/media/Files/Datasheets/ACS724-Datasheet.ashx`. (Accessed: 05-14-2020).

[99] Managing External Magnetic Field Interference When Using ACS71x Current Sensor ICs. https://www.allegromicro.com/en/Insights-and-Innovations/Technical-Documents/Hall-Effect-Sensor-IC-Publications/Managing-External-Magnetic-Field-Interference-ACS71x-Current-Sensor-ICs.aspx. (Accessed: 05-11-2020.

[100] Anomadarshi Barua and Mohammad Abdullah Al Faruque. A Wolf in Sheep's Clothing: Spreading Deadly Pathogens Under the Disguise of Popular Music. In *29th ACM Conference on Computer and Communications Security (CCS)*, 2022.

[101] Lisa Ta, Laura Gosa, and David A Nathanson. Biosafety and biohazards: understanding biosafety levels and meeting safety requirements of a biobank. *Biobanking*, pages 213–225, 2019.

[102] George F Risi, Marshall E Bloom, Nancy P Hoe, Thomas Arminio, Paul Carlson, Tamara Powers, Heinz Feldmann, and Deborah Wilson. Preparing a community hospital to manage work-related exposures to infectious agents in biosafety level 3 and 4 laboratories. *Emerging infectious diseases*, 16(3):373, 2010.

[103] Raymond YW Chinn and Lynne Sehulster. Guidelines for environmental infection control in health-care facilities; recommendations of cdc and healthcare infection control practices advisory committee (hicpac). 2003.

[104] PE Paul Ninomura and PE Richard Hermans. Ventilation standard for health care facilities. *ASHRAE Journal*, 50(10):52–57, 2008.

[105] Shelly L Miller, Nicholas Clements, Steven A Elliott, Shobha S Subhash, Aaron Eagan, and Lewis J Radonovich. Implementing a negative-pressure isolation ward for a surge in airborne infectious patients. *American journal of infection control*, 45(6):652–659, 2017.

[106] Chris P Underwood. *HVAC control systems: Modelling, analysis and design*. Routledge, 2002.

[107] Model srpmroom pressure monitor, 2020. https://www.setra.com/hubfs/Product_Data_Sheets/Setra_Model_SRPM_Data_Sheet.pdf?t=1516657591048&hsLang=en. (Accessed: 05-01-2022).

[108] MB Wilkinson and M Outram. Principles of pressure transducers, resonance, damping and frequency response. *Anaesthesia & Intensive Care Medicine*, 10(2):102–105, 2009.

[109] Ivan Bajsić, Jože Kutin, and Tomaž Žagar. Response time of a pressure measurement system with a connecting tube. *Instrumentation Science and Technology*, 35(4):399–409, 2007.

[110] Ying-Huang Tsai, Gwo-Hwa Wan, Yao-Kuang Wu, and Kuo-Chien Tsao. Airborne severe acute respiratory syndrome coronavirus concentrations in a negative-pressure isolation room. *Infection Control & Hospital Epidemiology*, 27(5):523–525, 2006.

[111] Guidelines for environmental infection control in health-care facilities, 2003. https://www.cdc.gov/infectioncontrol/guidelines/environmental/background/air.html. (Accessed: 05-01-2022).

[112] Wuhan lab leak theory: How fort detrick became a centre for chinese conspiracies, 2021. https://www.bbc.com/news/world-us-canada-58273322. (Accessed: 05-01-2022).

[113] Judene M Bartley, Russell N Olmsted, and Janet Haas. Current views of health care design and construction: Practical implications for safer, cleaner environments. *American Journal of Infection Control*, 38(5):S1–S12, 2010.

[114] Institute of occupational safety and health (taiwan). recommended guidelines for inspection of isolation wards for sars patients, 2003. https://www.ilosh.gov.tw/1261/1274/1276/8875/?cprint=pt. (Accessed: 05-01-2022).

[115] Paul A Jensen, Lauren A Lambert, Michael F Iademarco, and Renee Ridzon. Guidelines for preventing the transmission of mycobacterium tuberculosis in health-care settings, 2005. 2005.

[116] American institute of architects guidelines for the construction of hospitals and health care facilities. washington: The institute, 2006. https://fgiguidelines.org/wp-content/uploads/2015/08/2001guidelines.pdf. (Accessed: 05-01-2022).

[117] Guidelines for the classification and design of isolation rooms in health care facilities, victorian advisory committee on infection control, 2007. https://galihendradita.files.wordpress.com/2019/11/australia_isolation_rooms_2007.pdf. (Accessed: 05-01-2022).

[118] Stanley Corrsin. Extended applications of the hot-wire anemometer. *Review of Scientific Instruments*, 18(7):469–471, 1947.

[119] Finn and Inc. Conway. Room pressure monitors and environmental monitors, 2020. https://finnandconway.com/news/18694/setra-critical-room-pressure-monitors. (Accessed: 05-01-2022).

[120] Avnet Abacus. Pressure sensors: The design engineers guide. *Avnet Reach Further*, 2021.

[121] Pressure sensing 101 – absolute, gauge, differential & sealed pressure, 2022. https://esenssys.com/differences-between-pressure-sensors/. (Accessed: 05-01-2022).

[122] Series rsm rom status monitor, 2020. https://www.dwyer-inst.com/PDF_files/P_3_RSM.pdf. (Accessed: 05-01-2022).

[123] One vue sense, 2020. https://www.primexinc.com/en/assets?download=Primex_OneVUE-DiffPressure.pdf. (Accessed: 05-01-2022).

[124] Room status monitor, 2020. https://www.dwyer-inst.com/PDF_files/RSME.pdf. (Accessed: 05-01-2022).

[125] Room pressure monitor, 2020. https://sid.siemens.com/v/u/A6V10322677. (Accessed: 05-01-2022).

[126] Sensocon series a1, 2020. https://www.sensocon.com/uploads/Files/Install16/A1-Digital-Differential-Pressure-Gauge-IOM.pdf. (Accessed: 05-01-2022).

[127] Guardian space pressure monitor, 2022. https://paragoncontrols.com/wp-content/uploads/2021/07/SPM-1000-IOM.pdf. (Accessed: 05-01-2022).

[128] Theory of second-order systems, 2022. https://www.uml.edu/docs/Second-Theory_tcm18-190098.pdf. (Accessed: 05-01-2022).

[129] David Halliday, Robert Resnick, and Jearl Walker. *Fundamentals of physics*. John Wiley & Sons, 2013.

[130] Introduction to dynamic pressure sensors, 2022. https://www.pcb.com/resources/technical-information/introduction-to-pressure-sensors. (Accessed: 05-01-2022).

[131] Anna Goldenberg, Galit Shmueli, Richard A Caruana, and Stephen E Fienberg. Early statistical detection of anthrax outbreaks by tracking over-the-counter medication sales. *Proceedings of the National Academy of Sciences*, 99(8):5237–5240, 2002.

[132] John G. Bartlett. 20 - bioterrorism. In Lee Goldman and Andrew I. Schafer, editors, *Goldman's Cecil Medicine (Twenty Fourth Edition)*, pages 84–88. W.B. Saunders, Philadelphia, twenty fourth edition edition, 2012. ISBN 978-1-4377-1604-7. doi: https://doi.org/10.1016/B978-1-4377-1604-7.00020-8. URL https://www.sciencedirect.com/science/article/pii/B9781437716047000208.

[133] The feynman lectures on physics vol. i ch. 47: Sound. the wave equation, 2006. https://www.feynmanlectures.caltech.edu/I_47.html. (Accessed: 05-01-2022).

[134] Samusng galaxy s10, 2022. https://www.samsung.com/global/galaxy/galaxy-s10//. (Accessed: 05-01-2022).

[135] Which loudspeakers are loudest?, 2022. https://www.razmobility.com/assistive-technology-blog/which-loudspeakers-are-loudest/. (Accessed: 05-01-2022).

[136] Keysight / agilent 33120a function / arbitrary waveform generator, 15 mhz, 2022. https://www.keysight.com/us/en/product/33120A/function--arbitrary-waveform-generator-15-mhz.html. (Accessed: 05-01-2022).

[137] Boss audio systems r1002 car amplifier - 2 channel, 200 watts max power, 2 4 ohm stable, class ab, full range, 2022. https://www.amazon.com/BOSS-Audio-R1002-Car-Amplifier/dp/B004S50ZB2/ref=sr_1_2?dchild=1&keywords=200+watt+audio+amplifier&qid=1588804890&sr=8-2. (Accessed: 05-01-2022).

[138] Goldwood sound inc. sound module, 2022. https://www.amazon.com/Goldwood-Sound-Inc-GT-300PB-1188-2/dp/B071R82KPS. (Accessed: 05-01-2022).

[139] Gt-1188 tweeter drivers replacements for ksn1188a, 2022. https://www.amazon.com/Goldwood-Sound-Inc-GT-300PB-1188-2/dp/B071R82KPS. (Accessed: 05-01-2022).

[140] Sound meter, 2022. https://play.google.com/store/apps/details?id=kr.sira.sound&hl=en. (Accessed: 05-01-2022).

[141] Ultrasonic signal generator module, 2022. https://www.kemo-electronic.de/en/Car/Modules/M048N-Ultrasonic-Generator.php. (Accessed: 05-01-2022).

[142] Piezoelectric tweeter horntotot, 2022. https://www.amazon.com/ToToT-Ultrasonic-Speaker-Loudspeaker-Piezoelectric/dp/B07RW7ZNB4/ref=sr_1_3?dchild=1&keywords=ultrasonic+speaker&qid=1588806704&sr=8-3. (Accessed: 05-01-2022).

[143] Data sheetp1kpressure sensor, 2022. https://datasheet.octopart.com/P1K-2-2X16PA-Kavlico-datasheet-81473203.pdf. (Accessed: 05-01-2022).

[144] Integrated silicon pressure sensor on-chip signal conditioned, temperature compensated and calibrated, 2022. https://media.digikey.com/pdf/Data%20Sheets/Freescale%20Semi/MPVZ5004G.pdf. (Accessed: 05-01-2022).

[145] The sdp800 series, 2022. https://sensirion.com/media/documents/099567E0/6166D20B/Sensirion_Differential_Pressure_Sensors_Chart_SDP800Series.pdf. (Accessed: 05-01-2022).

[146] Basic board mount pressure sensors, 2022. https://www.mouser.com/datasheet/2/187/honeywell_sensing_board_mount_pressure_tbp_nbp_ser-1837963.pdf. (Accessed: 05-01-2022).

[147] P993 low range differential pressure pcb mount sensor, 2022. https://www.sensata.com/sites/default/files/a/sensata-p993%20series-differential%20pressure%20mount%20sensor-datasheet.pdf. (Accessed: 05-01-2022).

[148] Trustability® board mount pressure sensors, 2022. https://www.mouser.com/datasheet/2/187/honeywell-sensing-trustability-board-mount-pressur-1228675.pdf. (Accessed: 05-01-2022).

[149] Series a1 digital differential pressure gauge, 2022. https://www.sensocon.com/uploads/Files/English/Sensocon-Series-A1-Digital-Differential-Pressure-Gauge-Datasheet.pdf. (Accessed: 05-01-2022).

[150] Ek-p5: Differential pressure evaluation kit sdp8xx series, 2022. https://sensirion.com/products/catalog/EK-P5/. (Accessed: 05-01-2022).

[151] Improving differential pressure diaphragm seal system performance and installed cost, 2022. https://www.emerson.com/documents/automation/white-paper-improving-differential-pressure-diaphragm-seal-system-performance-insta pdf. (Accessed: 05-01-2022).

[152] Robert E Curry and Glenn B Gilyard. Experimental characterization of the effects of pneumatic tubing on unsteady pressure measurements. *NASA Technical Memorandum*, 41:71, 1990.

[153] Clear vinyl tubing, 2022. https://www.homedepot.com/p/UDP-3-16-in-I-D-x-5-16-in-O-D-x-20-ft-Clear-Vinyl-Tubing-T10007004/304185167. (Accessed: 05-01-2022).

[154] Static pressure pickup, 2022. https://www.dwyer-inst.com/Product/Pressure/RoomStatusMonitors/SeriesRSME#accessories. (Accessed: 05-01-2022).

[155] Sound waves — university physics volume 1, 2016. https://courses.lumenlearning.com/suny-osuniversityphysics/chapter/17-1-sound-waves/. (Accessed: 05-01-2022).

[156] Anomadarshi Barua and Mohammad Abdullah Al Faruque. Premsat: Preventing magnetic saturation attack on hall sensors. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 438–462, 2022.

[157] Anomadarshi Barua and Mohammad Abdullah Al Faruque. Halc: A real-time in-sensor defense against the magnetic spoofing attack on hall sensors. In *Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses*, RAID '22, page 185–199, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450397049. doi: 10.1145/3545948.3545964. URL https://doi.org/10.1145/3545948.3545964.

[158] Anomadarshi Barua and Mohammad Abdullah Al Faruque. Sensor Security: Current Progress, Research Challenges, and Future Roadmap (Invited Paper). In *International Conference on Computer-Aided Design (ICCAD 2022)*, 2022.

[159] Sujit Rokka Chhetri, Jiang Wan, and Mohammad Abdullah Al Faruque. Cross-domain security of cyber-physical systems. In *2017 22nd Asia and South Pacific design automation conference (ASP-DAC)*, pages 200–205. IEEE, 2017.

[160] Ishtiaq Rouf, Rob Miller, Hossen Mustafa, Travis Taylor, Sangho Oh, Wenyuan Xu, Marco Gruteser, Wade Trappe, and Ivan Seskar. Security and privacy vulnerabilities of {In-Car} wireless networks: A tire pressure monitoring system case study. In *19th USENIX Security Symposium (USENIX Security 10)*, 2010.

[161] Yazhou Tu, Vijay Srinivas Tida, Zhongqi Pan, and Xiali Hei. Transduction shield: A low-complexity method to detect and correct the effects of emi injection attacks on sensors. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 901–915, 2021.

[162] Renchi Yan, Teng Xu, and Miodrag Potkonjak. Semantic attacks on wireless medical devices. In *SENSORS, 2014 IEEE*, pages 482–485, 2014. doi: 10.1109/ICSENS.2014.6985040.

[163] Connor Bolton, Sara Rampazzi, Chaohao Li, Andrew Kwong, Wenyuan Xu, and Kevin Fu. Blue note: How intentional acoustic interference damages availability and integrity in hard disk drives and operating systems. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 1048–1062. IEEE, 2018.

[164] Jan Hjelmgren. Dynamic measurement of pressure.-a literature survey. 2002.

[165] Xiangguang Han, Qi Mao, Libo Zhao, Xuejiao Li, Li Wang, Ping Yang, Dejiang Lu, Yonglu Wang, Xin Yan, Songli Wang, et al. Novel resonant pressure sensor based on piezoresistive detection and symmetrical in-plane mode vibration. *Microsystems & nanoengineering*, 6(1):1–11, 2020.

[166] JC Greenwood and DW Satchell. Miniature silicon resonant pressure sensor. In *IEE Proceedings D (Control Theory and Applications)*, volume 135, pages 369–372. IET, 1988.

[167] Xun Shen, Yahui Zhang, and Tielong Shen. Cylinder pressure resonant frequency cyclic estimation-based knock intensity metric in combustion engines. *Applied Thermal Engineering*, 158:113756, 2019.

[168] Tian Wang, Meihui Gong, Xiaoyu Yu, Guangdong Lan, and Yunbo Shi. Acoustic-pressure sensor array system for cardiac-sound acquisition. *Biomedical Signal Processing and Control*, 69:102836, 2021.

[169] A Nagiub, Elias Soupos, and Hassan Nagib. Characterization of a mems acoustic/pressure sensor. In *37th Aerospace Sciences Meeting and Exhibit*, page 520, 1999.

[170] Anomadarshi Barua and Mohammad Abdullah Al Faruque. The hall sensor security. 2021.

[171] Anomadarshi Barua and Mohammad Abdullah Al Faruque. Special session: Noninvasive sensor-spoofing attacks on embedded and cyber-physical systems. In *2020 IEEE 38th International Conference on Computer Design (ICCD)*, pages 45–48. IEEE, 2020.

[172] Anomadarshi Barua, Lelin Pan, and Mohammad Abdullah Al Faruque. Bayesimposter: Bayesian estimation based.bss imposter attack on industrial control systems. In *Proceedings of the 38th Annual Computer Security Applications Conference*, ACSAC '22, page 440–454, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450397599. doi: 10.1145/3564625.3564638. URL `https://doi.org/10.1145/3564625.3564638`.

[173] Bianca Scholten. *The road to integration: A guide to applying the ISA-95 standard in manufacturing*. Isa, 2007.

[174] Christoph Jan Bartodziej. The concept industry 4.0. In *The concept industry 4.0*, pages 27–50. Springer, 2017.

[175] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business & information systems engineering*, 6(4):239–242, 2014.

[176] Yuping Xing and Yongzhao Zhan. Virtualization and cloud computing. In *Future wireless networks and information systems*, pages 305–312. Springer, 2012.

[177] Reinhard Langmann and Leandro F Rojas-Peña. A plc as an industry 4.0 component. In *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 10–15. IEEE, 2016.

[178] Omid Givehchi, Jahanzaib Imtiaz, Henning Trsek, and Juergen Jasperneite. Control-as-a-service from the cloud: A case study for using virtualized plcs. In *2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014)*, pages 1–4. IEEE, 2014.

[179] Yuhui Deng, Xinyu Huang, Liangshan Song, Yongtao Zhou, and Frank Z Wang. Memory deduplication: An effective approach to improve the memory system. *Journal of Information Science and Engineering*, 33(5):1103–1120, 2017.

[180] Reinhard Langmann and Michael Stiller. The plc as a smart service in industry 4.0 production systems. *Applied Sciences*, 9(18):3815, 2019.

[181] Bernard Friedland. *Control system design: an introduction to state-space methods*. Courier Corporation, 2012.

[182] Kaveh Razavi, Ben Gras, Erik Bosman, Bart Preneel, Cristiano Giuffrida, and Herbert Bos. Flip feng shui: Hammering a needle in the software stack. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 1–18, 2016.

[183] Antonio Barresi, Kaveh Razavi, Mathias Payer, and Thomas R Gross. {CAIN}: Silently breaking {ASLR} in the cloud. In *9th {USENIX} Workshop on Offensive Technologies ({WOOT} 15)*, 2015.

[184] Erik Bosman, Kaveh Razavi, Herbert Bos, and Cristiano Giuffrida. Dedup est machina: Memory deduplication as an advanced exploitation vector. In *2016 IEEE symposium on security and privacy (SP)*, pages 987–1004. IEEE, 2016.

[185] Marco Oliverio, Kaveh Razavi, Herbert Bos, and Cristiano Giuffrida. Secure page fusion with vusion: https://www. vusec. net/projects/vusion. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 531–545, 2017.

[186] *Factory Simulation 24V*. `https://www.fischertechnik.de/en/service/elearning/simulating/fabrik-simulation-24v`. (Accessed: 03-22-2022).

[187] *SIMATIC S7-1500*. `https://cache.industry.siemens.com/dl/files/914/59191914/att_86487/v1/s71500_cpu1516_3_pn_dp_manual_en-US_en-US.pdf`.

[188] Anam Sajid, Haider Abbas, and Kashif Saleem. Cloud-assisted iot-based scada systems security: A review of the state of the art and future challenges. *IEEE Access*, 4:1375–1384, 2016.

[189] Michael Tiegelkamp and Karl-Heinz John. *IEC 61131-3: Programming industrial automation systems*, volume 14. Springer, 1995.

[190] Jarno Ruotsalainen. Hardening and architecture of an industrial control system in a virtualized environment. Master's thesis, 2018.

[191] *Siemens: How to connect to a PLC with TIA Portal in a Virtual Machine.* `https://web.awc-inc.com/siemens-how-to-connect-to-a-plc-with-tia-portal-in-a-virtual-machine/`.

[192] Chao-Rui Chang, Jan-Jan Wu, and Pangfeng Liu. An empirical study on memory sharing of virtual machines for server consolidation. In *2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications*, pages 244–249. IEEE, 2011.

[193] Diwaker Gupta, Sangmin Lee, Michael Vrable, Stefan Savage, Alex C Snoeren, George Varghese, Geoffrey M Voelker, and Amin Vahdat. Difference engine: Harnessing memory redundancy in virtual machines. *Communications of the ACM*, 53(10):85–93, 2010.

[194] *Data Deduplication Overview*, . `https://docs.microsoft.com/en-us/windows-server/storage/data-deduplication/overview`.

[195] Thomas Goldschmidt, Mahesh Kumar Murugaiah, Christian Sonntag, Bastian Schlich, Sebastian Biallas, and Peter Weber. Cloud-based control: A multi-tenant, horizontally scalable soft-plc. In *2015 IEEE 8th International Conference on Cloud Computing*, pages 909–916. IEEE, 2015.

[196] Noëlle Rakotondravony, Benjamin Taubmann, Waseem Mandarawi, Eva Weishäupl, Peng Xu, Bojan Kolosnjaji, Mykolai Protsenko, Hermann De Meer, and Hans P Reiser. Classifying malware attacks in iaas cloud environments. *Journal of Cloud Computing*, 6(1):26, 2017.

[197] S Annapoorani, B Srinivasan, and GA Mylavathi. Analysis of various virtual machine attacks in cloud computing. In *2018 2nd international Conference on Inventive Systems and Control (ICISC)*, pages 1016–1019. IEEE, 2018.

[198] Ercan Nurcan Ylmaz, Bünyamin Ciylan, Serkan Gönen, Erhan Sindiren, and Gökçe Karacayılmaz. Cyber security in industrial control systems: Analysis of dos attacks against plcs and the insider effect. In *2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG)*, pages 81–85. IEEE, 2018.

[199] Seungoh Choi, Jongwon Choi, Jeong-Han Yun, Byung-Gil Min, and HyoungChun Kim. Expansion of {ICS} testbed for security validation based on {MITRE} att&ck techniques. In *13th {USENIX} Workshop on Cyber Security Experimentation and Test ({CSET} 20)*, 2020.

[200] *Affordable MQTT Broker Pricing*. https://www.bevywise.com/mqtt-broker/pricing.html.

[201] *EMQ X Broker*. https://www.emqx.io/downloads#broker.

[202] *mosquitto*. https://mosquitto.org/.

[203] *MQTT-C*. https://github.com/LiamBindle/MQTT-C.

[204] *eMQTT5*. https://github.com/X-Ryl669/eMQTT5.

[205] *wolfMQTT*. https://github.com/wolfSSL/wolfMQTT.

[206] Matt Pietrek. Peering inside the pe: a tour of the win32 (r) portable executable file format. *Microsoft Systems Journal-US Edition*, 9(3):15–38, 1994.

[207] *How to use TIA Portal Cloud*. https://new.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal/highlights/tia-portal-cloud.html.

[208] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. Flipping bits in memory without accessing them: An experimental study of dram disturbance errors. *ACM SIGARCH Computer Architecture News*, 42(3):361–372, 2014.

[209] Microsoft. *Large-Page Support*. https://docs.microsoft.com/en-us/windows/win32/memory/large-page-support.

[210] Mark Seaborn and Thomas Dullien. Exploiting the dram rowhammer bug to gain kernel privileges. *Black Hat*, 15:71, 2015.

[211] *Linux kernel 2.6.32, Section 1.3. Kernel Samepage Merging (memory deduplication)*. https://kernelnewbies.org/Linux_2_6_32#Kernel_Samepage_Merging_.28memory_deduplication.29.

[212] Konrad Miller, Fabian Franz, Marc Rittinghaus, Marius Hillenbrand, and Frank Bellosa. Xlh: More effective memory deduplication scanners through cross-layer hints. In *2013 USENIX Annual Technical Conference USENIX ATC 13)*, pages 279–290, 2013.

[213] *Chapter 7. KSM*. https://docs.fedoraproject.org/en-US/Fedora/18/html/Virtualization_Administration_Guide/chap-KSM.html.

[214] Gangyong Jia, Guangjie Han, Joel JPC Rodrigues, Jaime Lloret, and Wei Li. Coordinate memory deduplication and partition for improving performance in cloud computing. *IEEE Transactions on Cloud Computing*, 7(2):357–368, 2015.

[215] Lucian Cojocar, Kaveh Razavi, Cristiano Giuffrida, and Herbert Bos. Exploiting correcting codes: On the effectiveness of ecc memory against rowhammer attacks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 55–71. IEEE, 2019.

[216] Barbara Aichinger. Ddr memory errors caused by row hammer. In *2015 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–5. IEEE, 2015.

[217] Mark Lanteigne. How rowhammer could be used to exploit weaknesses in computer hardware. *SEMICON China*, 2016.

[218] D. Gruss, M. Lipp, M. Schwarz, D. Genkin, J. Juffinger, S. O'Connell, W. Schoechl, and Y. Yarom. Another flip in the wall of rowhammer defenses. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 245–261, 2018. doi: 10.1109/SP.2018.00031.

[219] Onur Mutlu. The rowhammer problem and other issues we may face as memory becomes denser. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, pages 1116–1121. IEEE, 2017.

[220] Philip G Emma, William R Reohr, and Mesut Meterelliyoz. Rethinking refresh: Increasing availability and reducing power in dram for cache applications. *IEEE micro*, 28(6):47–56, 2008.

[221] Dae-Hyun Kim, Prashant J Nair, and Moinuddin K Qureshi. Architectural support for mitigating row hammering in dram memories. *IEEE Computer Architecture Letters*, 14(1):9–12, 2014.

[222] Patrick J Meaney, Luis Alfonso Lastras-Montaño, Vesselina K Papazova, Eldee Stephens, JS Johnson, Luiz C Alves, James A O'Connor, and William J Clarke. Ibm zenterprise redundant array of independent memory subsystem. *IBM Journal of Research and Development*, 56(1.2):4–1, 2012.

[223] Zelalem Birhanu Aweke, Salessawi Ferede Yitbarek, Rui Qiao, Reetuparna Das, Matthew Hicks, Yossi Oren, and Todd Austin. Anvil: Software-based protection against next-generation rowhammer attacks. *ACM SIGPLAN Notices*, 51(4):743–755, 2016.

[224] Andrew Kwong, Daniel Genkin, Daniel Gruss, and Yuval Yarom. Rambleed: Reading bits in memory without accessing them. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 695–711. IEEE, 2020.

[225] Pietro Frigo, Emanuele Vannacc, Hasan Hassan, Victor Van Der Veen, Onur Mutlu, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi. Trrespass: Exploiting the many sides of target row refresh. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 747–762. IEEE, 2020.

[226] Ali Abbasi and Majid Hashemi. Ghost in the plc designing an undetectable programmable logic controller rootkit via pin control attack. *Black Hat Europe*, 2016: 1–35, 2016.

[227] Luis Garcia, Ferdinand Brasser, Mehmet Hazar Cintuglu, Ahmad-Reza Sadeghi, Osama A Mohammed, and Saman A Zonouz. Hey, my malware knows physics! attacking plcs with physical model aware rootkit. In *NDSS*, 2017.

[228] Alexander Bolshev, Jason Larsen, Marina Krotofil, and Reid Wightman. A rising tide: Design exploits in industrial control systems. In *10th {USENIX} Workshop on Offensive Technologies ({WOOT} 16)*, 2016.

[229] Ralf Spenneberg, Maik Brüggemann, and Hendrik Schwartke. Plc-blaster: A worm living solely in the plc. *Black Hat Asia*, 16:1–16, 2016.

[230] Johannes Klick, Stephan Lau, Daniel Marzin, Jan-Ole Malchow, and Volker Roth. Internet-facing plcs-a new back orifice. *Blackhat USA*, pages 22–26, 2015.

[231] Zachry Basnight, Jonathan Butts, Juan Lopez Jr, and Thomas Dube. Firmware modification attacks on programmable logic controllers. *International Journal of Critical Infrastructure Protection*, 6(2):76–84, 2013.

[232] Dillon Beresford. Exploiting siemens simatic s7 plcs. *Black Hat USA*, 16(2):723–733, 2011.

[233] Anomadarshi Barua and Mohammad Abdullah Al Faruque. *The Hall Sensor Security*, pages 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg, 2019. ISBN 978-3-642-27739-9. doi: 10.1007/978-3-642-27739-9_1652-1. URL `https://doi.org/10.1007/978-3-642-27739-9_1652-1`.

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

[235] Stephen McLaughlin and Saman Zonouz. Controller-aware false data injection against programmable logic controllers. In *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 848–853. IEEE, 2014.

[236] Daniel Gruss, Clémentine Maurice, and Stefan Mangard. Rowhammer. js: A remote software-induced fault attack in javascript. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 300–321. Springer, 2016.

[237] Andrei Tatar, Radhesh Krishnan Konoth, Elias Athanasopoulos, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi. Throwhammer: Rowhammer attacks over the network and defenses. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 213–226, 2018.

[238] Andrea Ajbl, Marc Pastre, and Maher Kayal. A fully integrated hall sensor microsystem for contactless current measurement. *IEEE Sensors Journal*, 13(6):2271–2278, 2013.

[239] Radivoje Popovic and Christian Schott. Sensor for detecting the direction of a magnetic field in a plane, June 26 2007. US Patent 7,235,968.

[240] Hrishikesh Mehta, Ujjwala Thakar, Vrunda Joshi, Kirti Rathod, and Pradeep Kurulkar. Hall sensor fault detection and fault tolerant control of pmsm drive system. In *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, pages 624–629. IEEE, 2015.

[241] Ersan Kabalci and Yasin Kabalci. A wireless metering and monitoring system for solar string inverters. *International Journal of Electrical Power & Energy Systems*, 96: 282–295, 2018.

[242] Yuan-Pin Tsai, Kun-Long Chen, and Nanming Chen. Design of a hall effect current microsensor for power networks. *IEEE Transactions on Smart Grid*, 2(3):421–427, 2011.

[243] Yue Xu, Hong-Bin Pan, Shu-Zhuan He, and Li Li. A highly sensitive cmos digital hall sensor for low magnetic field applications. *Sensors*, 12(2):2162–2174, 2012.

[244] Dong-Hyun Nam, Woo-Beom Lee, You-Sik Hong, and Sang-Suk Lee. Measurement of spatial pulse wave velocity by using a clip-type pulsimeter equipped with a hall sensor and photoplethysmography. *Sensors*, 13(4):4714–4723, 2013.

[245] Anomadarshi Barua and Mohammad Abdullah Al Faruque. Hall Spoofing: A Non-Invasive DoS Attack on Grid-Tied Solar Inverter. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1273–1290. USENIX Association, August 2020. ISBN 978-1-939133-17-5. URL `https://www.usenix.org/conference/usenixsecurity20/presentation/barua`.

[246] Edward Ramsden. *Hall-effect sensors: theory and application*. Elsevier, 2011.

[247] Martin H. Weik. *passive sensor*, pages 1235–1235. Springer US, Boston, MA, 2001. ISBN 978-1-4020-0613-5. doi: 10.1007/1-4020-0613-6_13692. URL `https://doi.org/10.1007/1-4020-0613-6_13692`.

[248] Xingguo Cheng, Zhenjun Sun, Xiaoyan Wang, and Sheng Liu. Open-loop linear differential current sensor based on dual-mode hall effect. *Measurement*, 50:29–33, 2014.

[249] Latham Alexander. *Common Mode Field Rejection in Coreless Hall-Effect Current Sensor ICs*, 2019. `https://www.allegromicro.com/-/media/files/application-notes/an269123-common-mode-field-rejection-in-coreless-current-sensor-ics.pdf`. (Accessed: 03-22-2022).

[250] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. Inaudible voice commands: The long-range attack and defense. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 547–560, Renton, WA, April 2018. USENIX Association. ISBN 978-1-939133-01-4.

[251] Martin H. Weik. *active sensor*, pages 21–21. Springer US, Boston, MA, 2001. ISBN 978-1-4020-0613-5. doi: 10.1007/1-4020-0613-6_258. URL `https://doi.org/10.1007/1-4020-0613-6_258`.

[252] 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*, pages 1004–1015, 2015.

[253] Littelfuse. *Magnets and Actuators.* https://www.littelfuse.com/~/media/electronics/datasheets/magnetic_actuators/littelfuse_magnetic_actuators_datasheet.pdf.pdf. (Accessed: 03-22-2022).

[254] Christina Garman, Matthew Green, Gabriel Kaptchuk, Ian Miers, and Michael Rushanan. Dancing on the lip of the volcano: Chosen ciphertext attacks on apple imessage. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 655–672, 2016.

[255] Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986.

[256] *uxcell DC12V 100KG Force Electric Lifting Magnet Electromagnet Solenoid Lift Holding.* https://www.amazon.ca/uxcell%C2%AE-Solenoid-Electric-Holding-Electromagnet/dp/B07FKNJ7CZ?th=1. (Accessed: 03-22-2022).

[257] S Geetha, KK Satheesh Kumar, Chepuri RK Rao, M Vijayan, and DC Trivedi. EMI shielding: Methods and materials—A review. *Journal of applied polymer science*, 112 (4):2073–2086, 2009.

[258] BP Lathi and Zhi Ding. Modern analog and digital communication systems. *Third generation OXFORD University Press, NEW YORK*, pages 50–51, 1998.

[259] *Cortex-M3 Reference Manual: EFM32 Microcontroller Family.* https://www.silabs.com/documents/public/reference-manuals/EFM32-Cortex-M3-RM.pdf. (Accessed: 03-22-2022).

[260] *Tektronix 119-6609-00.* https://www.testequipmentdepot.com/tektronix/accessories/cables-hardware/attachments/flexible-monopole-antenna-119660900.htm?ref=gbase. (Accessed: 03-22-2022).

[261] Werner Kutzelnigg, Giuseppe Del Re, and Gaston Berthier. Correlation coefficients for electronic wave functions. *Physical Review*, 172(1):49, 1968.

[262] *ACS718 datasheet*, . https://www.allegromicro.com/~/media/Files/Datasheets/ACS718-Datasheet.ashx. (Accessed: 03-22-2022).

[263] *ACS710 datasheet*, . https://www.allegromicro.com/~/media/Files/Datasheets/ACS710-Datasheet.ashx. (Accessed: 03-22-2022).

[264] *ACS715 datasheet*, . https://www.allegromicro.com/~/media/Files/Datasheets/ACS715-Datasheet.ashx. (Accessed: 03-22-2022).

[265] *ACS724 datasheet*, . https://www.allegromicro.com/~/media/Files/Datasheets/ACS724-Datasheet.ashx. (Accessed: 03-22-2022).

[266] SS49/SS19 datasheet, . https://www.digikey.com/htmldatasheets/production/809995/0/0/1/SS49-SS19-Series-Install-Instr.pdf. (Accessed: 03-22-2022).

[267] *SS39ET datasheet*, . https://www.digikey.com/htmldatasheets/production/43186/0/0/1/SS39ET-49E-59ET-Series-Datasheet.pdf. (Accessed: 03-22-2022).

[268] *SS490 datasheet*, . https://prod-edam.honeywell.com/content/dam/honeywell-edam/sps/siot/en-us/products/sensors/magnetic-sensors/linear-and-angle-sensor-ics/ss490-series-linear-sensor-ics/documents/sps-siot-sensors-linear-hall-effect-ics-ss490-series-datasheet-005843-2-en-ciid-50pdf. (Accessed: 03-22-2022).

[269] *DRV5053 datasheet*, . https://www.ti.com/document-viewer/DRV5053/datasheet. (Accessed: 03-22-2022).

[270] *LTSR 6-NP datasheet*, . https://www.lem.com/sites/default/files/products_datasheets/ltsr_6-np.pdf. (Accessed: 03-22-2022).

[271] *LV 25-P datasheet*, . http://www.farnell.com/datasheets/51633.pdf. (Accessed: 03-22-2022).

[272] *MU METAL SPECIFICATIONS*. http://www.mu-metal.com/technical-data.html. (Accessed: 03-22-2022).

[273] Silicon Labs. *Energy Profiler*. https://docs.silabs.com/simplicity-studio-5-users-guide/1.0/using-the-tools/energy-profiler/. (Accessed: 12-10-2021).

[274] *Allegro MicroSystems : Hall-effect sensors consume very little power*. https://www.eetimes.com/allegro-microsystems-hall-effect-sensors-consume-very-little-power/. (Accessed: 03-22-2022).

[275] *Grid-tied Solar Micro Inverter with MPPT*. https://www.ti.com/tool/TIDM-SOLARUINV. (Accessed: 03-22-2022).

[276] *MCP4252 datasheet*, . https://ww1.microchip.com/downloads/en/DeviceDoc/22060b.pdf. (Accessed: 03-22-2022).

[277] Alvaro A Cárdenas, Saurabh Amin, Zong-Syun Lin, Yu-Lun Huang, Chi-Yen Huang, and Shankar Sastry. Attacks against process control systems: risk assessment, detection, and response. In *Proceedings of the 6th ACM symposium on information, computer and communications security*, pages 355–366, 2011.

[278] 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, 2016.

[279] Yasser Shoukry, Pierluigi Nuzzo, Nicola Bezzo, Alberto L Sangiovanni-Vincentelli, Sanjit A Seshia, and Paulo Tabuada. Secure state reconstruction in differentially flat systems under sensor attacks using satisfiability modulo theory solving. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 3804–3809. IEEE, 2015.

[280] F Caricchi, F Giulii Capponi, F Crescimbini, and L Solero. Sinusoidal brushless drive with low-cost linear hall effect position sensors. In *2001 IEEE 32nd Annual Power Electronics Specialists Conference (IEEE Cat. No. 01CH37230)*, volume 2, pages 799–804. IEEE, 2001.

[281] Baoping Chen, Guoqing Lu, Ronghua Hao, Liqiang Hu, and Xiushu Tian. Intelligent speed and mileage measurement system for vehicles based on hall sensor. In *2009 First International Workshop on Education Technology and Computer Science*, volume 2, pages 272–274. IEEE, 2009.

[282] *Hall-Effect Current Sensor Market by Type, Technology, Output (Linear and Threshold), Industry (Industrial Automation, Automotive, Consumer Electronics, Telecommunication, Utilities, Medical, Railways), and Region - Global Forecast to 2023.* `https://www.marketsandmarkets.com/Market-Reports/hall-effect-current-sensor-market-201606539.html`. (Accessed: 03-22-2022).

[283] Joe Gilbert. Technical advances in hall-effect sensing. *Allegro MicroSystems technical paper STP 00-1. http://www. allegromicro. com/techpub2/stp/stp00-1. pdf*, 2006.

[284] MC Gold and DA Nelson. Variable magnetic field hall effect measurements and analyses of high purity, hg vacancy (p-type) hgcdte. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, 4(4):2040–2046, 1986.

[285] Gaetano Palumbo and Salvatore Pennisi. *Feedback amplifiers: theory and design*. Springer Science & Business Media, 2002.

[286] William H Hayt Jr, John A Buck, and M Jaleel Akhtar. *Engineering Electromagnetics— (SIE)*. McGraw-Hill Education, 2020.

[287] Paul Peter Urone, Roger Hinrichs. College Physics. 2012. `https://openstax.org/books/college-physics/pages/22-6-the-hall-effect`. (Accessed: 03-22-2022).

[288] Yoichi Ogo, Hidenori Hiramatsu, Kenji Nomura, Hiroshi Yanagi, Toshio Kamiya, Masahiro Hirano, and Hideo Hosono. p-channel thin-film transistor using p-type oxide semiconductor, sno. *Applied Physics Letters*, 93(3):032113, 2008.

[289] Charles I Hubert. *Electric Machines: Theory, Operation, Applications, Adjustment & Control*. Pearson Education, 2.00 edition, 2009.

[290] F.P. Miller, A.F. Vandome, and J. McBrewster. *Lorentz Force*. Alphascript Publishing, 2009. ISBN 9786130074920. URL https://books.google.com/books?id=Nrf5QQAACAAJ.

[291] Rames C Panda. *Introduction to PID controllers: theory, tuning and application to frontier areas*. BoD–Books on Demand, 2012.

[292] Otto Föllinger and Mathias Kluwe. *Laplace-und Fourier-Transformation*. Elitera Berlin, 1977.

[293] STP4NK80Z. https://media.digikey.com/pdf/Data%20Sheets/ST%20Microelectronics%20PDFS/ST(D,P)4NK80Z(-1,FP).pdf. (Accessed: 03-23-2022).

[294] Arduino Uno. https://store-usa.arduino.cc/products/arduino-uno-rev3/. (Accessed: 03-23-2022).

[295] Marian K Kazimierczuk, Giuseppe Sancineto, Gabriele Grandi, Ugo Reggiani, and Antonio Massarini. High-frequency small-signal model of ferrite core inductors. *IEEE Transactions on Magnetics*, 35(5):4185–4191, 1999.

[296] F Herrmann. Teaching the magnetostatic field: Problems to avoid. *American Journal of Physics*, 59(5):447–452, 1991.

[297] Vibration Engineering Consultant. EMI Interference: Understanding and Mitigating AC and DC Magnetic Fields. 2019. https://www.vibeng.com/blog/emi-interference-understanding-and-mitigate-the-ac-and-dc-emi-interference. (Accessed: 03-23-2022).

[298] Y. Shibuya. Magnetic Shielding Effect of a Spherical Shell. 2014. https://demonstrations.wolfram.com/MagneticShieldingEffectOfASphericalShell/. (Accessed: 03-23-2022).

[299] According to Ferroxcube (formerly Philips) Soft Ferrites data., . https://www.ferroxcube.com/zh-CN/download/download/21. (Accessed: 03-22-2022).

[300] *TO-220*. https://www.jameco.com/z/MJE8502-Motorola-TO-220-NPN-Power-Transistor-700V-2281319.html. (Accessed: 03-22-2022).

[301] Mn-Zn Toroid, . https://content.kemet.com/datasheets/KEM_E5003_ESD-R-H.pdf. (Accessed: 03-23-2022).

[302] G Ott, J Wrba, and R Lucke. Recent developments of mn–zn ferrites for high permeability applications. *Journal of Magnetism and Magnetic Materials*, 254:535–537, 2003.

[303] Honeywell. HALL EFFECT SENSING AND APPLICATION. 2019. http://www.rsp-italy.it/Electronics/Databooks/Honeywell/_contents/Honeywell%20-%20Hall%20Effect%20Sensing%20and%20Application%201998.pdf. (Accessed: 03-23-2022).

[304] Yong Wang, Zhaoyan Xu, Jialong Zhang, Lei Xu, Haopei Wang, and Guofei Gu. Srid: State relation based intrusion detection for false data injection attacks in scada. In *European Symposium on Research in Computer Security*, pages 401–418. Springer, 2014.

[305] Anomadarshi Barua and Mohammad Abdullah Al Faruque. Maghop: Magnetic spectrum hopping for securing voltage and current magnetic sensors. In *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust*, HOST '23, 2023.

[306] AR AM Makky, H Abo-Zied, FN Abdelbar, and P Mutschler. Design of the instrument current transformer for high frequency high power applications. In *2008 12th International Middle-East Power System Conference*, pages 230–233. IEEE, 2008.

[307] Sonia Dhia, Alexandre Boyer, Bertrand Vrignon, Mikaël Deobarro, and Thanh Vinh Dinh. On-chip noise sensor for integrated circuit susceptibility investigations. *IEEE transactions on instrumentation and measurement*, 61(3):696–707, 2011.

[308] Franco Fiori. A sensor signal amplifier resilient to emi. *IEEE Sensors Journal*, 16(18): 7008–7015, 2016.

[309] Orazio Aiello, Paolo Crovetti, and Franco Fiori. Investigation on the susceptibility of hall-effect current sensors to emi. In *10th International Symposium on Electromagnetic Compatibility*, pages 368–372. IEEE, 2011.

[310] Youqian Zhang and KB Rasmussen. Detection of electromagnetic interference attacks on sensor systems. In *IEEE Symposium on Security and Privacy (S&P)*, 2020.

[311] B Cogiore, J Pierre Keradec, and Jean Barbaroux. The two winding ferrite core transformer: An experimental method to obtain a wide frequency range equivalent circuit. In *1993 IEEE Instrumentation and Measurement Technology Conference*, pages 558–562. IEEE, 1993.

[312] C Chien. *The Hall effect and its applications*. Springer Science & Business Media, 2013.

[313] CR8320 datasheet, 2022. `https://www.crmagnetics.com/Assets/ProductPDFs/CR8300%20Series%20Specification%20Page.pdf`. (Accessed: 03-18-2023).

[314] Grove datasheet, 2022. `https://www.mouser.com/datasheet/2/744/Seeed_101020073-1217554.pdf`. (Accessed: 03-18-2023).

[315] Joyce H Wu, Jörg Scholvin, Jesús A del Alamo, and Keith A Jenkins. A faraday cage isolation structure for substrate crosstalk suppression. *IEEE Microwave and Wireless Components Letters*, 11(10):410–412, 2001.

[316] E Oran Brigham. *The fast Fourier transform and its applications*. Prentice-Hall, Inc., 1988.

[317] Solomon W Golomb and Guang Gong. *Signal design for good correlation: for wireless communication, cryptography, and radar*. Cambridge University Press, 2005.

[318] Maximal Length LFSR Feedback Terms. http://users.ece.cmu.edu/~koopman/lfsr/. (Accessed: 03-18-2023).

[319] AJ Salazar, G Bahubalindruno, GR Locharla, HS Mendonça, JC Alves, and JM Da Silva. A study on look-up table based sine wave generation. *Proceedings of the Regional Echomail Coordinator, Porto, Portugal*, pages 3–4, 2011.

[320] Zedboard datasheet, 2022. https://digilent.com/reference/_media/zedboard:zedboard_ug.pdf. (Accessed: 03-18-2023).

[321] ADV7125V datasheet, 2022. https://www.analog.com/media/en/technical-documentation/data-sheets/ADV7125.pdf. (Accessed: 03-18-2023).

[322] Circuit Design: How to make an amplitude modulated wave, 2022. https://www.engineersgarage.com/circuit-design-how-to-make-an-amplitude-modulated-wave/. (Accessed: 03-18-2023).

[323] EFM32 Giant Gecko datasheet, 2022. https://www.silabs.com/documents/public/data-sheets/efm32gg-datasheet.pdf. (Accessed: 03-18-2023).

[324] CSNS300 datasheet, 2022. https://datasheet.octopart.com/CSNS300M-Honeywell-datasheet-68416259.pdf. (Accessed: 03-18-2023).

[325] CTF-5RL-0050 datasheet, 2022. https://cdn.automationdirect.com/static/specs/acuampsolidcore.pdf. (Accessed: 03-18-2023).

[326] CR8410 datasheet, 2022. https://www.crmagnetics.com/Assets/ProductPDFs/CR8400%20Series%20Specification%20Page.pdf. (Accessed: 03-18-2023).

[327] MET-28-T datasheet, 2022. http://catalog.triadmagnetics.com/Asset/MET-28-T.pdf. (Accessed: 03-18-2023).

[328] MET-42-T datasheet, 2022. http://catalog.triadmagnetics.com/Asset/MET-42-T.pdf. (Accessed: 03-18-2023).

[329] Richard Taylor. Interpretation of the correlation coefficient: a basic review. *Journal of diagnostic medical sonography*, 6(1):35–39, 1990.

[330] Jiang et al. Benefits and costs of power-gating technique. In *2005 International conference on computer design*, pages 559–566. IEEE, 2005.

[331] OMO Gatous and José Pissolato. Frequency-dependent skin-effect formulation for resistance and internal inductance of a solid cylindrical conductor. *IEE Proceedings-Microwaves, Antennas and Propagation*, 151(3):212–216, 2004.

[332] Bora et al. Outstanding absolute electromagnetic interference shielding effectiveness of cross-linked pedot: Pss film. *Advanced Materials Interfaces*, 6(22):1901353, 2019.

[333] Tahar Merizgui, Bachir Gaoui, Tamer A Sebaey, and VR Arun Prakash. High content silver/zinc oxide nanoparticle and cobalt nanowire in caryota urens fibre-epoxy composites for enhanced microwave shielding. *Journal of Magnetism and Magnetic Materials*, 536:168118, 2021.

[334] Anomadarshi Barua, Deepan Muthirayan, Pramod P Khargonekar, and Mohammad Abdullah Al Faruque. Hierarchical temporal memory based one-pass learning for real-time anomaly detection and simultaneous data prediction in smart grids. *IEEE Transactions on Dependable and Secure Computing*, 2020.

[335] Jeff Hawkins and Subutai Ahmad. Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Frontiers in neural circuits*, 10:23, 2016.

[336] Ramin Moghaddass and Jianhui Wang. A hierarchical framework for smart grid anomaly detection using large-scale smart meter data. *IEEE Transactions on Smart Grid*, 9(6):5820–5830, 2017.

[337] Mengze Zhou, Yuhui Wang, Anurag K Srivastava, Yinghui Wu, and P Banerjee. Ensemble-based algorithm for synchrophasor data anomaly detection. *IEEE Transactions on Smart Grid*, 10(3):2979–2988, 2018.

[338] G Napier, EM Davidson, SDJ McArthur, and Member JR McDonald. An automated fault analysis system for SP energy networks: Requirements, design and implementation. In *2009 IEEE Power & Energy Society General Meeting*, pages 1–7. IEEE, 2009.

[339] Vivek Kumar Singh and Manimaran Govindarasu. Decision tree based anomaly detection for remedial action scheme in smart grid using pmu data. In *2018 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2018.

[340] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[341] Max Landauer, Markus Wurzenberger, Florian Skopik, Giuseppe Settanni, and Peter Filzmoser. Dynamic log file analysis: an unsupervised cluster evolution approach for anomaly detection. *computers & security*, 79:94–116, 2018.

[342] Jorge Valenzuela, Jianhui Wang, and Nancy Bissinger. Real-time intrusion detection in power system operations. *IEEE Transactions on Power Systems*, 28(2):1052–1062, 2012.

[343] Yuxun Zhou, Reza Arghandeh, Ioannis Konstantakopoulos, Shayaan Abdullah, Alexandra von Meier, and Costas J Spanos. Abnormal event detection with high resolution micro-PMU data. In *2016 Power Systems Computation Conference (PSCC)*, pages 1–7. IEEE, 2016.

[344] S Brahma, R Kavasseri, H Cao, Nilanjan Ray Chaudhuri, T Alexopoulos, and Y Cui. Real-time identification of dynamic events in power systems using pmu data, and potential applications—models, promises, and challenges. *IEEE transactions on Power Delivery*, 32(1):294–301, 2016.

[345] Jui-Sheng Chou and Ngoc-Tri Ngo. Smart grid data analytics framework for increasing energy savings in residential buildings. *Automation in Construction*, 72:247–257, 2016.

[346] Fenghua Gao, James S Thorp, Anamitra Pal, and Shibin Gao. Dynamic state prediction based on auto-regressive (AR) model using PMU data. In *2012 IEEE Power and Energy Conference at Illinois*, pages 1–5. IEEE, 2012.

[347] J Sia, E Jonckheere, Laith Shalalfeh, and Paul Bogdan. Pmu change point detection of imminent voltage collapse and stealthy attacks. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6812–6817. IEEE, 2018.

[348] Helin Yang, Arokiaswami Alphones, Wen-De Zhong, Chen Chen, and Xianzhong Xie. Learning-based energy-efficient resource management by heterogeneous RF/VLC for ultra-reliable low-latency industrial IoT networks. *IEEE Transactions on Industrial Informatics*, 16(8):5565–5576, 2019.

[349] Helin Yang, Xianzhong Xie, and Michel Kadoch. Machine Learning Techniques and A Case Study for Intelligent Wireless Networks. *IEEE Network*, 34(3):208–215, 2020.

[350] Keith Hollingsworth, Kathryn Rouse, Jin Cho, Austin Harris, Mina Sartipi, Sevin Sozer, and Bryce Enevoldson. Energy Anomaly Detection with Forecasting and Deep Learning. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4921–4925. IEEE, 2018.

[351] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017.

[352] Wong J. Netflix Surus GitHub. `https://github.com/Netflix/Surus2015`, 2015.

[353] Nikolay Laptev, Saeed Amizadeh, and Ian Flint. Generic and scalable framework for automated time-series anomaly detection. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1939–1947. ACM, 2015.

[354] Alexander Lavin and Subutai Ahmad. Evaluating Real-Time Anomaly Detection Algorithms–The Numenta Anomaly Benchmark. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 38–44. IEEE, 2015.

[355] Nelson Spruston. Pyramidal neurons: dendritic structure and synaptic integration. *Nature Reviews Neuroscience*, 9(3):206, 2008.

[356] Sen Song, Kenneth D Miller, and Larry F Abbott. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience*, 3(9):919, 2000.

[357] Sean Peisert, Reinhard Gentz, Joshua Boverhof, Chuck McParland, Sophie Engle, Abdelrahman Elbashandy, and Dan Gunter. LBNL Open Power Data. 2017.

[358] Emma Stewart, Anna Liao, and Ciaran Roberts. Open μPMU: A real world reference distribution micro-phasor measurement unit data set for research and application development. 2016.

[359] Scott Purdy. Encoding data for HTM systems. *arXiv preprint arXiv:1602.05925*, 2016.

[360] Yuwei Cui, Subutai Ahmad, and Jeff Hawkins. The HTM spatial pooler—A neocortical algorithm for online sparse distributed coding. *Frontiers in computational neuroscience*, 11:111, 2017.

[361] George K Karagiannidis and Athanasios S Lioumpas. An improved approximation for the gaussian q-function. *IEEE Communications Letters*, 11(8):644–646, 2007.

[362] Jeff Hawkins, Subutai Ahmad, and Donna Dubinsky. Hierarchical temporal memory including HTM cortical learning algorithms. *Techical report, Numenta, Inc, Palto Alto http://www. numenta. com/htmoverview/education/HTM_CorticalLearningAlgorithms. pdf*, 2010.

[363] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. Robust random cut forest based anomaly detection on streams. In *International conference on machine learning*, pages 2712–2721, 2016.

[364] Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

[365] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[366] Markus Schneider, Wolfgang Ertel, and Fabio Ramos. Expected similarity estimation for large-scale batch and streaming anomaly detection. *Machine Learning*, 105(3): 305–333, 2016.

[367] Chengwei Wang, Krishnamurthy Viswanathan, Lakshminarayan Choudur, Vanish Talwar, Wade Satterfield, and Karsten Schwan. Statistical techniques for online anomaly detection in data centers. In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pages 385–392. IEEE, 2011.

[368] John C Robertson, Ellis W Tallman, and Charles H Whiteman. Forecasting using relative entropy. *Journal of Money, Credit, and Banking*, 37(3):383–401, 2005.

[369] Eugene A Feinberg and Dora Genethliou. Load forecasting in: Applied mathematics for restructured electric power systems": Optimization, control, and computational intelligence. In *Power Electronics and Power Systems*, pages 269–285. Springer US, 2005.

[370] Douglas Henderson. An advanced electronic load governor for control of micro hydroelectric generation. *IEEE Transactions on Energy Conversion*, 13(3):300–304, 1998.

[371] JE Wesen, V Vermehren, and HM de Oliveira. Adaptive Filter Design for Stock Market Prediction Using a Correlation-based Criterion. *arXiv preprint arXiv:1501.07504*, 2015.

[372] Kevin J Lang, Alex H Waibel, and Geoffrey E Hinton. A time-delay neural network architecture for isolated word recognition. *Neural networks*, 3(1):23–43, 1990.

[373] Yuwei Cui, Subutai Ahmad, and Jeff Hawkins. Continuous online sequence learning with an unsupervised neural network model. *Neural computation*, 28(11):2474–2504, 2016.

[374] James R Zeidler. Performance analysis of LMS adaptive prediction filters. *Proceedings of the IEEE*, 78(12):1781–1806, 1990.

[375] Alex Waibel. Consonant recognition by modular construction of large phonemic time-delay neural networks. In *Advances in neural information processing systems*, pages 215–223, 1989.

[376] Numenta. `https://github.com/numenta/htmresearch/blob/master/projects/sequence_prediction/continuous_sequence/run_lstm_suite.py`, 2016.