Detecting Unauthorized Activity in Lightweight IoT Devices

by

Fatih Karabacak

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2020 by the
Graduate Supervisory Committee:

Sule Ozev, Co-Chair
Umit Ogras, Co-Chair
Jennifer Blain-Christen
Jennifer Kitchen

ARIZONA STATE UNIVERSITY

August 2020

ABSTRACT

The manufacturing process for electronic systems involves many players, from chip/board design and fabrication to firmware design and installation. In today's global supply chain, any of these steps are prone to interference from rogue players, creating a security risk. Manufactured devices need to be verified to perform only their intended operations since it is not economically feasible to control the supply chain and use only trusted facilities. It is becoming increasingly necessary to trust but verify the received devices both at production and in the field.

Unauthorized hardware or firmware modifications, known as Trojans, can steal information, drain the battery, or damage battery-driven embedded systems and lightweight Internet of Things (IoT) devices. Since Trojans may be triggered in the field at an unknown instance, it is essential to detect their presence at run-time. However, it isn't easy to run sophisticated detection algorithms on these devices due to limited computational power and energy, and in some cases, lack of accessibility. Since finding a trusted sample is infeasible in general, the proposed technique is based on self-referencing to remove any effect of environmental or device-to-device variations in the frequency domain. In particular, the self-referencing is achieved by exploiting the band-limited nature of Trojan activity using signal detection theory. When the device enters the test mode, a predefined test application is run on the device repetitively for a known period. The periodicity ensures that the spectral electromagnetic power of the test application concentrates at known frequencies, leaving the remaining frequencies within the operating bandwidth at the noise level. Any deviations from the noise level for these unoccupied frequency locations indicate the presence of unknown (unauthorized) activity. Hence, the malicious activity can

differentiate without using a golden reference or any knowledge of the Trojan activity attributes.

The proposed technique's effectiveness is demonstrated through experiments with collecting and processing side-channel signals, such as involuntarily electromagnetic emissions and power consumption, of a wearable electronics prototype and commercial system-on-chip under a variety of practical scenarios.

DEDICATION

To my parents,

for their love, endless support and encouragement.

To my wife,

whose sacrificial care for me and our children

made it possible for me to complete this work.

ACKNOWLEDGMENTS

First and foremost, I should express my sincere gratitude to my advisors, Sule Ozev, and Umit Ogras to guide and support me over the years. I would not have been able to do this dissertation without their help. I felt very fortunate to have worked with and learned from them. I am indebted to them a lot, and I cannot thank them enough.

I would like to thank my thesis committee members, Dr. Jennifer Kitchen and Dr. Jennifer Blain Christen, for the consideration to be members of the committee and allocating their precious time to do that; your ideas and feedback have been invaluable.

I'd like to thank my fellow graduate students, Md Muztoba, Tanvir Mustofa, Ganapati Bhat, and collaborators, who contributed to this research. I am very grateful to all of you.

I am also grateful to my brother Murat Karabacak, for supporting and constant encouragement I have gotten over the years, my parents Dondu Karabacak and Recep Karabacak, for giving birth to me at the first place and supporting me spiritually throughout my life.

Finally, I sincerely thank my wife, Halime, who listened and discussed this thesis's ideas on many occasions. Her true love and support have always been my strength. Without her help, I would not have been able to complete much of what I have done and become who I am. It will be ungrateful on my part if I thank her in these few words. I am thankful to my two children, Selime Betul and Humeyra, who for giving me happiness during this journey.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Motivation

Owning and operating a foundry with advanced capabilities can no longer be afforded by most design companies. Splitting design and manufacturing into separate businesses has enabled small and innovative design companies to flourish. However, such a distributed flow is also vulnerable to various forms of security and quality threats in the supply chain. These threats can include poor quality control, overbuild ICs, reverse engineering, malicious modification, and so forth. Among these threats, malicious modifications are arguably the most significant concern and have garnered considerable attention (Tehranipoor and Koushanfar 2010; Yier Jin and Yiorgos Makris 2008; Chakraborty, Narasimhan, and Bhunia 2009).

Systems-on-chip (SoC) design based on reusable IP is now a pervasive practice in the semiconductor industry due to the dramatic reduction in design/verification cost and time it offers. Cost and development time are two major problems faced by the designers of low-volume SoCs. Hence, they utilize third-party IP cores from dozens of different vendors to amortize the cost and shrink design turn-around time. Moreover, the boards and firmware for these devices can also be developed and installed by third-party vendors.

System integration is a big task, and no single company can do on their own. With a global supply chain consisting of many players, as seen in Figure 1, ensuring the security of wearable and IoT devices is a daunting challenge. It is becoming

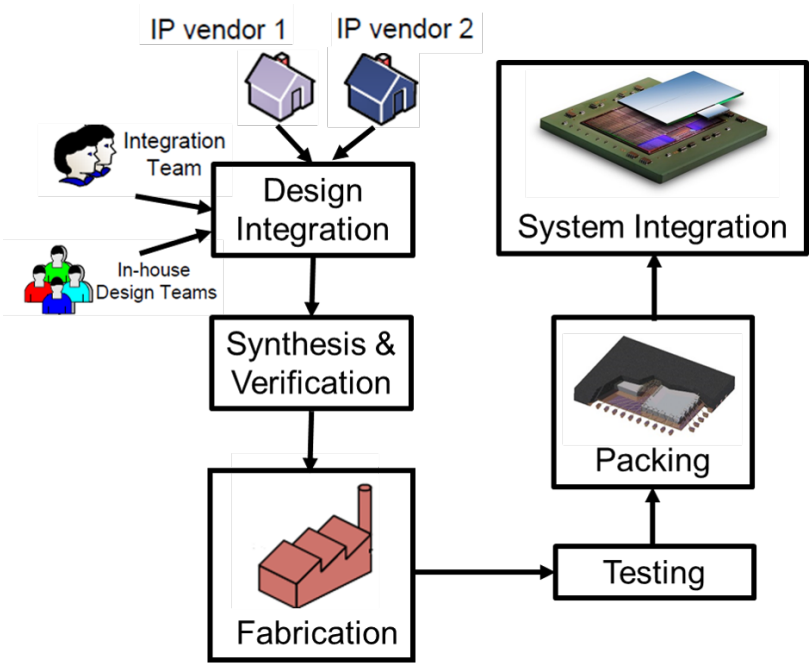


Figure 1: Major steps in the IC design flow.

increasingly necessary to trust but verify the received devices both at production and in the field.

Battery-driven embedded systems and small form-factor devices such as wearable and lightweight IoT devices are the resource-constrained embedded devices that only have limited computing power and battery capacity. These limitations make existing cybersecurity mechanisms, such as anti-virus software and anomaly detectors (Shila, Geng, and Lovett 2016), too costly to implement. Thus, ensuring the security of wearable devices with acceptable overhead is a new challenge that requires cost-conscious solutions. The security of embedded devices becomes more critical with the increasing interconnection of devices and the usual presence in any networks.

1.2 Malicious Modifications

A Trojan is defined as a malicious, intentional modification of circuit design or firmware code that results in undesired behavior when the circuit is deployed. Attacks on IoT devices can be realized in the form of malicious hardware or firmware modifications. The Trojans are designed in such a way that they trigger on rare conditions. Due to the wide range of activation mechanisms and lack of knowledge of Trojan's architecture or effect, the detection of Trojans is a daunting task.

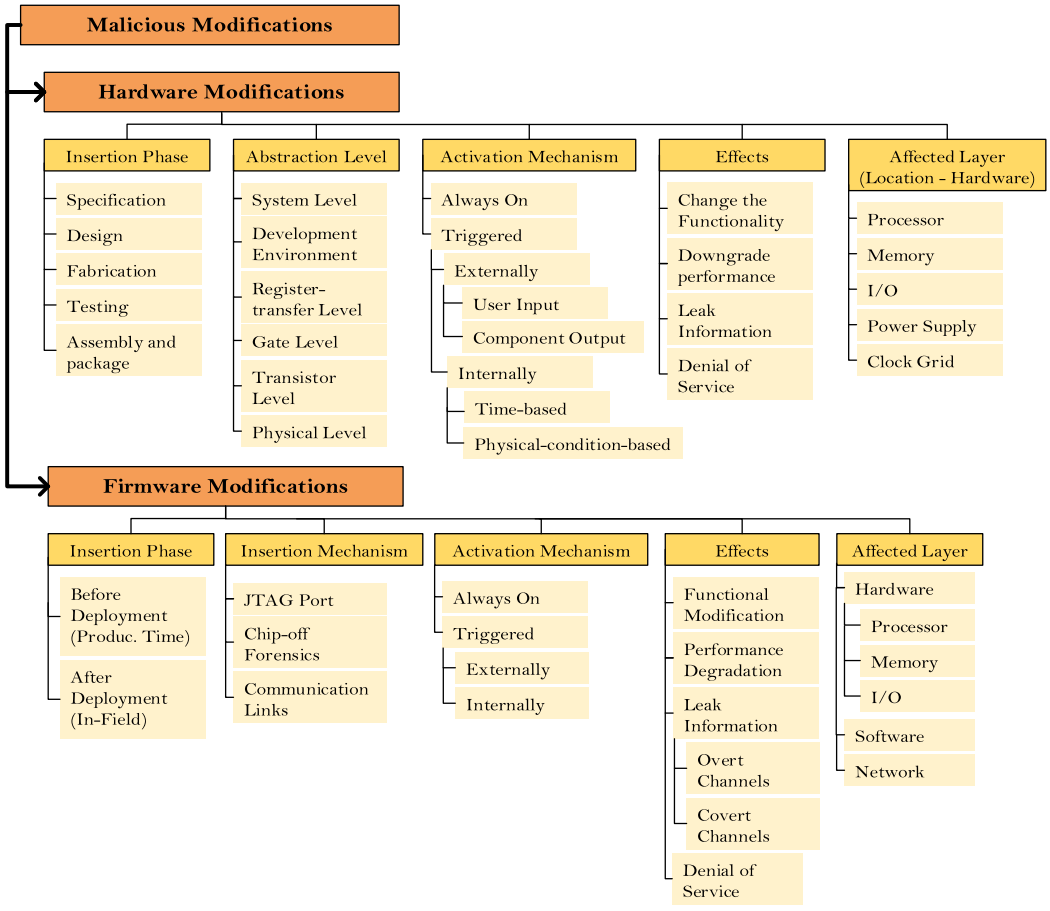


Figure 2: Hardware and Firmware Trojan taxonomy.

Research in the past decade has shown that hardware or firmware modifications can be inserted into the design without noticeable performance and hardware impact for nefarious purposes, such as stealing information, or causing malfunction (Narasimhan et al. 2012; Nowroz et al. 2014). An extensive taxonomy of hardware modifications (called "Hardware Trojans") and firmware modifications (called "Firmware Trojans") appears in Karri et al. 2010; Konstantinou, Keliris, and Maniatakos 2016 and presented in Figure 2, respectively. The classification is happened according to the time at which the Trojan is injected; how to insert the Trojans and impacted layer; activation types; effects; and location. The Trojans are sorted based on the different attributes that indicate several Trojans properties, such as potential impact and size of Trojan, etc.

1.3 Unauthorized Activity Detection Techniques

Malicious modifications have become a serious security threat for integrated circuits (ICs), especially systems used in critical applications and cyberinfrastructure with the globalization of semiconductor design and fabrication. Several malicious activity detection techniques can address or mitigate potential threats in the supply chain with recent research works over the past few years. Trojan detection aims to verify the existing designs and fabricated ICs without any additional circuitry. Those detection techniques are classified into two main categories, and can be organized into several subcategories, as shown in Figure 3 (Xiao et al. 2016). They are performed either at the design stage (i.e., pre-silicon) to validate IC designs or after the manufacturing stage (i.e., post-silicon) to verify fabricated ICs. The manufactured ICs can be tested with destructive or non-destructive techniques.

Destructive techniques generally use reverse-engineering techniques to obtain

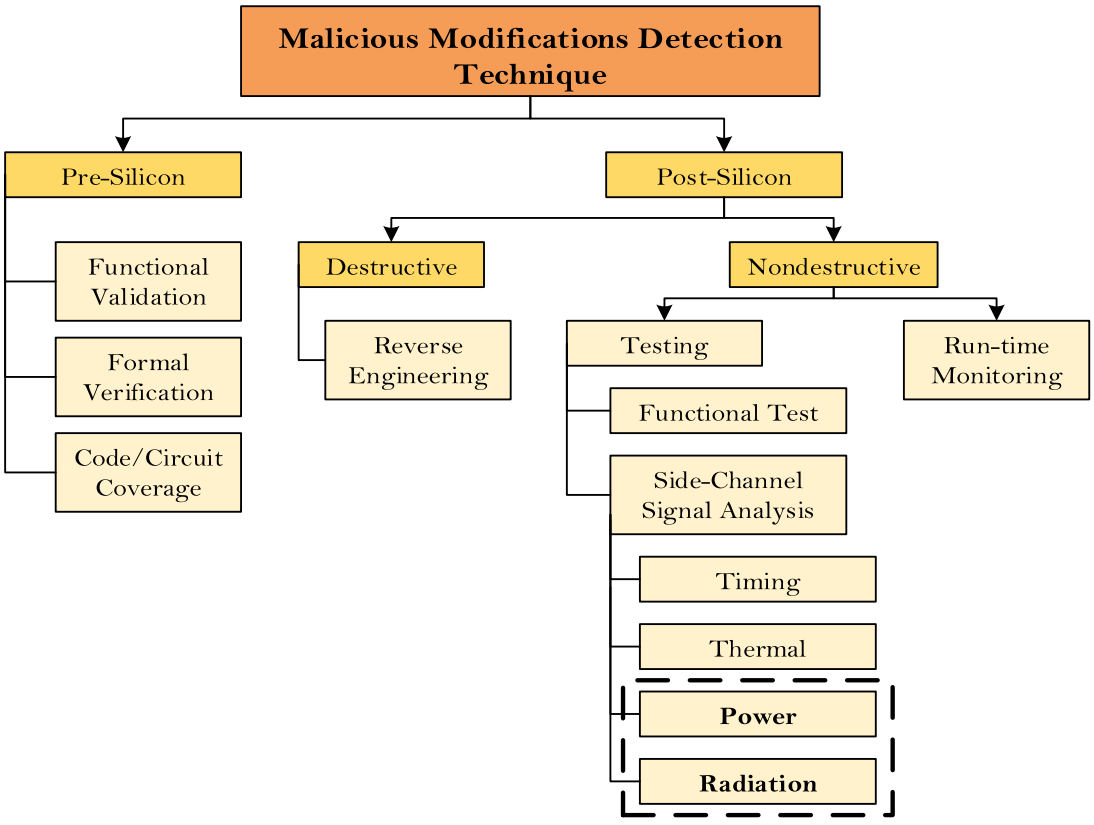


Figure 3: Trojan Detection Classification.

the characteristic of a Trojan free ICs, which is referred to as a golden reference. They promise a high percent rate to detect any malicious modification in the IC, as they are costly and time-consuming. Furthermore, the process variations can have a significant effect on the detection rate and Trojan free chips based on a golden reference. Non-destructive techniques can be classified into run-time monitoring and testing. Run-time monitoring is used to monitor for any abnormal behavior of IC continuously. This approach can utilize any pre-existing redundancy or supplemental on-chip structure to monitors the IC. However, this structure can increase the chip area, power, and delay to reduced circuit performance. Any modification to a circuit should be reflected in the parameters of the chip. Side-channel signal analysis is based

on the parameters changes such as dynamic power, leakage current, path-delay, and electromagnetic (EM) radiation. The advantage of the side-channel analysis over other techniques is not having to activate the Trojan to detect. However, this technique relies on trusted samples and does sensitivity to errors due to process variations and noise. This research work is focused on the side-channel analysis method based on self-referencing to remove any effect of environmental or device-to-device variations.

1.4 Contributions

The primary contributions of this research are summarized as follows:

- Proposed a self-referenced malicious activity detection technique applicable to not only sinusoidal excitation, but also to repetitive patterns to remove the effects of process and environmental variations.
- Introduced an algorithm for self-referencing the repetitive signature of the authorized activity.
- Established a methodology for limited-bin spectral analysis to detect of unauthorized activity to reduce the computational burden of the detection technique.
- Presented a technique to place the design under test (DUT) in a repetitive state to limit the frequency response of its authorized activity signature in a small number of frequency bins.
- Experimented applicability of proposed self-referenced malicious activity detection technique for power and electromagnetic (EM) parameter side-channel analysis.
- Developed an algorithm for low-cost analytical classification that can flag unauthorized activity automatically without a golden reference.

- Presented a methodology for time-domain signal stitching to collect side-channel signal information on repetitive primary activity to reduce test duration.
- Modelled side-channel parameter (power and EM) as a sum of the primary circuit, malicious activity, and environmental noise. Then, it showed how this model could be used for self-referencing through signal processing techniques.
- Showed thorough experimental evaluation on an embedded platform with a detection distance ranging from 30cm to 200cm for the stand-off methodology.
- Evaluated the proposed approach while running gesture recognition and Wi-Fi applications without requiring a trusted sample.
- Performed extensive experiments using a wearable electronics prototype (Gupta, Park, et al. 2017) and a multiprocessor system-on-chip (MpSoC) (ODROID, n.d.), and demonstrate the effectiveness of the proposed detection technique.

1.5 Thesis Organization

This thesis is organized into four chapters other than the introduction chapter; Chapter 2 discusses a review of relevant literature. Chapter 3 introduces remote detection of unauthorized activity processing involuntarily available electromagnetic emissions via spectral analysis on a separate detection apparatus, which is physically decoupled from the device under test. Chapter 4 proposes a malicious activity detection technique in lightweight wearable and IoT devices that relies on repetitious side-channel sample collection of the device, time-domain stitching, and frequency domain analysis. Finally, conclusions are discussed in Chapter 5.

Chapter 2

LITERATURE REVIEW

Battery-driven embedded systems have limited computing power and battery capacity. These constraints only worsen when the system is subject to the demands of security. There is no available security mechanism for embedded devices like anti-virus and anomaly detectors found on general-purpose computers, making them attractive targets of present-day sophisticated and innovative attacks. Those attacks can be realized in the form of hardware (Antonopoulos, Kapatsori, and Makris 2018) and firmware modifications (Kocher et al. 2004) as known Trojans. This wide range of attack space has resulted in exponentially increasing security problems.

Hardware Trojans are small scale circuit structures that perform a malicious function not intended by the designer of the system (Tehranipoor and Koushanfar 2010). They can be inserted at multiple points in the supply chain, for instance, by the production foundry (Xiao, Forte, and Tehranipoor 2014), or by a third-party IP provider (Bidmeshki and Makris 2015). Techniques have been developed to detect these Trojans at test time or at run time. Out of these, run-time detection approaches are most relevant to our work. Run-time detection approaches for hardware Trojans can modify the entire architecture of the structure that they aim to protect (Kim, Villasenor, and Koç 2009), or use multiple modular redundancies at the software level to enable accomplishing of a task even in the presence of malicious hardware (Yu and Frey 2013). Test-time approaches can be based on either functional verification (Kim, Villasenor, and Koç 2009) or side channel measurements (Agrawal et al. 2007; Banga and Hsiao 2009). Functional verification based techniques aim at statistically

determining test patterns that have the highest probability of detecting Trojans that result in logic failure.

Side-channel measurement-based techniques aim to infer the presence of a Trojan through measurement of parameters that would be modified by the Trojan's presence. These parameters can be path delays, transient supply current, or average supply currents. A big challenge in side-channel measurement-based Trojan detection is the lack of a known reference. This reference is generally established via transistor-level simulations (Cha and Gupta 2013), or it is assumed that trusted samples of the circuit under test can be obtained. Even when a reference can be created, noise in the environment and measurement errors pose another daunting challenge. Often, the symptom of the Trojan (in terms of delay deviation and/or current consumption) can be comparable to variations due to process and environment noise. Most common run-time detection approaches rely on statistical analysis on the measured side-channel parameters (Du et al. 2010; Narasimhan et al. 2011), to suppress the effect of process variations. Multi-dimensional analysis can be used to detect the presence of Trojan under process variations (Hu et al. 2013; Xiao, Zhang, and Tehranipoor 2013). If Trojan-free samples are available, more sophisticated training tools can be used to enhance detection capability (Liu, Jin, and Makris 2013). However, statistical techniques may fail if the deviation of the circuit parameters due to a Trojan's presence is comparable to that of differences with respect to within-die variations, and/or Trojan response is comparable to noise. In Du et al. (2010) and Narasimhan et al. (2011), this problem is alleviated by using the same input pattern and comparing the instantaneous and average supply current. In He et al. (2015), the authors propose a side-channel hardware Trojan detection method using frequency domain analysis based on a golden reference.

Side channel parameters include power consumption patterns, involuntary electromagnetic (Sauvage, Guilley, and Mathieu 2009; De Mulder et al. 2005), acoustic (Backes et al. 2010), and thermal emissions, network, cache, or memory access patterns (Al Faruque et al. 2016), and even chassis vibrations (Genkin, Pipman, and Tromer 2014). It has been shown that encryption keys can be extracted using EM emissions (Sauvage, Guilley, and Mathieu 2009; De Mulder et al. 2005). In Standaert, Malkin, and Yung 2009, the authors present a generic method to analyze security algorithms in terms of their level of involuntary emissions. This study shows that formal analysis methods can be used to quantitatively compare software blocks with respect to their identifying properties when monitored non-intrusively. In (Peeters, Standaert, and Quisquater 2007), the authors demonstrate that by placing the coil near the CPU/memory connections, they can determine the time when an algorithm transfers data. Similarly, the work presented in (Callan, Zajic, and Prvulovic 2014) shows that by measuring multiple side-channel signals (EM emissions, power fluctuations, and acoustic emissions), it is possible to determine program activity at the instruction level. A metric that quantifies the side channel signal's dependence on the instructional-level program execution differences is presented. The authors propose a measurement method to reduce the relative error and can be performed at much lower frequencies using commercially available instruments. The proposed method also relies on a periodic steady-state, which is created by altering instruction-level codes. The analysis of golden signatures can help select what instruction sequences provide the best spectral signatures to detect unauthorized activity. However, these signatures will still be subject to process variations, the suppression of which is the focus of this paper. Finally, it has been also demonstrated that acoustic emissions

could be used to extract RSA keys (Genkin, Shamir, and Tromer 2014) and identify activity in 3D printers (Faruque et al. 2016).

Most hardware Trojan detection techniques rely on an intimate knowledge of architecture and circuit-level design of the design under test, require trusted samples, or require direct hardware control. In the case of already deployed IoT nodes with hardware supplied from many different vendors, this may be infeasible. In contrast to prior work, our proposed techniques are based on signal processing and signal detection theory and do not require an established baseline.

In contrast, the firmware is easily distributed, making a much easier target for someone intent on compromising the embedded systems. Since embedded devices run on firmware, we need to understand how the firmware works. The firmware provides necessary information for how the hardware device communicates with other devices. It is found on all kinds of computer hardware, but where it is the most vulnerable is embedded devices that generate or exchange vast amounts of privacy-sensitive and security-critical information (Ravi et al. 2004). Since an increasing demand to connected embedded devices on the emerging IoT (O'Neill 2016), in addition to continually evolving attack techniques in recent decades, firmware security has become more critical than ever to organizations such as a bank, government, and businesses (Keoh, Kumar, and Tschofenig 2014; Sadeghi, Wachsmann, and Waidner 2015).

We can basically classify firmware attacks as static and dynamic attacks. While static firmware attack focuses on modifying the firmware code residing in memory with hardware modification or during the firmware update or patching process (Miller 2011; Cui, Costello, and Stolfo 2013; Bachy et al. 2015), dynamic firmware attack tries to exploit dynamic memory components like as stack and heap to change the behavior of the control flow of firmware (Bletsch et al. 2011). The leveraged vulnerabilities in

firmware for malicious modification has been addressed in several research studies from battery-powered personal health monitor devices to conventional industrial control systems (Rieck 2016; Konstantinou and Maniatakos 2015; McLaughlin et al. 2016).

Konstantinou and Maniatakos (2015) introduce firmware modifications as a new class of cyber-physical attacks against the smart grid environment and demonstrate how attackers can exploit design flaws and disrupt the operation of Circuit Breakers (CBs) by injecting spurious data to the breaker controller. The authors present two attack vectors based on the openings and closing status of CB vulnerability and the inability of the relay to sense a fault and initiate a trip to the CB. The authors further upload a modified firmware to the relay controller and reveal that the update validation employs a trivial checksum function. The researchers analyze different attack scenarios and conclude that maliciously firmware could cause a cascade of power outages. Bencsáth, Buttyán, and Paulik (2011) demonstrate hidden firmware modification attacks on embedded devices that have Web-based management interfaces vulnerable to Cross-Site Scripting type attacks whereby some malicious code can be injected in the Web pages stored on the device. They present a general framework of firmware modification attacks to show how the malicious script can install an arbitrarily modified firmware to enable botnet clients. The vulnerabilities indeed exist in commercially available devices. The framework has been implemented on Planex wireless routers to demonstrate how this vulnerability can create an entry point to install malicious code to turn the devices into bots in coordinated botnets. Programmable logic controllers (PLCs) are prevalent programmable devices that manage and control physical system components based on user-provided inputs and requirements. Basnight et al. (2013) investigate and assess the vulnerability of a conventional industrial PLC to counterfeit firmware updates. The PLC vulnerability

is examined by first using reverse engineering techniques to infer the firmware update validation method. The firmware update validation method is then analyzed for weaknesses that facilitate firmware modification and counterfeiting. The failings are subsequently exploited to create a counterfeit firmware sample that is uploaded and executed on a PLC. Tekeoglu and Tosun (2016) propose a low-cost testbed using publicly available open source software and inexpensive off-the-shelf hardware for investigation of a wide range of topics on security and privacy of IoT devices. The proposed testbed is designed to investigate security and privacy issues in IoT devices that use WiFi or Bluetooth by capturing transmitted packets. In Costin et al. (2014), the authors conduct a detailed analysis of security threats in embedded firmware. A correlation technique is used to determine whether the firmware contains any known vulnerabilities. The analysis reveals 693 firmware images known to have been affected by at least one vulnerability and reported 38 new firmware vulnerabilities. However, this approach cannot notify previously unknown firmware modifications. Recent research has revealed and exploited firmware update vulnerabilities in popular health monitoring devices, such as FitBit or Withings Activite activity trackers, which can be used to inject malicious code (Rieck 2016; Rahman, Carbunar, and Banik 2013). Cui et al. present a case study of firmware modification attacks against HP LaserJet printers that allow an adversary with only printing permission to update the printer with modified firmware (Cui, Costello, and Stolfo 2013).

On the detection and identification side, however, there is no much research work available. In these limited works, detection approaches are also classified as signature-based that is looking for signatures of known attacks or anomaly-based that is modeling the normal behavior of firmware and detecting deviations from this referenced model. A recent study focused on a low-cost technique to detect

the malicious firmware modification at the embedded devices using readily available registers (Wang et al. 2015). The proposed framework needs exhausted offline profiling for generating a referenced database. Moreover, this detection mechanism relies on the write-protected memories components, which are still vulnerable to alteration with hardware modification. The authors of Duflot et al. (2010) described a firmware vulnerability in the network adapter that a remote attacker on the network can gain full access to the victim's machine. They propose practical detection techniques that detect any unexpected change in the control flow when a return value is modified in the network adapter (Duflot, Perez, and Morin 2011). In Shila, Geng, and Lovett (2016), anomaly analysis for embedded firmware by employing source code instrumentation techniques is proposed. Any deviation with used defined threshold from the referenced run of the firmware is flagged as anomalous. The proposed technique needs to reference model and run the instrumented firmware extensively offline. However, it has much overhead for the computationally intensive tasks.

Our proposed technique fills the gap on the detection side of the malicious modification that can not be detectable with the existing detecting mechanism. We prove that how powerful our approach with applicable experiments.

Chapter 3

REMOTE DETECTION OF UNAUTHORIZED ACTIVITY VIA SPECTRAL ANALYSIS

3.1 Introduction

Today, there are about 20 billion devices in the world that interact with each other, and by 2025 it is estimated to go up to 75 billion devices (Evans 2011). These devices will have vastly different processing capabilities ranging from straightforward Internet of Things (IoT) end nodes, such as sensors, to high-end computing systems (Bogdan et al. 2016). For example, smart objects are already in use for supply chain management (Yan and Huang 2009), smart city management (Vlacheas et al. 2013), health care (Dohr et al. 2010), industrial management (Aref et al. 2007), and smart grid (North American Electric Reliability Council, New Jersey 2009). These devices pose new security challenges, mainly due to hardware, computation, power limitations, and, in some cases, lack of accessibility (Graves et al. 2015). Consequently, securing them is a daunting task that developers are currently paying scant attention to (Sklavos et al. 2017; Regazzoni and Polian 2017).

Research in the past decade has shown that small hardware or firmware modifications (also known as hardware Trojans) can be inserted into the design without noticeable performance and hardware impact for nefarious purposes, such as stealing information, or causing malfunction (Narasimhan et al. 2012; Nowroz et al. 2014). Hardware Trojans can be inserted into the circuit at the layout, through third party intellectual property (IP) cores, or at the firmware, such as FPGA codes.

In addition to continually evolving attack techniques, the firmware has become a popular target for security vulnerabilities (O'Neill 2016; Keoh, Kumar, and Tschofenig 2014; Sadeghi, Wachsmann, and Waidner 2015). With increasing complexity and functionality added to the underlying hardware, IoT devices are commonly designed with a firmware update feature (Kachman and Balaz 2016). This feature enables end-users or vendors with appropriate access to patch bugs and upgrades firmware without requiring physical changes to the hardware. Unfortunately, the same functionality also leads to another form of attack, where malicious firmware code can be inserted into the device via a firmware update. Sophisticated security mechanisms, such as viruses and anomaly detectors found on general-purpose computers, are often unavailable for embedded devices and IoT nodes, which makes them attractive targets for innovative attacks (Kocher et al. 2004). Large scale deployment of these devices has already occurred and exploited by attackers (Leander 2016; Costin et al. 2014). Such malicious firmware modifications have been demonstrated for battery-powered personal health monitor devices and industrial control systems (Rieck 2016; Konstantinou and Maniatakos 2015).

We define unauthorized activity (*malicious activity*) as activity due to any firmware or hardware modification not authorized by the user. The unauthorized activity, i.e., an attack, can be triggered at an unknown time during regular operation. Detection of such unauthorized activity for IoT nodes requires a different approach than the protection of sophisticated computer systems. For IoT nodes, unauthorized modifications to the policy may be below the operating system level, rendering most software-based protection systems ineffective. Since we cannot rely on the hardware to be Trojan-free, ideally, detection should be achieved in a manner decoupled from the device under test (DUT).

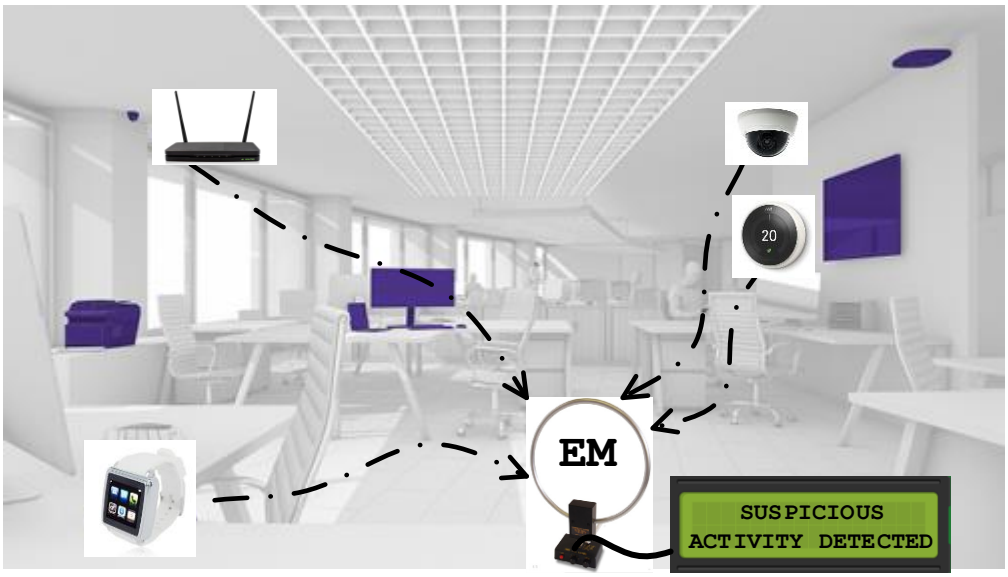


Figure 4: Stand-off security verification illustration of multiple IoT devices using gateway EM capture.

Decoupled detection can be achieved via observing side-channel information, such as current (power) consumption, electromagnetic (EM) emissions, and acoustic emissions. While measuring power would provide the best resolution in terms of information, it also requires contact with the IoT node, which may not be feasible. Other forms of side-channel information have been used to learn information, such as encryption keys, from the device (Ghosh et al. 2016; Heuser et al. 2016). The same type of involuntary emissions, which have been used to attack the device, can also be used to protect it from malicious activity. To this effect, using electromagnetic emissions as a signature from the DUT is a potential solution for this problem that has received attention recently (Standaert, Malkin, and Yung 2009). If the Trojan signature is known, cross-correlation based techniques can be used for its detection (Bhasin et al. 2013). However, reliance on this knowledge would limit the applicability of the detection

and make this a reactive process. Similarly, if golden signatures of the DUT can be obtained, a one-class classifier can be used to detect unknown Trojans (Bao, Forte, and Srivastava 2014). However, obtaining golden (Trojan-free) samples of the DUT is difficult if not infeasible. Moreover, process and environmental variations would limit the efficacy of golden signatures. As illustrated in Figure 4, our vision to protect the IoT devices is to have a separate detection device that serves many IoT devices, and that can be used to scan the environment for EM signatures periodically.

The detection hardware communicates with each of the IoT devices to put them into the test mode, run the test application, and capture and process the EM signature. As an example, Figure 5 shows the EM emissions of two applications running on the same device. These two applications are identical, except for a Trojan code that is only a few lines. While there are minute differences in the signals, it would be challenging to differentiate the unauthorized activity from an authorized event without knowledge on Trojan or golden signature.

This work aims at filling this gap and proposes a method that relies on neither golden signatures nor Trojan signatures for detection. We present a complete stand-off detection system for unauthorized activity in IoT devices. We aim to detect unexpected behavior, which can be due to any Trojan. The detection tool resides on a separate piece of hardware, which is physically decoupled from the DUT. It receives the EM emissions from the DUT using a low-frequency antenna located within a 120 cm distance. Our key observation is that *multiple runs* of a predefined application on the *same* IoT device produces similar EM emissions regardless of process variations. We place the DUT into a test mode when it is not in use. During this mode, we run a predefined test application repetitively with a fixed period.

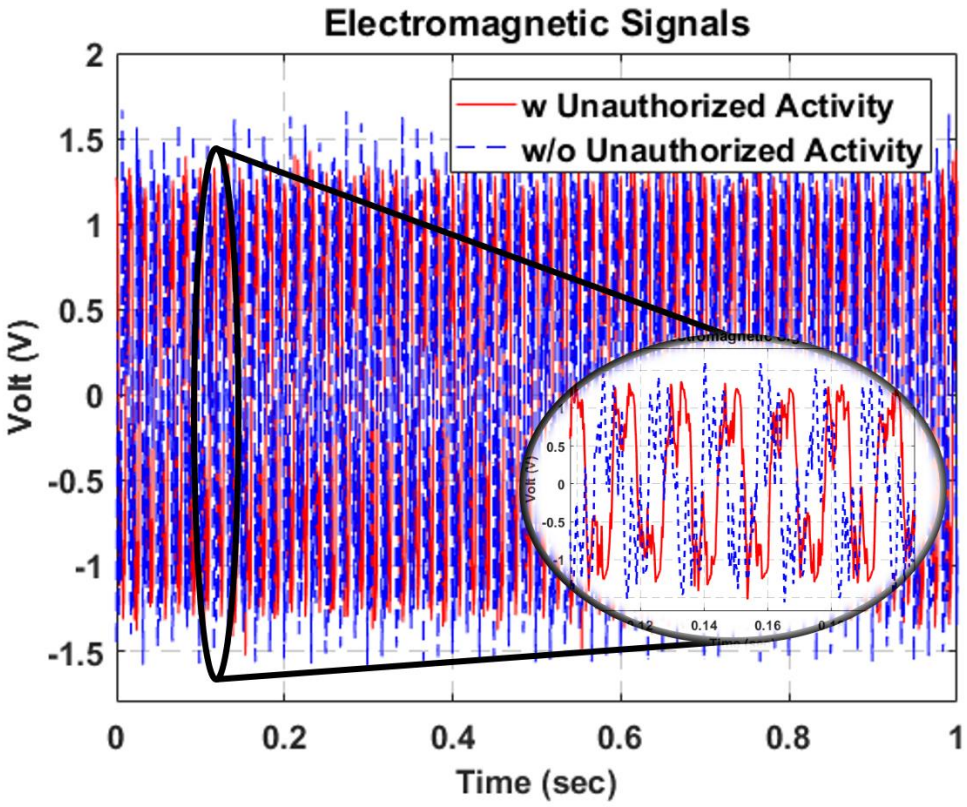


Figure 5: Comparison of measured electromagnetic emission from experimental device under test with and without unauthorized activity. The test application run time is around 3ms.

Figure 5 shows the time domain EM signature of repetitive authorized activity (test application), along with the EM signature of the same application with a Trojan that runs at random intervals. Environment noise and jitter would make detection unreliable in the time domain. However, the EM response due to this authorized activity occurs at known frequencies, which are the harmonics of the application period. It is unlikely that the period of the Trojan activity will match the application periodicity. Therefore, they will inevitably occupy other frequency bins. As a result, we can infer the presence (or absence) of unauthorized activity by comparing the

power in these bins with the noise level. Finally, we develop an analytical comparison method that averages noise power in higher frequency bins and determines a threshold for unoccupied frequency bins. This post-processing method enables self-referencing using the repetitive signature as well as the noise level. We call this the analytical detection method (ADM). We note that the test mode is a small fraction of the time the DUT is used (<1 minute). Our approach does not place any requirements on the applications running on the platform at other times. We performed our experiments on a commercial system-on-chip which run Android OS (ODROID, n.d.).

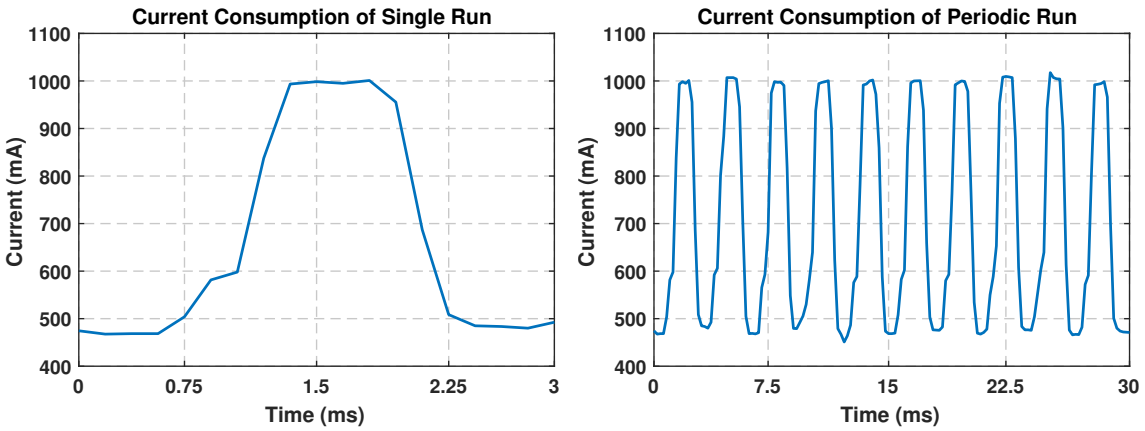
The major contributions of this work are as follows:

- A technique to place the DUT in the repetitive state to limit the frequency response of its authorized activity signature in a small number of frequency bins,
- An algorithm for self-referencing the repetitive signature of the authorized activity,
- An algorithm for low-cost analytical classification that can flag unauthorized activity automatically without a golden reference,
- Thorough experimental evaluation on an embedded platform with a detection distance ranging from 30 cm to 200 cm.

The rest of the chapter is organized as follows. Section 3.2 describes the unauthorized activity detection framework in detail and compares the proposed classification algorithm to a solid baseline classifier. Section 3.3 overviews the experimental setup and presents detailed evaluations. Section 3.4 discusses the usage model of the proposed stand-off detection technique. Finally, the limitations of the proposed approach appear in Section 3.5.

3.2 Unauthorized Activity Detection

Differentiating legitimate activity from an unauthorized activity is a daunting task due to variations in the hardware behavior resulting from the process, voltage, and environmental factors. To suppress these variations, we propose to use *self-referencing*, where we compare the DUT response within one time period to its response at another period for the same program phase. Unless there is an unauthorized activity, the periodic execution should display near-identical behavior. Unauthorized activity, however, is likely to cause variations from one execution to another. Our goal is to identify these variations to detect potentially unauthorized activity.



(a) The single run **(b)** The periodic run

Figure 6: Measured current consumption when test application runs a single and periodically run of the test application.

EM emissions are primarily due to current consumption. In order to detect any unauthorized activity in the DUT with its EM response, we need to understand its overall current consumption. The current consumption of the DUT, $I(t)$, can be expressed as:

$$I(t) = I_{app}(t) + I_{ua}(t) + n(t) \tag{3.1}$$

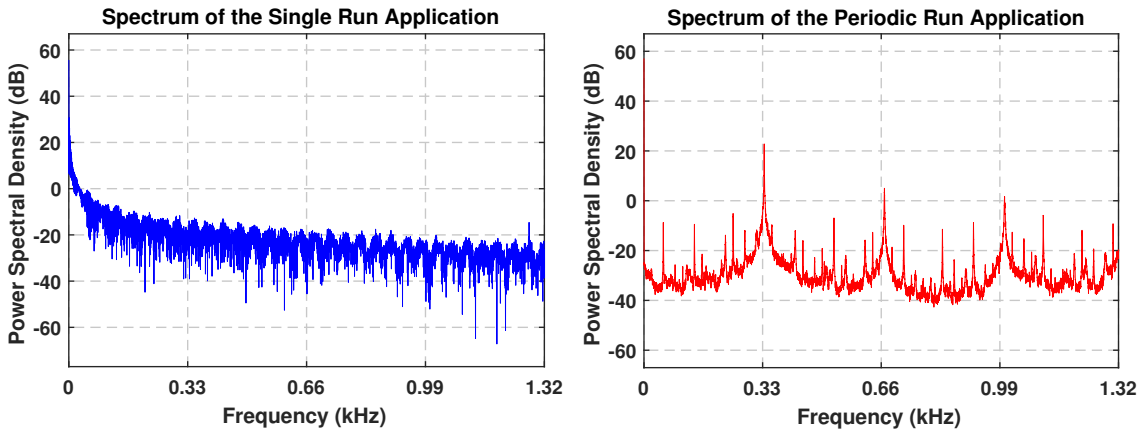
where $n(t)$ is random noise, I_{app} and I_{ua} denote the currents drawn by test application and unauthorized activity, respectively. As an example, the current consumption of the single run of an application is shown in Figure 6a. Figure 6b shows measured current consumption of the repetitive run of the application, which can be obtained by repeating the single run after some fixed period of time, T_0.

$$I_{app}(t) = I_{sr}(t) * \left(\sum_{m=1}^{M_{app}} \delta(t - mT_0) \right) \tag{3.2}$$

$$I_{app}(t) = \sum_{m=1}^{M_{app}} I_{sr}(t - mT_0) \tag{3.3}$$

The current consumption of the periodic run, I_{app}, can be expressed in terms of the current consumption of the single run, I_{sr}, as in Equation 3.2, where M_{app} is the number of single runs in the measurement duration. For a given single run, the current consumption, I_{sr}, is a continuous function that is 0 everywhere except for the duration of the run, T_0: $0 \leq t < T_0$. Thus, the convolution reduces to a Dirac train in the time domain, as in Equation 3.3.

The spectrum of the overall current is the sum of the spectra of its components. Since $n(t)$ represents the noise, its power will be spread over the entire measurement bandwidth. We denote the frequency domain representation of the single run of the application with $S_{sr}(f)$. Figure 7a plots the frequency spectrum of single run shown in Figure 6a. The frequency-domain expression of the repetitive run of the application is given in Equation 3.5 and plotted in Figure 7b.



(a) The single run **(b)** The periodic run

Figure 7: Current consumption spectrum of the single run and the periodic run of test application.

$$S_{sr}(f) = \int_{-\infty}^{+\infty} I_{sr}(t) e^{j2\pi ft} dt \tag{3.4}$$

$$S_{app}(f) = \frac{1}{T_0} \sum_{m=0}^{M_{app}-1} S_{sr}(f) \cdot \delta(f - mf_0)) \tag{3.5}$$

In a practical setting, there will be some variations in $I_{sr}(t)$ from period to period, due to system clock jitter, and other unknown system or environmental changes. This variation will result in the EM power of the application to be spread around its original location. However, most of the application EM power will be concentrated at or around harmonics of the fundamental frequency, f_0. If the bins at and around these harmonic frequencies are taken out of consideration, the remaining spectral signature can be referred to as the self-referencing signature. In the absence of any additional activity, this signature should display a flat frequency spectrum that is only due to noise. The unauthorized activity current consumption, $I_{ua}(t)$, is an unknown

signal. However, its switching speed is limited by the system clock. Therefore, the unauthorized activity will present with a spectral signature that is different than noise. Assuming that the unauthorized activity is uncorrelated with the test application, its spectrum will also occupy bins that are expected to be unoccupied. This will enable decoupling the signature of the unauthorized activity from the test application activity (Karabacak, Ogras, and Ozev 2016). Figure 8 shows the self-referenced power spectrum of the EM signatures with and without unauthorized activity.

3.2.1 Detection and Classification Algorithms

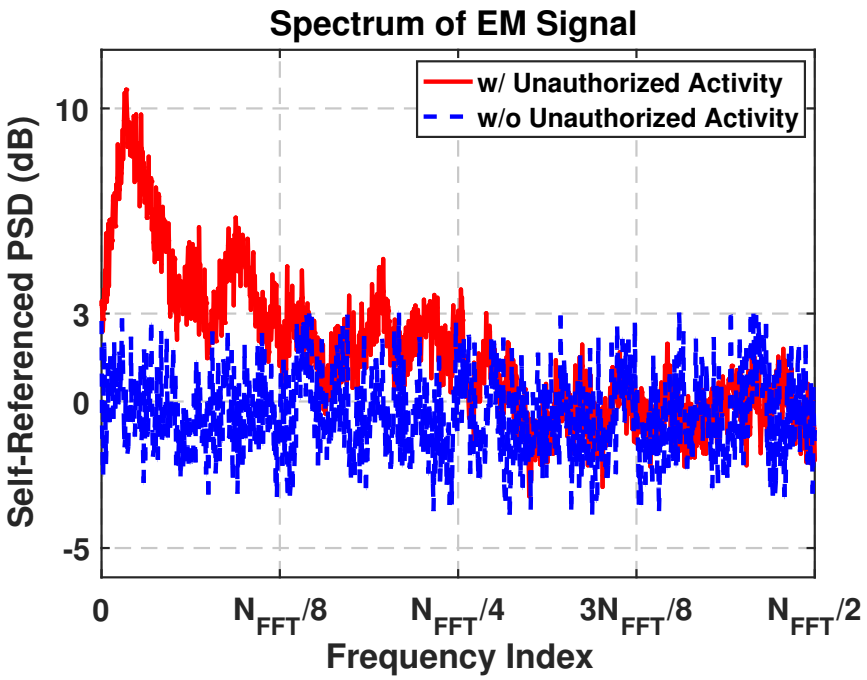


Figure 8: Self-referenced power spectrum of the EM signal.

Visual inspection of Figure 8 clearly reveals that there is a significant difference between the spectra with and without unauthorized activity. However, two issues need

to be addressed to turn this visual inspection into a detection algorithm. First, we need to develop a numerical metric to express this difference. Second, we need to avoid reliance on any signature information from the application or unauthorized activity. As noted earlier, once all occupied frequency bins are removed from consideration, the remaining signature should display the same characteristics as noise (i.e., flat spectrum). This expected behavior is the one that we will use for the detection of any unauthorized activity.

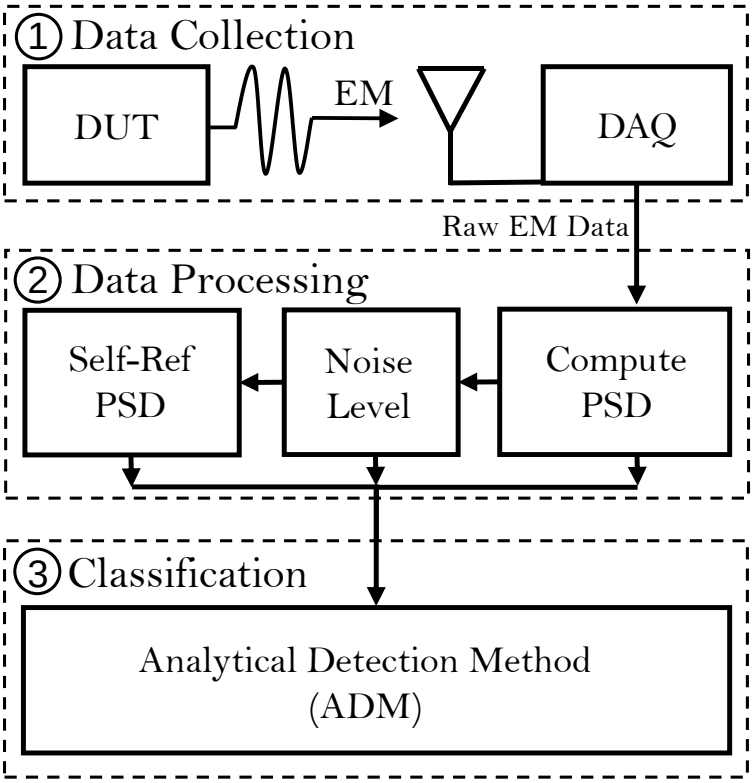


Figure 9: Flow cart of detection algorithm.

The proposed detection algorithm is presented in Figure 9. First, the EM signals

Table 1: Confusion matrix of the one-class classification.

		Actual Results	
		With Unauthorized Activity	No Unauthorized Activity
Detection Results	Detected	*True Positive (TP)*	*False Positive (FP)*
	Not Detected	*False Negative (FN)*	*True Negative (TN)*

captured by the antenna are saved to the data acquisition (DAQ) device. Then, the power spectral density is computed using the recorded data. The EM spectrum includes expected device frequency components, S_{app}, measurement and environment noise, as well as other unintended frequency components. It is evident in Figure 8 that the power in high-frequency bins is mainly due to noise. Therefore, we can use the high-frequency fraction of the measurement bandwidth to produce the reference noise level, which can be expressed as

$$\mu_n = \frac{8}{N_{FFT}} \sum_{f=\frac{3}{8}N_{FFT}}^{\frac{4}{8}N_{FFT}} S_{EM}(f) \tag{3.6}$$

where N_{FFT} is the number of samples in Fourier transform, and S_{EM} is the spectrum of the EM signal. Due to random characteristics of the unauthorized activity and its band-limited nature, its spectrum is condensed at the lower frequencies (Stepanov and Venetsanopoulos 2008). In Section 3.3, we conduct experiments on various levels of unknown activity and determine at which point the unknown activity cannot be detected. These limitations are discussed in Section 3.5. The self-referenced data, along with the extracted noise level, is used for the classification. This classification can lead to both false negatives and positives, which are defined in the class-confusion matrix given in Table 1.

3.2.1.1 Analytical Detection Method (ADM)

In addition to the average noise power, we can also determine the the standard deviation, σ_n, of noise power per bin using Equation 3.7.

$$\sigma_n = \sqrt{\frac{8}{N_{FFT}} \sum_{f=\frac{3}{8}N_{FFT}}^{\frac{4}{8}N_{FFT}} (S_{EM}(f) - \mu_n)^2} \tag{3.7}$$

To minimize false positives due to noise, we set a threshold of $3\sigma_n$ to process the self-referenced EM frequency signature and determine the total number of frequency bins with power exceeding this threshold level as shown in Figure 10. This is the single numeric metric we will use for automated detection.

The selection of 3σ is based on statistical norms; 3σ covers 99.7% of samples when samples exhibit Gaussian distribution. Since we are mostly concerned with noise, and noise does exhibit Gaussian distribution, we chose 3σ for the threshold. However, even with a Gaussian distribution, 0.3% of samples will naturally be outside of this confidence threshold. We account for this statistical variation by allowing 0.5% of the observed frequency bins to exceed the set threshold (0.5% stems from 0.3% with an

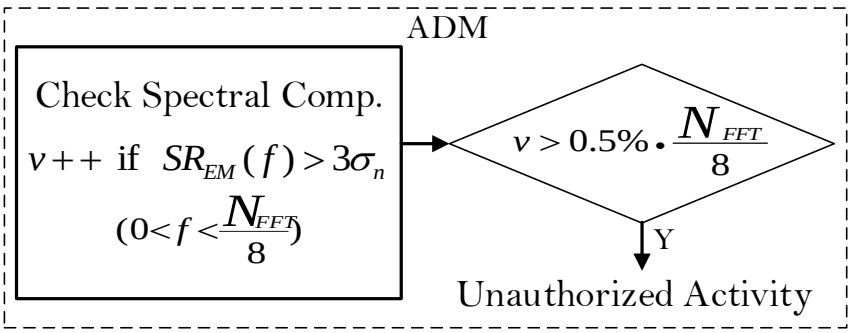


Figure 10: Flow cart of Analytical Detection Method.

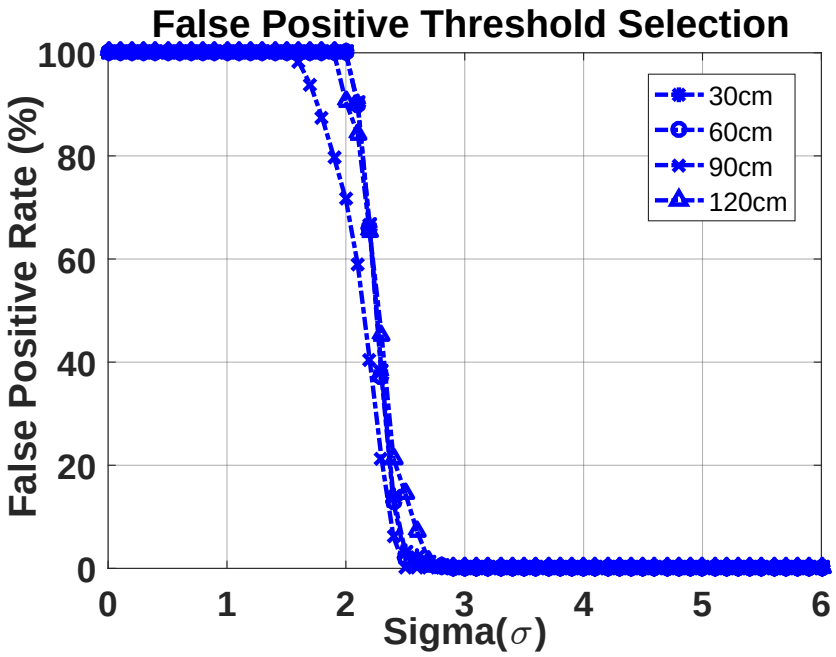


Figure 11: False positive rate is analyzed through 6-σ.

additional margin of 0.2%) without reporting suspicious activity. To explore further, we conduct 500 tests, each ranging from 30 cm to 120 cm. The spectrum thresholds are set by using the standard deviation ranging from 0 to 6σ. As expected, the false positive rate reduces sharply around 3σ, as seen in Figure 11. Since increasing the threshold also hampers the detection capability, this experiment confirms that the choice of 3σ is justified.

Since the additional error is always possible due to unexpected frequency spurs, we accept up to 0.5% violations in the self-referenced spectrum. If the violation count is higher than this limit, we mark the DUT with unauthorized activity. This classification algorithm that we call ADM is based on analytical analysis of expected noise behavior without knowing the precise power levels of the application signature.

3.2.1.2 One-Class Support Vector Machine (SVM)

We emphasize that the proposed ADM approach *does not* require any trusted sample. To compare ADM to a solid baseline, we also implemented a one-class support vector machine (SVM) classifier (Tax 2013), which has shown great potential in the area of anomaly detection (Maglaras and Jiang 2014).

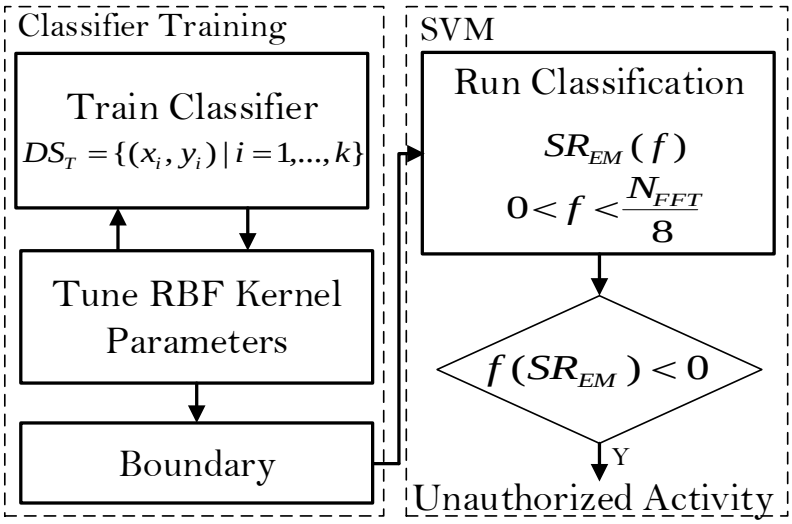
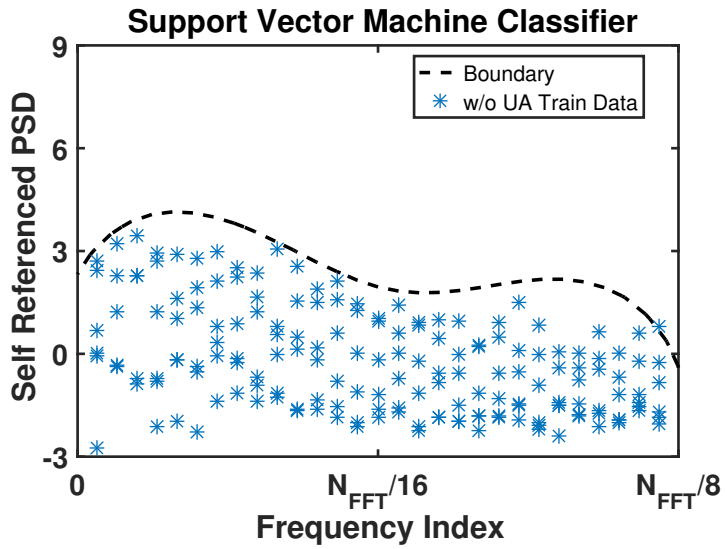


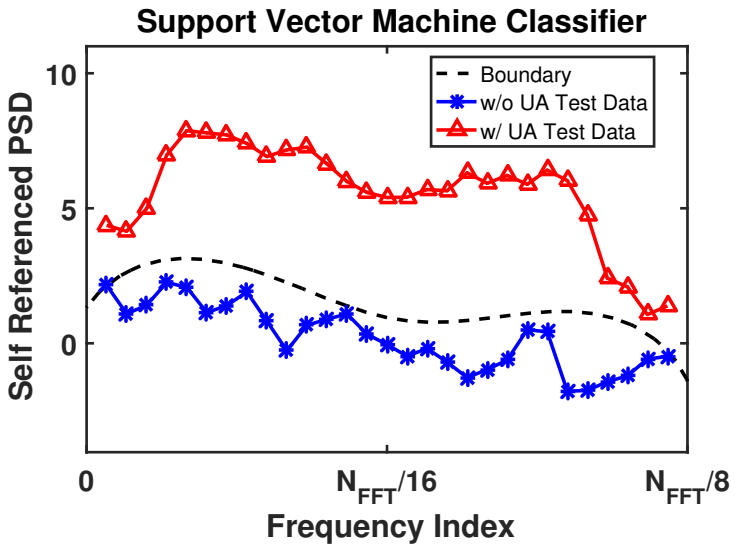
Figure 12: Flow cart of One-class Support Vector Machine.

In the SVM classifier, we used only the data sets without unauthorized activity for the training. Hence, this classifier would need golden samples. We use a Gaussian Radial Basis Function (RBF) kernel. The output of one-class SVM is binary, i.e., positive and negative. One of the important steps in the training phase is tuning a set of parameters that is used in one-class RBF kernel SVM as shown in Figure 12.

Such parameters have significant impacts on the accuracy of the developed models. For example, the γ parameter defines how far the influence of an individual training



(a) Training results using only inputs without unauthorized activity (i.e., no Trojan data is used during training).



(b) Test results with unknown Trojan activity.

Figure 13: SVM one-class classifier.

example reaches, and the ν parameter basically sets an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors. We tuned the rejection fraction ν in the range (0, 1) with step of 0.001 and the RBF kernel width γ

in the range (0, 50) with step of 0.5. It should be noted that the training data did not include any unauthorized activity. The trained classifier is shown Figure 13a. To investigate its best achievable performance, we trained one-class SVM using the entire training set, including multiple distances. The performance of the SVM classifier is shown in Figure 13b.

3.3 Experimental Evaluation

The measurement setup has a big impact on the effectiveness of the analysis and the data acquiring is one of the most important pieces of this setup. To evaluate the proposed approach, we construct the experimental setup shown in Figure 14. The experiments are performed in a standard lab setup without any specific attempt to generate a low-noise environment to generate a realistic scenario. On the contrary, there are numerous other devices and test equipment running in the lab during the set of experiments.

The experimental setup consists of the DUT, a low-frequency antenna system, and a digital oscilloscope. The loop antenna is used to provide higher gain at lower frequencies. As the DUT, we employ a mobile hardware platform (ODROID, n.d.). We position the antenna to measure the EM emissions from the DUT at distances ranging from 30 cm to 200 cm. After acquiring the EM signal by the loop antenna, the signal is transmitted to a receiver that consists of a current-mode amplifier and an audio amplifier with broadband loop equalization (LF Engineering Company, n.d.). The antenna receiver output is connected to the digital oscilloscope to capture and transfer the EM signal to the PC.

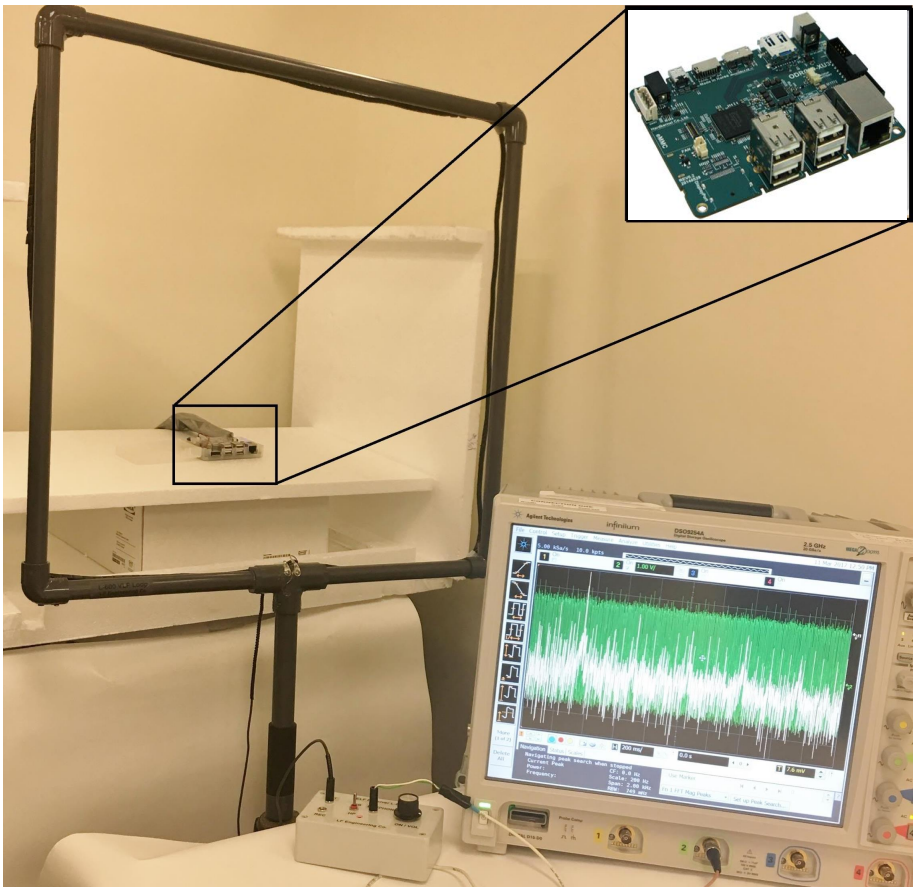


Figure 14: Experimental setup with the DUT, antenna with VLF receiver, and digital storage oscilloscope.

3.3.1 Periodic Test Application

One of the critical features of the proposed approach is to run a test activity periodically to concentrate the power spectrum on known harmonics. We achieve this using the procedure outlined in Algorithm 1. The test duration is given as the input to the periodic test application. After setting the required time flags, Lines 5-8 run the designated test application continuously with a period T_0. The period is set such that all activity of the test application completes within this duration. The periodic test application can be invoked at different frequencies by changing

the test application activation period, T_0. This period can also be set following a pseudo-random sequence at run-time, where the sequence is not known to anyone other than the system integrator. Hence, the detection algorithm can be run at arbitrary frequencies unknown to the attacker. Therefore, even if the attacker has the design, s/he does not necessarily know the test strategy. As a result, it is very unlikely that the test program repetition frequency will coincide with the attacker's repetition frequency. Moreover, it can be changed from one test period to another to avoid any incidental coincidence.

ALGORITHM 1: Periodic Test Application

```
1  procedure PERIODIC APP (TestDuration, T_0)
2      t_test_start ← t_current_time;
3      t_test_end ← t_current_time + TestDuration;
4      i ← 1;
5      while t_current_time < t_test_end do
6          run The Test App;
7          while t_current_time < t_test_start + i * T_0 do
8              sleep
9          end
10         i ← i + 1;
11     end
12     return;
13 end
```

In our implementation, we used a simple matrix multiplication routine, whose run-time ranges between 1 ms and 1.5 ms as the test application. We note that any other routine that can run in the firmware could also be employed. The repetition period needs to be larger than the duration of the application to avoid interference between different periods. On the one hand, a larger period is desirable to better separate the runs from each other. On the other hand, as the period increases, the frequency harmonics are compressed, and the number of occupied frequency bins

increases. Hence, we set the repetition period to the minimum duration where the application can safely enter and exit. Due to the application run overhead, we set the repetition period to 3 ms. This test will be performed periodically for each IoT device. The underlying trade-off in determining the period is the power consumption overhead of the test application against the Trojan detection probability. If the testing is done more frequently, there is a higher chance of having the Trojan activity during testing, but also the power consumption overhead would be higher. For instance, if the test is performed once daily, 10s-15s/1 day is the overhead. Many IoT platforms are active only for a very short time separated by long idle intervals during which the system can be placed low-power or sleep mode (Hrishikesh Jayakumar et al. 2014). Hence, the test can be scheduled during an idle period.

3.3.2 Unauthorized Activity

To model the unauthorized activity, we implement a malicious modification that reads the temperature sensor information and writes it to another memory location that is accessible by the attacker. The malicious code takes only 6 lines of code to implement. This unauthorized code may activate at random times and with a random activation probability. We experiment with activation probability ranging from 1% (i.e malicious code is active every 1 ms with a 1% probability) to 99% (i.e. the malicious code is active every 1 ms with a 99% probability.)

We first average the acquired EM signal data of many data sets, to make possible the detection of even small EM contribution from the infected device with unauthorized activity compared to the genuine device. Also, the noise level can be reduced. However, when we compare these measurements that are captured in the same testing

Table 2: Lowest energy of malicious activity that can be detected.

	30 cm	60 cm	90 cm	120 cm	150 cm
Unauthorized Activity Energy (mJ)	0.863	1.319	1.554	5.431	11.509
Foreground Application Energy (mJ)	11261.25				

environment, which is shown in Figure 5. We cannot find any noticeable characteristics between these measurements, which may result from the low activity level of the unauthorized activity. Thus we cannot tell the infected device from the genuine device by directly in time domain comparison.

Table 2 shows the analysis of detectable Trojan energy at various distances and compares this to the energy of the foreground application. Typical applications on the same platform consume around 1 to 2 W within 1 second this corresponds to approximately 2 J/sec (Gupta, Patil, et al. 2017). Note that, unlike hardware Trojans, firmware Trojans cannot run on extremely low power as they employ the entire system even if it is for a few lines of code. Also, note that the foreground application energy is more than 1000 times higher compared to the detectable Trojan energy. The proposed technique detects EM activity due to the power consumption fluctuation during the test time. If there is no switching activity due to the Trojan, there will be no spectral signature. Under this circumstance, the Trojan cannot be detected.

3.3.3 Detection Capability as a Function of Distance

To evaluate the detection capability of the proposed techniques as a function of distance, we vary the distance between the DUT and antenna from 30 cm to 200 cm, while keeping the unauthorized activation probability at 50%. The number of captured

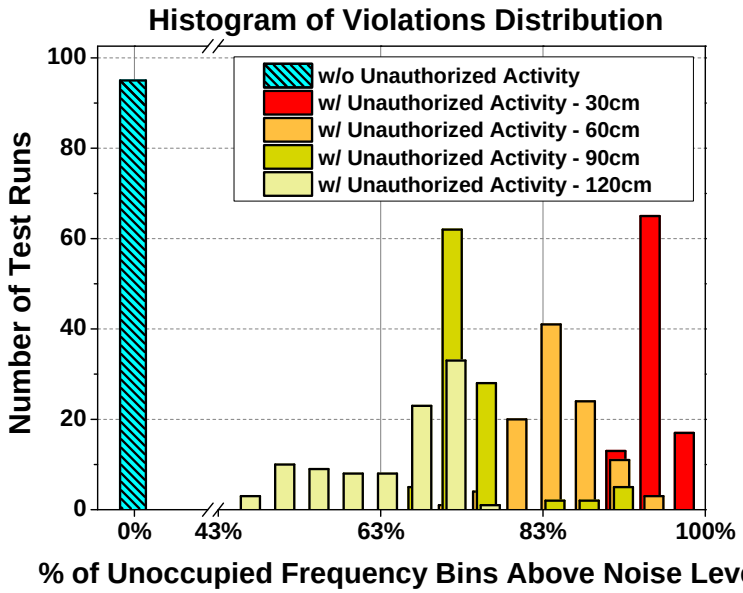


Figure 15: Histogram of spectral violations for 30 cm, 60 cm, 90 cm, and 120 cm distance between device under test and antenna with and without unauthorized activity with ADM classifier.

data sets is primarily limited by the memory depth of the scope. This allowed us to collect 95 different data sets with and 95 data sets without unauthorized activity for a given distance.

Figure 15 shows the percentage of the self-referenced frequency spectrum bins that are determined to be above the noise level using the detection algorithm outlined in Section 3.2. Figure 15 clearly shows that the proposed approach is very effective in detecting unauthorized activity for up to 120 cm distance. In particular, 100% of the data sets without unauthorized activity are identified correctly. Furthermore, all of the data sets with unauthorized activity present with a large number of frequency bins above the noise level. The percentage of the unoccupied bins for all 95 data sets is much higher than the 0.5% threshold to flag unauthorized activity. Hence, it is

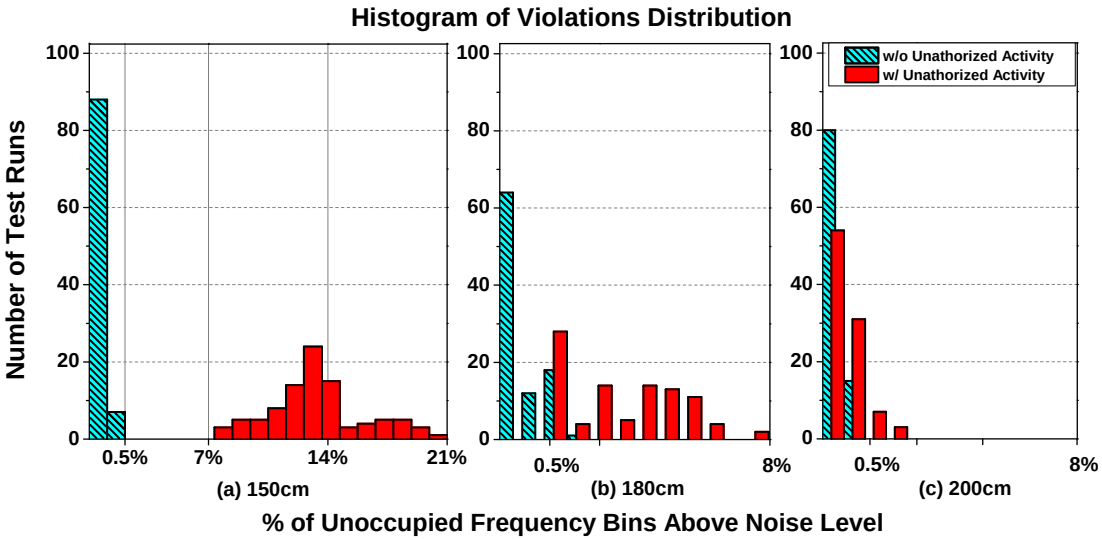


Figure 16: Histogram of spectral violations for (a) 150 cm, (b) 180 cm and (c) 200 cm distance between device under test and antenna with and without unauthorized activity.

possible to detect unauthorized activities with 100% accuracy up to 120 cm, when the unauthorized activity is 50%.

When we repeat the same experiment at a 150 cm distance, we observe that the gap between the data sets with and without unauthorized activity starts reducing, as shown in Figure 16 (a). In particular, the highest violation percentage of the frequency bins is around 21% and the lowest is around 7% for the unauthorized activity. We can still differentiate the data sets with unauthorized activities with 100% accuracy. Increasing the distance further hampers detection capability, as seen in Figure 16 (b) and (c). At 180 cm, we clearly see false positives and negatives, and at 200 cm, there is no difference in the spectrum with or without unauthorized activity. It should be noted that there are no false positives at any distance up to 150 cm.

3.3.4 The Role of the Signal-to-Noise Ratio in Detection Precision and Accuracy

Signal-to-noise ratio (SNR) is a measure used to compare the level of the desired signal to the background noise level. The signal-to-noise ratio often limits the accuracy with which some measurements can be done. For our measured EM signals, it can limit the reliability of detecting correctly. To this end, the proposed detection technique is analyzed how the detection rate is changed with environment noise vary the distance between the DUT and antenna. The main challenge to finding the relation of signal-to-noise ratio with the detection accuracy is the calculation of the initial signal-to-noise ratio of the data since the measured data includes signal and noise of the environment. In this way, the signal power is calculated with the taking first harmonic of the foreground signal, and the noise is calculated with the using Equation 3.6 and Equation 3.7. Once signal and background noise power are found, SNR can be calculated with the Equation 3.8.

$$SNR = \frac{P_{Signal}}{P_{Noise}} \tag{3.8}$$

We can find the signal root mean square (RMS) voltage with Equation 3.9a. The acquiring only signal power is not feasible for our measured EM data, we can

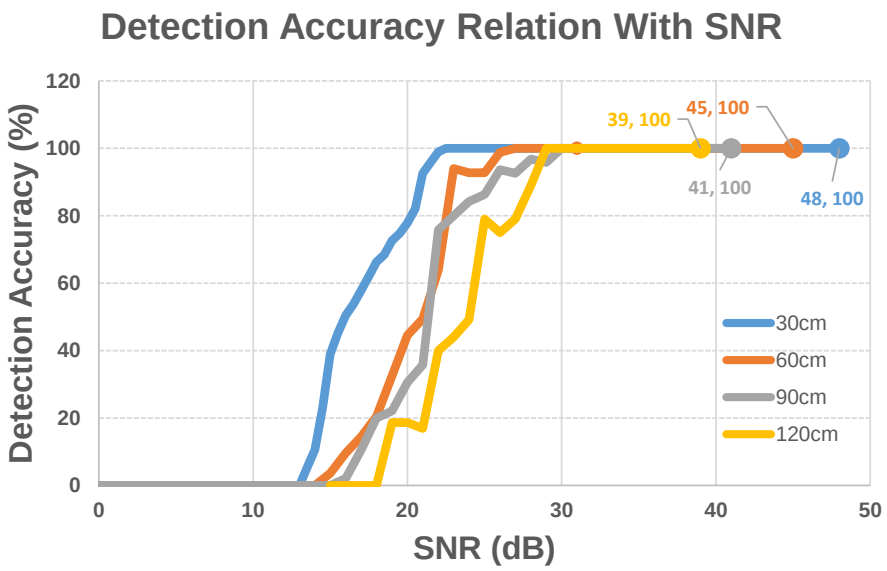


Figure 17: Detection accuracy relation with SNR from 30 cm to 120 cm distance between device and antenna.

assume the collected data as only signal power to calculate background noise standard deviation with Equation 3.9b to sweep SNR value for finding the accuracy relation.

$$V_{Signal\_RMS} = \sqrt{\frac{1}{N_{FFT}} \sum_{t=0}^{N_{FFT}-1} |\, V(t)\,|^2} \tag{3.9a}$$

$$V_{noise} = \frac{V_{Signal\_RMS}}{10^{\frac{SNR}{20}}} \tag{3.9b}$$

To sweep the signal to noise ratio, we add calculated standard deviation with zero mean noise to our measured data. Figure 17 shows the analysis results for measurement distances from 30 cm to 120 cm. The accuracy of the detection starts to decrease around 30 dB and reaches the not detectable area around 13 dB. The results give us an understanding of how SNR affects the detection rate of malicious activity with EM side-channel data. Due to using the measured data that includes noise can create some imperfection on the analysis, we have to aware of this assumption. In

conclusion, the detection and classification of measured EM signals can be solved, maximizing the signal-to-noise ratio (SNR), as seen Figure 17.

3.3.5 Detection Accuracy as a Function of Activation Probability

Next, we analyze the effectiveness of the proposed approach when the activation probability of the unauthorized activity varies. We collect and process 6365 data sets. Figure 18 shows the box plot of percentage of bins above noise level as a function of activation probability from 30 cm to 150 cm distance. We observe that with as little as 3% activation probability, the unauthorized activity can be detected with almost 100% accuracy at 30 cm. As expected, with increasing distance, the unauthorized activity is not detectable at the lower activation probability levels from 1% to 5%.

Table 3 provides detailed results on the false negatives for the proposed detection method, at various activation probabilities and detection distances. We observe that the proposed method achieves more than 90% detection rate for unauthorized activity at or above 5% activation probability up to the 90 cm distance. The proposed approach is still effective up to 120 cm distance for unauthorized activity with higher than 10% activation probability.

To compare our analytical classification approach, we also used a one-class support vector machine to classify EM signatures as explained in Section 3.2.1.2. Note that to train the SVM, a golden signature is required. In other words, we need EM signatures of the DUT that is known to be free of unauthorized activity.

Table 4 details the false negative rate using the same EM signatures, but with an

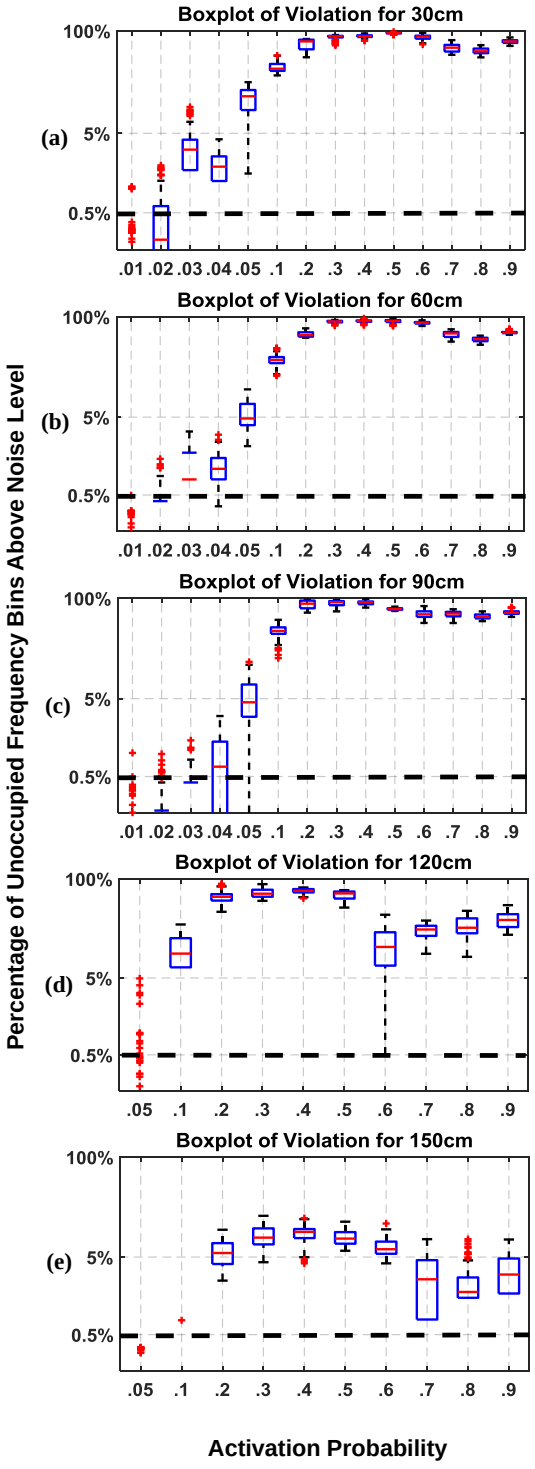


Figure 18: Analysis of the relationship between activation probability and spectral violations with 95 data sets at each of activation probability level.

Table 3: False negative of analytical detection method

FN (ADM)		Distance				
		30 cm	**60 cm**	**90 cm**	**120 cm**	**150 cm**
Activation Probability	**0.01**	94.74%	100.00%	98.95%	100.00%	100.00%
	0.02	67.37%	76.84%	89.47%	100.00%	100.00%
	0.03	2.11%	44.21%	84.21%	100.00%	100.00%
	0.04	1.05%	3.16%	42.11%	100.00%	100.00%
	0.05	0.00%	0.00%	9.47%	86.32%	100.00%
	0.10	0.00%	0.00%	0.00%	4.21%	98.95%
	0.20	0.00%	0.00%	0.00%	0.00%	0.00%
	0.30	0.00%	0.00%	0.00%	0.00%	0.00%
	0.40	0.00%	0.00%	0.00%	0.00%	0.00%
	0.50	0.00%	0.00%	0.00%	0.00%	0.00%
	0.60	0.00%	0.00%	0.00%	0.00%	0.00%
	0.70	0.00%	0.00%	0.00%	0.00%	2.11%
	0.80	0.00%	0.00%	0.00%	0.00%	3.16%
	0.90	0.00%	0.00%	0.00%	0.00%	4.21%

Table 4: False negative of Support Vector Machine One-class classifier

FN (SVM)		Distance				
		30 cm	**60 cm**	**90 cm**	**120 cm**	**150 cm**
Activation Probability	**0.01**	29.47%	46.32%	67.37%	100.00%	100.00%
	0.02	0.00%	9.47%	23.16%	100.00%	100.00%
	0.03	0.00%	8.42%	14.74%	100.00%	100.00%
	0.04	0.00%	0.00%	1.05%	100.00%	100.00%
	0.05	0.00%	0.00%	0.00%	12.63%	100.00%
	0.10	0.00%	0.00%	0.00%	0.00%	81.05%
	0.20	0.00%	0.00%	0.00%	0.00%	0.00%
	0.30	0.00%	0.00%	0.00%	0.00%	0.00%
	0.40	0.00%	0.00%	0.00%	0.00%	0.00%
	0.50	0.00%	0.00%	0.00%	0.00%	0.00%
	0.60	0.00%	0.00%	0.00%	0.00%	0.00%
	0.70	0.00%	0.00%	0.00%	0.00%	0.00%
	0.80	0.00%	0.00%	0.00%	0.00%	0.00%
	0.90	0.00%	0.00%	0.00%	0.00%	0.00%

Table 5: False positive and false negative rates at higher distances (180 cm and 200 cm).

$\alpha = 0.5$	**180 cm**		**200 cm**	
	FP	**FN**	**FP**	**FN**
ADM	6.32%	0.00%	1.05%	49.47%
SVM	11.58%	0.00%	3.16%	42.11%

SVM classifier. The SVM results are slightly better at higher distances, but the SVM classifier is also limited to maximum 120 cm distance and minimum 5% activation probability. This is due to the inherent lack of information at a higher distance. This comparison shows that despite not having any golden signatures, and with only limited knowledge of the test application, the proposed ADM classifier performs as well as the SVM classifier that relies on the golden signature.

Table 5 details the results for higher measurements distances. As mentioned earlier, the detection distance of the proposed approach is limited to 120 cm. The SVM classifier is also limited to the same distance. This limitation is primarily due to signal power falling below measurement sensitivity level. Finally, Table 6 compares the false positive rates of the two classifiers, the ADM classifier, and the SVM classifier. Again, the ADM classifier performs as well as the SVM classifier within the 120 cm distance.

Table 6: Comparison of false positive rates for the two classifiers.

FP	**30 cm**	**60 cm**	**90 cm**	**120 cm**	**150 cm**
ADM	0.00%	0.00%	0.00%	0.00%	0.00%
SVM	0.00%	0.00%	0.00%	0.00%	2.11%

3.4 Usage Scenario of the Proposed Detection Technique

Unquestionably many industries can benefit from IoT technology. That is why more and more businesses are beginning to embrace smart technology applications, such as home automation, supply chain management, transportation, etc. On the one hand, the enabling IoT systems are adaptive to meet the customers' ongoing changing needs, and the installation of smart products provides convenience and savings of time, money, and energy. On the other hand, the growing presence and demand of devices enables new attack methods and attack surfaces for attackers to exploit, posing serious security and privacy issues. For instance, security researchers showed how they could hack a smart home IoT device and used it as a back door to obtain network password (Chapman 2014). It is very evident that the fact these devices are poses serious threats, which could have significant real-world consequences. This is one of the many challenges in providing IoT device security.

There are many new security technologies and solutions currently being developed that can help to address IoT device security problems. Those solutions are like isolating the IoT device from the network, reverse engineering the physical device, etc. They may also need physical access to run their detection algorithm. Without losing any benefits of the IoT systems, our proposed malicious activity detection technique can relieve the security challenges of the lightweight IoT devices.

IoT devices are mostly installed at a fixed (permanent) location, and malicious modifications may become active only the connected network and location. The detection technique needs to run for any possible malicious change without removing the IoT device at the environment to avoid losing the status of the questioned device.

The devices are installed in a close proximity environment due to their network

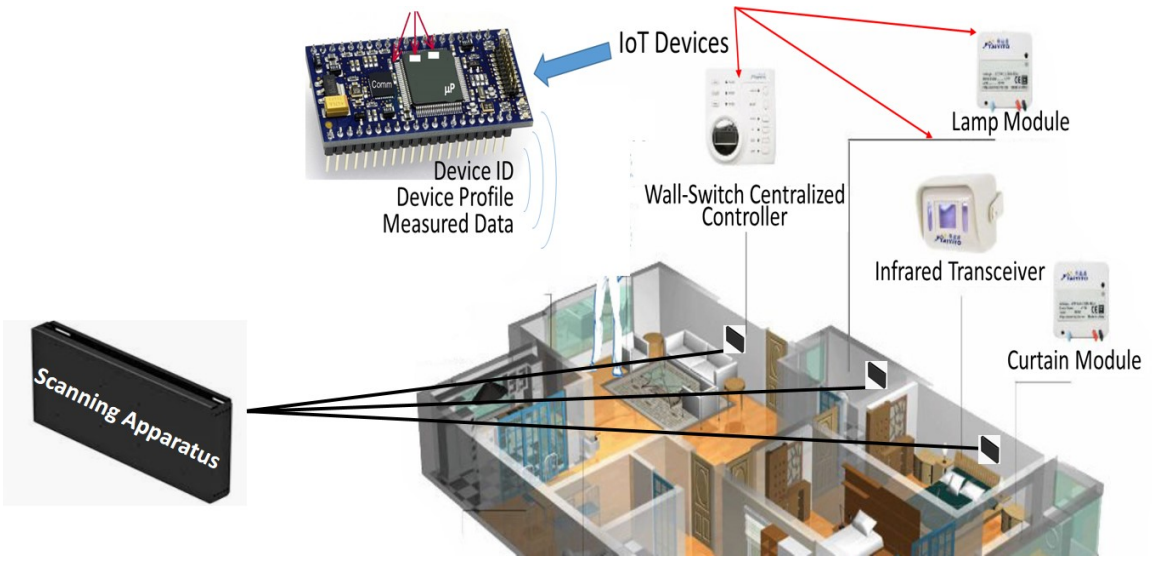


Figure 19: Stand-off detection of unauthorized activity: Home Automation IoT.

protocol needs or device operating range. We install the proposed detection setup (scanning apparatus) within a 1.5m radius to detect any abnormality of the encircling IoT devices, as seen in Figure 19. The detection apparatus can enable the proposed stand-off detection technique. And also, this decoupled device works with the gateway to enable testing for surrounding IoT devices.

3.5 Limitations

The proposed technique relies on EM emissions by authorized and unauthorized activities. As such, these EM emissions need to be above the measurement sensitivity. This requirement sets a maximum distance requirement for detection. In our experiments, we have found that the proposed detection works reasonably well up to the 120 cm distance. Detection capability is also limited by the overall energy consumed by the unauthorized activity. This energy is spread within an observation bandwidth.

Depending on the rate of operation, this energy, spread over a broad spectrum, may fall below the measurement sensitivity. We have also found experimentally that activation probabilities below 3% are rarely detectable.

We have made no assumptions on the functionality of the unauthorized activity or the test application. However, as a limitation, we should note that if a Trojan that is not activated within the test duration will not generate a spectral signature and cannot be detected via the proposed method. Besides, a Trojan that is active in a short burst can also escape detection since its energy will fall below the detectable level. This scenario is analogous to the activation probability limitation.

Chapter 4

DETECTING MALICIOUS ACTIVITY IN LIGHTWEIGHT WEARABLE AND IOT DEVICES

4.1 Introduction

The Internet of Things (IoT) refers to the ever-growing network of computing devices that have vastly different processing capabilities ranging from simple IoT end nodes, such as sensors, to high-end computing systems. For example, smart objects are already in use for supply chain management, smart city management, health care, automation of various home chores, and smart grid. While IoT offers a promising future for automation, convenience, and machine-to-machine interaction without human intervention, thereby improving the quality of life, it poses new security challenges, particularly due to its limitations on hardware, compute resource, and power.

Systems-on-chip (SoC) design based on reusable IP is now a pervasive practice in the semiconductor industry due to the dramatic reduction in design/verification cost and time it offers. Cost and development time are two major problems faced by the designers of low-volume SoCs. Hence, they utilize third-party IP cores from dozens of different vendors to amortize the cost and shrink design turn-around time. Moreover, the boards and firmware for these devices can also be developed and installed by third-party vendors.

With a global supply chain consisting of many players, ensuring the security of

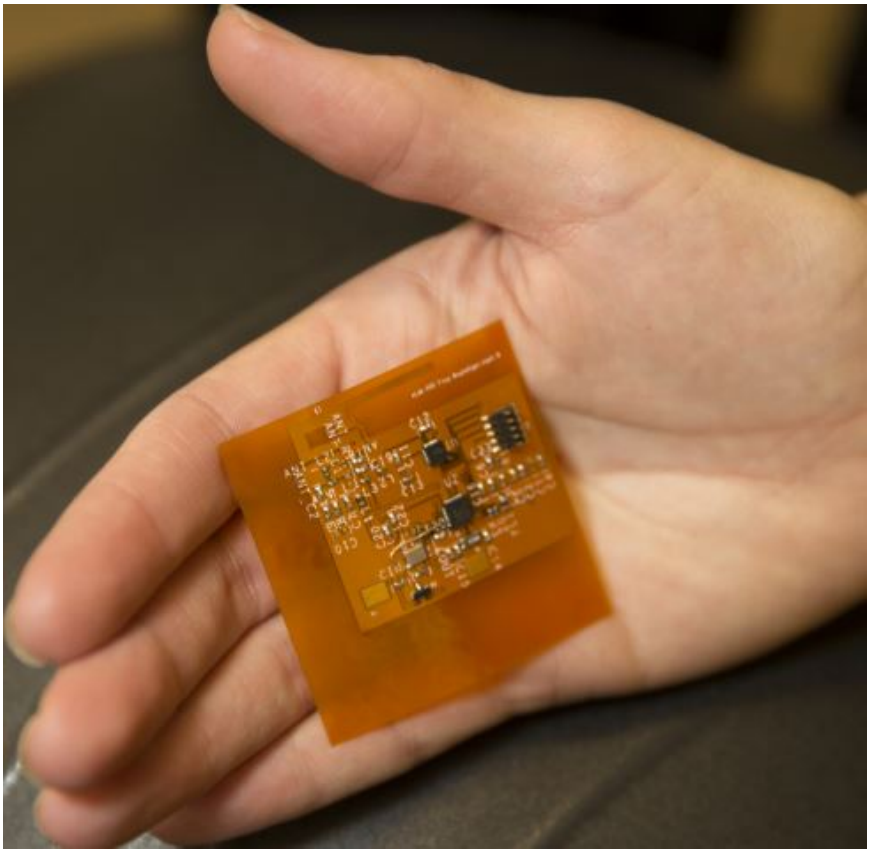


Figure 20: Wearable Electronics Prototype.

wearable and IoT devices is a daunting challenge. It is becoming increasingly necessary to *trust but verify* the received devices both at production time and in the field.

Battery-driven embedded systems and small form-factor devices such as wearable devices are resource-constrained embedded devices, which have limited computing power and battery capacity. These limitations make existing cybersecurity mechanisms such as anti-virus software and anomaly detectors (Shila, Geng, and Lovett 2016) too costly to implement. Thus, ensuring the security of wearable devices with acceptable overhead is a new challenge that requires cost-conscious solutions. The security of embedded devices becomes even more critical because the increased inter-connectivity provides more space for attackers to introduce the Trojans (Hamdioui et al. 2014).

Attacks on wearable devices can be realized in the form of malicious hardware or firmware modifications, also known as Trojans. The wide range of attack space has resulted in exponentially increasing security problems. To evade detection, Trojans inflict damage through small, well-hidden circuitry or firmware modifications, which activate either randomly, or after specific trigger incidents (Rostami et al. 2013). Therefore, the symptoms of the Trojan are not always observable. Moreover, when a Trojan is active, its impact on measurable factors such as system performance or power consumption is typically negligible due to the subtle modification it makes. Due to a wide range of trigger mechanisms that activate the Trojan and payload that distorts the system under attack, detection of hardware Trojans is a daunting task. This difficulty is multiplied due to the diversity of the IP cores on the chip. Hence, ensuring the integrity of each resource is becoming increasingly difficult. Finally, the behavior of a compromised chip cannot be compared against a trusted sample, since finding a golden reference may not always be feasible.

This chapter presents a technique to detect malicious activity in lightweight wearable and IoT devices. The proposed detection technique does not rely on trusted samples; rather, it establishes a baseline for each individual device based on its periodic steady-state (PSS) behavior. By referencing the detection threshold to baseline characteristics of each individual device, environment, and process variations can be removed.

In order to collect data without affecting the operation of the device, the proposed technique uses a signal stitching technique in which side-channel information of repetitious code sequences is sampled and processed to monitor device activity.

The proposed technique is demonstrated on the wearable device prototype shown in Figure 20, which runs gesture recognition software, including an arbitrary repetitive

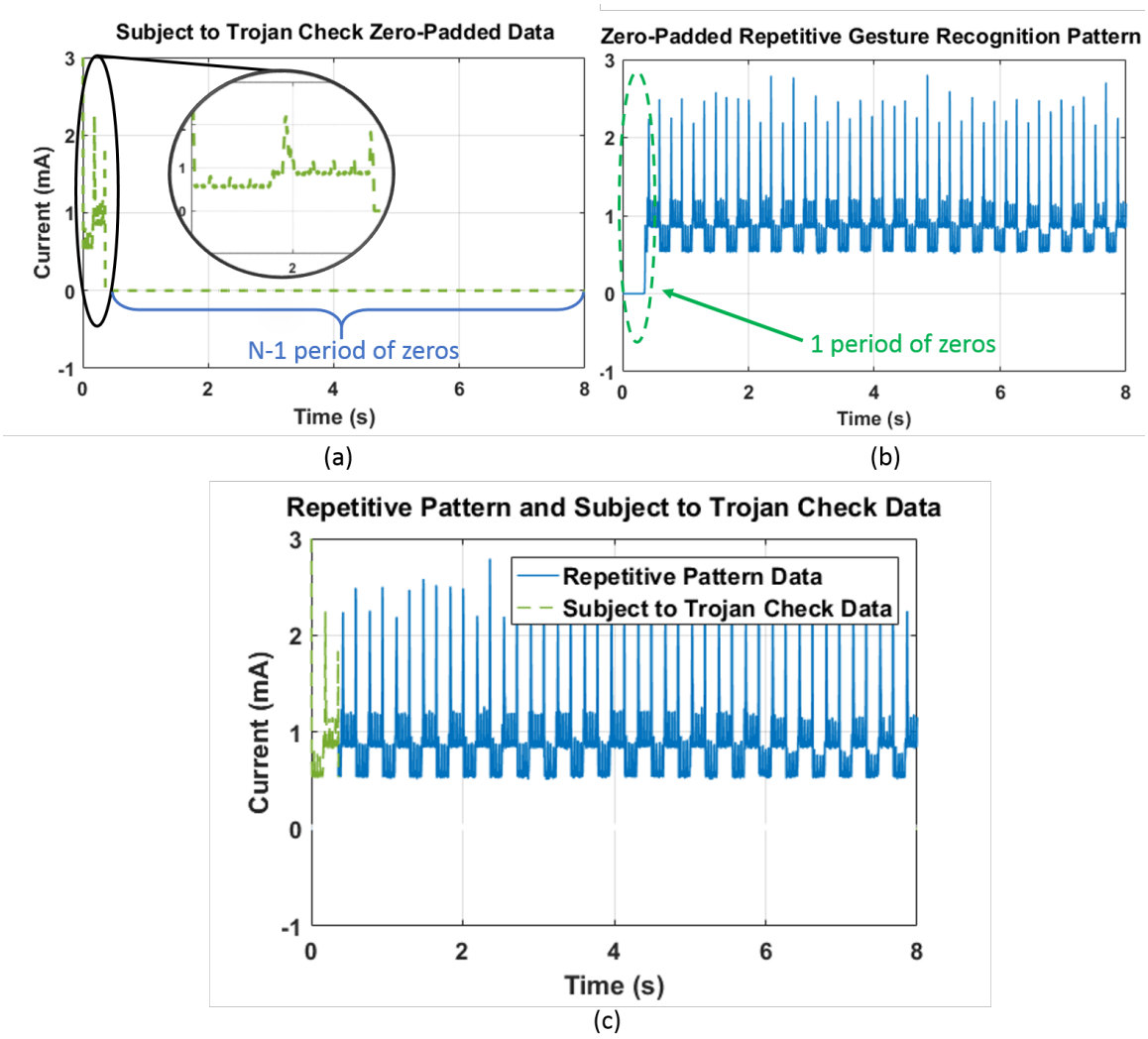


Figure 21: (a) Zero padded data that is subjected to Trojan check. (b) Zero-padded repetitive gesture recognition pattern with one period of zero padding at the starting. (c) Stitched one period of data to repetitive gesture recognition data.

gesture recognition algorithm. By collecting data on repetitive patterns, side-channel signals, such as power consumption, of the device are driven into a periodic steady-state (PSS). By collecting data on this repetitious pattern and stitching it with existing data, the overall operation pattern of the device can be established.

As an example, Figure 21a shows the run-time data collected from the device in one instance with zero padding. This data is stitched to pre-existing data (collected

earlier) shown in Figure 21b, to obtain the complete data sequence, shown in Figure 21c. The complete data can be analyzed in the frequency domain to establish criteria for flagging suspicious activity.

The collected side-channel data can be expressed as the sum of the primary system response (i.e., the power consumption without a Trojan), the Trojan activity, and environment and measurement noise. Putting the device into a PSS concentrates the primary signal power at a known frequency and its harmonics. This leaves a large portion of the signal spectrum unoccupied and available for detection. The Trojan activity, if there is any, would occupy a wider band since it is unlikely to be correlated with primary activity. Thus, the unoccupied bins of the spectrum can be analyzed to determine whether there is unauthorized activity.

The major contributions of this chapter are as follows:

- We present a methodology for time-domain signal switching to collect side-channel signal information on repetitive primary activity to reduce test duration.
- We present a methodology for limited-bin spectral analysis for the detection of unauthorized activity to reduce the computational burden of the detection technique.
- We present a self-referenced malicious activity detection technique applicable to not only sinusoidal excitation, but also to repetitive patterns to remove the effects of process and environmental variations.
- We evaluate the proposed approach while running gesture recognition and Wi-Fi applications without requiring a trusted sample.
- We perform extensive experiments using a wearable electronics prototype (Gupta, Park, et al. 2017) and a commercial multiprocessor system-on-chip (Mp-

SoC) (ODROID, n.d.), and demonstrate the effectiveness of the proposed detection technique.

The rest of the chapter is organized as follows. The threat model is explained in Section 4.2. The proposed malicious activity detection technique is described in Section 4.4. Finally, extensive experimental evaluation is presented in Section 4.5.

4.2 Threat Model

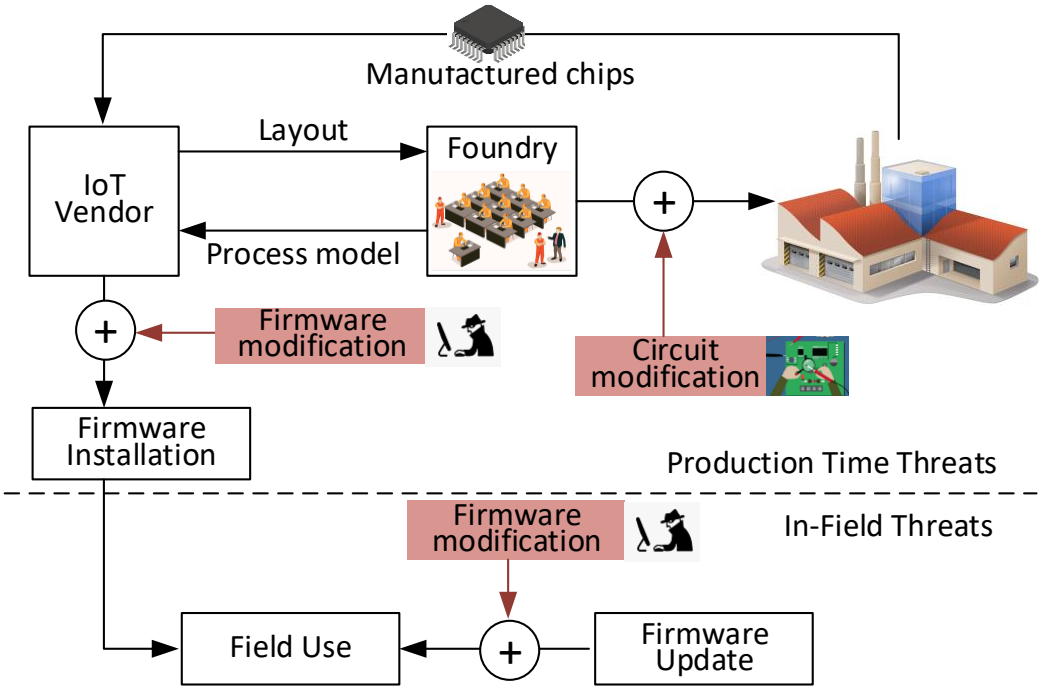


Figure 22: Firmware threats can be added during firmware installation at the IoT vendor or other third party company. The attacker can also make firmware changes during field updates.

Many wearable electronic and IoT devices run their own operating system and

applications. As these devices become more common, they will also become bigger targets for hackers. Trojans could be targeted at firmware to insert malicious code that gains privileged access to steal information or cause harm to the device. They could take the form of passive hardware entities which help malicious software bypass pre-existing hardware protection systems.

Figure 22 illustrates possible attack models for potential hardware and firmware Trojan threats. The IoT vendor receives the process model from the foundry and produces layout of the circuit based on this process model. The malicious modifications can be added at production time or while the device is in use in the field. The threats can either originate at the foundry or during production firmware installation. There can be circuit modifications and firmware modifications. Hardware threats originate at the foundry from malicious attackers (third-party consultants, rogue employees, etc.) who modify the hardware to insert a malicious circuit. Firmware threats can be added during firmware installation at the IoT vendor or other third party company. The attacker can also make firmware changes during field updates. Hence, all chips manufactured by an untrusted foundry are suspect, and every product in field needs periodic monitoring to verify recent changes made to the system.

4.3 Lightweight IoT Device Characteristics

The number of IoT devices is continually increasing for several different reasons such as low production cost, easily fabricated and available electronics in the market, improving the quality of life through technology, availability of wireless and wired communication networks. The term things in the Internet of Things is a piece of equipment having sensing, actuating, storage, or processing capability. These devices

Table 7: Example of Lightweight IoT Devices.

Device Name	MCU/CPU	Storage (ROM/FLASH)	Memory (RAM)	Power
Flexible Wearable IoT Device (Bhat et al. 2018)	CC2650 SimpleLink Multistandard Wireless MCU 32-bit ARM Cortex M3 processor runs at 48 MHz	128 Kbyte	20KB of Ultralow- Leakage SRAM	Battery Operated - The gesture recognition application consumes12.5 mW power for recognizing a single activity
IoT patient monitoring system MICAz Wireless Measurement System (Akkaş, Sokullu, and Çetin 2020)	ATmega128 8-bit Atmel microcontroller	128 Kbyte	4 Kbytes Internal SRAM	Battery Operated - 2AA Batteries
Senior LifeStyle System Qorvo-Sensara (Qorvo 2020)	XAP5 high performance 16/32-bit microcontroller Arm Cortex - M4 processor	512 Kbyte	64 Kbyte Low Leakage Retention RAM	Battery Operated
Vibration Alert Bracelet for Notification of the Visually and Hearing Impaired (Conley et al. 2019)	ESP8266EX - Tensilica L106 32-bit RISC processor	512 Kbyte	<50KB	Battery Operated
IoT patient activity tracker TEMPO - Care Predict (CarePredict 2017)	nRF52832 32-bit ARM Cortex M4F processor	512 Kbyte	64 KB RAM	Battery Operated

possess unique characteristics such as limited memory, battery capacity, and processing power. They don't have the memory in gigabytes, nor processing power in terms of cores. Most devices have an MCU rather than a CPU, and that MCU is running at speeds measured in megahertz, not gigahertz. Lightweight IoT devices have a fraction of the computational power of a server, desktop, or even a smartphone.

The energy budget for IoT nodes usually consists of small battery sources or energy harvesters. Thus battery life is frequently one of the most significant gating factors in designing an IoT device. The limited power of an IoT device, and the need to preserve that power impact MCU choice, RAM size, and network type. Storage is on an embedded ROM chip, with ROM storage typically in the range of 96kB to 256kB, to perhaps 512kB. Table 7 shows an example of the hardware specification of lightweight IoT devices for senior care ambient assisted living. We can itemize the challenges that make it challenging to secure IoT devices.

- Limited Computational Power
- Limited Power
- Limited Memory (RAM) and Storage (ROM/Flash)
- Limited Network Speed and Bandwidth
- Limited Security Tools

The lightweight MCU/CPU limited-resource devices that make IoT devices so versatile also limit the use of conventional security solutions. This lack of robust security solutions is especially troubling because, by their very nature, IoT devices are generally far removed from the traditional firewalls and layered security of the enterprise. IoT Security has always been a challenge. But the problem is even more significant when working within the resource constraints inherent to IoT devices (CENTRI IoTAS 2019).

4.4 Malicious Activity Detection

4.4.1 Run-time Testing and Signal Stitching Technique

It is highly desirable to detect hardware or firmware Trojans before chips are deployed in the field. Still, existing techniques cannot guarantee comprehensive coverage for all types and sizes of Trojans. If a Trojan attack is introduced after production, such as insertion during a firmware update, or was not detected at production time, in-field activity monitoring and run-time testing can significantly reduce its risk. In exchange for some performance overhead, these approaches can flag the device or disable it upon detecting malicious activity. However, the challenge is that these testing and monitoring activities, like measuring device current, power consumption, or memory usage, should not interfere with regular device operation.

Many IoT platforms are active only for very short intervals, between which the system is placed into a low-power or sleep mode (H. Jayakumar et al. 2014). Hence, the idle periods can readily be used for test data collection. The same set of applications that run during regular operation can be used for PSS generation in test mode during idle time. Therefore malicious modification(s) cannot evade detection when testing is performed.

The proposed detection approach uses a run-time signal stitching technique to collect data without affecting regular operation. The device is monitored, and data is collected during idle time. Although data could be collected during active periods, these resource-constrained wearable or IoT devices have limited processing capability (Cheng et al. 2015), so data collection while doing regular operation would reduce device efficiency and usability. As long as the device reaches a periodic steady-state, it does

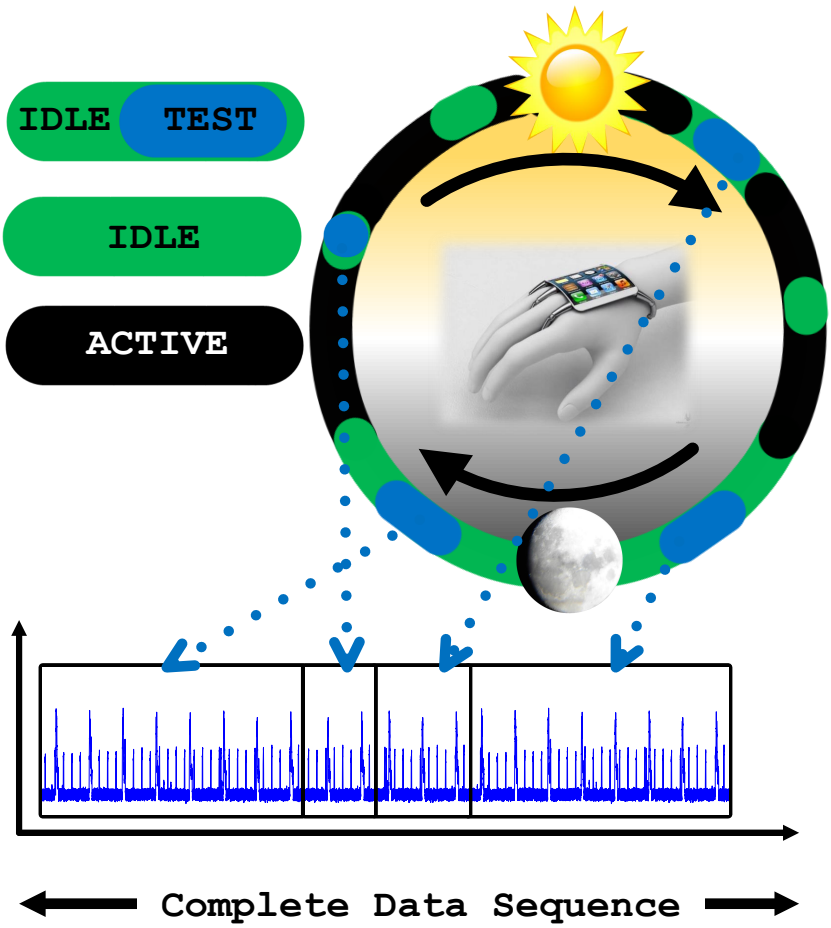


Figure 23: Illustration of device usage in a day and measuring current of device in idle stage for data stitching.

not matter when it is collected. Hence, we can select any suitable time-span for testing, and the best choice is the idle stage of the device. By collecting data during repetitive patterns and stitching it with existing data, the device's overall operational model can be established. After the entire data sequence is assembled, the proposed detection algorithm can be run to determine whether the collected data contains known or unknown authorized activity. The signal stitching technique also enables

the collection of data during different times of the day. Hence the technique is able to detect a Trojan that has a larger bandwidth than the length of a single measurement.

Memory in wearable devices, in particular, has always been a tricky balancing act. Unlike memory for PCs, tablets, and other devices, memory for wearables is smaller and less dense, but still must be adequate to support the functions of the device and its firmware. With their expansive growth in popularity, users are expecting more from wearable and IoT devices than ever before. And while developers are always looking for new and exciting features to add, those features cannot come at the expense of basic functionality. In other words, expanded or more accurate security monitoring must also come with improved processing and memory. So the memory is a requirement for the proposed malicious activity detection. However, performing data collection and analysis during idle times means there will free memory that is sufficient for running the detection algorithm.

As seen in Figure 23, the device is monitored, and during its idle time, part of the data is measured. This data is stitched with previously collected data samples to synthesize a complete data sequence. When the required data sequence is collected, frequency domain analysis can be run.

4.4.2 Optimized Fast Fourier Transform (FFT) Algorithm for Zero-Padded Data

IoT or wearable devices typically run on limited battery-operated power and therefore present unique challenges in terms of availability of computational resources. Thus, a new challenge to ensuring the security of wearable devices is power awareness (Leabman and Brewer 2018). Activity monitoring, measurement of complete

data sequences, and detection algorithm computation add significant overhead that requires further power-aware optimization.

First, the proposed detection method relies upon the identification of a repetitive pattern in the main application to effectively decouple the fundamental frequency and its harmonics from other unexpected activities. Thus, some monitoring time is required for the detection algorithm to function. However, we can minimize the monitoring time by reusing previously collected and analyzed data that has not been flagged for suspicious activity. Figure 24a shows a shorter period of data that has been collected and zero-padded to match the size of the complete data sequence for Fourier transform analysis. In Figure 24b, the collected data is zero-padded for a period of samples, and then its Fourier Transform is saved. The Fourier Transform of saved and zero-padded data can be summed as seen in Figure 24c to assemble a complete spectrum that can be analyzed for suspicious activity.

Second, the computation time and power consumption need to be minimized to support a seamless, energy-efficient security monitoring system. The Recursive Fast Fourier Transform (FFT) algorithm can be optimized by skipping zero-padded data in Algorithm 2, where Input represents the zero-padded period of measured data, NFFT represents the sample size of the test data, and Size represents the original number of sample points before zero-padding. For optimal efficiency, the number of samples in the stitched data set must be a power of two. The optimized FFT algorithm is 4.2 times faster than the standard recursive FFT algorithm for NFFT = 1024 and Size = 32 samples. We will discuss more the computation time of the FFT in experimental data in Section 4.5.

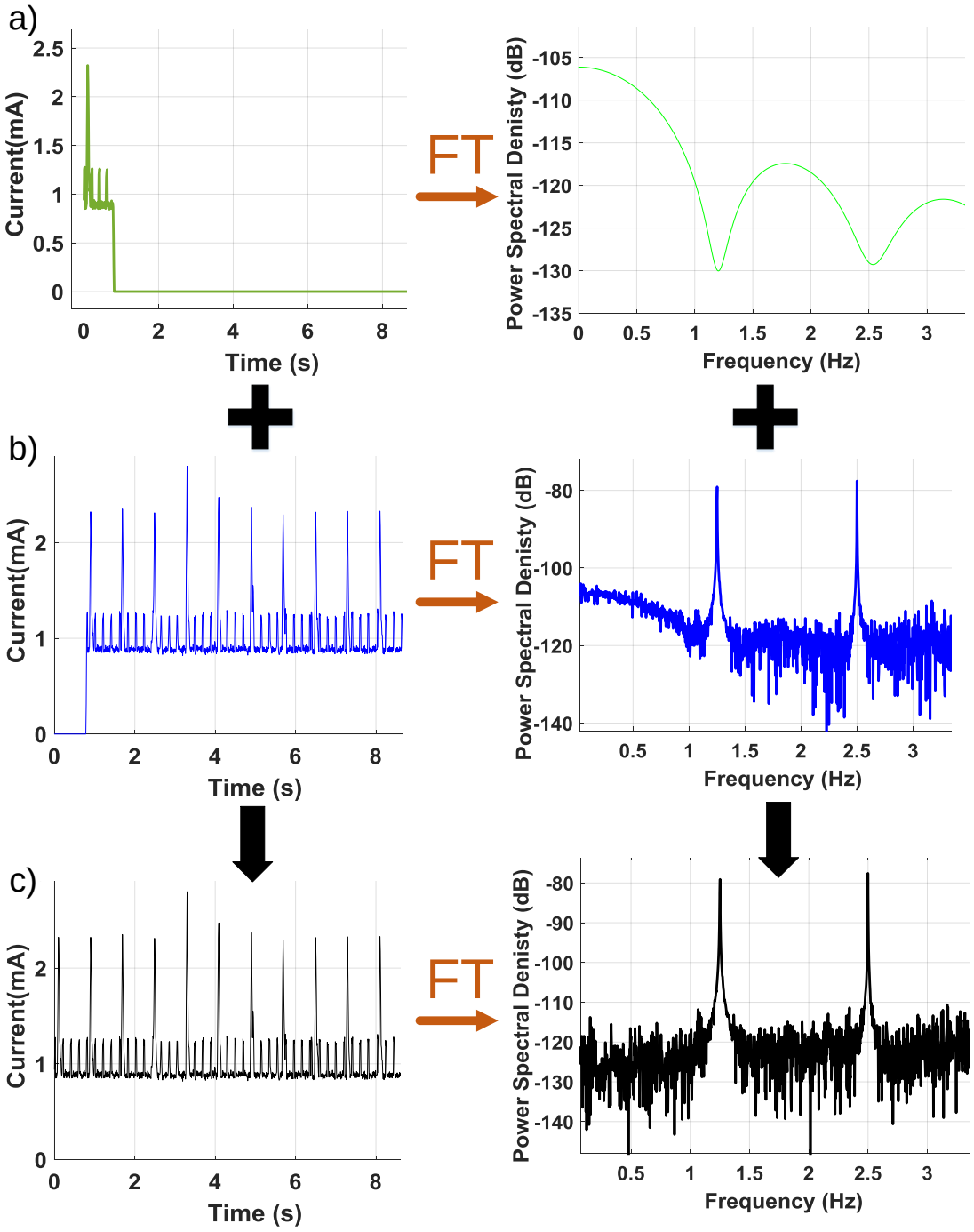


Figure 24: Illustration of superposition of Fourier Transform that is using saved and a period of data to determine suspicious activity.

ALGORITHM 2: Optimized Recursive Fast Fourier Transform for Zero-Padded Data

```
1  procedure FFT (Input, N, Size)
2      for k = 0; k < N/2; k = k + 1 do
3          FFT^0 = Input[k];
4          FFT^1 = Input[2k + 1];
5      end
6      y^0 ← FFT(FFT^0, N/2, Size);
7      y^1 ← FFT(FFT^1, N/2, Size);
8      if NFFT/Size ≤ N then
9          if NFFT/Size == N then
10             for k = 0; k < N; k = k + 1 do
11                 y_k ← FFT^0[0];
12             end
13         else
14             for k = 0; k < N/2; k = k + 1 do
15                 if not(y_k^0 == 0 AND y_k^1 == 0) then
16                     y_k ←  y_k^0 + w y_k^1;
17                     y_{k+N/2} ←  y_k^0 − w y_k^1;
18                 else
19                     y_k ←  0;
20                     y_{k+N/2} ←  0;
21                 end
22             end
23         end
24     return y;
25 end
```

4.4.3 Proposed Detection Technique Overview

The major problem in detecting unauthorized activity is obtaining a golden signature, which is extremely difficult, and in some cases, unfeasible. Moreover, Process-Voltage-Temperature (PVT) variations and environmental noise can mask the effect of the Trojan circuit on measured parameters (e.g, power (Narasimhan et al. 2011), even if multi-dimensional analysis is used (Hu et al. 2013; Cha and Gupta

2013)). Therefore, a decision on the presence (or absence) of a Trojan should ideally, be made without the need for a golden or trusted reference. To detect small deviations in power due to malicious activity, the primary circuit response needs to be decoupled from that of the Trojan and any environmental noise. We achieve this by generating a test sequence such that the *signal* (i.e., the response of the device) has spectral properties that can be differentiated from the Trojan activity. We can denote the measured power $P(t)$ as follows:

$$P(t) = P_0 \cdot sin(\omega_0 t) + \varepsilon(t) + n(t) + P_{Tr}(t) \tag{4.1}$$

where $P_0 \cdot sin(\omega_0 t)$ is the sinusoidal power consumption of the primary circuit, $\varepsilon(t)$ denotes the error that we make in setting up the sine wave, $n(t)$ is random noise, and $P_{Tr}(t)$ is the power consumption of the Trojan circuit (Karabacak, Ogras, and Ozev 2016). The power spectrum of the primary signal, i.e., $P_0 sin(w_0 t)$ is concentrated in one frequency location. The noise signal has a flat spectral signature. While the specific details of the Trojan activity are unknown, its switching speed is clearly limited by the system clock, and it is unlikely that the period of the Trojan activity will match with the primary signal periodicity. Therefore, the FFTs of Trojan and primary activity will inevitably occupy different frequency bins. As a result, the bandwidth of the Trojan signal will be limited, making it different from white noise as well as the primary signal. The spectrum of $P(t)$ is the sum of the spectra of its components. The primary signal, $P_0 \cdot sin(\omega_0 t)$ is concentrated in one frequency location. The error signal, $\varepsilon(t)$, can be modeled as a pulse train:

$$\varepsilon(t) = (\sum_{i=1}^{N_p} \varepsilon_i \cdot p(t - iT_s)) * (\sum_{k=1}^{N_s} \delta(t - kT_0)) \tag{4.2}$$

where N_p is the number of samples in the sine wave, T_s is the sampling period, ε_i is the magnitude of the approximation error, $p(t)$ is a pulse with duration T_s, N_s is the number of sine wave periods in the measurement duration, and T_0 is the period of the sine wave. Note that while the error of the sine wave approximation within a single period can be random, the error signal itself is also periodic with the same period as the original sine wave. Hence, the power of the error signal will be concentrated at the harmonics of f_0, as in Equation 4.3.

$$S_\epsilon(f) = \sum_{i=1}^{N_p} \frac{2\varepsilon_i}{N_s}\delta(f) + \sum_{n=1}^{N-1}(\sum_{i=1}^{N_p} \frac{2\varepsilon_i}{n\pi} sin(\frac{n\pi}{N_s}))\delta(f - nf_0) \tag{4.3}$$

$n(t)$ is the noise signal, hence, its power will be spread through the *entire spectrum*. Finally, $P_{Tr}(t)$ is the *unknown Trojan signal*. While the specific details of the Trojan activity is unknown, its switching speed is clearly limited by the system clock. We can also safely assume that it is uncorrelated to the primary signal. As a result, the bandwidth of the Trojan signal will be limited, making it different from white noise as well as the primary signal as seen in Figure 25.

Figure 25 shows that Trojan activity has a distinct signature. However, we cannot assume knowledge of this spectrum, since we do not know how the Trojan operates. In order to detect an unknown signature, we will focus on what is expected from the device and flag if there is any suspicious activity. This will enable us to determine whether the spectrum deviates significantly from its expected form.

The detection algorithm uses a run-time signal stitching technique that eliminates hardware variation, and the self-referencing procedure removes the effect of process and environmental change. It can detect any activity which consumes power. The detection method utilizes a periodic run of the device application(s) to create a steady-

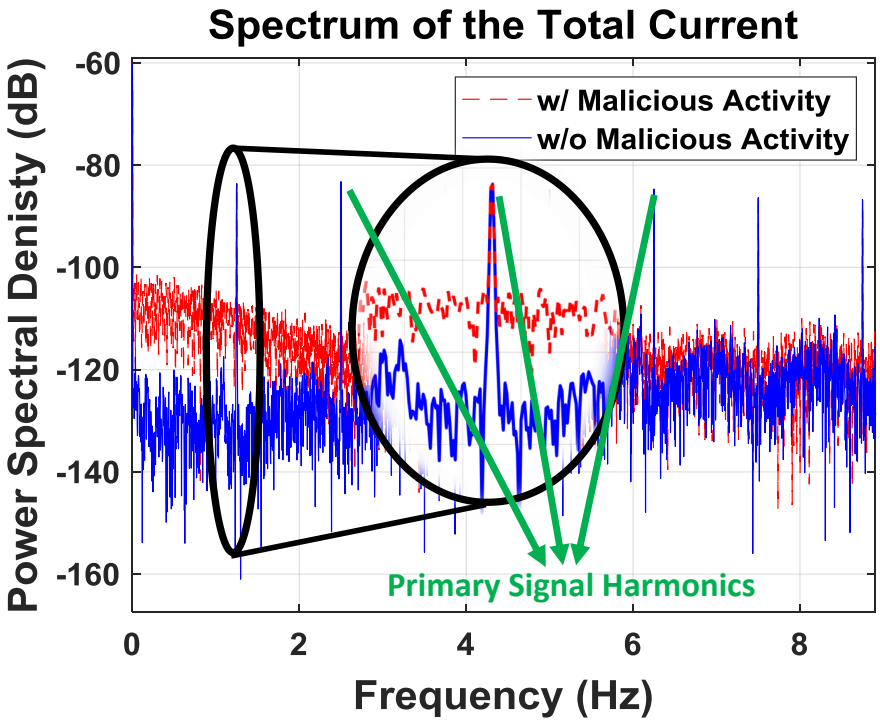


Figure 25: Spectrum of the Total Current.

state concentrating the known signal power at a known frequency. Moreover, using the native app (s) of the device for steady-state characterization allows for regular operation and triggering the Trojan conditions. In addition, coverage of triggering conditions can be increased by designing patterns that enable different hardware and firmware portions. In summary, if there is a Trojan that is activated while the application is running, the proposed algorithm can detect this abnormal behavior.

The proposed detection algorithm is shown in Figure 26. We sample and process the composite power signal $P(t)$ in the digital domain. We first pass the power signal through a low-pass filter, whose cut-off frequency is chosen to include harmonics of the primary signal that are above the noise floor. The resulting signal $P_{ref}(t)$ is

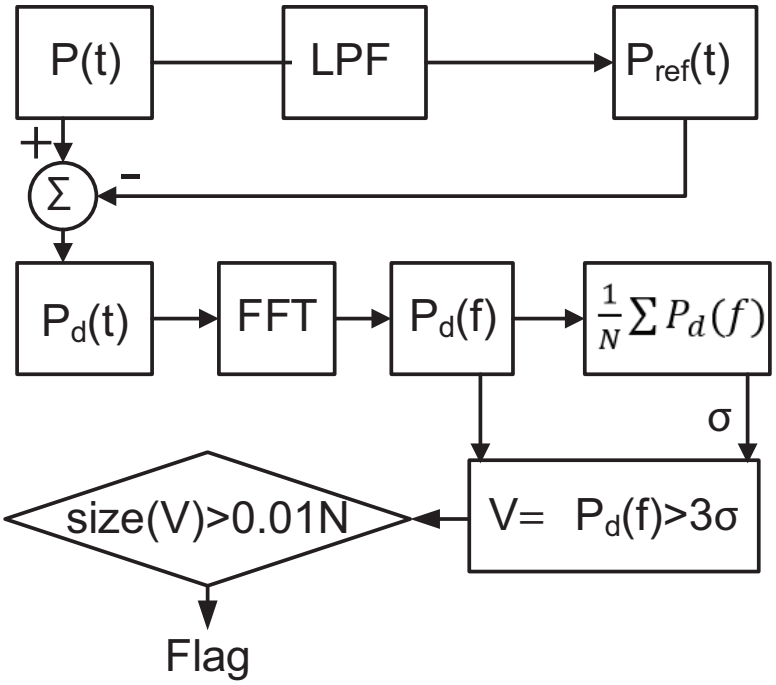


Figure 26: Detection Algorithm

subtracted from that of $P(t)$ to exclude primary signal components from the analysis. The resulting residual time-domain signal $P_d(t)$ contains noise, the majority of the Trojan signal energy, and some additional components due to modeling effects. The noise level, σ_n, can be determined from the residual spectrum $P_d(f)$ by taking an average of the spectrum. To avoid false positives due to noise, we set a spectrum threshold of $3\sigma_n$ (corresponding to a confidence level of nearly 99.7%), and record spectral components (V) exceeding this threshold.

We extend the mathematical model to remove the need for sinusoidal excitation, which may not be practical in an Internet of Things (IoT) context. Similarly, there is a need for techniques to detect malicious activity in lower power IoT devices. To this end, this work uses an arbitrary repetitive pattern to drive the device under test such that its output, e.g., its power consumption, is driven into a PSS, while the Trojan

is excited in an uncorrelated fashion. In this way, we decouple the response of the device from that of the Trojan circuit and noise.

$$S_f(f) = \sum_{n=1}^{\infty} \frac{2P_f}{n\pi} sin(\frac{n\pi d}{T})\delta(f - nf_0) \tag{4.4}$$

Detection of unknown malicious activity can be generalized to any periodic foreground signal, including a periodic pulse train, as in Equation 4.4, where P_f represents the power of each pulse, d represents the duration of each pulse, and T represents the period of the pulse train. Hence, it is possible to use this self-referencing technique where the foreground (legitimate) activity is repetitive, resulting in a power signature in the shape of a pulse train. This can be achieved either by intentionally repeating a program segment or identifying a program segment that repeats and adjusting the sampling window accordingly. If the same foreground frame is captured several times, it is possible to obtain a spectral signature of the foreground activity. The spectral signature of each window then is compared to this average to determine whether the signature contains the expected flat spectrum (due to noise) or unknown unauthorized activity in addition to noise, which would result in a non-flat spectrum. We use the high-frequency portion of the spectrum as a reference to judge the *flatness* of the resulting signature. By placing a threshold on the spectral signature, we can determine how many bins violate this threshold. The number of such bins should not exceed 1% (due to the 3σ band). Hence, violations beyond this number are treated as evidence of unauthorized malicious activity, as seen in Figure 28 (d).

4.5 Experimental Evaluation

4.5.1 Experimental Setup

To evaluate the proposed approach, we perform two sets of experiments, one on an IoT device prototype (Gupta, Park, et al. 2017) and another on a commercial MpSoC (ODROID, n.d.). The wearable IoT device runs a gesture recognition algorithm under a limited energy budget (Bhat, Park, and Ogras 2017) and is representative of low-power IoT devices used for human-computer interaction and mobile health monitoring. The second application scenario uses Wi-Fi communication, which is omnipresent in many mobile application scenarios, ranging from smartphones to simple Wi-Fi enabled IoT devices.

The wearable and IoT device is capable of performing multiple tasks and is configured to fulfill one or more needs of a specific target group. It can support applications such as gesture recognition, temperature reading, location reporting, heart rate reading, and health/fitness monitoring, all of which create a periodic steady-state without requiring any special arrangements during the design stage. We have to remember that an exhaustive testing strategy is universally acknowledged as prohibitive because designs of even moderate complexity need hundreds of years to test comprehensively.

These various periodic applications can be selected to reduce test generation effort and will adequately account for continuous working and environmental changes. In addition to these functional applications, dedicated firmware that concentrates the power spectrum on known harmonics can be added and run periodically. Such an application can run at arbitrary frequencies unknown to the attacker, thereby obscuring

the test strategy, even if the attacker has knowledge of the design. Changing the run frequency from one test period to another will further help prevent compromising the detection system. In our implementation, we use gesture recognition and Wi-Fi applications, which are expected to create a repetitive pattern while the device is in use in the environment.

4.5.2 Malicious Activity

A Trojan, when inserted into a chip, will most likely consume power. However, the Trojan's contribution to the total power consumption of the circuit depends heavily on its size and type. We know that full activation of a Trojan uses structural and functional patterns. It would be extremely challenging considering that the size and type of Trojan are unknown to us. Trojans often consist of two parts: a trigger circuit and a payload circuit. The trigger circuit judges whether the condition for the Trojan payload is satisfied or not using the trigger inputs and/or internal states of the circuit. The trigger part is often considered to be always-on circuitry that monitors the operation of the contaminated system for the activation sequence. The payload circuit creates malfunctions such as leakage of information, downgrading the circuit's performance, or catastrophic failure of the system.

At the system level testing or side-channel analysis, the output effect of the Trojan that is the power consumption in our case is only the consideration for analysis. For our analysis, we will consider the Trojan output effect on power consumption while the trigger circuit or payload circuit is working. Based on observation of power consumption, we are able to detect abnormalities or malicious activity at the device. Thus, we must be able to separate the components of current that are likely to be

Table 8: Types of Trojans.

Trojan Type	**Activity Duration (∼ms)**	**Trojan/Total Energy (%)**
Type I	5	1
Type II	10	2
Type III	15	3
Type IV	30	6

consumed by a Trojan in the infested device. In this work, we present four types of Trojans and their energy ratio in Table 8. From the viewpoint of the trigger circuit, the Trojans tend to increase power consumption or redundant processes, and consequently, they lead to faults or gradual degradation of the quality of the product. Their effect on the regular circuit at a given time is minimal, and therefore their detection is difficult.

In an ideal case, our malicious activity detection method applies circuit input vectors, triggers the malicious activity, observes the unintended behavior, and reports it to the design owner. However, in practice, the activation of malicious activity is challenging and sometimes is impossible, as we do not have enough information about the Trojan's features, including its location (firmware or hardware), the trigger condition, and the malicious functionality. Therefore, we propose focusing on Trojan's side-channel signal as the primary detection method as the power consumption of the Trojan is a substantial side-channel signal analysis parameter. In this article, we pay particular attention to determining the smallest detectable Trojan, i.e., the lowest energy that a Trojan may have and still be detected, using one of these four types of malicious activities that are enabled at random times. Instead of focusing on identifying triggering and activation mechanisms, our proposed method is intended to detect a malicious addition of any kind. We should also note that the always-on

type of Trojan that shows constant DC current consumption is not the target of the proposed detection method. This type of Trojan can easily be found by a time-domain power trace detection method (Bhunia and Tehranipoor 2019).

4.5.3 Comparisons to Existing Trojan Detection Methods

By comparison, time-domain analysis of the same current measurement data for our proposed Trojan types cannot distinguish between Trojan-free or Trojan-infested runs. This occurs because the Trojan's current consumption is hidden within the margins allowed for process and environment variations (Y. Jin and Y. Makris 2010). In this way, the malicious activity can be hidden within the run from the system level examination. Figure 27a plots the current consumption when a period of gesture recognition application is run by 100 of the Trojan-free runs, as well as the $\mu \pm 3\sigma$ envelope of the current consumption for 500 of the Trojan-free runs. Figure 27b and Figure 27c plot the current consumption when a period of gesture recognition application run by the Type-III and Type-IV Trojan-infested runs, respectively. This demonstrates that malicious activity is still within the $\mu \pm 3\sigma$ envelope and cannot be detected by time-domain analysis.

4.5.4 Proposed Detection Algorithm Optimization

The assumption of rareness is explored in many Trojan detection proposals. The objective is to reduce the monitoring and computation time of the test (i.e., the size of the collected information from the device) while increasing the confidence level that the device under test is Trojan-free. This is very desirable as it lessens the test time

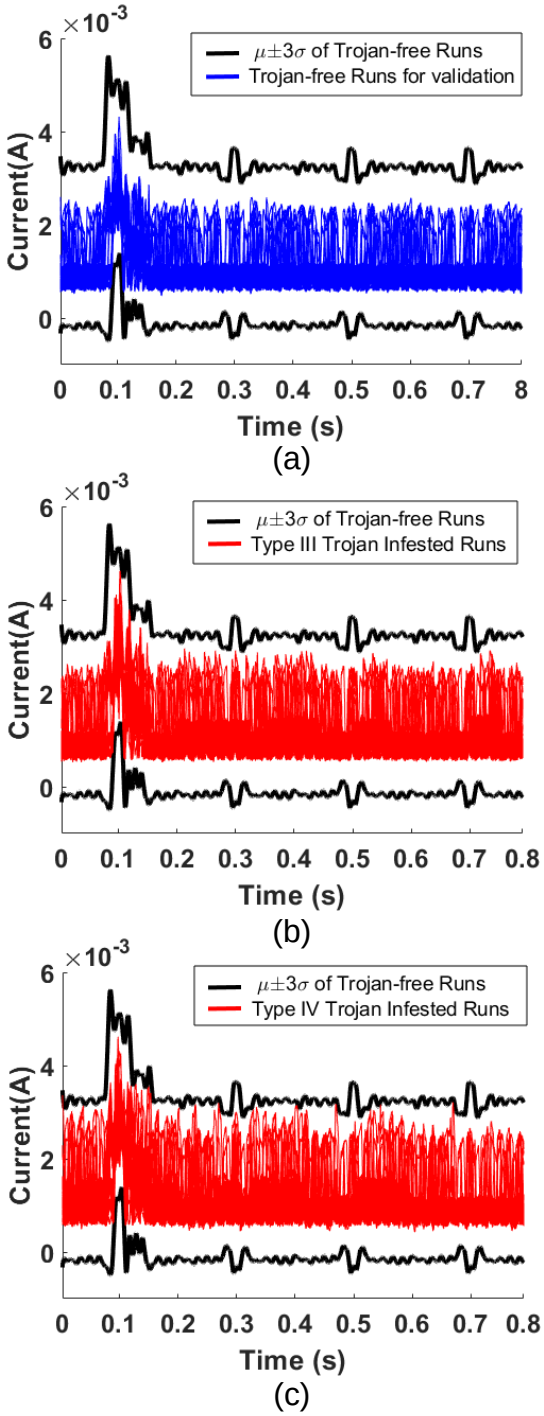


Figure 27: The $\mu \pm 3\sigma$ current consumption envelope of 500 Trojan-free runs and the current consumption of 100 Trojan-free runs (a), the current consumption of 100 Type-I Trojan-infested runs (b), and the current consumption of 100 Type-II Trojan-infested runs (c).

Table 9: Proposed Detection Method Test Modes.

Test Mode	**Signal Stitching**	**Min Monitoring Time (s)**	**Computation Time (s)**	**FFT**
Mode 1	No	20	60	Regular
Mode 2	Yes	0.8	60	Regular
Mode 3	Yes	0.8	14	Optimized

(faster testing is needed for resource-constrained device) and eliminates the stress of continuous testing.

As seen in Table 9, our detection algorithm can be used with a different set of test modes. To use the proposed detection algorithm, different test modes can be selected based on device usage. We also evaluate the sensitivity of our detection method in the presence of measurement and background noise. Our proposed test requires a collection of 20 seconds of data that can be collected without system interruption for Test Mode 1. When the complete data sequence is collected, the analysis of the data takes approximately 60 seconds to run with a standard recursive FFT algorithm. Since test Mode 1 may not be practical during periods when the device is in heavy use, we apply a signal stitching technique to eliminate the need for continuous data collection. Thus, the monitoring time can be reduced to approximately 0.8 seconds. Once the full sequence of data is collected, a regular FFT can be run, which takes approximately 60 seconds to complete the analysis for the Test Mode 2.

Battery-powered devices have limited resources to support regular operation, so security or monitoring functions need to minimize their resource requirements. The proposed technique consumes 80 seconds for Mode 1 testing, which can be done during device idle time. The Mode 1 test also represents the most exhaustive method that was explored. As seen in Table 9, other test modes require much less time and achieve

acceptable coverage. The computation time of a regular FFT can be further optimized for our proposed detection technique, as seen in Test Mode 3. Part of the data from the Fourier Transform is saved, as seen in Figure 24b. A limited period of data is measured and zero-padded to the same size as the complete data sequence. If the Fourier Transform of the zero-padded data is added to the saved Fourier Transform, which was previously analyzed and not flagged for any suspicious activity, we can safely analyze one period of data using our proposed optimized FFT. Hence, the single test computation time will go down approximately 4.2 times with optimized FFT, as seen in Table 9. Test Mode 3 is used to analyze a short period of data collected recently, while Mode 2 waits until the device gathers and stitches a full data sequence for analysis. Test Mode 2 works effectively with data collected throughout the day to detect the abnormalities and provides higher detection coverage. In contrast, test Mode 3 can run the detection algorithm based on only a short period of available data.

4.5.5 Gesture Recognition Application

The gesture recognition algorithm runs on the IoT prototype attached to the wrist of a user with a period of 0.8 seconds (Park et al. 2018). The wearable prototype includes a microprocessor TI CC2650 and a motion processing unit, Invensense MPU-9250. It features test ports for the microprocessor power consumption, which was measured using NI PXIe-4081 and PXIe-4080 digital multimeter systems with a 5 kHz sampling frequency. During the recognition algorithm running, the motion processing unit records the accelerometer and gyroscope readings. Then, the micro-controller processes the sensor data using a neural network to recognize the user gesture. Finally,

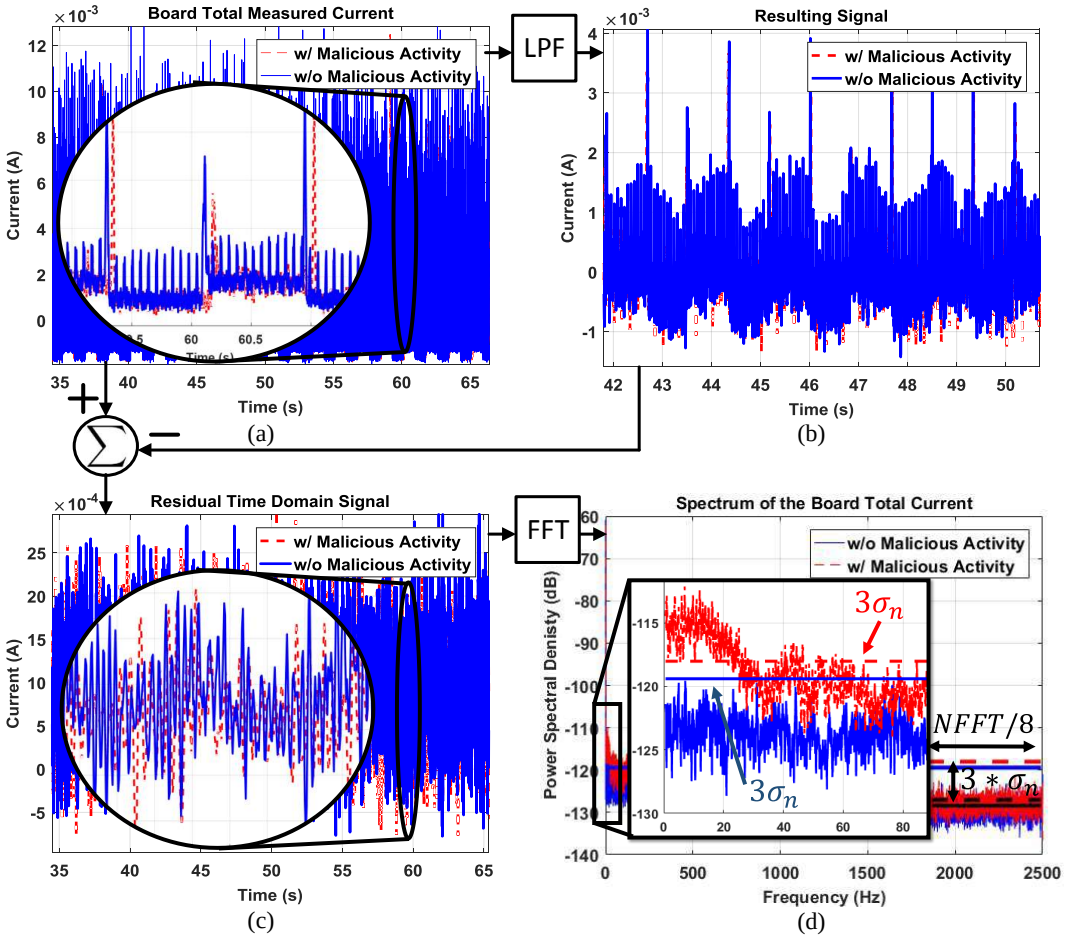


Figure 28: (a) The total current drawn by the IoT prototype. (b) The current signal after the low-pass filter. (c) The resulting residual time domain signal, which contains the noise signal and the majority of the malicious activity energy. (d) The residual signal spectrum with calculated noise threshold levels.

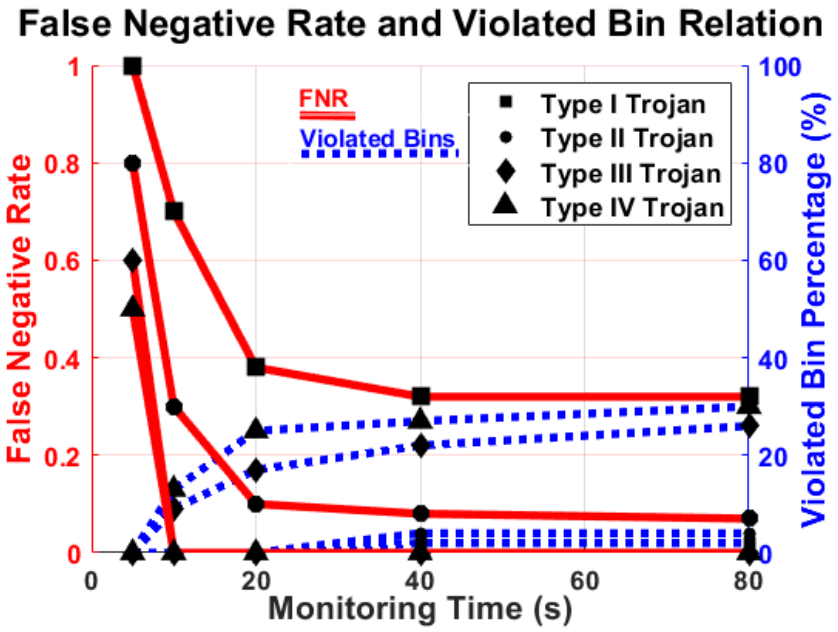
the recognized gesture is transmitted via Bluetooth Low Energy (BLE) communication protocol. While the application runs in the foreground, various malicious programs, as seen in Table 8, are launched randomly on the micro-controller.

The total current is drawn by the IoT prototype with and without Trojan activity are depicted in Figure 28a. We observe that the time domain signals are almost identical, also highlighted by the zoomed-in section of the plot. The current signals

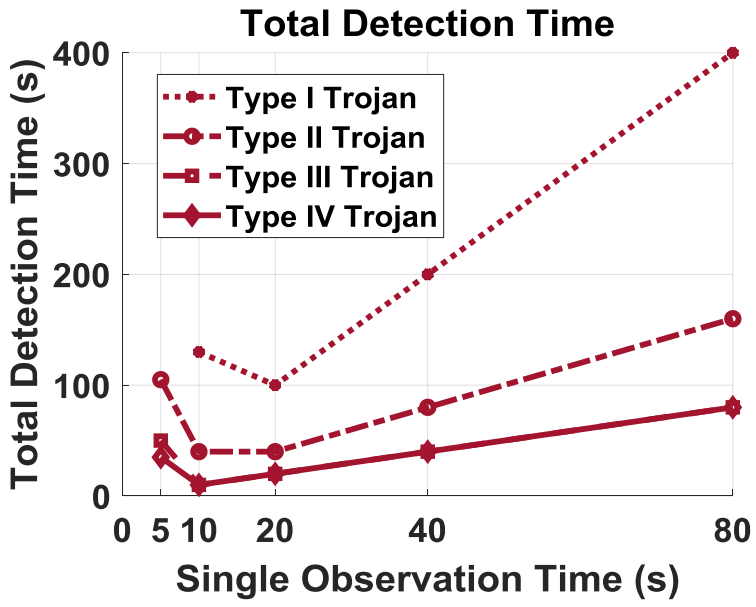
after low-pass filtering are plotted in Figure 28b. The difference between the original and filtered signal gives the residual signal, as shown in Figure 28c. This residual time-domain signal contains the noise and the majority of the malicious activity energy, but the malicious activity is still not differentiable. Finally, we provide the power spectrum of the current with and without Trojan activity in Figure 28d. The frequency-domain data clearly shows that malicious activity exhibits a unique signature at low frequencies, which is easily differentiable from the spectrum without any malicious activity.

To show the robustness of the detection technique, we ran our test with four different Trojans for 5, 10, 20, 40, and 80 seconds. In our experiments, we collected 740 spectra data, which had no malicious activity and 110 different spectra for each type of Trojan. The experiments were performed at different times of day to improve confidence in detection techniques for environmental changes, such as temperature. The signal stitching technique was used in these experiments to combine data gathered at a different time during the day and, therefore, with different temperatures. In each of these cases, the Trojans activate randomly, resulting in different time/frequency signatures. The experiments were performed at different times of the day to improve confidence in the fact that there are ***no false-positives*** generated by the system. This ensures that potential changes in the environment do not affect the evaluation. As mentioned earlier, even without malicious activity, a small percentage of the spectrum may be polluted due to harmonics not related to the signal. Therefore, the threshold level for violations is set to 1% of the compared bins of the lower frequency and will flag suspicious activity only if violations exceed the limit.

As seen in Figure 29a, the red lines show the false-negative rate of Trojans, and the blue lines depict the minimum violated bin percentage. The percentage of minimum



(a)



(b)

Figure 29: (a) Relation of False negative rate (FNR) (red) and minimum number of violated bin (blue) with respect to monitoring time. There are no false positives. (b) Total time to detect malicious activity with respect to single observation time.

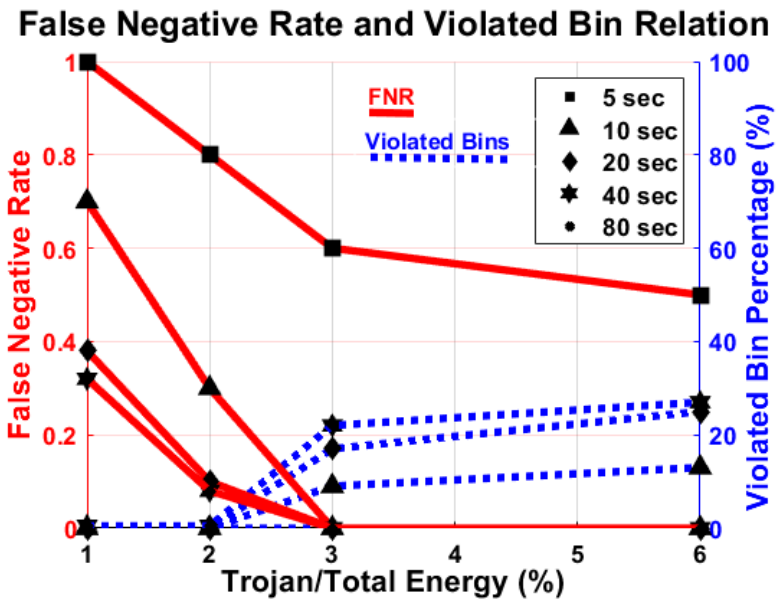


Figure 30: Relation of False negative rate (FNR) (red) and minimum number of violated bin (blue) with respect to Trojan energy. There are no false positives.

violated bin is flat if we go beyond 20 seconds of monitoring time. The number of bins over the self-referenced threshold does not linearly increase with monitoring time. Figure 29b plots the total detection time with respect to the duration of a single observation to evaluate the cost of testing, based on the number of tests required to detect unauthorized activity. In order to make sure the system is secure, the confidence level is set to 99% for completely automated malicious activity detection. The detection time decreases up to 20 seconds of observation, but increases after that. Based on the observation, we see that a single test can detect type III and IV Trojans if the monitoring time is 10 seconds or more. The Type I Trojans require five repetitions, and Type II Trojans need only two repetitions of 20 seconds or more to detect, as seen in Figure 29b. Based on our experimental outcome, we decided to a monitoring time of 20 seconds, which is optimal for all types of Trojans under consideration.

Power consumption by malicious activity is minimal compared to the rest of the operation of the original system. In particular, we focus on determining the smallest detectable Trojan, i.e., the lowest energy that a Trojan may have and still be detected, using one of these four types of malicious activities that are randomly enabled with various activity duration. We run the Trojans standalone 1000 times and take the average of the total to determine energy to calculate the energy of the Trojans. As seen in Figure 30, the false-negative rate is drastically reduced when the Trojan energy increased from 1% to 2%, and we reliably detect the Trojan if it is 3% or more of the total power of the system. In addition, the percentage of violated bins significantly increases when the Trojan energy increases from 2% to 3% but does not change much after that.

4.5.6 Wi-Fi Application

This section demonstrates the proposed approach using a Wi-Fi application running on a commercial MpSoC (Odroid-XU3) (ODROID, n.d.). The foreground application reads sensor information and transmits the data over Wi-Fi to a target destination. To model the malicious activity, we employ a simple firmware Trojan that copies the sensor data to a separate location at random instances. We assume that the Trojan is always active and does not rely on a particular trigger mechanism.

First, we randomize the patterns of the Wi-Fi application, just like the firmware Trojan. Figure 31a shows the residual power pattern (after averaging) under this scenario. Since there is no repeating pattern to the foreground application, the signatures with and without Trojan are identical. Next, we repeat the Wi-Fi application with the same frame, whereas the Trojan is unaltered. Figure 31b shows the residual

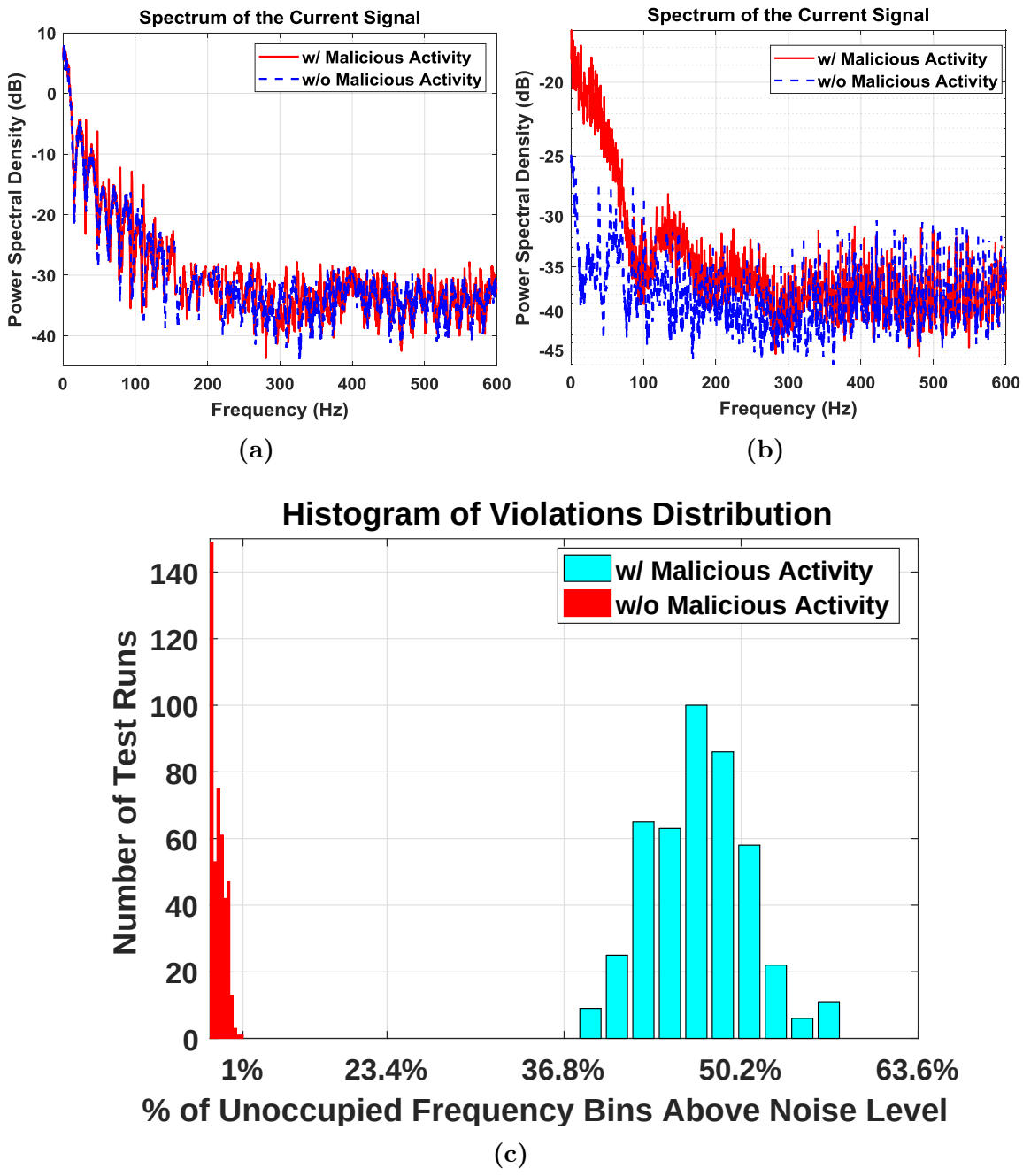


Figure 31: (a) Residual spectrum of random WiFi activity signatures with and without malicious activity are identical. (b) Residual spectrum of Periodic WiFi activity clearly reveals that there is a significant difference. (c) Histogram of number of spectrum violations for 1000 residual spectra out of which 500 had malicious activity with minimum 36.97% of spectrum violation, while without malicious activity experiments spectra with maximum 0.95% of spectrum violation.

Table 10: Percentage of violations in Data Sets (DS) 1–5.

0.5 % Threshold	w/Trojan (%)			w/outTrojan (%)			False (%)	
	Max	Min	Ave	Max	Min	Ave	Pos.	Neg.
DS1	54.42	36.97	46.99	0.21	0.0	0.12	0.0	0.0
DS2	45.87	39.25	42.32	0.12	0.09	0.10	0.0	0.0
DS3	58.38	38.62	49.23	0.38	0.0	0.09	0.0	0.0
DS4	49.48	37.17	42.85	0.95	0.0	0.17	0.0	0.0
DS5	47.50	38.36	43.37	0.21	0.0	0.11	0.0	0.0

spectrum of the remainder signal after averaging and filtering. Due to the repetitive nature of the foreground signal, it can be referenced with respect to itself, leaving only small variations due to noise and other factors. The malicious activity due to the Trojan is observable under this scenario. Note that, for detection, we do not need to compare the spectrum with a golden signature; the expectation is to have a flat spectrum, regardless of the power levels. We only need to analyze the spectrum of the measured signal to deduce whether it contains only noise or if there is an unwanted activity in addition to the primary circuit current signal and noise.

We performed 500 different experiments, with and without malicious activity. For each experiment, the IoT device is driven into a periodic steady-state for approximately 30 seconds by sending repetitive Wi-Fi messages. We apply the noise threshold explained in Section 4.4, and count the number of frequency bins above the noise level, which are referred to as violations. The histogram of the number of violations is depicted in Figure 31c. We can observe that the spectrum without malicious activity had a significantly lower number of violations as expected. In fact, there is an apparent separation between the histograms of spectrum violations with and without malicious activity.

Table 10 shows the number of violations for five data sets as a percentage of the

number of frequency bins. We set a threshold of 1% violations due to the random nature of noise. In Figure 31c, we can clearly classify all the spectra with less than 1% violations as Trojan free.

Chapter 5

CONCLUSIONS

In this thesis, we propose a self-referenced malicious activity detection method with two different side-channel parameters. First, we present a complete stand-off detection methodology for unauthorized activity in the Internet of things (IoT) nodes and other lightweight embedded systems. The unauthorized activity can be due to malicious modifications at the hardware, firmware, or software level.

We place the design under test (DUT) in a test mode by periodically running a test application and measuring the EM emission from the DUT. The repetitive nature of the test application produces a frequency spectrum where the power is concentrated on the repetition period's harmonics. We make no further assumptions on the test application in terms of how much power is expected at a given frequency. Furthermore, we assume that the periodicity of the unauthorized activity is different than that of the test application. This assumption can be relaxed by using two different test applications with different periods. By removing the harmonic frequencies of the test application out of consideration and using noise at higher frequencies as a reference, we develop an analytical detection methodology. Experimental results show that the proposed method is effective in detecting unauthorized activity with an activation probability above 5% at or below 120cm distance.

Second, we present a novel malicious Trojan activity detection technique using noise-based self-referencing that utilizes a signal stitching technique to reduce test time by a factor of 25 and analysis computation by more than four times with an optimized recursive Fast Fourier Transform (FFT). Self-referencing enables the use of

parametric measurements, such as power consumption and current, to detect malicious Trojan activity without using a trusted golden reference. This is important, since obtaining a golden signature is extremely difficult, and, in some cases, unfeasible. Self-referencing is achieved by putting the design under test into a periodic steady state. This periodic behavior concentrates the primary circuit signal power into a known frequency and reveals malicious activity using spectral analysis.

We also evaluate the sensitivity of our detection method in the presence of measurement and background noise. We have performed separate experiments on an IoT device and a commercial system-on-a-chip (SoC) with randomly activated and randomly switching Trojans. These experiments demonstrate that the proposed approach can successfully detect Trojan activity without causing false alarms. Detection capability is limited by the overall energy consumed by the unauthorized activity spread within an observation bandwidth. Depending on the rate of activity, this energy, spread over a wide spectrum, may fall below the measurement sensitivity. We should note that a Trojan that is not activated within the test duration will not generate a spectral signature and cannot be detected via the proposed method. In addition, a Trojan that is active in a very short burst may also escape detection since its energy will fall below the detectable level.

NOTES

REFERENCES

Agrawal, Dakshi, Selcuk Baktir, Deniz Karakoyunlu, Pankaj Rohatgi, and Berk Sunar. 2007. "Trojan detection using IC fingerprinting." In *Security and Privacy, 2007. SP'07. IEEE Symposium on,* 296–310.

Akkaş, M., Radosveta Sokullu, and Hüseyin Çetin. 2020. "Healthcare and Patient Monitoring Using IoT." *Internet of Things* (February): 100173. doi:10.1016/j.iot.2020.100173.

Al Faruque, Mohammad Abdullah, Sujit Rokka Chhetri, A Canedo, and J Wan. 2016. *Forensics of thermal side-channel in additive manufacturing systems.* Technical report. Tech. Rep., 2016.[Online]. Available: http://cecs. uci. edu/files/2016/01/CECS-TR-01-16. pdf.

Antonopoulos, Angelos, Christiana Kapatsori, and Yiorgos Makris. 2018. "Hardware Trojans in Analog, Mixed-Signal, and RF ICs." In *The Hardware Trojan War,* 101–123. Springer.

Aref, SH, H Latifi, MI Zibaii, and M Afshari. 2007. "Fiber optic Fabry–Perot pressure sensor with low sensitivity to temperature changes for downhole application." *Optics communications* 269 (2): 322–330.

Bachy, Yann, Frédéric Basse, Vincent Nicomette, Eric Alata, Mohamed Kaâniche, Jean-Christophe Courrège, and Pierre Lukjanenko. 2015. "Smart-TV security analysis: practical experiments." In *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks,* 497–504. IEEE.

Backes, Michael, Markus Dürmuth, Sebastian Gerling, Manfred Pinkal, and Caroline Sporleder. 2010. "Acoustic Side-Channel Attacks on Printers." In *USENIX Security Symposium,* 307–322.

Banga, Mainak, and Michael S Hsiao. 2009. "A novel sustained vector technique for the detection of hardware Trojans." In *VLSI Design, 2009 22nd International Conference on,* 327–332. IEEE.

Bao, Chongxi, Domenic Forte, and Ankur Srivastava. 2014. "On application of one-class SVM to reverse engineering-based hardware Trojan detection." In *Quality Electronic Design (ISQED), 2014 15th International Symposium on,* 47–54. IEEE.

Basnight, Zachry, Jonathan Butts, Juan Lopez Jr, and Thomas Dube. 2013. "Firmware modification attacks on programmable logic controllers." *International Journal of Critical Infrastructure Protection* 6 (2): 76–84.

Bencsáth, Boldizsár, Levente Buttyán, and Tamás Paulik. 2011. "XCS based hidden firmware modification on embedded devices." In *SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks,* 1–5. IEEE.

Bhasin, Shubhendu, Jean-Luc Danger, Sylvain Guilley, Xuan Thuy Ngo, and Laurent Sauvage. 2013. "Hardware trojan horses in cryptographic ip cores." In *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on,* 15–29. IEEE.

Bhat, Ganapati, Ranadeep Deb, Vatika Vardhan Chaurasia, Holly Shill, and Umit Y Ogras. 2018. "Online human activity recognition using low-power wearable devices." In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD),* 1–8. IEEE.

Bhat, Ganapati, Jaehyun Park, and Umit Y Ogras. 2017. "Near optimal energy allocation for self-powered wearable systems." In *Proceedings of International Conference on Computer Aided Design.*

Bhunia, Swarup, and Mark Tehranipoor. 2019. "Chapter 16 - System Level Attacks & Countermeasures." In *Hardware Security,* 419–448. Morgan Kaufmann. doi:https://doi.org/10.1016/B978-0-12-812477-2.00006-X.

Bidmeshki, Mohammad-Mahdi, and Yiorgos Makris. 2015. "Toward automatic proof generation for information flow policies in third-party hardware IP." In *Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on,* 163–168. IEEE.

Bletsch, Tyler, Xuxian Jiang, Vince W Freeh, and Zhenkai Liang. 2011. "Jump-oriented programming: a new class of code-reuse attack." In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security,* 30–40. ACM.

Bogdan, Paul, Miroslav Pajic, Partha Pratim Pande, and Vijay Raghunathan. 2016. "Making the internet-of-things a reality: from smart models, sensing and actuation to energy-efficient architectures." In *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis,* 25. ACM.

Callan, Robert, Alenka Zajic, and Milos Prvulovic. 2014. "A practical methodology for measuring the side-channel signal available to the attacker for instruction-level events." In *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture,* 242–254.

CarePredict. 2017. *Nordic-powered wireless wearable provides caregivers of seniors feedback on potential health concerns.* https://www.nordicsemi.com/News/2017/07/Tempo-from-CarePredict.

CENTRI IoTAS. 2019. *White Paper - Heavyweight Security for Lightweight Devices.* https://www.centritechnology.com/wp-content/uploads/2019/02/Heavyweight-Security-for-Lightweight-Devices.pdf.

Cha, Byeongju, and Sandeep K Gupta. 2013. "Trojan detection via delay measurements: A new approach to select paths and vectors to maximize effectiveness and minimize cost." In *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE),* 1265–1270. IEEE.

Chakraborty, Rajat Subhra, Seetharam Narasimhan, and Swarup Bhunia. 2009. "Hardware Trojan: Threats and emerging solutions." In *Proc. of Intl. High Level Design Validation and Test Workshop,* 166–171.

Chapman, Alex. 2014. "Hacking into internet connected light bulbs." http://contextis.com/resources/blog/hacking-internet-connected-light-bulbs, 2014 July.

Cheng, Z., P. Li, J. Wang, and S. Guo. 2015. "Just-in-Time Code Offloading for Wearable Computing." *IEEE Transactions on Emerging Topics in Computing* 3, no. 1 (March): 74–83. doi:10.1109/TETC.2014.2387688.

Conley, Kelsey, Alex Foyer, Patrick Hara, Tom Janik, Jason Reichard, Jon D'Souza, Chandana Tamma, and Cristinel Ababei. 2019. "Vibration Alert Bracelet for Notification of the Visually and Hearing Impaired." *Journal of Open Hardware* 3 (October). doi:10.5334/joh.17.

Costin, Andrei, Jonas Zaddach, Aurélien Francillon, and Davide Balzarotti. 2014. "A large-scale analysis of the security of embedded firmwares." In *23rd USENIX Security Symposium (USENIX Security 14),* 95–110.

Cui, Ang, Michael Costello, and Salvatore J Stolfo. 2013. "When Firmware Modifications Attack: A Case Study of Embedded Exploitation." In *NDSS.*

De Mulder, Elke, Pieter Buysschaert, Sıddıka Berna Örs, Peter Delmotte, Bart Preneel, Guy Vandenbosch, and Ingrid Verbauwhede. 2005. "Electromagnetic analysis at-

tack on an FPGA implementation of an elliptic curve cryptosystem." In *Computer as a Tool, 2005. EUROCON 2005. The International Conference on,* 2:1879–1882. IEEE.

Dohr, Angelika, R Modre-Opsrian, Mario Drobics, Dieter Hayn, and Günter Schreier. 2010. "The internet of things for ambient assisted living." In *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on,* 804–809. Ieee.

Du, Dongdong, Seetharam Narasimhan, Rajat Subhra Chakraborty, and Swarup Bhunia. 2010. "Self-referencing: a scalable side-channel approach for hardware trojan detection." In *Cryptographic Hardware and Embedded Systems, CHES 2010,* 173–187. Springer.

Duflot, Loic, Yves-Alexis Perez, and Benjamin Morin. 2011. "What if you can't trust your network card?" In *International Workshop on Recent Advances in Intrusion Detection,* 378–397. Springer.

Duflot, Loic, Yves-Alexis Perez, Guillaume Valadon, and Olivier Levillain. 2010. "Can you still trust your network card." *CanSecWest/core10:* 24–26.

Evans, Dave. 2011. *The Internet of Things. How the next evolution of Inetnet is changing everything. Cisco Internet Business Solutions Group (IBSG).*

Faruque, Al, Mohammad Abdullah, Sujit Rokka Chhetri, Arquimedes Canedo, and Jiang Wan. 2016. "Acoustic side-channel attacks on additive manufacturing systems." In *Proceedings of the 7th International Conference on Cyber-Physical Systems,* 19. IEEE Press.

Genkin, Daniel, Itamar Pipman, and Eran Tromer. 2014. "Get your hands off my laptop: physical side-channel key-extraction attacks on PCs." In *Cryptographic Hardware and Embedded Systems–CHES 2014,* 242–260. Springer.

Genkin, Daniel, Adi Shamir, and Eran Tromer. 2014. "RSA key extraction via low-bandwidth acoustic cryptanalysis." In *Advances in Cryptology–CRYPTO 2014,* 444–461. Springer.

Ghosh, Swaroop, Mohammad Nasim Imtiaz Khan, Asmit De, and Jae-Won Jang. 2016. "Security and privacy threats to on-chip non-volatile memories and countermeasures." In *Proceedings of the 35th International Conference on Computer-Aided Design,* 10. ACM.

Graves, Ricardo, Giorgio Di Natale, Lejla Batina, Shivam Bhasin, Baris Ege, Apostolos Fournaris, Nele Mentens, Stjepan Picek, Francesco Regazzoni, Vladimir Rozic, et al. 2015. "Challenges in designing trustworthy cryptographic co-processors." In *Circuits and Systems (ISCAS), 2015 IEEE Int. Symp.* 2009–2012.

Gupta, Ujjwal, Jaehyun Park, Hitesh Joshi, and Umit Y Ogras. 2017. "Flexibility-Aware System-on-Polymer (SoP): Concept to Prototype." *IEEE Transactions on Multi-Scale Computing Systems* 3 (1): 36–49.

Gupta, Ujjwal, Chetan Arvind Patil, Ganapati Bhat, Prabhat Mishra, and Umit Y Ogras. 2017. "DyPO: Dynamic Pareto-Optimal Configuration Selection for Heterogeneous MpSoCs." *ACM Transactions on Embedded Computing Systems (TECS)* 16 (5s): 123.

Hamdioui, Said, Jean-Luc Danger, Giorgio Di Natale, Fethulah Smailbegovic, Gerard van Battum, and Mark Tehranipoor. 2014. "Hacking and protecting IC hardware." In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE),* 1–7. IEEE.

He, Chunhua, Bo Hou, Liwei Wang, Yunfei En, and Shaofeng Xie. 2015. "A failure physics model for hardware Trojan detection based on frequency spectrum analysis." In *Reliability Physics Symposium (IRPS), 2015 IEEE International,* PR.1.1–PR.1.4. doi:10.1109/IRPS.2015.7112822.

Heuser, Annelie, Stjepan Picek, Sylvain Guilley, and Nele Mentens. 2016. "Side-channel Analysis of Lightweight Ciphers: Does Lightweight Equal Easy?" In *RFIDSec.*

Hu, Kangqiao, Abdullah Nazma Nowroz, Sherief Reda, and Farinaz Koushanfar. 2013. "High-sensitivity hardware trojan detection using multimodal characterization." In *Proc. of DATE,* 1271–1276.

Jayakumar, H., K. Lee, W. S. Lee, A. Raha, Y. Kim, and V. Raghunathan. 2014. "Powering the Internet of Things." In *2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED),* 375–380. August. doi:10.1145/2627369.2631644.

Jayakumar, Hrishikesh, Kangwoo Lee, Woo Suk Lee, Arnab Raha, Younghyun Kim, and Vijay Raghunathan. 2014. "Powering the internet of things." In *Proceedings of the 2014 international symposium on Low power electronics and design,* 375–380. ACM.

Jin, Y., and Y. Makris. 2010. "Hardware Trojans in Wireless Cryptographic ICs." *IEEE Design Test of Computers* 27, no. 1 (January): 26–35. doi:10.1109/MDT.2010.21.

Jin, Yier, and Yiorgos Makris. 2008. "Hardware Trojan detection using path delay fingerprint." In *Proc. of Intl. Workshop on Hardware-Oriented Security and Trust,* 51–57.

Kachman, Ondrej, and Marcel Balaz. 2016. "Optimized differencing algorithm for firmware updates of low-power devices." In *Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2016 IEEE 19th International Symposium on,* 1–4. IEEE.

Karabacak, Fatih, Umit Y Ogras, and Sule Ozev. 2016. "Detection of malicious hardware components in mobile platforms." In *17th International Symposium on Quality Electronic Design, ISQED 2016.* IEEE Computer Society.

Karri, R., J. Rajendran, K. Rosenfeld, and M. Tehranipoor. 2010. "Trustworthy Hardware: Identifying and Classifying Hardware Trojans." *Computer* 43 (10): 39–46.

Keoh, Sye Loong, Sandeep S Kumar, and Hannes Tschofenig. 2014. "Securing the internet of things: A standardization perspective." *IEEE Internet of Things Journal* 1 (3): 265–275.

Kim, Lok-Won, John D Villasenor, and Cetin K Koç. 2009. "A Trojan-resistant system-on-chip bus architecture." In *Proc. of IEEE Military Communications Conf.* 1–6.

Kocher, Paul, Ruby Lee, Gary McGraw, Anand Raghunathan, and Srivaths Moderator-Ravi. 2004. "Security as a new dimension in embedded system design." In *Proceedings of the 41st annual Design Automation Conference,* 753–760. ACM.

Konstantinou, C., A. Keliris, and M. Maniatakos. 2016. "Taxonomy of firmware Trojans in smart grid devices." In *2016 IEEE Power and Energy Society General Meeting (PESGM),* 1–5.

Konstantinou, Charalambos, and Michail Maniatakos. 2015. "Impact of firmware modification attacks on power systems field devices." In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm),* 283–288. IEEE.

Leabman, Michael A, and Gregory Scott Brewer. 2018. *Battery life of portable electronic devices.* US Patent App. 14/586,062, June.

Leander, Gregor. 2016. "Intrinsic Code Attestation by Instruction Chaining for Embedded Devices." In *Security and Privacy in Communication Networks: 11th*

International Conference, SecureComm 2015, Dallas, TX, USA, October 26-29, 2015, Revised Selected Papers, 164:97. Springer.

LF Engineering Company. n.d. "Model L-600S ELF/VLF H-Field Loop Receiving System." https://www.lfengineering.com/files/pdf/L-600.pdf.

Liu, Yu, Yier Jin, and Yiorgos Makris. 2013. "Hardware Trojans in wireless cryptographic ICs: silicon demonstration & detection method evaluation." In *Proceedings of the International Conference on Computer-Aided Design,* 399–404. IEEE Press.

Maglaras, Leandros A, and Jianmin Jiang. 2014. "Intrusion detection in scada systems using machine learning techniques." In *Science and Information Conference (SAI), 2014,* 626–631. IEEE.

McLaughlin, Stephen, Charalambos Konstantinou, Xueyang Wang, Lucas Davi, Ahmad-Reza Sadeghi, Michail Maniatakos, and Ramesh Karri. 2016. "The Cybersecurity Landscape in Industrial Control Systems." *Proceedings of the IEEE* 104 (5): 1039–1057.

Miller, Charlie. 2011. "Battery firmware hacking." *Black Hat USA:* 3–4.

Narasimhan, Seetharam, Xinmu Wang, Dongdong Du, Rajat Subhra Chakraborty, and Swarup Bhunia. 2011. "TeSR: A robust temporal self-referencing approach for hardware trojan detection." In *Proc. of Intl. Symp. on Hardware-Oriented Security and Trust,* 71–74.

Narasimhan, Seetharam, Wen Yueh, Xinmu Wang, Saibal Mukhopadhyay, and Swarup Bhunia. 2012. "Improving IC security against Trojan attacks through integration of security monitors." *IEEE Design & Test of Computers* 29 (5): 37–46.

North American Electric Reliability Council, New Jersey. 2009. *NERC Disturbance Reports.*

Nowroz, Abdullah Nazma, Kangqiao Hu, Farinaz Koushanfar, and Sherief Reda. 2014. "Novel Techniques for High-Sensitivity Hardware Trojan Detection Using Thermal and Power Maps." *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 33 (12): 1792–1805.

O'Neill, Maire. 2016. "Insecurity by Design: Today's IoT Device Security Problem." *Engineering* 2 (1): 48–49.

ODROID. n.d. "Platforms, ODROID – XU3." http://www.hardkernel.com/main/products/prdt_info.php?g_code=G140448267127.

Park, Jaehyun, Ganapati Bhat, Cemil S Geyik, Umit Y Ogras, and Hyung Gyu Lee. 2018. "Energy-Optimal Gesture Recognition using Self-Powered Wearable Devices." In *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS),* 1–4. IEEE.

Peeters, Eric, François-Xavier Standaert, and Jean-Jacques Quisquater. 2007. "Power and electromagnetic analysis: Improved model, consequences and comparisons." *Integration, the VLSI journal* 40 (1): 52–60.

Qorvo. 2020. *Qorvo's Senior Lifestyle and Family Lifestyle Services.* https://www.qorvo.com/applications/internet-of-things/lifestyle-systems.

Rahman, Mahmudur, Bogdan Carbunar, and Madhusudan Banik. 2013. "Fit and vulnerable: Attacks and defenses for a health monitoring device." *arXiv preprint arXiv:1304.5672.*

Ravi, Srivaths, Anand Raghunathan, Paul Kocher, and Sunil Hattangady. 2004. "Security in embedded systems: Design challenges." *ACM Transactions on Embedded Computing Systems (TECS)* 3 (3): 461–491.

Regazzoni, Francesco, and Ilia Polian. 2017. "Securing the hardware of cyber-physical systems." In *Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific,* 194–199. IEEE.

Rieck, Jakob. 2016. "Attacks on Fitness Trackers Revisited: A Case-Study of Unfit Firmware Security." *arXiv preprint arXiv:1604.03313.*

Rostami, M, F Koushanfar, J Rajendran, and R Karri. 2013. "Hardware security: Threat models and metrics." In *Proc. of Intl. Conf. on Computer-Aided Design,* 819–823.

Sadeghi, Ahmad-Reza, Christian Wachsmann, and Michael Waidner. 2015. "Security and privacy challenges in industrial internet of things." In *Proceedings of the 52nd Annual Design Automation Conference,* 54. ACM.

Sauvage, Laurent, Sylvain Guilley, and Yves Mathieu. 2009. "Electromagnetic radiations of fpgas: High spatial resolution cartography and attack on a cryptographic module." *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* 2 (1): 4.

Shila, Devu Manikantan, Penghe Geng, and Teems Lovett. 2016. "I can detect you: Using intrusion checkers to resist malicious firmware attacks." In *Technologies for Homeland Security (HST), 2016 IEEE Symposium on,* 1–6. IEEE.

Sklavos, Nicolas, Ricardo Chaves, Giorgio Di Natale, and Francesco Regazzoni. 2017. *Hardware Security and Trust: Design and Deployment of Integrated Circuits in a Threatened Environment.*

Standaert, François-Xavier, Tal G Malkin, and Moti Yung. 2009. "A unified framework for the analysis of side-channel key recovery attacks." In *Annual International Conference on the Theory and Applications of Cryptographic Techniques,* 443–461. Springer.

Stepanov, Sander, and Anastasios Venetsanopoulos. 2008. "Random pulse train spectrum calculation unleashed." In *Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on,* 523–526. IEEE.

Tax, DMJ. 2013. *Data description toolbox dd tools 2.0. 0.*

Tehranipoor, Mohammad, and Farinaz Koushanfar. 2010. "A Survey of Hardware Trojan Taxonomy and Detection." *IEEE Design and Test of Computers* 27 (1).

Tekeoglu, Ali, and Ali Şaman Tosun. 2016. "A testbed for security and privacy analysis of IoT devices." In *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS),* 343–348. IEEE.

Vlacheas, Panagiotis, Raffaele Giaffreda, Vera Stavroulaki, Dimitris Kelaidonis, Vassilis Foteinos, George Poulios, Panagiotis Demestichas, Andrey Somov, Abdur R Biswas, and Klaus Moessner. 2013. "Enabling smart cities through a cognitive management framework for the internet of things." *Communications Magazine, IEEE* 51 (6): 102–111.

Wang, Xueyang, Charalambos Konstantinou, Michail Maniatakos, and Ramesh Karri. 2015. "ConFirm: Detecting firmware modifications in embedded systems using hardware performance counters." In *Computer-Aided Design (ICCAD), 2015 IEEE/ACM International Conference on,* 544–551. IEEE.

Xiao, K., D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor. 2016. "Hardware Trojans: Lessons Learned after One Decade of Research." *ACM Trans. Des. Autom. Electron. Syst.* (New York, NY, USA) 22, no. 1 (May). doi:10.1145/2906147.

Xiao, Kan, Domenic Forte, and Mohammad Tehranipoor. 2014. "A Novel Built-In Self-Authentication Technique to Prevent Inserting Hardware Trojans." *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 33 (12): 1778–1791.

Xiao, Kan, Xuehui Zhang, and Mohammad Tehranipoor. 2013. "A Clock Sweeping Technique for Detecting Hardware Trojans Impacting Circuits Delay." *Design Test, IEEE* 30 (2): 26–34.

Yan, Bo, and Guangwen Huang. 2009. "Supply chain information transmission based on RFID and internet of things." In *Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on,* 4:166–169. IEEE.

Yu, Qiaoyan, and Jonathan Frey. 2013. "Exploiting error control approaches for Hardware Trojans on Network-on-Chip links." In *Proc. of Intl. Symp. on Defect and Fault Tolerance in VLSI and NanoTech. Systems,* 266–271.