Leaky Hardware

Modeling and Exploiting Imperfections in Embedded Devices

Ilias Giechaskiel

Kellogg College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2019

Knowledge is in the end based on acknowledgement.

— Ludwig Josef Johann Wittgenstein

Acknowledgements

Although conducting research and collating results in a long, coherent document can be a lonely process, I was extremely lucky to have the support I needed during my years at Oxford. I would thus like to use this space to express my appreciation for those who kept me sane when I could not see light at the end of the tunnel.

First and foremost, I would like to extend my sincerest gratitude to my family, particularly my parents, Nathan and Elisavet Giechaskiel, who have taught me to value the pursuit of knowledge, and to strive for excellence.

To the Oxford University Greek Society (OUGS) football club: thank you for stretching me to my athletic limits on the field, and for being great friends off the pitch. Jantje Silomon, thank you for making RHB102 a welcoming place, and for sharing many meals, laughs, and pints of diet coke with me when we needed a break. Sophia Sklaviadis, your patience and insights when repeatedly looking over my application material and thesis drafts cannot be overstated.

I owe a big thank you to my partner and editor-in-chief, Dr. Eleni Philippou, for always being encouraging, and for putting up with my many sleepless nights lost in front of a dimly-lit computer screen.

I would like to thank my research group for their helpful feedback on papers and presentations alike. Youqian Zhang, in particular, you have been a great co-author, and a master of turning data and ideas into meaningful figures. I would also like to acknowledge Shanquan Tian's help in getting me off the ground with FPGA experiments on the cloud.

Throughout my DPhil, I have had the pleasure of great supervision. I therefore thank Prof. Cas Cremers for a nine-week mini-project, which turned into a wonderful year-long collaboration; Prof. Ken Eguro for taking a chance with an ambitious internship project at Microsoft Research; Prof. Jakub Szefer for hosting me at Yale and supporting my research for five months; and Prof. Kasper B. Rasmussen for allowing me to explore my academic interests within and outside of Oxford.

This thesis was made possible with the financial support of the Clarendon Fund; the Engineering and Physical Sciences Research Council (EPSRC); Kellogg College; the Oxford Department of Computer Science; and the Center for Doctoral Training (CDT) in Cyber Security, whose grants I used to purchase equipment, travel to conferences, as well as cover university fees and day-to-day expenses.

As a final note, I would like to thank Prof. Andrew Martin for bringing the CDT to life, as well as David Hobbs and Maureen York, whose help in navigating the Oxford landscape has been immeasurable.

Computer Science is a science of abstraction.

— Alfred Vaino Aho

Abstract

Embedded systems are found in many safety- and security-critical applications, and bring aspects of the physical world to the digital one and vice versa. However, imperfections in this hardware bridge can break the integrity of sensor inputs into an embedded device, causing it to act upon the wrong data. For instance, malicious electromagnetic transmissions can trick systems into inducing defibrillation shocks and raising the temperature of infant incubators, both with potentially severe health consequences.

Unfortunately, such attacks which alter sensor outputs without changing the property being measured itself have so far only been studied in an ad-hoc manner. In my thesis, I address this shortcoming in two ways. First, I create a taxonomy of these "out-of-band" signal injection attacks and defenses. Second, I propose a framework that quantifies security in their context through a system model, mathematical definitions, and an algorithm that can compare the "security level" of off-the-shelf systems.

In my thesis, I also investigate Field-Programmable Gate Arrays (FPGAs), which are available on public cloud infrastructures, and are also integrated in many consumer end-products, such as smartphones and laptops. As FPGAs are often used in sensitive applications, including genome processing, cryptography, and financial modeling, it is necessary to ensure that they can maintain the secrecy of the data that they process.

However, the confidentiality of FPGA data can be broken, as I demonstrate through three new sources of information leakage due to hardware imperfections. The first source exists between "long wires" within seven families of Xilinx FPGAs. I explain how to exploit long-wire leakage for covert- and side-channel attacks, both locally, and on two commercial FPGA clouds through novel ring oscillators structures that bypass currently-deployed countermeasures.

The second source of leakage operates even when different FPGA users are isolated to distinct dies of the same chip. These unintended interactions demonstrate that current FPGA architectures are not well-suited for multi-tenancy, despite the physical isolation of user logic. Finally, I show that assigning dedicated FPGAs to different users is still not enough to prevent cross-FPGA communication: shared Power Supply Units (PSUs) leak information between physically distinct FPGA, CPU, and GPU boards, which can be detected via means of a novel receiver design and classification metric.

Overall, in my thesis, I highlight that the underlying electrical properties of embedded devices often fall short of protecting the integrity and the confidentiality of the data that they process, and allow remote attackers to spoof sensor measurements or infer cryptographic keys and other types of data.

Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

ΔC . . . . Absolute Count Difference 111, 229

Δd_{RO} . . . Absolute Delay Difference 139, 229

ΔRC . . . Relative Count Difference 107, 139, 229

$\Delta\Sigma$ . . . . Delta-Sigma 14, 91

$\Sigma\Delta$ . . . . Sigma-Delta 14

`HLONG` . . . Horizontal Long 23

`VLONG` . . . Vertical Long 23, 100, 137, 231

AAC . . . Advanced Audio Coding 89

ABS . . . . Anti-Lock Braking System 38

AC . . . . Alternating Current 18, 40, 184, 223

ADC . . . Analog-to-Digital Converter 5, 12, 39, 76, 158, 186, 215

AES . . . . Advanced Encryption Standard 27, 123, 182

AM . . . . Amplitude Modulation 36, 37

ASIC . . . Application-Specific Integrated Circuit 26, 130

AWS . . . Amazon Web Services 7, 21, 134

AXI4 . . . Advanced eXtensible Interface 4 22

BLT . . . . Baum–Liu–Tesche 17

BRAM . . Block Random-Access Memory 20, 103

CERT . . Computer Emergency Readiness Team 30

CERT . . Cyber Emergency Response Team 30

CLB . . . Configurable Logic Block 22, 103, 138, 233

CPU . . . Central Processing Unit 3, 19, 156, 158, 187

CUDA . . Compute Unified Device Architecture 164

DC . . . . Direct Current 14, 38, 78, 184, 216

DCM . . . Digital Clock Manager 22, 106

DIP . . . . Dual Inline Package 92
DoS . . . . Denial-of-Service 45
DPI . . . . Direct Power Injection 33
DRAM . . Dynamic Random-Access Memory 22
DRC . . . Design Rule Check 28, 155
DSP . . . . Digital Signal Processing 20, 37, 103
DTMF . . Dual-Tone Multi-Frequency 36
DVFS . . . Dynamic Voltage and Frequency Scaling 182
EC2 . . . . Elastic Cloud Compute 21
ECDSA . . Elliptic Curve Digital Signature Algorithm 182
ECG . . . Electrocardiogram 1, 35
EM . . . . Electromagnetic 1, 11, 34
EMC . . . Electromagnetic Compatibility 16, 32
EMI . . . Electromagnetic Interference 16, 33, 95
ENOB . . Effective Number of Bits 14
ESD . . . . Electrostatic Discharge 14, 39, 78
FACS . . . FPGA-Accelerated Cloud Server 21
FF . . . . Flip-Flop 20, 135
FIFO . . . First In, First Out 20
FIR . . . . Finite Impulse Response 64
FM . . . . Frequency Modulation 36, 37
FPGA . . . Field-Programmable Gate Array . . . 3, 12, 42, 53, 100, 133, 157, 185
FSPL . . . Free-Space Path Loss 17
GPIO . . . General Purpose Input/Output 38
GPU . . . Graphics Processing Unit 4, 156, 158, 187
HDD . . . Hard-Disk Drive 48
HDL . . . Hardware Description Language 19
HLS . . . High-Level Synthesis 19
HT . . . . Hardware Trojan 26, 102
HW . . . . Hamming Weight 100, 176
IC . . . . . Integrated Circuit 12, 39, 91, 182

IEMI . . . Intentional Electromagnetic Interference 32
ILA . . . . Integrated Logic Analyzer 106
IMD . . . Implantable Medical Device 1, 34
I/O . . . . Input/Output 20, 40, 106
IP . . . . . Intellectual Property 4, 20, 100, 188
IR . . . . . Infrared 51
LD . . . . Latch 135
LED . . . Light-Emitting Diode 52, 117
LFSR . . . Linear Feedback Shift Register 108
LiDAR . . Light Detection and Ranging 31
LNA . . . Low-Noise Amplifier 221
LPF . . . . Low-Pass Filter 14, 47, 94
LUT . . . Lookup Table 20, 105, 135, 161, 229
MEMS . . Micro-Electro-Mechanical System 18, 44, 48, 96
MMCM . Mixed-Mode Clock Manager 22, 136
MUX . . . Multiplexer 20
NTC . . . Negative Temperature Coefficient 41
PCB . . . Printed Circuit Board 12, 41, 76
PCC . . . Pearson Correlation Coefficient 89, 215
PCIe . . . Peripheral Component Interconnect Express 22, 136, 159
PDF . . . . Probability Distribution Function 79
PDN . . . Power Distribution Network 27, 158
PLL . . . . Phase-Locked Loop 22, 136
PM . . . . Phase Modulation 37
PSU . . . . Power Supply Unit 4, 28, 158, 187
PUF . . . . Physical Unclonable Function 26, 135
PVT . . . Process, Voltage, and Temperature 26, 139
PWM . . . Pulse Width Modulation 38
RAM . . . Random-Access Memory 27
RF . . . . Radio Frequency 34
RFI . . . . Radio Frequency Injection 33

RMS . . . Root-Mean-Square 87, 217
RO . . . . Ring Oscillator 6, 12, 42, 53, 104, 134, 158, 186, 228
RSA . . . Rivest-Shamir-Adleman 182
RTD . . . Resistance Temperature Detector 41
RTT . . . Receiver-Transmitter-Transmitter 119
SAR . . . Successive Approximation 14, 91
SDR . . . Software-Defined Radio 72, 129
S/H . . . . Sample-and-Hold 14
SLL . . . . Super Long Line 236
SLR . . . . Super Logic Region 4, 24, 134, 182, 186, 236
SoF . . . . Start-of-Frame 122
SPI . . . . Serial Peripheral Interface 92
SPL . . . . Sound Pressure Level 45
SSD . . . . Solid-State Drive 3, 19, 60
SSI . . . . Stacked Silicon Interconnect 19
STR . . . . Self-Timed Ring 52
TACC . . . Texas Advanced Computing Center 21
TDC . . . Time-to-Digital Converter 27, 155, 158, 236
TRNG . . True Random Number Generator 26, 42, 44, 53
TRT . . . Transmitter-Receiver-Transmitter 120
TSOT . . . Thin Small-Outline Transistor 92
TSSOP . . Thin-Shrink Small-Outline Package 92
TSV . . . . Through-Silicon Via 25
UART . . . Universal Asynchronous Receiver/Transmitter . 91, 106, 136, 160, 223
UAV . . . Unmanned Aerial Vehicle 51
USB . . . . Universal Serial Bus 41
USRP . . . Universal Software Radio Peripheral 129
UV . . . . Ultraviolet 51
VISA . . . Virtual Instrument Software Architecture 91
VR . . . . Virtual Reality 45
XADC . . Xilinx Analog-to-Digital Converter 25, 224

A ship in harbor is safe, but that is not what ships are built for.

— John Augustus Shedd

1 Introduction

Contents

Embedded systems are integrated into products that typically interface with physical processes [203], and thus often contain sensors converting physical properties into electrical features. As shown in Figure 1.1, sensors (and, by extension, embedded systems) are ubiquitous, and directly or indirectly influence the daily lives of many. They can be found everywhere from motion detectors in home security systems and heart rate monitors in smartwatches, to speedometers in cars and altimeters in airplanes. Moreover, they are present in research and industrial applications, including thermal infrared sensors on satellites, and pressure sensors in factories. Their measurements guide many safety- and security-critical actions, so attacking them or the embedded systems in which they are integrated can have serious consequences for the people that depend on them.

Indeed, in 2013 Foo Kune et al. demonstrated that adversarial electromagnetic (EM) emissions could cause electrocardiograms (ECGs) to erroneously report heart beats, and further result in defibrillation shocks being delivered by Implantable Medical Devices (IMDs) [94]. These adversarial signal injection attacks were unlike prior

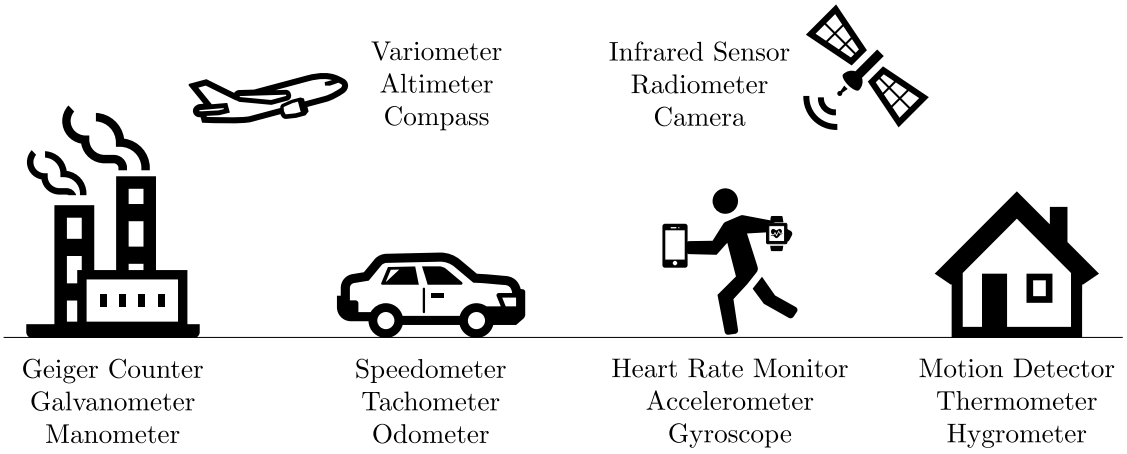


Figure 1.1: Sensors in embedded devices are a part of most people's daily lives.

work which simply exploited insecure IMD interfaces that were explicitly made for communication [133]: although appropriate cryptographic protocols can protect digital interfaces, it is harder to defend analog interfaces which cannot be authenticated.

In the years since Foo Kune et al.'s results, more types of such "out-of-band" signal injection attacks have started to surface. Although some still use EM interference as the method of injection—for example, so as to remotely trigger smartphone voice commands [153]—other research achieves similar goals through other means. For instance, acoustic transmissions can control the output of accelerometers [336] or cause speech recognition systems to respond to inaudible voice commands [380]. However, these remote attacks on the integrity of data inputs to embedded systems have so far only been studied in an ad-hoc fashion, without extensive insights into their commonalities.

To address this limitation, this thesis first systematizes such attacks and related defenses, highlighting previously-unexplored connections between them and work in electromagnetic interference and side-channel attacks (Chapter 3). In doing so, this thesis uncovers gaps in their experimental methodology, as well as challenges for future research. One of these challenges, namely the lack of a way to quantify the security of different systems in the context of out-of-band signal injection attacks, is solved through a framework which includes a system model, mathematical definitions, and an algorithm that can compare the "security level" of off-the-shelf systems (Chapter 4).

In its first part, this thesis thus shows that since many attacks are due to hardware imperfections (e.g., unintentional antennas in the wires connecting sensors to micro-controllers, or non-linearities in the analog-to-digital conversion process), in order to protect the integrity of measurements, low-level details about the hardware necessarily "leak" into abstractions about the system itself. As Joel Spolsky put it, "all non-trivial abstractions, to some degree, are leaky" [306]. As a result, in order to build secure systems, one must design around the hardware limitations.

In its second part, this thesis investigates how less-than-perfect devices also leak information in a more traditional sense: side-channel analysis against embedded devices can reveal secret keys by exploiting data-dependent power consumptions, EM emanations, or other unintentional behaviors of the systems' hardware [102]. In other words, having investigated how hardware imperfections impact the integrity of data inputs to embedded systems, this thesis turns to similar vulnerabilities to break the confidentiality of the data being processed. Unlike prior work which has so far typically required physical access to the system under attack and external equipment such as oscilloscopes and EM probes to measure the information leakage [102], this thesis is instead concerned with a stronger threat model: that of remote attacks on Field-Programmable Gate Arrays (FPGAs).

FPGAs make for an attractive target, as they are often integrated in consumer end-products, including Solid-State Drives (SSDs) [17], smartphones [320], and laptops [144]. In addition, they are now becoming increasingly available due to the proliferation of cloud FPGAs [361] and FPGA hybrids, such as multi-die Central Processing Units (CPUs) with integrated FPGAs [18, 67], or FPGAs with embedded hard [365] and soft [211] processors. Given that FPGAs are often used in sensitive applications, including genome processing, cryptography, and financial modeling [11], it is necessary to ensure that they can maintain the data that they process secret.

This thesis, however, demonstrates three new sources of information leakage that can remotely break the confidentiality of FPGA data, despite various countermeasures. The covert- and side-channel attacks introduced are possible even though the victim and adversary circuits are logically isolated. Moreover, they do not make use of system monitors or other privileged primitives that would normally be inaccessible to user logic in

cloud and other setups. Finally, they are present not only without access to or modification of the FPGA boards, but also without having to account for noisy environments, i.e., without maintaining the voltage or temperature of the FPGA constant.

The first attack requires the source and sink designs to use adjacent interconnect resources, called "long wires". This thesis shows for the first time that long wires leak information about their state in a way which can be measured fully on-chip and used for data exfiltration in multiple generations of FPGA devices, and in various experimental setups (Chapter 5). Through novel ring oscillator structures bypassing existing defense mechanisms, long-wire leakage can be detected even on cloud FPGAs. As a result, this attack highlights the need for physical isolation of potentially-adversarial logic.

The second scenario enforces strong physical isolation assumptions between the two communicating parties as a possible countermeasure. This approach takes advantage of high-end FPGA architectures to place different users on separate dies of the same FPGA chip, called Super Logic Regions (SLRs). Even in this case, however, a cross-SLR high-bandwidth voltage-based covert channel can be created both locally and on cloud FPGAs (Chapter 6). In other words, this attack shows that current device architectures are unsuitable for multi-tenant cloud occupancy.

The third and final setup explores whether dedicating FPGA boards on a per-user basis can prevent cross-board leakage in single-tenant FPGAs. However, this thesis demonstrates that leakage through shared Power Supply Units (PSUs) can break separation of privilege, allowing a sink FPGA to detect levels of activity from a source FPGA, Central Processing Unit (CPU), or Graphics Processing Unit (GPU). More concretely, this thesis introduces the first FPGA-to-FPGA, CPU-to-FPGA, and GPU-to-FPGA covert channels through a novel receiver design and measurement metric (Chapter 7).

In its second part, this thesis thus highlights that architectural FPGA improvements are needed to protect users from malicious logic in many different arrangements. These setups include Intellectual Property (IP) cores that share the same routing resources, physically-isolated occupants in multi-tenant environments, or even users on completely distinct FPGA boards. The contributions of this thesis are explained in more detail in Section 1.1, with a list of publications in Section 1.2, and a statement of originality in Section 1.3.

1.1 Research Goals & Contributions

The aim of this thesis is to *demonstrate how unintentional hardware properties can be exploited by remote adversaries attacking the integrity and confidentiality of information processed by embedded devices*. To do so, after introducing some relevant background and terminology (Chapter 2), this thesis *identifies gaps in the literature of out-of-band signal injection attacks and defenses* (Chapter 3, based on [111]). Specifically, through a comprehensive survey of related work, it unifies the diverse terminology used by different works to create a common language through which to discuss attack and defense mechanisms. It further systematizes the state-of-the-art through a chronological and thematic evolution of out-of-band signal injections, and creates a classification of sources of vulnerability and countermeasures. While highlighting cross-influences between electromagnetic, conducted, acoustic, and optical attacks, this thesis places out-of-band signal injection attacks and defenses in the wider context of side-channel leakage and electromagnetic interference. In doing so, gaps in the experimental approach of published research are identified, and concrete steps to overcome these challenges in the future are proposed.

One of the theoretical gaps that this thesis addresses is that of *providing a unified framework through which to evaluate the effects of out-of-band signal injection attacks* (Chapter 4 and Appendix A, based on [116]). This is done through a system model which abstracts away from engineering concerns associated with remote transmissions, and mathematical definitions for security against adversarial signal injection attacks. These definitions address effects ranging from mere disruptions of sensor readings to precise waveform injections of attacker-chosen values. The usability of the framework (model and definitions) is demonstrated in practice through an algorithm which can calculate the "security level" of real, off-the-shelf systems, and an investigation into the unintentional demodulation properties of Analog-to-Digital Converters (ADCs). It is further shown that the proposed model can be used to inform circuit design choices and evaluate defense mechanisms.

Having mapped the space of integrity attacks on embedded devices, this thesis then investigates remote confidentiality attacks, specifically to *evaluate whether it is possible to conduct on-chip side-channel analysis, and estimate the resulting information leakage* (Chapter 5 and Appendix B, based on [110, 112]). This is shown to be the case by discovering that "long" wires in FPGAs leak information about their state in a way which can be measured fully on-chip. Specifically, this thesis highlights the previously-unknown fact that if a long wire carries a logic 1, the delays of nearby long wires are slightly shorter than when it carries a logic 0, even when the driven value remains constant. This long-wire leakage is first characterized across six families of Xilinx FPGAs, and the effect is shown to be independent of the device used, the location and orientation of the transmitter and receiver, and the pattern of transmission. By exploiting this effect, it is possible to create a covert channel with bandwidth upwards of 6 kbps and accuracy of 99.9%. Side-channel attacks using the same effect can recover signals that are kept constant for as low 1.3 µs with an accuracy of more than 98.4%. It is further shown that the channel remains accurate when there is significant dynamic activity on the device or when multiple simultaneous transmissions are taking place, even without accounting for changes in the environment, such as temperature and voltage fluctuations.

Having demonstrated this phenomenon on local boards, the next natural step is to *investigate the feasibility of multi-tenant attacks on cloud FPGAs* (Chapter 6, based on [113, 114]). In doing so, this thesis introduces novel Ring Oscillator (RO) structures bypassing countermeasures which are currently deployed by commercial clouds, and compares these new RO designs to traditional combinatorial loop ROs. This is done by first illustrating that the long-wire leakage persists in a seventh family of Xilinx devices (present on commercial FPGA clouds), using a novel metric to measure femtosecond-scale changes in the delay of the long wires. As the long-wire leakage requires co-located circuits in the FPGA device, this thesis investigates a possible countermeasure through strong physical isolation between different users onto separate dies of the same FPGA, called Super Logic Regions (SLRs). However, it shows that current FPGA architectures cannot prevent information leaks across the SLR dies, and are therefore not well-suited for multi-tenant cloud setups. Specifically, this thesis contains the first cross-SLR attack,

which is demonstrated with FPGAs locally, on Amazon Web Services (AWS), and on Huawei Cloud. The bandwidth of the ensuing covert channel is characterized both analytically and experimentally, with multi-bit simultaneous transmissions reaching bandwidths of up to 4.6 Mbps with 97.6% accuracy. The strength of the leakage is measured across various experimental setups with different sizes, locations, and types of receivers and transmitters, while software- and hardware-level countermeasures are proposed to prevent and mitigate the impact of these discoveries.

As multi-tenant setups are shown to be insecure, the final research question that is addressed in this thesis is to *evaluate whether it is possible to implement remote cross-device attacks in dedicated, single-tenant FPGAs* (Chapter 7, based on [115]). This is answered in the affirmative by identifying the sharing of Power Supply Units (PSUs) as a new source of vulnerability, even for unprivileged FPGA designs without access to voltage or temperature system monitors. Specifically, this thesis introduces the first remote covert-channel attack between FPGAs on distinct physical boards, as well as the first CPU-to-FPGA and GPU-to-FPGA covert channels using high loads of activity on the respective processors. Using a novel classification metric and measurement setup, a bandwidth-accuracy tradeoff analysis is performed, and it is shown that the covert channel reaches accuracies of up to 100% across two PSUs and four Xilinx FPGA boards. Finally, defense mechanisms are proposed based on the resulting insights.

Overall, this thesis identifies several ways in which imperfect hardware can break the integrity and confidentiality of secrets processed by embedded devices. It highlights the former through an extensive taxonomy of out-of-band signal injection attacks and a new framework that quantifies the security of systems in their context. It demonstrates the latter through three new sources of intra- and inter-chip leakage on FPGAs. In conclusion (Chapter 8), to protect against either type of attack, designers must account for hardware limitations, effectively trading information leakage for leaky abstractions.

1.2 Published Results

The research I conducted for my DPhil in Cyber Security has resulted in the following peer-reviewed conference and journal publications:

1. I. Giechaskiel, K. B. Rasmussen, and K. Eguro. Leaky wires: Information leakage and covert communication between FPGA long wires. In *ACM ASIA Conference on Computer and Communications Security (ASIACCS)*, 2018 [112].

2. I. Giechaskiel, K. Eguro, and K. B. Rasmussen. Leakier wires: Exploiting FPGA long wires for covert- and side-channel attacks. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 12(3):1–29, Sep 2019 [110].

3. I. Giechaskiel, K. B. Rasmussen, and J. Szefer. Measuring long wire leakage with ring oscillators in cloud FPGAs. In *International Conference on Field-Programmable Logic and Applications (FPL)*, 2019 [113].

4. I. Giechaskiel, Y. Zhang, and K. B. Rasmussen. A framework for evaluating security in the presence of signal injection attacks. In *European Symposium on Research in Computer Security (ESORICS)*, 2019 [116]. **Best Paper Award**.

5. I. Giechaskiel, K. B. Rasmussen, and J. Szefer. Reading between the dies: Cross-SLR covert channels on multi-tenant cloud FPGAs. In *IEEE International Conference on Computer Design (ICCD)*, 2019 [114].

6. I. Giechaskiel and K. B. Rasmussen. Taxonomy and challenges of out-of-band signal injection attacks and defenses. *IEEE Communications Surveys & Tutorials (COMST)*, 22(1):645–670, Mar 2020 [111].

7. I. Giechaskiel, K. B. Rasmussen, and J. Szefer. C^{3}APSULe: Cross-FPGA covert-channel attacks through power supply unit leakage. In *IEEE Symposium on Security and Privacy (S&P)*, 2020 [115].

Moreover, during the course of my time at Oxford, I have also produced the following research, which is outside the scope of this thesis, and is therefore not discussed further:

8. I. Giechaskiel, C. Cremers, and K. B. Rasmussen. On Bitcoin security in the presence of broken cryptographic primitives. In *European Symposium on Research in Computer Security (ESORICS)*, 2016 [108].

9. I. Giechaskiel, C. Cremers, and K. B. Rasmussen. When the "crypto" in cryptocurrencies breaks: Bitcoin security under broken primitives. *IEEE Security & Privacy*, 16(4):46–56, Aug 2018 [109].

1.3 Statement of Originality

As sole first author of all publications produced during my DPhil, I have been responsible for the conception, execution, and write-up of the ideas, experimental setup, and analysis in my research. Co-authors have provided invaluable feedback throughout my research and the publication process, but all writing in this thesis is mine.

I have been the sole student author in all but one publication, which is discussed in Chapter 4 and Appendix A. The figures in these chapters were mostly produced by Youqian Zhang, who also designed and performed the smartphone injection attacks. Although Prof. Kasper B. Rasmussen provided the motivation for investigating ADC vulnerabilities, the entire framework (idea and execution) is my own work, including the system model, the security definitions, the algorithm, and the signal injection experiments into ADCs.

Moreover, I am entirely responsible for the idea and execution of the literature survey on out-of-band signal injection attacks, including the choice and definition of the term, the resulting taxonomy of attacks and defenses, as well as the identification of research challenges and open questions in the area (Chapter 3).

The investigation into long wires (Chapter 5 and Appendix B) arose organically after I noticed the effect during a more general examination of side-channel leakage in FPGAs under Prof. Ken Eguro of the University of Washington and Microsoft Research. The insight of using latches and flip-flops to create ring oscillators bypassing combinatorial loop restrictions in cloud FPGAs originated with Prof. Takeshi Sugawara of the University of Electro-Communications, but the flip-flop-based RO proposed in this thesis (Chapter 6) was conceptually designed and implemented in practice by me.

Finally, Prof. Jakub Szefer of Yale University proposed the general direction for covert-channel attacks on cloud and local FPGAs (Chapters 6 and 7), with the work developing with a more detailed focus during my investigations. Crucially, the research into SLRs,

the multi-bit transmission approach (Chapter 6), the conception of the specific FPGA-to-FPGA, CPU-to-FPGA, and GPU-to-FPGA channels, as well as the novel classification metric and measurement setups (Chapter 7) are all my own ideas, implementations, and analyses. To the best of my knowledge, I have cited all relevant sources which have influenced or are related to my thesis and publications.

A little inaccuracy sometimes saves tons of explanation.

— Hector Hugh Munro (Saki)

2 Background

Contents

This chapter introduces the relevant background to explain how imperfections and other unintentional hardware properties lead to exploitable security vulnerabilities which can remotely break the integrity and confidentiality of data processed by an embedded device. Starting with the integrity of sensor measurements, Section 2.1 discusses concepts and terminology surrounding out-of-band signal injection attacks, which form Chapters 3 and 4 of this thesis. As Section 2.1 explains, out-of-band signal injection attacks exploit non-linearities in the analog-to-digital conversion process, unintentional antennas that pick up electromagnetic (EM) transmissions, as well as imperfections in the sensors

themselves for acoustic and other attacks.

Section 2.2 then provides the necessary background to understand the confidentiality attacks of Chapters 5, 6, and 7, which are performed within and between Field-Programmable Gate Arrays (FPGAs). Specifically, by expanding on the high-level design process which is used for local and cloud FPGA deployments, Section 2.2 motivates and supports the threat models of subsequent chapters. Moreover, it presents details about the internal layout and interconnect of the FPGA Integrated Circuits (ICs), which are necessary for both the routing attacks of Chapter 5, and the cloud attacks of Chapter 6. It also describes the structure and properties of circuits called Ring Oscillators (ROs), which are used as covert- and side-channel receivers throughout the three chapters. Section 2.2 further places the rest of this thesis in context by discussing related work on remote FPGAs attacks. Section 2.3 finally summarizes the key insights of this chapter.

2.1 Fundamentals of Out-of-Band Signal Injections

Embedded systems depending on sensors to interface with their external environment require the conversion of a physical property (for instance, temperature or speed) to an electrical quantity, such as voltage or resistance. These electrical measurements are typically analog in nature, and need to be digitized by an Analog-to-Digital Converter (ADC) before they are processed. Although modern cryptography has mostly solved the problem of secure communication between digital interfaces, there is no way to authenticate the measurements themselves or the analog component of the connection between the sensor and the embedded device. This lack of authentication, coupled with hardware imperfections in the conversion process (Section 2.1.1), can be exploited for out-of-band signal injection attacks.

For EM attacks in particular, the signal transmitted by an adversary undergoes circuit-specific transformations. For instance, the wires or Printed Circuit Board (PCB) traces between the sensor and the ADC act as unintentional low-power, low-gain antennas, which are resonant at high frequencies related to the inverse of the wire length [94, 188]. As a result, attacker signals need to be transmitted over high-frequency carriers in order to be picked up by the circuit traces (Section 2.1.2). They are then demodulated into

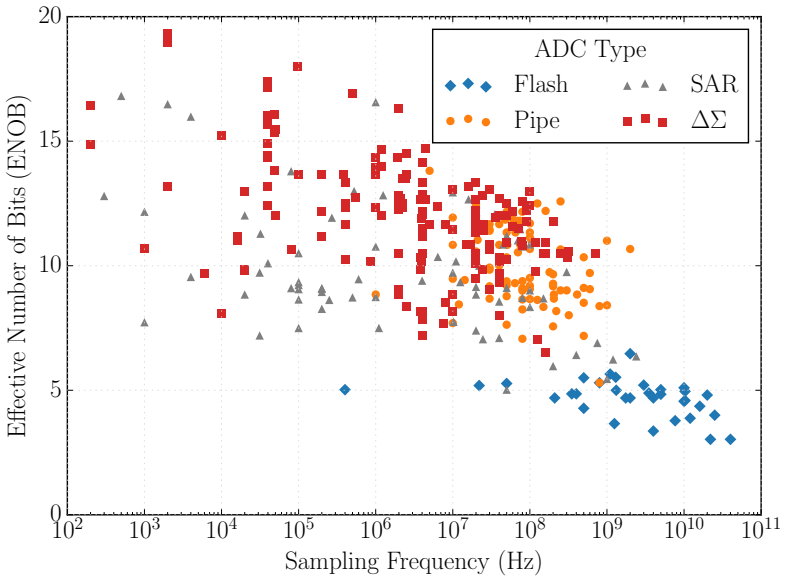


Figure 2.1: Analog-to-Digital Converter (ADC) trends: Effective Number of Bits (ENOB) and sampling frequency for different ADC types, using data compiled by Murmann [225].

low-frequency, meaningful waveforms due to non-linearities in the ADC or are rectified due to non-linearities in other components, such as transistors [250].

It should be noted that although, conceptually, the sensor, the ADC, and the interfacing logic perform distinct functions, all three can be fully integrated into the same IC. Despite the fact that this chip presents a digital interface to third parties (which can be protected by cryptographic protocols), the sensor itself can still be vulnerable to out-of-band signal injection attacks. For example, acoustic attacks targeting the resonant frequencies of gyroscopes and accelerometers have proven to be effective even against digital ICs [301, 336, 337]. These types of vulnerabilities are discussed in Section 2.1.3.

2.1.1 Analog-to-Digital Conversion

Embedded systems with sensors require ADCs to convert analog measurements to digital ones. However, ADCs contain components which may cause a mismatch between the "true" value at their input and the digitized output. This section describes how these components can affect the digitization process.

There are multiple variants of ADC architectures, which offer different tradeoffs between their *sampling rate* in Hz (i.e., how many samples per second the ADC can take) and their *resolution N* in bits, allowing the ADC to represent 2^N different values.

For instance, Delta-Sigma (ΔΣ) ADCs (also called Sigma-Delta (ΣΔ) ADCs) offer the highest resolutions at the slowest speeds, while Flash and Pipeline ADCs the opposite, with Successive Approximation (SAR) ADCs in the middle, as shown in Figure 2.1.[1] These ADC types are commonly used in many embedded systems, and Chapter 4 and Appendix A conduct experiments against them to identify differences in their demodulation characteristics.

However, despite differences in the specific internal composition of different ADC types (which are not relevant for this thesis), ADCs are composed of some common building blocks which result in similarities in their demodulation properties. Specifically, every ADC contains three basic components: a "sample- or track-and-hold circuit where the sampling takes place, the digital-to-analog converter and a level-comparison mechanism" [244], as shown in Figure 2.2. The sample-and-hold component acts as a low-pass filter, and makes it harder for an adversary to inject signals modulated at high frequencies. However, the level-comparison mechanism is essentially an amplifier with non-linearities which induces Direct Current (DC) offsets, and allows low-frequency intermodulation products to pass through. These non-linearities, along with shifts caused by Electrostatic Discharge (ESD) diodes, can produce meaningful low-frequency waveforms out of carefully-crafted signals. These waveforms tend to be modulated over high-frequency carriers in order to enter the targeted circuit [94].

Sample-And-Hold Filter Characteristics: A Sample-and-Hold (S/H) mechanism is an *RC* circuit connected to the analog input, with the resistor and the capacitor connected in series. The transfer function of the voltage across the capacitor is $H_{S/H}(j\omega) = \frac{1}{1+j\omega RC}$, and the magnitude of the gain is $G_{S/H} = \frac{1}{\sqrt{1+(\omega RC)^2}}$. As the angular frequency $\omega = 2\pi f$ increases, the gain is reduced: the S/H mechanism acts as a Low-Pass Filter (LPF). The $-3\,\mathrm{dB}$ cutoff frequency is thus $f_{cut} = \frac{1}{2\pi RC}$, which is often higher than the ADC sampling rate (see Chapter 4 and Appendix A). Hence, "aliasing" occurs when signals beyond the *Nyquist frequency* (equal to half the sampling rate) are digitized by the

[1] Figure 2.1 uses data that has been compiled by Murmann from state-of-the-art publications [225]. It estimates resolution through the Effective Number of Bits (ENOB) metric, which can account for measurement distortions due to quantization errors, thermal noise, etc.

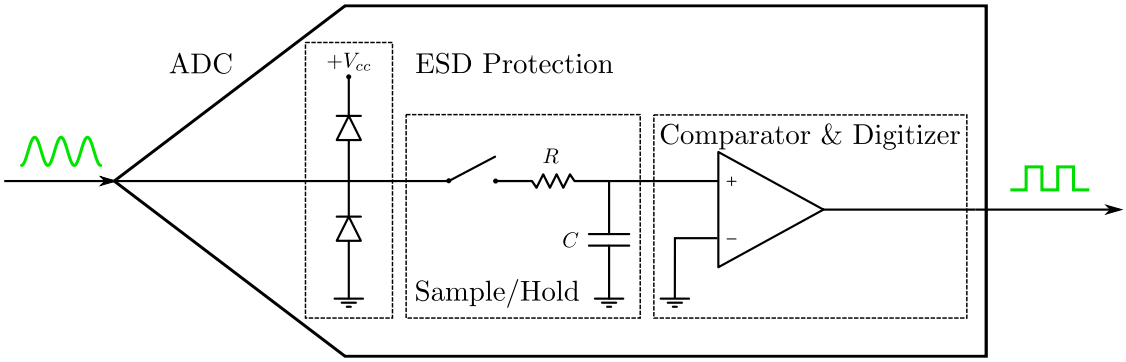


Figure 2.2: Analog-to-Digital Converter (ADC) model: Non-linearities due to Electrostatic Discharge (ESD) protection diodes and amplifiers (e.g., comparators) counteract low-pass filtering effects of the Sample-and-Hold (S/H) mechanism. These non-linearities can unintentionally demodulate high-frequency input signals, which are then processed as legitimate measurements.

ADC: high-frequency signals become indistinguishable from low-frequency signals that the ADC can sample accurately.

Amplifier Non-Linearities: Every ADC contains amplifiers: a comparator, and possibly buffer and differential amplifiers. Many circuits also contain additional external amplifiers to make weak signals measurable. All these components have harmonic and intermodulation non-linear distortions [265], which an adversary can exploit. Harmonics are produced when an amplifier transforms an input v_{in} to an output $v_{out} = \sum_{n=1}^{\infty} a_n v_{in}^n$. In particular, if $v_{in} = \hat{v} \cdot \sin(\omega t)$, then:

$$v_{out} = \left(\frac{a_2\hat{v}^2}{2} + \frac{3a_4\hat{v}^4}{8} + \cdots\right) + (a_1\hat{v} + \cdots)\sin(\omega t) - \left(\frac{a_2\hat{v}^2}{2} + \cdots\right)\cos(2\omega t) + \cdots \quad (2.1)$$

Equation (2.1) shows that "the frequency spectrum of the output contains a spectral component at the original (fundamental) frequency, [and] at multiples of the fundamental frequency (harmonic frequencies)" [265]. Moreover, the output includes a DC component, which depends only on the even-order non-linearities of the system. Besides harmonics, intermodulation products arise when the input signal is a sum of two sinusoids: $v_{in} = \hat{v}_1 \cdot \sin(\omega_1 t) + \hat{v}_2 \cdot \sin(\omega_2 t)$. In that case (for instance when the injected signal sums with the sensor signal), the output signal contains frequencies of the form $n\omega_1 \pm m\omega_2$ for integers $n, m \neq 0$. These non-linearities demodulate attacker waveforms, even when they are modulated on high-frequency carriers.

Diode Rectification: The input to the ADC can contain reverse-biased diodes to ground and V_{cc} to protect it from ESD. When the input to the ADC is negative, or when it exceeds

V_{cc}, the diodes clamp it, causing non-linear behavior. When the real sensor signal is non-zero, this behavior is also asymmetric, causing a DC shift [44, 265, 286], which compounds with amplifier non-linearities.

Conclusion: Although there are many types of ADCs, they all contain the same basic building blocks. Even though the sample-and-hold mechanism should attenuate high-frequency signals beyond the maximum sampling rate of the ADC, non-linearities due to ESD diodes and amplifiers in the ADC cause DC offsets and the demodulation of signals through harmonics and intermodulation products. Chapter 4 and Appendix A exemplify these effects through experiments with the major types of ADCs.

2.1.2 Radiated and Conducted Paths

The concept of wires and PCB traces acting as antennas has been extensively studied in the field of Electromagnetic Compatibility (EMC), both to verify that devices do not cause undue interference in nearby devices, but also to ensure that devices are immune to Electromagnetic Interference (EMI) exposure "while maintaining a predefined performance level" [220]. The aspect of how traces act as unintentional receiving antennas falls within the realm of immunity, and is therefore central to out-of-band EM signal injection attacks. Specifically, *back-door coupling* [53, 94, 117, 229, 356], where the "radiation couples through imperfections (apertures) in an electromagnetic shield, giving rise to a diffuse and complex field pattern within the shielded structure" [53] is often responsible for the vulnerability of systems to out-of-band EM attacks.

However, modeling the susceptibility of systems against back-door coupling is a hard task "without detailed testing, although properties averaged over frequency bands can be predicted" [37]. More precisely, although the resonant behavior of simple geometric structures (e.g., lines and rectangles) has been extensively studied [251], when non-linear components such as diodes are included, systematic experiments are necessary to identify the extent to which intermodulation products appear in the output [356]. Such intermodulation products can act as potential envelope detectors causing systems to take the wrong safety-critical actions [94], demonstrating that these coupling effects are a concern for more than just compliance with EMC regulations.

That said, mathematical formulas have been developed to predict the response of PCB traces to external EM fields, showing an inverse relationship between the length of the microstrip trace and resonant frequencies [188]. Many such models are based on transmission line theory, the Baum–Liu–Tesche (BLT) equation [32], and a (quasi) transverse electromagnetic propagation model [125, 187, 188, 290, 345]. Although the details of these models and the theory behind them are beyond the scope of this thesis, it is worth noting that transmission line theory [226, 227] and the BLT equation [128, 323, 325] can also be used to predict the susceptibility of circuits to *conducted* attacks, where disturbances are propagated through cables and other structures [240].

The parameters considered for these models typically include the "field incidence, polarization angles, and the magnitude and phase of the impedances loading the microstrip terminations" [177]. Since analog interconnects (e.g., to ADCs) "can be highly mismatched", signal outputs can be affected through "distortion in nonlinear components like diodes and transistors" [178]. As there is "a large variety of analog circuit topologies and interconnect geometries", it is not easy to *a priori* predict the response of circuits, and therefore analysis "should be performed on a case-by-case basis" [178].

Finally, it should be noted that, in the far field, the Free-Space Path Loss (FSPL) (derived from the Friis transmission formula [97]) is:

$$\text{FSPL} = \left(\frac{4\pi d}{\lambda}\right)^2 = \left(\frac{4\pi df}{c}\right)^2 \tag{2.2}$$

where f is the transmission frequency, $\lambda = c/f$ is the signal wavelength, and d is the distance between the transmitter and the receiver [251]. When either the frequency or the distance doubles, the path loss quadruples. As a result, to maintain the received voltage level constant, the effective radiated power (equal to transmitter power multiplied by the antenna gain) also needs to quadruple.

2.1.3 Other Non-Linearities

Out-of-band signal injection attacks that do not depend on electromagnetic emissions often exploit vulnerabilities in the sensors themselves. Specifically, acoustic transmissions target resonant mechanical frequencies and non-linearities in microphones and speakers.

Control over gyroscopes and accelerometers [301, 336, 337] lies in the former category, while inaudible voice commands into smartphones [272, 380] rely on the latter; both are discussed in this section.

Gyroscopes and accelerometers in modern devices are examples of Micro-Electro-Mechanical System (MEMS) sensors. Gyroscopes, in particular, operate through "vibrating mechanical elements to sense rotation" [22]. In other words, MEMS gyroscopes contain oscillating structures which, when rotated, appear to have a measurable force (called the *Coriolis force*) exerted on them [1]. These mechanical resonators "generate and maintain a constant linear or angular momentum", so that "when the gyroscope is subjected to an angular rotation, a sinusoidal Coriolis force at the frequency of drive-mode oscillation is induced in the sense direction" [1]. This force is exerted in a different direction from the moving direction, and, depending on the type of the gyroscope, the angular rotation can be estimated through changes in capacitance, piezoresistive effects, etc. [1, 22]. MEMS accelerometers, on the other hand, consist of spring-mass systems, with acceleration resulting in a deflection of the seismic mass. This deflection can also be detected through capacitive and piezoresistive means [175]. By transmitting sounds at the resonant frequency of the MEMS sensor, out-of-band acoustic signal injection attacks cause MEMS gyroscopes and accelerometers to report incorrect values.

By contrast, other research targets microphone non-linearities to cause inaudible sound (i.e., frequencies beyond the human-audible range of 20 kHz) to be interpreted as valid commands by speech recognition systems such as Apple Siri and Amazon Alexa. Microphones operate by converting the mechanical deformation of a membrane (caused by the air pressure of a sound wave) into a capacitive change, which produces an Alternating Current (AC) signal [380]. This process also has non-linearities that produce second-order components, including harmonics and intermodulation products, which transform modulated ultrasound transmissions to executable voice commands [380].

2.2 Field-Programmable Gate Arrays

Field-Programmable Gate Arrays (FPGAs) are Integrated Circuits (ICs) that implement reconfigurable hardware, and are often used in high-bandwidth, low-latency

applications, such as high-frequency trading [184], post-decay particle detection [126], acceleration of massively parallel computation [254], or the replacement of network cards [93] and Solid-State Drive (SSD) controllers [238]. Besides permeating distributed systems and critical infrastructure, FPGA chips are also often integrated in consumer electronics, such as SSDs [17], smartphones [320], and laptops [144], making it necessary to ensure that their computations are performed in a trustworthy manner.

This section gives the relevant background for discussing how to break the security of FPGA applications, starting with an overview of the design flow for FPGAs (Section 2.2.1), and the deployment process on public FPGA cloud providers (Section 2.2.2). This section also introduces architectural details of the internal hardware layout and routing interconnect (Section 2.2.3) of the devices used in the experiments conducted in this thesis, with the Stacked Silicon Interconnect (SSI) layout of Virtex UltraScale+ devices discussed separately (Section 2.2.4). This section then describes the circuit structure and properties of Ring Oscillators (ROs) (Section 2.2.5), which are used as covert-channel receivers in most of the FPGA-based experiments of this thesis. Finally, the contributions of this thesis are better contextualized within related work in Section 2.2.6, which summarizes remote attacks on FPGAs, with or without the use of ROs.

2.2.1 Design Process

The FPGA design process differs significantly from the traditional Central Processing Unit (CPU) programming model, since FPGA code gets translated to a physical circuit layout. This is typically accomplished through a Hardware Description Language (HDL), such as SystemVerilog or VHDL, though recent advances in High-Level Synthesis (HLS) allow a subset of C code to be translated to a legal FPGA instantiation. FPGA code primarily describes two types of circuits: *combinational logic*, i.e., functions whose outputs depend only on the circuits' current inputs, and *sequential logic*, whose current outputs also depend on past outputs [134]. An example of the former is an adder or the `XOR` function, while the latter includes clocked, stateful processes, such as counters which increment or decrement after each clock cycle.

After HDL logic gets translated to Boolean functions in a process called *synthesis*, the next step is to *map* the logic gates into manufacturer- and architecture-dependent primitives, i.e., common building blocks for a specific chip. Specifically, combinational logic in FPGAs mainly uses reconfigurable Lookup Tables (LUTs) and multiplexers (MUXes) to represent the truth table for stateless functions. Modern FPGAs also contain additional primitives to optimize the power and timing efficiency of other functions, and include Digital Signal Processing (DSP) blocks and adder carry chains in the reconfigurable fabric. Sequential logic, on the other hand, requires memory elements, which typically come in the form of D-Flip-Flops (FFs). Larger sections of memory, which can also be used for First In, First Out (FIFO) algorithms, are provided through Block Random-Access Memory (BRAM) structures. After mapping, the primitives need to be *placed* on the grid-like FPGA layout, and connections need to be *routed* between different logic blocks and external Input/Output (I/O) connections. The final step is *generating the bitstream* that gets loaded onto the device, and which configures the FPGA resources. It should be noted that designs are frequently functionally *verified* to ensure they meet specifications, while *timing analysis* is conducted, often iteratively at various points during the process, to guarantee that physical delay constraints (e.g., setup and hold times) between stages are met.

Although more low-level details on the specific primitives, physical layouts, and routing resources in Xilinx FPGAs are discussed in Section 2.2.3, it is worth highlighting some high-level properties of typical FPGA designs, and their corresponding security implications. First of all, although the conversion from HDL code to a physical implementation is typically handled by the manufacturer's tools (e.g., Xilinx Vivado or Intel Quartus), compiler directives are available for the manual placement and routing of signals. Moreover, due to the complexity and size of FPGAs (in the last decade high-end devices have effectively grown 8× in size [365, 368]), user circuits often incorporate third-party implementations of various protocols, data structures, and algorithms. These licensed designs, called Intellectual Property (IP) cores, may also come in a pre-routed black-box format to eliminate the variability of on-the-fly routing and attain a known clock frequency. As a result, as the placement and routing of different blocks is often

opaque to circuit designers, logic created by different parties can end up using adjacent physical resources, and exploit unintentional hardware interactions for data exfiltration through covert- and side-channel attacks.

Besides communication between IP cores of different security guarantees integrated in the same design [141, 142], multi-user setups will also present further threats in future FPGA deployments: CPU/FPGA hybrids are now surfacing, and include Intel Xeon CPUs with integrated FPGAs [18, 67], Xilinx Zynq FPGAs with hard ARM processors [365], and Microsemi FPGAs with soft RISC-V processors [211]. Moreover, FPGAs in cloud environments such as Amazon Web Services (AWS) are also becoming common, necessitating that cloud resources be protected from potentially adversarial user logic. These concerns are further discussed in Section 2.2.2. It should be noted that FPGA attacks which do not exploit unintentional hardware properties are also possible, for instance through bitstream modifications [60, 222, 223]. However, such attacks are outside the scope of this thesis and are therefore not expanded upon any further.

2.2.2 Cloud Deployment

FPGAs on the cloud have been available since at least 2015, when the Texas Advanced Computing Center (TACC), in collaboration with Project Catapult by Microsoft Research, gave researchers access to a cluster with 384 Intel Stratix V FPGAs [326]. Since then, the number of Intel and Xilinx FPGA cloud offerings has drastically increased, with Intel Arria 10 FPGA boards powering Alibaba Cloud F1 instances [9] and machine learning applications on Microsoft Azure [212]. Xilinx-based public FPGA clouds have also been available since 2016, when AWS announced Elastic Cloud Compute (EC2) FPGA F1 instances with Xilinx Virtex UltraScale+ FPGAs [13]. Virtex UltraScale+ FPGAs are also available on Huawei Cloud FPGA-Accelerated Cloud Server (FACS) FP1 instances [369] and on Alibaba Cloud F3 instances [9]. Moreover, Kintex UltraScale boards can be found in closed beta on Baidu FPGA Cloud Compute servers [27] and Tencent Cloud FX2 instances [322]. Finally, Xilinx Alveo Accelerator Cards are present on the Nimbix Cloud Platform [234].

These cloud FPGA offerings have resulted in a new programming model, where only part of the FPGA is dedicated to users, with the remaining resources reserved by a cloud-provided "shell". This shell is responsible for the low-level interfaces of the FPGA fabric to the exterior of the chip, including Peripheral Component Interconnect Express (PCIe) and Dynamic Random-Access Memory (DRAM) controllers, physical pinouts, etc. Due to this shell region, temperature and system monitors, device identification primitives, as well as other security features are off-limits to user logic, which is forced to interact with the shell (and external hardware) through a high-level interface, such as the Advanced eXtensible Interface 4 (AXI4). This new set of constraints raises the bar for attackers, who do not have physical access to the FPGA boards, and cannot in any way modify them.

Security research on cloud FPGAs has primarily focused on single-tenant applications, for example protecting IP designs in untrusted cloud infrastructures [86], and the cloud provider from potentially malicious user logic [334]. However, although cloud FPGAs are currently allocated on a per-user basis, it is likely that they will eventually become sharable commodity resources. Indeed, to more efficiently share the underlying hardware, several designs have been proposed to accommodate for multi-tenant occupancy of physical FPGA resources [341]. Logical isolation is often a key component of multi-tenant approaches, though physical isolation is not always enforced [52, 62, 351]. Unfortunately, even designs with physical isolation [158, 162, 163] do not consider or protect against side- and covert-channel attacks, despite their use of strict "fencing" regions [334]. As this thesis shows (Chapter 6), even strong physical isolation is not enough to protect against covert-channel communication between different tenants on the same cloud FPGA.

2.2.3 Layout & Routing Resources

Modern Xilinx FPGAs have a grid layout internally, whose fundamental building block is called a Configurable Logic Block (CLB). These CLBs are organized in columns, which are interspersed between columns containing other types of logic resources. As shown in Figure 2.3, these resources include BRAM, DSP blocks, Digital Clock Manager (DCM) resources such as Mixed-Mode Clock Managers (MMCMs) and Phase-Locked Loops (PLLs), as well as I/O transceivers. Each CLB is composed of two *slices*, each of

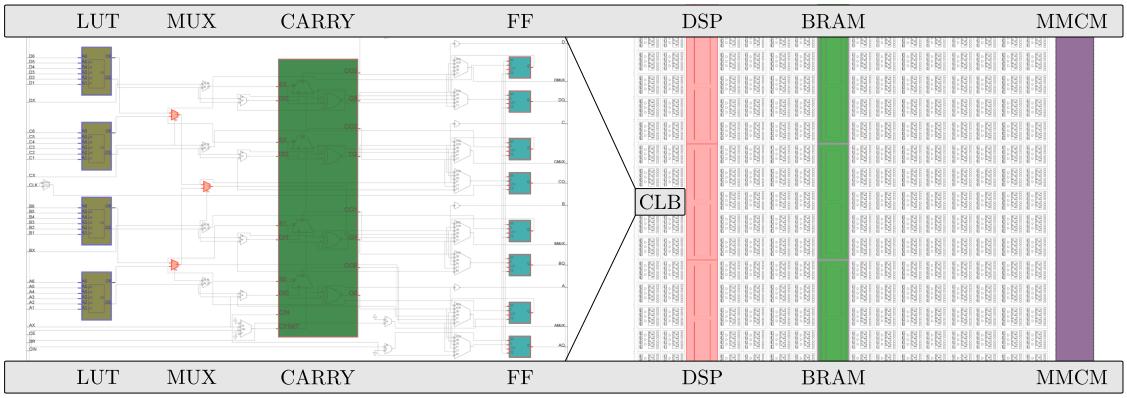


Figure 2.3: Artix 7 Digital Signal Processing (DSP), Block Random-Access Memory (BRAM), Mixed-Mode Clock Manager (MMCM), and Configurable Logic Block (CLB) columns. CLBs contain Lookup Tables (LUTs), Multiplexers (MUXes), Flip-Flops (FFs), and carry logic.

which contains four LUTs and eight registers (four registers for Virtex 5 devices), along with other resources such as MUXes and carry chains. It should be noted that in the Virtex UltraScale+ architecture, CLBs provide the same amount of resources, but the two slices have been combined into one larger slice with eight LUTs and sixteen registers.

Each CLB has an associated *switch matrix*, which contains routing resources to connect elements within a CLB, and enables CLBs to communicate with each other. There are multiple types of such communication wires with different orientations and lengths, but this thesis (Chapter 5 and Appendix B) is specifically concerned with *long* wires, and refers to shorter wires as *local routing*. Long wires are used to efficiently connect CLBs that are far apart, and can be *vertical*, between elements with the same x coordinate, or *horizontal*, having the same y coordinate. As FPGA devices have more rows than columns, this thesis primarily discusses Vertical Longs (`VLONG`s), although the phenomenon of long-wire leakage was also observed for Horizontal Longs (`HLONG`s).

The properties of `VLONG` wires have changed in the various iterations of the Xilinx architectures investigated in this thesis. For most devices, `VLONG`s can be driven from either end (i.e., they are bidirectional), and have intermediate read-only taps. Moreover, each CLB can only drive two `VLONG`s, one connecting it to CLBs above it and the other to CLBs below it. In these architectures, adjacent long wires also originate from adjacent CLBs. The length of the `VLONG`s varies per generation: Virtex 5 `VLONG`s span eighteen CLBs, and have two taps after the sixth and twelfth CLBs. Virtex 6 `VLONG`s span sixteen

Property	Virtex 5	Virtex 6	Series 7	Virtex US+
Reference	[367]	[368]	[358]	[365]
Node Size (nm)	65	40	28	16
# of Slices/CLB	2	2	2	1
# of LUTs/Slice	4	4	4	8
# of FFs/Slice	4	8	8	16
VLONG Bidirectional?	Yes	Yes	Yes	No
VLONG Span (# of CLBs)	18	16	18	12
# of VLONG Taps	2	1	1	0
# of VLONGs/CLB	2	2	2	2×8

Table 2.1: Layout and routing properties of the Xilinx families used in the long-wire experiments. Further details are provided in the text.

CLBs, and have just one intermediate tap after the eighth CLB, while Series 7 VLONGs span eighteen CLBs, but only have an intermediate tap after the ninth CLB.

The Virtex UltraScale+ architecture, however, is significantly different. VLONGs are unidirectional and tapless, only span twelve CLBs, and are organized in channels of eight. As a result, there are sixteen VLONGs originating from each CLB (eight for each direction). Moreover, adjacent long wires need to be driven from the same switch matrix, even if their driving signals (e.g., LUTs or FFs) are not in the same CLB. The properties of the various FPGA generations tested in this thesis are summarized in Table 2.1. Long-wire leakage in earlier FPGA generations is primarily investigated in Chapter 5 and Appendix B, but Chapter 6 demonstrates that despite fundamental changes in routing, long-wire leakage persists even in the Virtex UltraScale+ architecture.

2.2.4 Stacked Silicon Interconnect

High-end Xilinx boards, such as those found on AWS and Huawei Cloud, use Stacked Silicon Interconnect (SSI) technology as a way of creating much larger devices with a lower power envelope and more dedicated features [364, 365]. The SSI layout allows multiple dies, called Super Logic Regions (SLRs), to be integrated into one big FPGA chip. The Amazon and Huawei FPGA clouds investigated in this thesis use 16 nm Virtex UltraScale+ XCVU9P chips, which split their 90 clock regions equally into three separate SLR dies. These SLRs are adjacent to each other, and are connected through a silicon

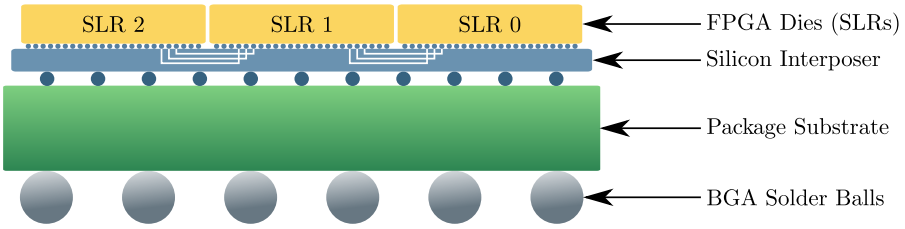


Figure 2.4: Stacked Silicon Interconnect (SSI) layout of Virtex UltraScale+ chips, adapted from a Xilinx User Guide [364]. Super Logic Regions (SLRs) are separate Field-Programmable Gate Array (FPGA) dies, connected and powered through the silicon interposer, which acts as a conduit to external Input/Output (I/O) connections through the package substrate.

interposer, as shown in Figure 2.4. This interposer is a passive layer which connects global clocking and general interconnect resources to the SLR dies [364]. It also acts as a conduit between SLR components and the package substrate, providing connectivity to I/O pins, as well as power and ground connections using Through-Silicon Vias (TSVs) [364].

Each SSI device has a master SLR die, which is responsible for "the primary configuration logic that initiates configuration of the device and all other SLR components" [364]. The master SLR also has access to some dedicated FPGA circuitry, including the Xilinx Analog-to-Digital Converter (XADC) and unique identifiers, such as the Device DNA and User eFUSE [364]. Cloud FPGAs making use of partial reconfiguration reserve portions of the master SLR die (and of slave SLRs) for their shell interface, which abstracts away concrete physical implementation details such as I/O pinouts, DRAM controllers, and clock logic. This SLR layout could also provide a natural partitioning mechanism for multi-tenant cloud FPGAs, with the cloud provider reserving the master SLR, and different user designs being restricted to separate SLR dies. This would result in better physical isolation between the different users compared to existing proposals which split logic along (or even within) clock regions on the same die [158, 162, 163]. However, as shown in Chapter 6, physical isolation of user logic to dedicated SLR dies cannot prevent cross-SLR information leakage between tenants.

2.2.5 Ring Oscillators

Ring Oscillators (ROs) are a type of circuit which consists of an odd number of `NOT` gates, chained together in a ring formation, i.e., with the output of the last gate being fed back as input to the first gate. ROs form a loop, whose output at

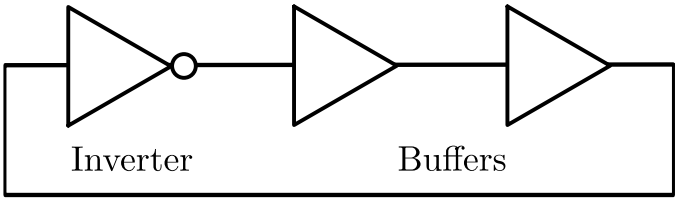


Figure 2.5: Ring Oscillator (RO) with one inverter and two buffer stages, implemented with Lookup Tables (LUTs), and used as a covert-channel receiver in the experiments of Chapter 5.

any stage oscillates between 1 and 0 (true and false). The frequency of oscillation depends on the number of stages in the RO, the delay between the stages, as well as Process, Voltage, and Temperature (PVT) variations in the environmental conditions and the manufacturing process [132]. The sensitivity of ROs to these variations makes them suitable for use as temperature monitors [41, 313, 389], True Random Number Generators (TRNGs) [165, 314, 344], and Physical Unclonable Functions (PUFs) [139, 154, 198, 199, 206, 312, 382].[2]

ROs can also be used to discover circuit-level modifications. For example, changes in RO frequencies can alert users to the addition of power measurement probes used in side-channel analysis [183]. Alternatively, they can also detect Hardware Trojans (HTs) in FPGAs and Application-Specific Integrated Circuits (ASICs): the dynamic activity of HTs results in local voltage drops. These drops affect ring oscillator frequencies, thus making HTs detectable without external equipment [155, 384], instead of needing physical measurements of power consumption [5] or EM emanations [28]. In addition, ROs respond well to chip aging, and can effectively detect recycled or counterfeit ICs [385]. For further information, the interested reader may wish to consult various surveys on supply-chain hardware security and related issues [20, 21, 152, 231, 269, 321, 381].

Figure 2.5 specifically depicts the RO design used in Chapter 5 to characterize long-wire leakage. It consists of one `NOT` gate (inverter) and two buffer stages, which are implemented using LUTs on Xilinx FPGAs. Due to their potential for abuse, these traditional LUT-based ROs (*combinatorial loops*) are prohibited by some cloud providers,

[2] PUFs can distinguish between otherwise identical devices, and are used in some cryptographic algorithms [138, 154, 196]. PUF constructions exploiting hardware imperfections also include the decay rate of DRAM [280, 371], its access latency [160, 236], and its response to the RowHammer effect [14].

such as AWS [12]. Nevertheless, as shown in this thesis (Chapter 6), alternative RO structures currently remain undetected, and can be used to achieve the same effects.

2.2.6 Remote Attacks

Due to the emergence of cloud FPGA computing platforms and the possibility of multi-tenant occupancy of the underlying hardware, recent research has investigated remote FPGA attacks, primarily using ROs as side- and covert-channel receivers and transmitters. For example, ROs have been shown to be effective heaters of the FPGA fabric [3, 332] and can also detect thermal changes caused by external sources [313], other concurrent users on the same FPGA [142], or the previous tenants of time-shared FPGAs [332]. ROs can also watermark circuits and protect IP cores [276], cause voltage fluctuations [252] and crash the FPGA [121], or inject faults into computations and extract cryptographic keys [167]. Additionally, as Section 3.3 discusses, ROs can bias TRNGs [197].

On the receiving end, ROs have also been used to infer the state of nearby long wires, first in Xilinx (Chapter 5) and later in Intel [253, 260] devices. ROs can also recover cryptographic keys in multi-tenant FPGAs, without requiring that routing resources be adjacent [387]. The attacker and victim circuits are physically separated in this case, but they are still located on the same single-die FPGA, unlike the experiments of Chapter 6 which reveal cross-SLR information leakage between distinct dies of the same IC.

It should be noted that other types of circuits can also cause faults through large switching activity [390], fast voltage transients [292], and concurrent write collisions in dual-port Random-Access Memory (RAM) [8]. Time-to-Digital Converters (TDCs) are also powerful side-channel receivers, and use tapped delay lines through chains of buffers and measurement latches. TDCs thus exploit the logic and routing delay dependence on FPGA voltage, and estimate it by monitoring how far signals have propagated.

Schellenberg et al., in particular, have used TDCs to recover Advanced Encryption Standard (AES) keys in multi-tenant [282] and cross-chip [283] setups. Unlike the attacks of Chapter 7, however, the chips used by Schellenberg et al. were located on the same FPGA board, and hence shared the same voltage regulator. This lack of additional intermediate components between the Power Distribution Networks (PDNs) of the two

FPGAs made information leakage easier. Moreover, the board on which experiments were conducted was explicitly "designed for external side-channel analysis research" [283]. By contrast, the cross-FPGA attacks of Chapter 7 use off-the-shelf Xilinx-designed boards, and work across the shared Power Supply Unit (PSU).

It should be noted that RO combinatorial loops and timing violations exploited by TDCs can be detected by bitstream-checking mechanisms [122, 168]. Although the proposed countermeasures do not necessarily identify the alternative RO designs of this thesis, some Design Rule Checks (DRCs) which can be used instead are proposed in Chapter 6.

2.3 Summary

This chapter provided the necessary background to understand and perform the attacks of the subsequent chapters of this thesis. Specifically, Section 2.1 discussed the prerequisites for the out-of-band signal injection attacks of Chapters 3 and 4. These include non-linearities in ADCs and microphones, the resonant behavior of MEMS devices, as well as unintentional antennas in the wires and PCB traces connecting sensors to microcontrollers. Section 2.2 then presented background details for Chapters 5–7. It summarized the FPGA design process locally and on the cloud, and expanded upon the layout and interconnect resources of FPGA devices. It also introduced the design for one of the RO types used in this thesis, and presented related work in remote FPGA attacks. Overall, this chapter showed that hardware details are often hidden away from designers, and that unintentional properties and imperfections are key for attacks on the confidentiality and integrity of data processed by embedded devices.

Chaos is merely order waiting to be deciphered.

— José de Sousa Saramago

3

Out-of-Band Signal Injection Taxonomy

Contents

This chapter is concerned with *out-of-band* signal injection attacks, which violate the integrity of sensor measurements by exploiting the hardware imperfections described in Section 2.1. As shown in Figure 3.1, these attacks can be performed using electromagnetic radiation targeting unintentional antennas [94, 286, 296], conducted signals through shared power lines [193], as well as optical [242, 375] and acoustic [42, 301, 380] emissions exploiting flaws in the conversion process from physical properties into electrical ones. The systems attacked have been equally diverse, and include medical devices [94], drones [301, 336], hard drives [42] and cameras [246], among many others. However, despite the wide range of attack methods and devices targeted, research in the

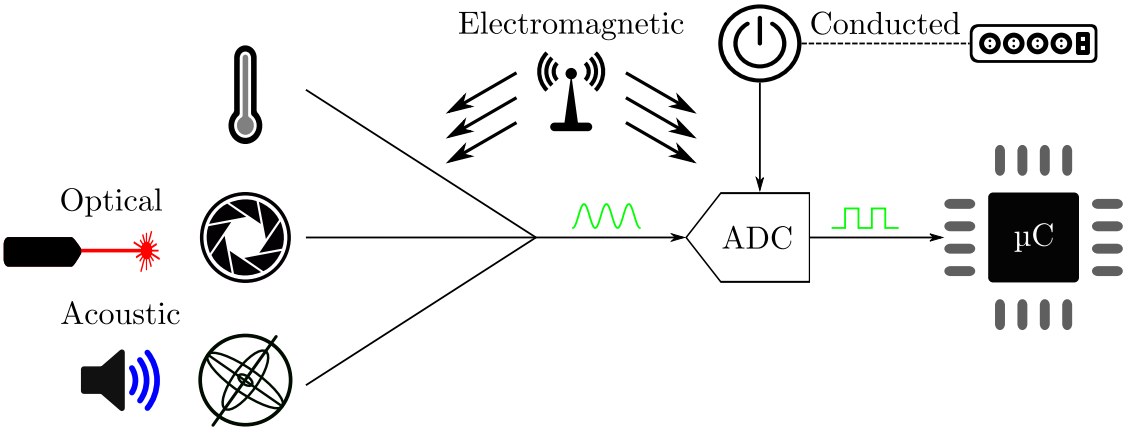


Figure 3.1: Out-of-band signal injection channels: remote and conducted adversarial electromagnetic emanations, optical emissions, and acoustic waves can attack the sensors themselves, or the interfaces connecting sensors to microcontrollers through Analog-to-Digital Converters (ADCs).

field thus far has been conducted in a disjointed manner.

Although out-of-band signal injection attacks have primarily only been (openly) conducted in a lab environment, they have garnered the interest of technological and mainstream news publications outside of the academic community [23, 95, 96, 328–330]. They have even prompted national agencies to issue Computer Emergency Readiness Team (CERT) advisories [70]. [3] Moreover, out-of-band signal injection attacks share the same potential for weaponization as mass-produced sonic repellents [233] or commercial [48] and military [331] jammers, despite differences in the techniques used. Finally, their effects can in some cases be fatal, for instance by tricking cardiac implantable electrical devices into causing pacing inhibition and defibrillation shocks [94]. As a result, to bring attention to these potentially severe issues, and to help designers better protect future hardware devices, this chapter:

1. Unifies the diverse terminology used by different works to create a common language through which to discuss attack and defense mechanisms (Section 3.1).

2. Highlights cross-influences between electromagnetic (Section 3.2), conducted (Section 3.3), acoustic (Section 3.4), and other (Section 3.5) attacks.

3. Proposes the first chronological and thematic evolution of out-of-band signal injection attacks (Section 3.6) and creates a taxonomy of countermeasures (Section 3.7).

[3] CERT can also stand for Cyber Emergency Response Team, depending on the agency.

4. Places attacks and defenses in the wider context of side-channel leakage and electromagnetic interference (Section 3.8).

5. Identifies gaps in the experimental approach of published research, and proposes concrete steps to overcome these challenges in the future (Section 3.9).

In summary (Section 3.10), this chapter unifies different types of out-of-band signal injection attacks and defenses, and provides insights into new research directions.

3.1 Choice of Terminology

Although many out-of-band signal injection attacks have been performed in the literature, the terms used to describe them have been inconsistent, or even non-existent [262, 349]. This section identifies and unifies the nomenclature, thereby providing a common language through which to compare different research. Indeed, as later sections demonstrate, the commonalities in the attack techniques highlight a need for an all-encompassing term irrespective of the method of injection. The term chosen for this unification can be defined as follows:

Definition 1 (Out-of-Band Signal Injection Attacks) *Out-of-band signal injection attacks are adversarial manipulations of interfaces not intended for communication involving sensors or actuators that cause a mismatch between the true physical property being measured or acted upon and its digitized version.*

The term *injection* was chosen because it captures the fact that values reported by a system are altered; it is not channel-specific; and it has already been adopted by different works [34, 50, 94, 116, 153, 193, 200, 201, 232, 237, 336, 337, 380]. The *out-of-band* qualifier is necessary to distinguish the attacks studied in this chapter from signal injection attacks on sensors using pulse reflections such as Light Detection and Rangings (LiDARs), radars, and sonars [61, 246, 294, 372, 375], signal injection attacks on the physical layer of communication protocols [150], and false data injection attacks [179, 191]. Such attacks do not depend on hardware imperfections, but instead use external communication interfaces, and are therefore not out-of-band. In other words, because digital interfaces

can be easily protected with cryptography, spoofing attacks of unauthenticated, digital communication interfaces [133, 189, 270, 379] are not studied in this chapter.

By contrast, the term "out-of-band signal injection" captures attacks which target interfaces using signals outside of their intended frequency of operation. It includes ultraviolet or infrared light against cameras which should only be recording the visible part of the spectrum and ultrasonic injections against microphones meant to be recording only audible sounds: these attacks transmit signals that are literally outside the operational band, and have a secondary goal of undetectability or *concealment* [297]. The term also encompasses electromagnetic signals against systems without any (intentional) antennas, and acoustic attacks against gyroscopes and accelerometers: these are also out-of-band, since they inject signals through channels other than the ones used by the sensor to measure the physical property. The use of the out-of-band modifier is therefore consistent with the definition for out-of-band covert communication [57]. It has also recently been used by Tu et al. [337] to describe acoustic attacks on inertial sensors, further motivating its choice in this thesis.

It should be noted that earlier work [294, 295] has proposed a subdivision of signal injections attacks into: (a) *regular-channel attacks*, which target the sensor structure itself by "using the same type of physical quantity sensed"; (b) *transmission-channel attacks*, which target the connection between the sensor output and the measurement setup; and (c) *side-channel attacks*, where the sensors themselves are targeted, but "by physical stimuli other than those they are supposed to sense". This thesis does not follow this categorization, as the various subdivisions generally correspond to the medium of injection (optical, electromagnetic, and acoustic respectively). Moreover, regular-channel attacks are usually in-band, while side-channel attacks have an overloaded meaning.

The term *(intentional) interference* [33, 34, 42, 94, 153, 193, 237, 301] is similarly unsuitable because: (a) it does not make it clear that the attackers can in some cases inject waveforms of their choosing; and (b) Intentional Electromagnetic Interference (IEMI) has an established meaning in Electromagnetic Compatibility (EMC) literature [256, 278]. IEMI attacks often use high-power, destructive transmissions (Section 3.8), and therefore have a different aim than out-of-band signal injection attacks.

Terminology	Example References
Injection	[94, 116, 153, 200, 232, 237, 336, 337]
Intentional Interference	[33, 34, 42, 94, 153, 193, 237, 301]
Non-Linearity	[271, 272, 374, 380]
Spoofing	[242, 246, 337, 375]
Other (See Text)	[98, 286, 289]

Table 3.1: Terminology used by different works to describe out-of-band signal injection attacks.

The term *(sensor) spoofing* [242, 246, 337, 375] was also avoided for similar reasons: it has an overloaded meaning in authentication contexts and with in-band signal injection attacks [71, 296]. Moreover, it does not capture the physical aspect of injections, and does not accurately describe coarse-grained attacks which lead to saturation of a sensor.

Other terms used have been specific to the particular channel which is being exploited, including *induction attacks* [286], *acoustic resonance* [289], and *(acoustic) non-linearity* [271, 272, 380]. Such terms were not selected because they are channel-specific, and focus on the mechanism of the attack, rather than the effect. Similarly, methodology-inspired terms which have been avoided include Radio Frequency Injection (RFI), Direct Power Injection (DPI), and other terminology that arises in immunity or susceptibility literature against (non-adversarial) Electromagnetic Interference (EMI) [24, 25, 44–46, 100, 101, 107, 156, 250, 315].

Finally, the term *transduction attacks*, proposed by Fu and Xu [98] to mean attacks which "exploit a vulnerability in the physics of a sensor to manipulate its output or induce intentional errors" has not yet received mainstream recognition. It also does not necessarily make it clear that the attack may target the interface between the sensor and the rest of the system, instead of just the sensor itself. The various terms which have been used to describe out-of-band signal injection attacks are shown in Table 3.1, along with example references. It should be noted that since the majority of the attacks in the literature are on sensors rather than actuators, the term sensor is used collectively in this chapter for brevity. In other words, this chapter distinguishes between the two only when it is necessary to do so, i.e., when there is a divergence in the attack methodology.

In summary, out-of-band signal injection attacks have four key features that differentiate them from related research. The **first** such property is that they aim to *change*

values processed by a system, rather than infer them, distinguishing them from side-channel [307] and fault-injection attacks [30, 151, 378]. The **second** feature is that out-of-band attacks do not change the measured quantity itself: in the language of Shoukry et al. [297], the property measured itself is *trusted*, although the measurement itself is not. For instance, using electromagnetic signals to change the audio recorded by a microphone [94] is an example of an out-of-band signal injection attack, but heating a temperature sensor with an open flame is not. The **third** property is that attacks are *non-invasive* [297], therefore precluding direct physical access to the system under attack. The **final** characteristic highlights the physical aspect of out-of-band signal injection attacks. In other words, the attacks studied in this chapter alter sensor measurements or actuator inputs at the hardware layer instead of the protocol layer. However, since related research areas can offer invaluable insight into novel injection techniques and possible countermeasures, cross-disciplinary connections are made throughout this chapter, but with a clear focus on how they impact out-of-band signal injection attacks and defenses.

3.2 Electromagnetic Transmissions

As Section 2.1.2 highlighted, the effects of unintentional antennas have been central in the security and EMC communities. Until now, unintentional *transmitting* antennas have been key for side-channel analysis: data-dependent emissions can reveal the information processed by a device to a remote attacker [164]. However, wires conversely acting as unintentional *receiving* antennas have only become a focal point of research more recently: out-of-band signal injection attacks have demonstrated that the antenna-like behavior of wires between sensors and microcontrollers can result in adversarial electromagnetic (EM) signals being interpreted as legitimate measurements.

Implantable Medical Devices (IMDs) are an example of a safety-critical system where external EMI can cause physical harm to people. As a result, there is extensive research on the EMI behavior of various IMDs [29, 51, 65, 68, 84, 137, 146, 186, 216, 217, 247, 248, 285]. Some of these works have even pinpointed the properties of "non-linear circuit elements" in pacemakers as the culprits for demodulating Radio Frequency (RF) signals produced by cell phones [29, 59]. However, the consequences of intentional

electromagnetic out-of-band signal injection attacks on IMDs were only first identified in 2009 by Rasmussen et al. in the context of a distance-bounding protocol [262].

The proposed protocol used ultrasound transmissions to place guarantees on the distance between two communicating parties. However, it was determined that an EM signal could "induce a current in the audio receiver circuit just as if the IMD received a sound signal" [262]. This would break protocol properties which depend on the speed of sound constant: adversarial transmissions propagating at the speed of light allow an adversary to operate from a longer distance. This attack is perhaps the first electromagnetic out-of-band signal injection, using unintentional antennas on the path "from the reception circuit to the piezo element" to attack an ultrasound protocol [262].

Although the attack by Rasmussen et al. was more of a side-note to an otherwise-secure protocol [262], Foo Kune et al.'s seminal 2013 "Ghost Talk" paper [94] specifically focused on such adversarial injections. It showed that EM emissions could affect electrocardiogram (ECG) measurements and cause IMDs to deliver defibrillation shocks [94]. Foo Kune et al. succeeded in injecting arbitrary analog measurements, making a marked improvement in the literature compared to coarse replay and jamming attacks on IMDs [124, 133, 189, 273]. Their approach also significantly differed from high-power IEMI leading to the transient upset or destruction of commercial equipment [47, 53, 76, 117, 140, 229, 230, 239, 256, 275].

The attack on IMDs by Foo Kune et al. [94] used low-power ($\leq 10\,\mathrm{W}$), low-frequency (kHz range) EM signals which coupled to the leads of IMDs and cardiac implantable electrical devices. In the open air, the distance achieved was up to 1.67 m, but when submerging the devices in saline (to approximate the composition of the human body), successful attacks were limited to less than 10 cm. The signals emitted targeted the baseband (i.e., the frequency of operation) of the IMDs directly, and thus did not make use of the non-linearities discussed in Section 2.1.1. However, this attack should still be considered out-of-band, as the IMD leads were meant to require physical contact for measurements, and should not be reacting to remote electromagnetic transmissions. In other words, this attack is more akin to coupling to a multimeter's probes to cause wrong voltage readings remotely, rather than causing interference to a Wi-Fi device by

transmitting at 2.4 GHz: although both require external stimuli, the mode of injection is different from the intended mode of operation.

Foo Kune et al. also conducted out-of-band attacks against webcams and Bluetooth headsets that were up to 1 m away from an 80 mW source [94]. The authors transmitted modulated signals over high-frequency carriers (in the hundreds of MHz) that make use of unintentional antennas on the path between the microphone and the amplifier. Non-linearities then demodulated the input signals and produced intelligible audio output. This output overpowered legitimate conversations via the headset and fooled automated dial-in services by emulating key presses via modulated Dual-Tone Multi-Frequency (DTMF) signals. Music identification services also correctly identified the transmitted song, despite it being modulated on a high-frequency carrier [94].

Kasmi and Lopes Esteves similarly targeted smartphone microphones, but with a goal of triggering voice commands (e.g., "OK Google", "Hey Siri") by emitting Amplitude Modulation (AM) signals [153]. These signals get picked up by the user's hands-free headset, are then demodulated due to non-linearities, and finally get executed by the software voice-processing service. It is interesting to note that, by default, "a long hardware button press is required for launching the service" [153]. However, a Frequency Modulation (FM) signal at the same frequency can also emulate this headphone button press [153], allowing the attack to be fully carried out remotely.

The attack by Kasmi and Lopes Esteves used *front-door coupling*, as the "radiation couples to equipment intended to communicate or interact with the external environment" [53]. This is because headphones can be used as FM antennas and can thus not be effectively shielded. It should be noted that the field strength required was in the order of $25 - 30\,\mathrm{V/m}$, which is close to the limit for human safety, and an order of magnitude higher than the required immunity level of $3\,\mathrm{V/m}$ [153]. This illustrates that high powers might still be required for reasonable attack distances: in a subsequent work, the authors noted that their attack requires a power of 40 W for a distance of 2 m, and 200 W for a distance of 4 m [193].

Figure 3.2 shows an example of an electromagnetic out-of-band signal injection, which summarizes the attacks against microphone sensors. It shows that the desired adversarial

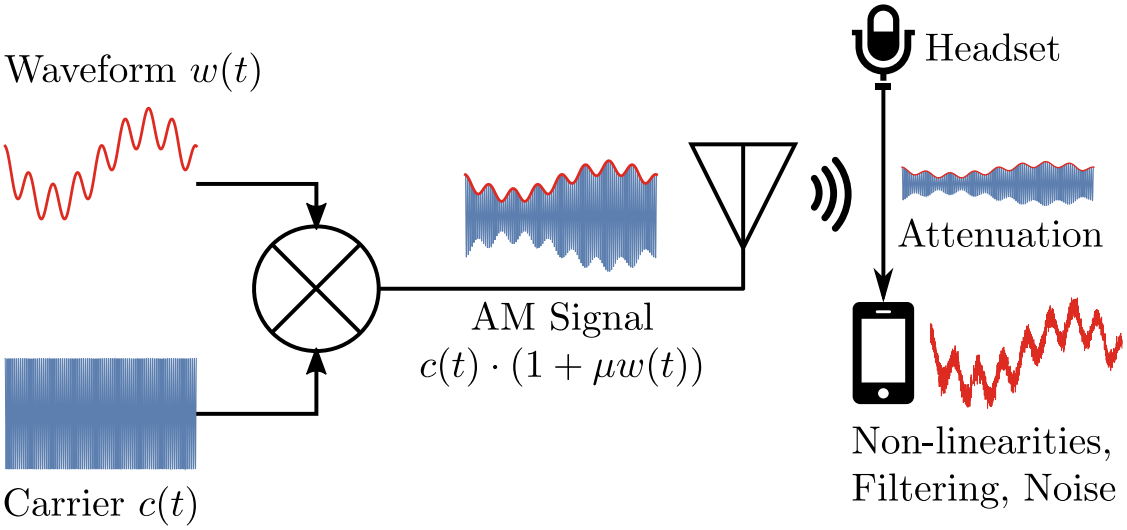


Figure 3.2: Basic operating principle of an electromagnetic out-of-band signal injection attack against a microphone. Amplitude Modulation (AM) signals are transmitted using an antenna, and are picked up (and attenuated) by headphone cables or Printed Circuit Board (PCB) traces. Non-linearities in amplifiers coupled with low-pass filters remove the carrier wave and down-convert the target signal. The demodulated signal can be used to break ultrasound protocols [262], fool music services [94], or inject voice commands [153], despite the additional noise introduced.

waveform $w(t)$ needs to be modulated over a high-frequency carrier $c(t)$, so that the signal can be picked up with relatively low attenuation by the victim device's wires. For example, amplitude modulation with a modulation depth $0 < \mu \leq 1$ can be used to couple to wired headphones. Through low-pass filtering effects and non-linearities in the phone's internal amplifier, the target waveform is preserved at the output of the Digital Signal Processing (DSP) chip. Although this process introduces noise, the demodulated signal can still be distinguished by online music services [94], imitate voice-initiated commands which are then executed by the phone [153], or break protocol guarantees [262]. Although the above works targeted microphones, electromagnetic out-of-band signal injections are not sensor- or device-specific. For example, subsequent sections discuss proof-of-concept EM-based injections against temperature sensors [338, 386]. However, most attacks have primarily used amplitude modulation, leading to a question that has yet not been addressed:

Open Question 1: What is the optimal modulation scheme for electromagnetic out-of-band signal injection attacks? Can Frequency Modulation (FM), Phase Modulation (PM), or other schemes replace Amplitude Modulation (AM)?

Magnetic emissions had remained largely-unexplored until Shoukry et al. used them to

confuse an Anti-Lock Braking System (ABS) sensor [296]. This was done by exposing the magnetic-based wheel speed sensors to an adversarial magnetic field at close proximity, resulting in altered speed measurements, potentially veering cars off the road [296]. Although the work by Shoukry et al. is in-band and only operates at a short distance, it inspired subsequent work in out-of-band attacks: Selvaraj et al. recently conducted the first attacks on actuators (rather than sensors) through EM transmissions [286]. Specifically, an unmodulated sawtooth waveform was chosen to cause a "sharp decrease, for a very small amount of time" at the target servo [286]. Because the servo is controlled using Pulse Width Modulation (PWM), a waveform of the same frequency (50 Hz) results in a one-way (clockwise) rotation.

This attack has a few limitations: changing the attacking frequency to 60 Hz causes the servo to "change positions randomly" [286]. Moreover, relatively high powers are required: a 10 V (peak-to-peak) waveform is insufficient, so a 50 W amplifier and a 1-to-6 step-up transformer are necessary. Moreover, one of the servo wires is wrapped around the toroid transferring the EM signal. Although "the same effect was observed when a length of the wire was placed within a solenoid", "producing an effect at a distance requires the proper selection of a field directivity element" [286]:

Open Question 2: How can one control actuators in both directions, with higher precision, and from a large distance?

Selvaraj et al. additionally proposed an analytical model of electromagnetic induction attacks for sensors and actuators, with a focus on the magnetic rather than the electric field [286]. To support their model, they further conducted experiments against General Purpose Input/Output (GPIO) pins of microcontrollers in analog and digital modes. They showed that 1.82 W transmissions of unmodulated signals at frequencies between $0 - 1$ GHz can result in a Direct Current (DC) offset, even when the microcontroller is at a distance of up to 1 m from the source. This indicates that an adversary can successfully inject signals over a wide range of frequencies, without having precisely determined the resonance behavior of the system.

Selvaraj et al. [286] were only concerned with the average power received and not time-dependent signals. This is in contrast to work on the demodulating effects of

amplifiers (Section 3.3), which depends on inter-modulation products and harmonics. Instead, Electrostatic Discharge (ESD) diodes were identified as the culprits for the resulting DC offset, due to their clipping non-linearities. However, it is not clear whether the same methodology can induce attacker-desired, time-varying waveforms through modulated (in amplitude or otherwise) transmissions: in the language of Chapter 4, Selvaraj et al. performed an *existential injection* which disturbs the Analog-to-Digital Converter (ADC) readings, but not a *selective injection* of attacker-chosen waveforms, unlike the earlier work of Foo Kune et al. [94].

As a final point of note, researchers have identified that coupling into the wiring interconnects within Integrated Circuits (ICs) is possible [177]. However, this disturbance "can be neglected up to several gigahertz", since ICs are "usually smaller than a few centimeters" [177]. This leads to another research question:

Open Question 3: Is it possible to conduct electromagnetic out-of-band signal injection attacks into digital Integrated Circuits (ICs) which unify sensors, Analog-to-Digital Converters (ADCs), and microcontrollers in one package?

Although this question has been answered in the affirmative for acoustic attacks (Section 3.4), it remains open for electromagnetic ones.

3.3 Conducted Signals

A different class of out-of-band signal injection attacks requires an indirect physical connection between the attacker and the victim, such as a shared power line. Unlike their radiated counterparts of Section 3.2, these *conducted* attacks do not require that signals be picked up by unintentional receiving antennas on the path between sensors and microcontrollers. Instead, signals are propagated along conductors primarily on the powering circuit, which can transfer through crosstalk or coupling to paths containing non-linearities. Much like electromagnetic attacks, conducted out-of-band signal injection attacks have also been primarily experimental in nature. Their methodology often follows that of susceptibility literature, which predicts a device's response to high-frequency radio signals. Systems tested include microcontrollers, ADCs, and other embedded

devices which contain Input/Output (I/O) and power pins. The goal of such research is to quantify immunity to radiated and conducted EM disturbances, and is typically concerned with the average power received by the embedded system, similar to the work by Selvaraj et al. [286] summarized in Section 3.2.

To avoid legal and practical considerations related to electromagnetic transmissions, the experimental approach often followed is known as Direct Power Injection (DPI). It consists of injecting harmonic disturbances from a few kHz to a couple of GHz and measuring the relationship between forward power and frequency. Multiple works have shown that as the frequency of the input increases, immunity to DPI also increases [24, 25, 44, 101, 156, 176].[4] In other words, higher frequencies generally require higher forward power injections for the same level of susceptibility. This mirrors the findings of the (remote) injections by Selvaraj et al. [286], discussed in Section 3.2.

A similar methodology can be applied to evaluate the demodulation characteristics of amplifiers and transistors [100, 107, 250, 315], and therefore better predict the fidelity with which attackers can inject target waveforms, both in the conducted and in the radiated settings. This is done in Chapter 4 and Appendix A for six ADCs, with a focus on how to exploit this demodulating effect for out-of-band signal injection attacks. It is shown that ADCs of three different types from four manufacturers, and with different resolutions and sampling frequencies can all demodulate AM waveforms. Moreover, the fundamental frequency persists along with its harmonics and some high-frequency components, even for carriers which are multiple times the ADCs' sampling and cutoff frequencies. As Appendix A shows, attacks can also be performed remotely, without following DPI methodology: a 10 dBm (10 mW) transmission can be demodulated by a receiver amplifier at small distances (5 cm).

In their "Trick or Heat" work, Tu et al. [338] also conducted DPI experiments on operational amplifiers, but with a view on how to exploit rectification effects for out-of-band signal injection attacks on temperature sensors. They, too, determined that, as the frequency increases, the magnitude of the Alternating Current (AC) voltage decreases, while "EMI signals at specific frequencies induce a significant DC offset" [338].

[4] The survey by Ramdani et al. [259] contains more information on RF immunity models.

Moreover, for a given frequency of injection, power and the induced DC offset are "locally proportional", though the rate of change "gradually decreases as the power of injected EMI signals grows" [338]. However, power and DC offset are not always positively correlated, even for remote transmissions: for some frequencies, the induced DC offset is negative [338].

Having characterized the behavior of individual amplifiers, Tu et al. turned their attention to different types of thermal sensors, including Negative Temperature Coefficient (NTC) thermistors, shielded and unshielded K-type thermocouples, and Resistance Temperature Detectors (RTDs). With a 35 dBm (3.2 W) electromagnetic source, Tu et al. succeeded in changing the reported temperature of various devices by at least 0.5 °C [338]. The systems attacked included, among others, newborn incubators, soldering irons, and 3D printers, which were placed at distances of up to 6 m.

In some experiments, a thick wall was present between the transmitting device and an infant incubator under attack. It was shown that, even in this setup, an adversary can increase the measured skin temperature by 3.4 °C or decrease it by 4.5 °C [338], again demonstrating the potentially fatal consequences of out-of-band attacks. Most of the attacks by Tu et al. used unmodulated transmissions, and therefore only looked at the relationship between frequency and DC offset, or the relationship between power and DC offset. However, when investigated jointly, amplitude modulation was capable of causing selective injections (Chapter 4). In other words, Tu et al. [338] spelled "HI" in the output of the temperature sensor by appropriately modulating the amplitude of the transmission.

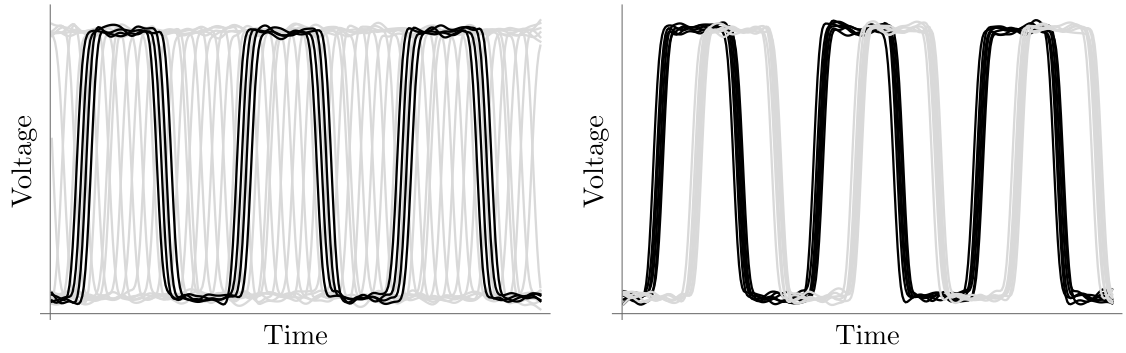
In a different strand of research, Lopes Esteves and Kasmi demonstrated how to inject voice commands ("OK Google") into a smartphone through conducted means [193]. Specifically, the attack exploited the fact that on the device's circuit board, the phone's Universal Serial Bus (USB) charging port is physically close to the audio frontend, where demodulation (envelope detection) can take place due to non-linearities [193]. As a result, back-door coupling occurs, either due to "a re-radiation of the interference from the USB circuitry bypassing the physical isolation by parasitic coupling (crosstalk) or the possible sharing of the V_{cc} and GND networks on the Printed Circuit Board (PCB)" [193].

Open Question 4: What properties of the power circuit and related layout considerations make systems vulnerable to conducted out-of-band signal injections?

The methodology used was inspired by experiments on the propagation of conducted disturbances and on EM injections into power cables. Specifically, amplitude-modulated signals were injected at various locations of the power network, i.e., at different plug points on the same strip and on extension cords. The phone was charging on a computer USB port or through a wall adapter. Experiments were repeated both with a magnetic injection probe (directly coupling to cables), and a "custom coupler made with capacitors, resistors and a high-frequency transformer" [193]. In all cases, it was determined that the smartphone can demodulate (and execute) commands carried on the 200 – 250 MHz range at distances of up to 10 m, even with only a 0.5 W source. Such conducted attacks therefore significantly lower the power requirements and increase the injection distance compared to the same authors' remote EM attack on smartphones [153].

True Random Number Generators (TRNGs) which are based on Ring Oscillators (ROs) are also vulnerable to conducted signal injection attacks: due to the dependency of the RO frequency on voltage, a suitable signal can lead to frequency locking of the oscillators [2, 207], removing the randomness in jitter differences. Markettos and Moore first conducted the attack in practice in 2009 by directly injecting 24 MHz signals into the power supply of two ring oscillators composed of discrete logic chips [200]. Moreover, they succeeded in biasing TRNGs even in secure microcontrollers and smartcards: a sinusoidal wave of 1 V peak-to-peak (2.5 mW) at 24.04 MHz was enough to cause a 5 V smartcard to fail statistical tests of randomness [200].

Time-varying signals are not always necessary: a constant (DC) power supply voltage can also lead to locking of ring oscillators in an under-volted Field-Programmable Gate Array (FPGA). This is because there is a "dependence of the frequency of one oscillator on the current peaks caused by rising and falling edges of the second oscillator" [40]. Figure 3.3 illustrates what happens when two ring oscillators frequencies lock: during normal operation (Figure 3.3a), the ring oscillator values "slide past each other, minimising the likelihood of two rings transitioning together" [200]. However, when a



(a) Unlocked **(b)** Locked

Figure 3.3: Example waveforms for two Ring Oscillators (ROs) with frequency locking (a) absent or (b) present.

frequency- or voltage-based attack causes the ROs to lock (Figure 3.3b), their relationship becomes predictable, biasing the TRNG.

It should be noted that although under- or over-volting the FPGA is usually considered a fault attack, in this case the ring oscillators are still functioning properly, but the entropy of the TRNG is reduced due to less jitter present [55, 201]. An interesting new class of such remote under-voltage attacks on TRNGs has recently surfaced. Because ring oscillators have the potential to increase the delay of FPGA elements by causing voltage drops, they can also cause timing violations, thereby reducing the randomness of TRNGs [197]. Such an attack does not require equipment for physical injections. Instead, the adversary only needs co-located (but logically and physically isolated) circuits on the same FPGA as the target TRNG. This setup reflects multi-tenant cloud designs [197], and presents new challenges for the protection of shared FPGAs against software-only attacks without physical access, such as those performed in Chapters 5 and 6.

Although the above attacks generally alter the power supply directly, the same outcome can be achieved through EM emanations targeting the wires connecting the various stages of the ring oscillators [33, 34, 50]. This requires micro-probes at very close proximity to the ring oscillators (in the order of 100 µm from the FPGA packaging), so as to localize the effects of the injection [33, 34]. However, TRNGs are also vulnerable against EM injections into power supply cables: Osuka et al. demonstrated that an injection probe wrapped around the DC power supply cable of a TRNG can also bias its randomness [237].

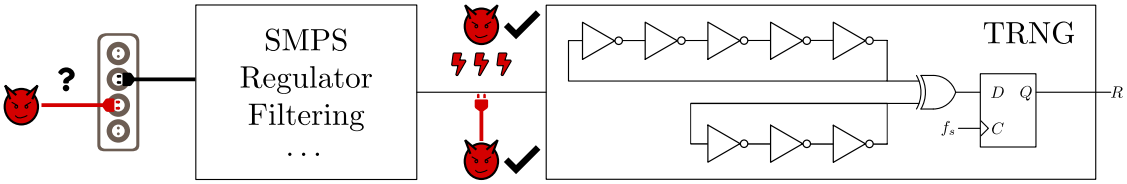


Figure 3.4: True Random Number Generators (TRNGs) based on Ring Oscillators (ROs) are vulnerable to frequency locking: electromagnetic and conducted signals into power supply cables can bias the randomness. Are attacks through a shared mains power supply network possible?

Although the design only used two ring oscillators composed of discrete logic chips,[5] injections were successful even when the probe was placed at a distance of 40 cm from the ROs, with a power of only 25.2 dBm (0.33 W).

It is worth highlighting that although the conducted voice command injection attack by Lopes Esteves and Kasmi was performed over shared power lines [193], all existing attacks on TRNGs bypass AC-to-DC rectification and voltage regulation. This leads to the following question for future research:

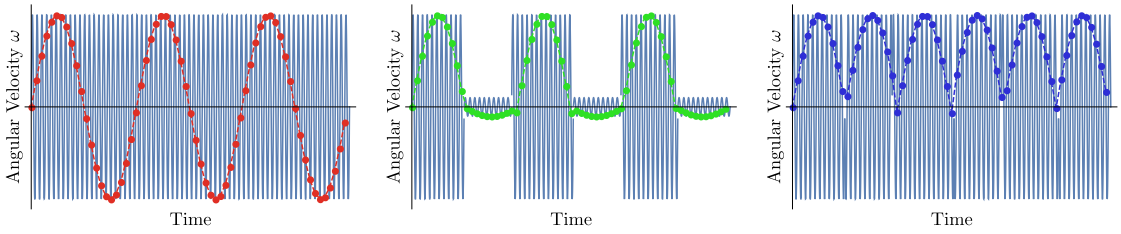
> **Open Question 5:** Is it possible to bias True Random Number Generators (TRNGs) through conducted out-of-band signal injection attacks on the primary side of power supplies (mains voltage), as shown in Figure 3.4?

3.4 Acoustic Emanations

Research into acoustic out-of-band signal injection attacks has primarily focused on: (a) attacking electro-mechanical devices by causing vibrations at their resonant frequencies; and (b) exploiting microphone non-linearities for inaudible voice commands. Both types of vulnerabilities were briefly introduced in Section 2.1.3. In the former category, Micro-Electro-Mechanical System (MEMS) gyroscopes and accelerometers have been a popular target for acoustic resonance attacks.

Early research into the properties of MEMS gyroscopes had shown that high-power acoustic noise at or near the resonant frequency can degrade the performance of the sensor [58, 74, 75]. However, the security effects of intentional sound transmissions were not explored until the 2015 "Rocking Drones" paper by Son et al. [301]. Initially, the effect

[5] Much like the original work by Markettos and Moore [200]. However, Bayon et al. [33, 34] targeted a more realistic TRNG composed of 50 ROs.



(a) Denial-of-Service **(b)** Side-Swing **(c)** Switching

Figure 3.5: Different acoustic injection approaches against gyroscopes. For (a) Denial-of-Service (DoS) attacks, a single-tone transmission at the gyroscope's resonant frequency suffices. This results in oscillating digital measurements of angular velocity, and can destabilize equipment [301, 337]. To remove the negative components, one can either (b) decrease the transmission amplitude in a *Side-Swing* attack [337], or (c) change the frequency of transmission for a *Switching* attack [337]. The cumulative effects of these approaches are shown in Figure 3.6.

was a simple Denial-of-Service (DoS) attack on drones. It was caused by the transmission of single-tone sound waves at the resonant frequency of drones' gyroscopes, so there was no control over their movements. The distance was also short, at 10 cm using a speaker producing a Sound Pressure Level (SPL) of up to 113 dB at the target frequencies.

Proof-of-concept control was then first demonstrated in a Black Hat conference presentation against gyroscopes in Virtual Reality (VR) headsets and self-balancing vehicles [349]. Tu et al.'s "Injected and Delivered" paper later became the first academic work to control gyroscopes in a much more fine-grained fashion [337]. This research allowed control for long periods of time (up to the minute range) and at large distances (up to 7.8 m with a maximum SPL of 135 dB) [337].

Tu et al. noticed that single-tone frequencies (like the ones used by Son et al. [301]) result in an oscillating discrete (digitized) output, which destabilizes equipment. In other words, a simple transmission at the resonant frequency is a type of DoS attack because the angular velocity (as measured by the gyroscope) fluctuates between positive and negative values (Figure 3.5a). However, it is possible to remove these negative components by decreasing the transmission amplitude during the corresponding measurements. This is called a *Side-Swing* attack, which "proportionally [amplifies] the induced output in the target direction and attenuate[s] the output in the opposite direction" [337] (Figure 3.5b).

Instead of attenuating signals during half of the transmission period, one can also control "the induced output by manipulating the phase of the digital signal with repetitive

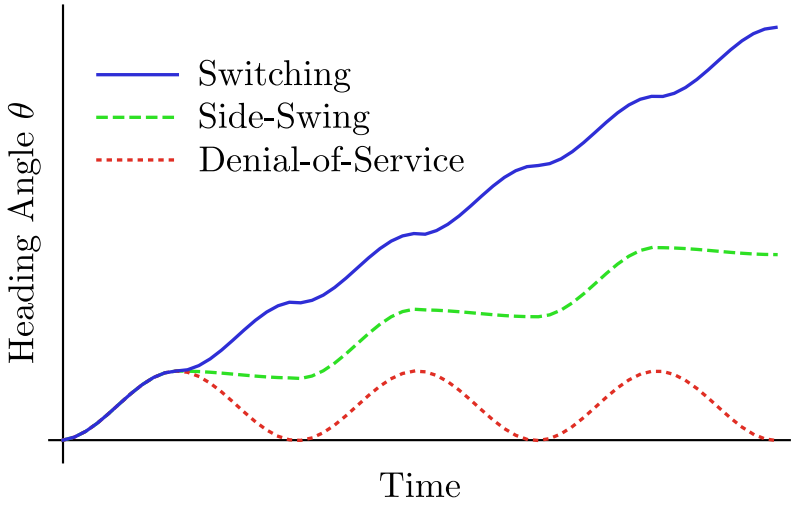


Figure 3.6: Effects of the three attacks of Figure 3.5. In a Denial-of-Service (DoS) attack, the accumulating heading angle fluctuates, while it continues increasing for both attacks proposed by Tu et al. [337]. In Switching attacks, the angle increases at twice the rate of Side-Swing attacks.

phase pacing" in a *Switching* attack [337] (Figure 3.5c). As this is accomplished in practice by changing the tonal frequency instead of attenuating the amplitude, a Switching attack contributes twice as much to the overall change in direction as a Side-Swing attack. This is shown by looking at the accumulating heading angle in Figure 3.6. It should be noted that by accounting for drifts in the sampling rate of the ADC (which are amplified during adversarial injections [337]), both attacks can control the gyroscopic output for longer periods of time.

In response to the rising interest in acoustic vulnerabilities, Khazaaleh et al. [159] created a mathematical model to explain the resonance response of gyroscopes. They showed that "the misalignment between the sensing and driving axes of the gyroscope is the main culprit behind the vulnerability of the gyroscope to ultrasonic attacks" [159]. More precisely, because "the sensing direction is not exactly orthogonal to the driving direction, some of the energy gets coupled to the sensing direction" [159]. This causes a false reading, which is typically corrected "by employing a demodulator in the readout circuit" [159]. When the transmission frequency is slightly different from the sensing frequency, the gyroscope generates "measurable output", whose frequency equals "the difference between the driving frequency and the frequency of the acoustic signal" [159]. As shown experimentally, this model also explains why it is better to transmit near

the resonant frequency rather than exactly at it [159]. It also suggests that low-pass filters or differential measurements through additional proof masses are ineffective countermeasures against acoustic out-of-band signal injection attacks [159].

Although Tu et al. succeeded in controlling gyroscopes in phones, scooters, stabilizers, screwdrivers, and VR headsets among others, only DoS attacks were successful against accelerometers [337]. According to Trippel et al., insecure amplifiers and Low-Pass Filters (LPFs) prior to the accelerometer ADCs can demodulate both AM and PM attacker injections [336]. These insecurities are the results of clipping non-linearities and permissive filtering respectively, and allow for both biasing and control attacks.

Trippel et al. used accelerometers to spell words, naming their work "WALNUT" for the output of the spoofed sensor measurements. They were also able to control off-the-shelf devices, such as remote-controlled cars and Fitbit fitness tracking wristbands [336]. Although spoofing step counts might seem innocuous, companies often offer financial rewards for health-related activity [336], so cheating devices (which do not yet exploit out-of-band effects) are already being sold [299]. Most attacks by Trippel et al. [336] were performed at distances of 10 cm, with a speaker producing an SPL of 110 dB. The duration of control over the output of the MEMS sensors was often limited to $1-2$ s (but up to 30 s) due to sampling rate drifts. Moreover, it was shown that the three axes do not behave identically under acoustic injections. For example, there are some MEMS devices for which only the x-axis responds to acoustic transmissions, while others are vulnerable in all three axes, but at different resonant frequencies.

Although Trippel et al. [336] attacked a single sensor in one direction at a time, Nashimoto et al.'s "Sensor CON-Fusion" investigated whether *sensor fusion* using a Kalman Filter can improve the robustness of measurements [232]. It was shown that "while sensor fusion introduces a certain degree of attack resilience, it remains susceptible" to combined acoustic and electromagnetic injections [232]. Specifically, Nashimoto et al. succeeded in simultaneously controlling the roll, pitch, and yaw (the three angular axes in aircraft nomenclature) by fusing the outputs of an accelerometer, a gyroscope, and magnetometer. However, in non-simulated environments, "there is an error in the roll angle", and "the resulting inclination does not last long" [232]. Although fusion is

further explored in the context of defense mechanisms (Section 3.7), the above discussion leads to the following research question:

> **Open Question 6:** Is it possible to use acoustic injections to precisely control Micro-Electro-Mechanical System (MEMS) gyroscope and accelerometer measurements in all three directions simultaneously and/or for longer periods of time?

In a different strand of research, ten years after a video demonstrating that shouting causes unusually high disk I/O latency [127], Shahrad et al. showed that acoustic transmissions can cause vibrations in Hard-Disk Drives (HDDs) [289]. These vibrations result in read and write errors at distances up to 70 cm with a sound level of 102.6 dBA [289]. As a result, they can make systems unresponsive, or even crash them [289].

Although Shahrad et al. primarily focused on the effect of the angle of transmission [289], research conducted in parallel more precisely pinpointed the root cause of the issue using Finite Element Analysis [42]. Specifically, it was shown that (audible) acoustic waves "can displace a read/write head or disk platter outside of operational bounds, inducing throughput loss"—even for displacements of only a few nanometers [42]. In addition, in their "Blue Note" paper, Bolton et al. also used ultrasonic transmissions to attack the shock sensors which are meant to protect HDDs during sudden drops by "parking the read/write head" [42]: modern HDDs contain "piezo shock sensors or MEMS capacitive accelerometers" to "detect sudden disturbances" [42], and can also be attacked through ultrasound transmissions at their resonant frequencies. Through these malicious acoustic attacks, SPL levels of up to 130 dB cause HDDs at distances of 10 cm to become unresponsive, thus disabling laptops and video recorders [42].

Not all acoustic attacks transmit at resonant frequencies. By contrast, other research targets microphone non-linearities to cause inaudible sound to be recorded. Early work on acoustic attacks primarily focused on adversarial control of machine learning in speech recognition systems [56, 340]. Such research did not take advantage of non-linearities and was in-band, as the transmissions were audible (although indecipherable by people). However, later investigations revolved around ensuring that the transmitted frequencies are beyond the human-audible range (20 kHz). This was first accomplished by Zhang

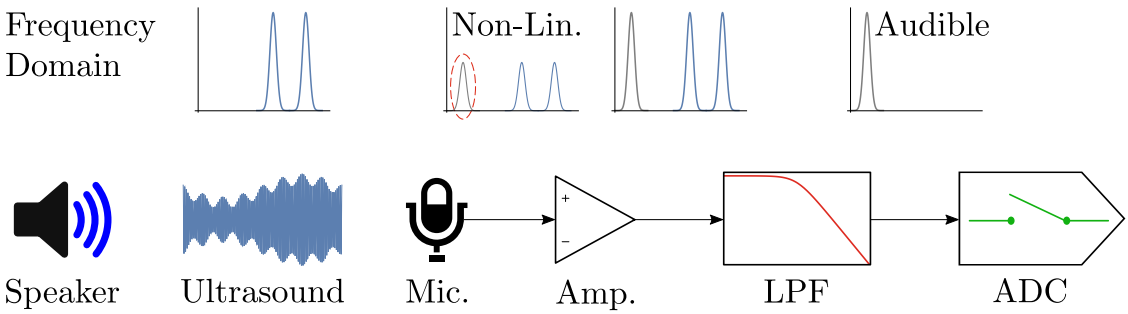


Figure 3.7: High-level overview of ultrasonic attacks against microphones [272, 303, 376, 380]. A speaker transmits inaudible tones, but intermodulation products are created due to non-linearities in the microphone and amplifier. A Low-Pass Filter (LPF) removes ultrasound frequencies, so the Analog-to-Digital Converter (ADC) records only audible byproducts.

et al. in their "DolphinAttack" paper, which exploited non-linearities in microphone sensors [380], as shown in Figure 3.7.

The work by Zhang et al. showed how to transform (modulated) ultrasound transmissions into valid commands which were executed by speech recognition systems such as Apple Siri, Google Now, Microsoft Cortana, and Amazon Alexa [380]. The same authors later expanded their attack by testing different setups, and increasing the attack distance from 1.7 m to 19.8 m [376]. This was done by replacing the 125 dB source with an ultrasonic transmitter array and amplifier outputting 1.5 W to increase sound pressure. Above this transmission power, the attack becomes audible due to non-linearities in the transmission medium (air) and the source speakers [380].

Earlier, Roy et al. had also noted that non-linearities in speakers make it harder for an adversary to increase the attack distance: "increasing the transmit power at the speaker triggers non-linearities at the speaker's own diaphragm and amplifier, resulting in an audible [output]" [272]. Instead, multiple speakers in the form of an ultrasonic array can be used to attack voice recognition systems including Amazon Alexa and Google Now at distances of up to 7.6 m using a 6 W source [272]. The attack works by partitioning the audio spectrum across the various speakers in a way that "reduces the audible leakage from any given speaker" while minimizing the total leakage power [272]. This prevents any of the non-linearities (and the transmitted signal itself) from being audible. It should be noted that if multiple non-cooperating ultrasonic sources are emitting

simultaneously, intermodulation distortions can create audible byproducts [374], allowing for the detection of potential attacks.

Parallel to the 2017 DolphinAttack paper [380], similar research was in progress at Princeton [303]: Song and Mittal also succeeded in injecting inaudible voice commands to an Amazon Echo and an Android phone. Although they accomplished relatively long distances (3.54 m with an input power of 23.7 W), their work remains in poster format. Moreover, prior to their inaudible voice commands work [272], Roy et al. also used inaudible ultrasound transmissions to record audible sounds. Specifically, in their "BackDoor" paper, they proposed a high-bandwidth covert channel that operates at up to 1.5 m using a 2 W source [271]. Although covert channels are not discussed extensively in this chapter, the research by Roy et al. is included due to the methodology which naturally led to their later work.

More concretely, instead of using amplitude modulation over a single frequency like Zhang et al. [380], Roy et al. simultaneously played two ultrasound tones whose shadows create audible sounds (only sensed by microphones) due to non-linearities [271]. They showed that amplitude modulation could not be used due to non-linearities in the ultrasound transmitters themselves, which would result in audible signals. Instead, further pre-computation was required to remove the "ringing effect", where "the transmitted sound becomes slightly audible even with FM modulation" [271]. These results point towards the next open question [376]:

Open Question 7: How can non-linear acoustics in the transmission medium (air and speakers) be avoided to further extend the range of inaudible attacks?

3.5 Optical and Thermal Manipulations

Although electromagnetic, conducted, and acoustic attacks form the majority of out-of-band signal injections, there has been some research on optical attacks, as well as temperature attacks which bias RO-based TRNGs. In the former category, adversaries exploit permissive filtering and poor shielding in interfaces which only expect ambient environmental conditions. Most papers so far have targeted sensors in an in-band fashion:

out-of-scope research includes attacks on LiDARs [246, 294, 375], as well as visible-light attacks on cameras in Unmanned Aerial Vehicles (UAVs) [71] and cars [246, 375].

Researchers had also hypothesized that excessive light injections would blind car cameras and confuse auto-controls [245]. Indeed, limited success against cameras has been achieved using ultraviolet (UV) and infrared (IR) lasers up to 2 m away [246]. However, attacks were only possible in dark environments, and the results were not reproducible with invisible lasers against other makes and models of cameras [375].

In a different strand of research, Park et al. showed that some medical infusion pumps are not well-protected against adversarial optical injections [242]. Specifically, in order to measure the flow rate of the medicine being administered, pumps are fitted with drip measurement sensors. These sensors consist of an IR emitter and receiver facing each other. When a drop passes through the sensor, the IR receiver temporarily senses less light due to diffusion, allowing for the rate to be measured. However, because the sensor is not well-enclosed, an adversary can shine an IR laser into the sensor, causing these drops to be undetected, thereby saturating the sensor. By then un-blinding the sensor, the attackers can also trick the firmware into detecting fake drops and bypassing alarms.

Adversaries can therefore selectively both over- and under-infuse a patient for an extended period of time, and for a variety of normal flow volumes. However, this can only be done with coarse-grained control over the real flow rate. Most of the experiments by Park et al. were conducted at a distance of 10 cm. Success was nevertheless reported up to 12 m away using a 30 mW IR laser pointer [242]. These results show that optical attacks can reach meaningful distances, but are, of course, limited by line-of-sight considerations.

It should be noted that the attack by Park et al. should be considered to be out-of-band, as the pump was not meant to receive external stimuli, in contrast to, for example, LiDARs, which are supposed to interact with external objects. In other words, a LiDAR depends on its surroundings to reflect its transmitted pulses and therefore infer the distance to the interfering objects. On the other hand, the drip sensor and (part of) the intravenous tube could be enclosed and shielded from the environment. This naturally leads to the defense mechanisms proposed by Park et al. [242], which are discussed in Section 3.7, and which beg the following question:

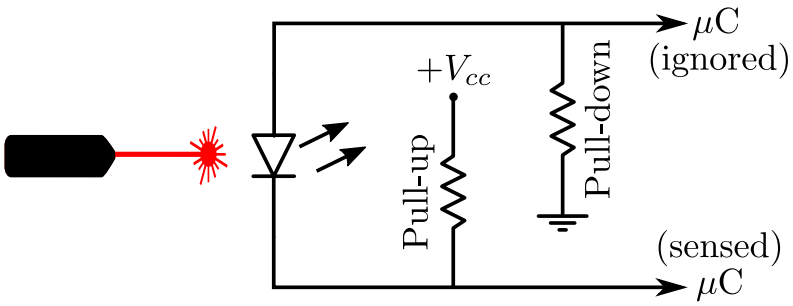


Figure 3.8: Unconventional circuit layout to convert a Light-Emitting Diode (LED) into a photodiode [194]. Pull-up and pull-down resistors can be internal to microcontroller input pins.

Open Question 8: Are there optical out-of-band attacks which exploit alternative hardware imperfections besides poor filtering and shielding?

Under somewhat unrealistic assumptions, the answer to the above question might be "yes". In their typical mode of operation, Light-Emitting Diodes (LEDs) convert current to light. However, they can also function in the reverse by producing current when illuminated [214, 215]. In preliminary experiments, Loughry recently showed that this behavior can be exploited for an optical covert channel [194]. The setup is certainly unconventional: both LED pins are connected to microcontroller GPIO pins, which are configured as inputs with (internal) pull-up and pull-down resistors, as shown in Figure 3.8. According to Loughry, seven out of ten LEDs tested responded to laser and light of different wavelengths, with some LEDs producing measurable current at both ends (anode and cathode) [194]. Whether out-of-band signal injections or fault attacks can exploit this effect has yet to be seen [194].

The final class of attacks exploits the dependence of ROs on temperature to reduce the entropy of the TRNG. It is only mentioned here for completeness, as distance requirements would dictate physical access to the device under attack. Early work in the area showed that statistical randomness tests of RO-based TRNGs would fail for certain FPGA temperatures [298]. More detailed experimentation conducted a few years later using different heat-transfer methods (resistor heater, Peltier cooler, and liquid nitrogen) then showed that "the hotter the temperature, the larger the bias" [304]. Martín et al. [201] also investigated the effect of temperature across multiple TRNG designs based on Self-Timed Rings (STRs), which do not exhibit the frequency locking

effects discussed earlier [66]. It was shown that the effects of higher temperatures on the randomness of STR-based TRNGs were not significant due to a combination of a decrease in frequency and an increase in jitter due to thermal noise [201].

Since devices were operating within their specifications, these biasing effects could be considered out-of-band attacks that operate at limited distances. This is in contrast to, for example, Martín et al.'s work which investigates the entropy of TRNGs in response to ionizing radiation [202]. However, remote conducted attacks on TRNGs are possible using local voltage drops [197]. Whether it is possible to reproduce these effects using heating circuits (e.g., [3]) remains an open question:

Open Question 9: Can software-only thermal effects bias True Random Number Generators (TRNGs) based on Ring Oscillators (ROs) in multi-tenant Field-Programmable Gate Arrays (FPGAs)?

Overall, the efficacy of optical and thermal out-of-band injections has been limited, due to the nature of vulnerable interfaces and proximity considerations respectively.

3.6 Taxonomy of Attacks

This section presents a taxonomy of out-of-band signal injection attacks, tracing their evolution through time and topic, and identifying commonalities in their methodology and sources of vulnerability. It first highlights thematic and evolutionary cross-influences between the various works studied in this chapter (Figure 3.9), and then categorizes the key hardware imperfections that make them possible (Table 3.2). Figure 3.9, specifically, is a citation graph of out-of-band attacks and related work, where an edge $X \rightarrow Y$ indicates that X is cited by (i.e., influences) Y. To reduce clutter, if a paper X is cited by both Y and Z, and Y is cited by Z, then no arrow from X to Y is drawn.

Figure 3.9 reveals that cross-influences are not limited to attacks. Instead, there is a general trend of earlier research observing the effects of non-adversarial interference, with later work actively exploiting the same phenomenon for signal injection attacks. For example, multiple works had identified the effects of electromagnetic interference on medical devices [137, 146, 186, 285] (with more papers discussed in Section 3.2), but Foo

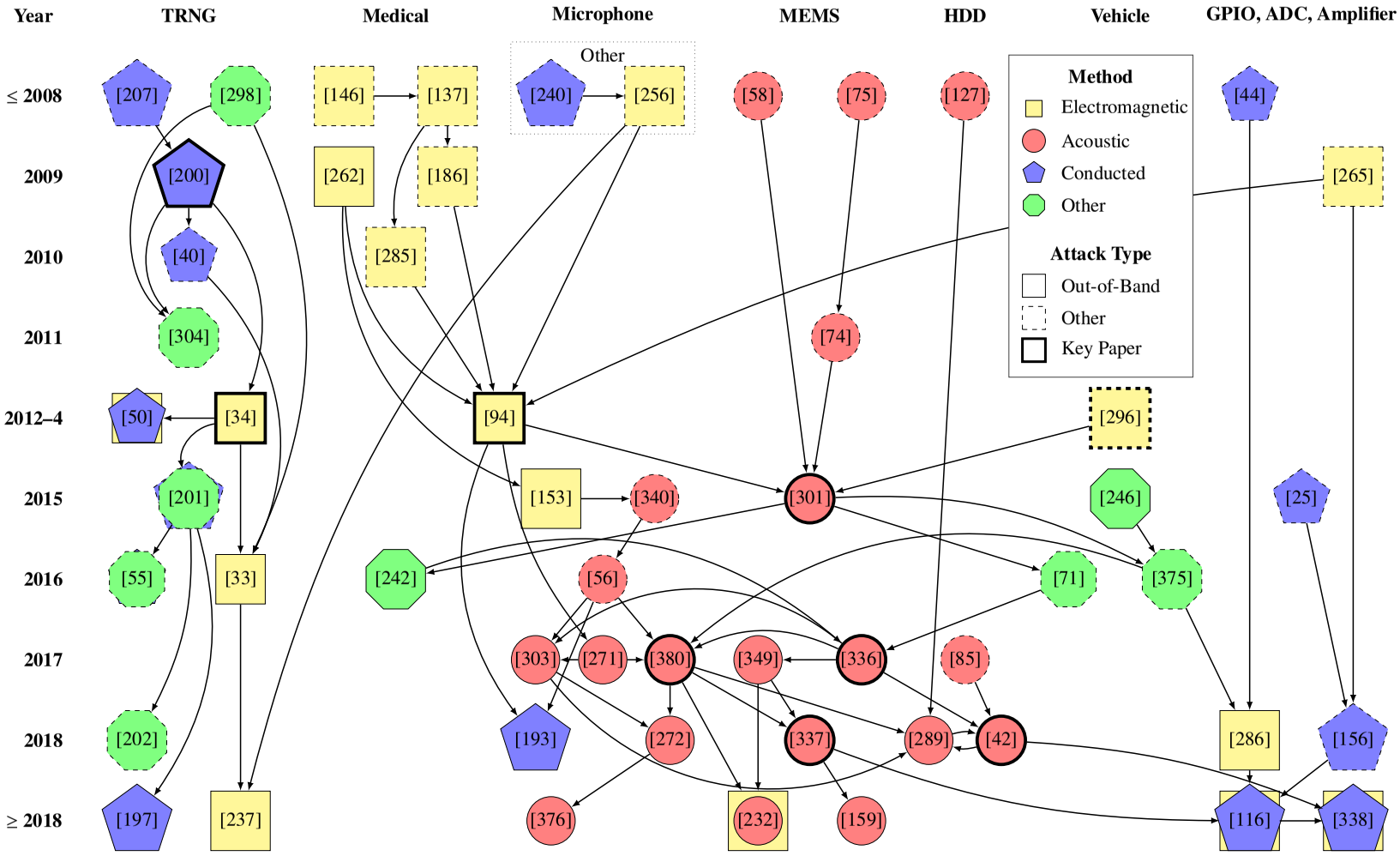


Figure 3.9: Evolutionary and thematic taxonomy of out-of-band signal injection attacks, categorized by topic and methodology. Cross-influences between out-of-band signal injections and some key related works are also shown. Highly influential papers are discussed more extensively in Section 3.6.

Kune et al. [94] were the first to recognize the effect as a security concern rather than a safety and reliability one. Similarly, Dean et al. commented on the effect of acoustic noise on gyroscopes [74, 75], but Son et al. [301] later used the same effect to destabilize drones.

Observation 1: Out-of-band signal injection attacks identify the effect of noise on systems and find novel ways to amplify it through hardware imperfections.

The graph further shows that out-of-band signal injection attack methodology has both been motivated by and has itself spurred research exploiting more traditional avenues of attack. For example, investigations into acoustic signal injection attacks [271, 272, 380] have been influenced by research on machine learning algorithms that respond to commands which are audible but indecipherable by humans [56, 340]. In addition, acoustic out-of-band attacks on MEMS sensors [301, 336] have both sparked and been inspired by in-band optical attacks on UAVs [71]. Moreover, after the effects of acoustic noise on the security of gyroscopes were first identified [301], subsequent work improved the level of control over the output [337], and attacked additional types of MEMS sensors [336] and electro-mechanical devices such as HDDs [42].

Observation 2: After an exploitable hardware imperfection has been identified, determining its root cause opens up new avenues of attacks across different domains, and with alternative methodologies.

Figure 3.9 further highlights a few key works which have played a central role in the development of the field. The first set of such papers [34, 94, 200, 301, 380] was chosen due to the high number of citations they have received overall (≥ 95) and from other out-of-band attack research (≥ 5).[6] Specifically, the works by Markettos and Moore [200] and by Bayon et al. [34] were chosen because they successfully biased TRNGs in the conducted and EM settings respectively, going beyond earlier theoretical work on oscillator locking [2, 207]. Their work led to the development of a new branch of attacks, which has developed independently of other signal injections, as shown in Figure 3.9.

Foo Kune et al.'s work on adversarial electromagnetic interference [94] also features prominently in Figure 3.9, having been cited by almost all out-of-band signal injection

[6] The citation counts are current as of 1 October 2019 according to Google Scholar.

attacks that were published after it (with the exception of TRNG research). Foo Kune et al. were the first to successfully exploit non-linearities and unintentional antennas in remote electromagnetic injection attacks, which were non-adversarial in prior work (e.g., [137, 146]), or only mentioned in passing [262].

With over 150 citations since 2017, the research by Zhang et al. [380] has been an influential acoustic out-of-band attack against microphones. Unlike earlier work on covert communication [271], and indecipherable-yet-audible commands [56, 340], Zhang et al.'s "DolphinAttack" exploited microphone non-linearities for inaudible injections.

Research on acoustic attacks is perhaps more mature against MEMS sensors, in great part due to early work by Son et al., who first showed how to disrupt gyroscopes [301]. Moreover, Trippel et al.'s research has also significantly furthered the state-of-the-art in acoustic injections by controlling the output of accelerometers for short periods of time [336]. As a result, the work of Tu et al., which showed how to extend the duration of control [337], is in the second set of articles highlighted in Figure 3.9. This set contains recent studies whose novelty and potential has not yet received mainstream attention.[7] For example, the techniques proposed by Tu et al. to overcome ADC sampling rate drifts should be applicable to other methods of injection and against different types of targets. The work by Bolton et al. [42] is in the same category, as it managed to bridge research on HDD attacks (e.g., [289]) with attacks on MEMS sensors, and contained significant insights into why resonance attacks against hard drives are possible.

The final work highlighted in Figure 3.9 is the in-band attack of Shoukry et al. against an ABS sensor [296]. It has been included not just for its high citation count (more than 130, of which 10 are out-of-band attacks), but because it is the first EM paper to focus on the magnetic field rather than the electric field. As a result, it served as inspiration to recent magnetic out-of-band attacks [232, 286], which will hopefully be explored more in the future (Section 3.9).

Table 3.2 synthesizes the extensive out-of-band signal injection literature by presenting the various attacks along with factors which contribute to them. It further notes the maximum power used and distance achieved for an attack, including the level of control

[7] Trippel et al's 2017 paper [336] lies between the two categories, already having over 70 citations, 12 of which are from other attack papers.

Year	Ref.	Power	Distance	Method	Effect	Resonance	Linearity	Filtering	Shielding	Algorithm
2009	[262]	(unspecified)	(unspecified)	📶	○	✓	–	–	✓	–
2015	[246]	(unspecified)	2.00 m	✷—	◔	–	–	✓	–	–
2016	[55]	(under-volt)	(direct)	ϟ	◐	✓	–	–	–	–
2017	[349]	(unspecified)	(unspecified)	🔊	●	✓	✓	✓	✓	–
2018	[232]	(unspecified)	(unspecified)	🔊📶	●	✓	–	✓	✓	✓
2019	[197]	(140K ROs)	(internal)	ϟ	◐	✓	–	–	–	–
2009	[200]	0.002 W	(direct)	ϟ	◐	✓	–	✓	–	✓
2012	[34]	0.003 W	100 µm	📶	◐	✓	–	–	–	–
2013	[50]	≤ 0.563 W	“near”	ϟ📶	◐	✓	–	–	–	–
2013	[94]	10.000 W	1.67 m	📶	●	✓	✓	✓	✓	–
2015	[153]	200.000 W	4.00 m	📶	●	✓	–	–	✓	✓
2016	[33]	0.003 W	100 µm	📶	◐	✓	–	–	–	–
2016	[242]	0.030 W	12.00 m	✷—	◐	–	–	–	✓	✓
2017	[271]	2.000 W	1.50 m	🔊	○	–	✓	–	–	–
2017	[303]	23.700 W	3.54 m	🔊	●	–	✓	–	–	–
2018	[193]	0.500 W	10.00 m	ϟ	●	✓	✓	✓	–	✓
2018	[237]	0.331 W	0.40 m	ϟ📶	◐	✓	–	–	–	–
2018	[272]	6.000 W	7.62 m	🔊	●	–	✓	–	–	✓
2018	[286]	1.820 W	1.00 m	📶	◔	✓	✓	✓	✓	–
2019	[116]	0.010 W	0.05 m	ϟ📶	○	✓	✓	✓	✓	✓
2019	[338]	3.162 W	6.00 m	ϟ📶	●	✓	✓	✓	✓	–
2019	[376]	1.500 W	19.80 m	🔊	●	–	✓	–	–	✓
2015	[301]	113 dB	0.10 m	🔊	◔	✓	–	–	✓	–
2017	[336]	110 dB	0.10 m	🔊	●	✓	✓	✓	✓	✓
2017	[380]	125 dB	1.75 m	🔊	●	–	✓	–	–	✓
2018	[42]	130 dB	0.10 m	🔊	◔	✓	–	–	✓	✓
2018	[289]	103 dBA	0.70 m	🔊	◔	✓	–	–	✓	–
2018	[337]	135 dB	7.80 m	🔊	●	✓	–	✓	✓	✓
2019	[159]	94 dB	0.11 m	🔊	◔	✓	–	–	✓	✓

Table 3.2: Out-of-band signal injection attack methods, vulnerabilities, and effects, along with maximum power and distance. 🔊 acoustic, ϟ conducted, 📶 electromagnetic, and ✷— optical attacks can ◔ disrupt, ◐ bias, and ● control the output, or ○ remain partially-realized or theoretical. A vulnerability either contributes (✓) to the attack, or it does not (–).

over the resulting signal. Effects range from theoretical attacks that are only partially realized to practical attacks which disrupt, bias, or completely control the output. As Table 3.2 indicates, information on the attack setup was often hard to find, sometimes completely missing, and typically had to be identified by looking up the datasheets of the signal generators, antennas, and amplifiers used. This lack of experimental details is further discussed in the context of future research in Section 3.9.

Out-of-band signal injection attacks exploit five key aspects of vulnerability: resonance; non-linearity; improper filtering; poor shielding; and insecure algorithms. All attacks which do not target microphones and optical sensors depend on resonance of some sort: this can be acoustic resonance of mechanical structures, electromagnetic resonant frequencies of unintentional antennas, or the existence of locking frequencies for ring oscillators. Other attacks depend on non-linearities of amplifiers, microphones, and speakers to demodulate high-frequency signals. This is because resonant frequencies are often much higher than those of desired injection waveforms.

In addition, many works identify improper filtering, particularly prior to ADCs and amplifiers, as well as poor shielding, to be crucial factors contributing to out-of-band attacks. Finally, in some cases, insecure sampling and processing algorithms exacerbate the problem by making it easier for an adversary to trick the system under attack into performing a dangerous action. These sources of vulnerability form the basis for many of the proposed countermeasures, which are discussed in detail in Section 3.7.

3.7 Analysis of Countermeasures

Although the literature on out-of-band attacks is quite broad, research on defenses has been more sparse. The works that have investigated countermeasures against out-of-band signal injection attacks have noted that a combination of prevention and detection techniques both in software and in hardware are necessary to improve security. The proposed defense mechanisms have been divided into six categories: more resilient hardware; improved sampling algorithms; sensor fusion and duplication; better filtering; additional shielding; and anomaly detection of measurements and the environment. Each category is discussed in detail below, with Table 3.3 summarizing the various proposals

Year	Ref.	Method	Hardware	Sampling	Fusion	Filtering	Shielding	Anomaly
2009	[200]	ϟ📶	✳	–	–	✳	✳	–
2009	[262]	📶	–	–	–	–	✳	–
2012	[34]	📶	–	–	–	–	–	–
2013	[50]	ϟ📶	–	–	–	–	–	–
2013	[94]	📶	✓	–	–	✓	✓	✓
2015	[153]	📶	✳	–	–	–	✳	✳
2015	[246]	✷—	–	–	✳	✳	–	–
2015	[297]	📶	–	✓	–	–	–	–
2015	[301]	🔊	✳	–	–	–	✓	–
2016	[33]	📶	–	–	–	–	–	–
2016	[55]	ϟ	–	–	–	–	–	–
2016	[242]	✷—	–	–	–	–	✳	✳
2016	[295]	✷—	–	–	✳	–	✳	–
2017	[271]	🔊	–	–	–	–	–	–
2017	[303]	🔊	–	–	–	–	–	–
2017	[336]	🔊	✳	✓	–	✳	✳	–
2017	[349]	🔊	✳	–	✳	–	✳	✳
2017	[380]	🔊	✓	–	–	–	–	✓
2018	[42]	🔊	✳	–	✳	–	✓	✓
2018	[193]	ϟ	–	–	–	✳	–	✳
2018	[232]	🔊📶	–	–	✓	–	–	–
2018	[237]	ϟ📶	–	–	–	✳	✳	–
2018	[272]	🔊	–	–	–	–	–	✓
2018	[286]	📶	–	–	–	✳	✳	–
2018	[289]	🔊	✳	–	–	–	✳	✳
2018	[337]	🔊	–	✳	✳	✳	✳	–
2019	[116]	ϟ📶	✳	✳	–	✳	✳	✳
2019	[159]	🔊	✳	–	–	–	✳	✳
2019	[197]	ϟ	–	–	–	–	–	✳
2019	[224]	📶	–	✓	–	–	–	✓
2019	[338]	ϟ📶	–	–	✳	✳	✳	✓
2019	[376]	🔊	✓	–	–	–	–	✓
2019	[327]	🔊📶	–	–	✓	–	–	✓
2020	[386]	📶	–	✓	–	✳	✳	–

Table 3.3: Summary of evaluated (✓), proposed (✳), and absent (–) countermeasures against 🔊 acoustic, ϟ conducted, 📶 electromagnetic, and ✷— optical out-of-band signal injection attacks.

per paper. As the table indicates, much of the discussion has been theoretical, with few works evaluating countermeasures in practice.

> **Observation 3:** The effectiveness of proposed countermeasures remains mostly theoretical, as practical implementations are often limited in scope, with superficial discussion of monetary and computational costs.

Robust Hardware: In response to resonance and non-linearity vulnerabilities, various works have proposed preventive improvements in the hardware itself to make it more robust and less susceptible to attacks. One of these improvements against electromagnetic attacks reduces asymmetries in differential inputs to a system. By doing so, attacker transmissions are received almost identically by the two unintentional receiving antennas and are severely attenuated. For example, Markettos and Moore recommend reducing the asymmetries in ring oscillators through "carefully balanced transistors", or the use of differential ones, which are "less affected by supply and substrate noise" [200]. Similarly, Foo Kune et al. found that using differential rather than single-ended comparators attenuated signals by up to 30 dB [94]. Although signals could still be injected, the power requirements to do so increased significantly, thereby raising the bar for attackers.

Another approach is to change the sensors themselves, rather than attempt to improve the physical layout of a circuit. For example, both Shahrad et al. [289] and Bolton et al. [42] note that replacing HDDs with Solid-State Drives (SSDs) thwarts acoustic resonance attacks due to a lack of moving parts. In a similar vein, Zhang et al. noted that the iPhone 6 Plus resisted their inaudible voice commands, since it is "designed to suppress any acoustic signals whose frequencies are in the ultrasound range" [380].

Finally, better frontends with fewer non-linearities are less sensitive to EMI noise [116, 153] and sonic injections [336, 349, 380]. They therefore make injections harder for adversaries, and are discussed in greater detail below, along with other general designs.

Better Sampling: Many papers have proposed improvements in sampling techniques to make it difficult for an adversary to predict how a high-frequency signal will be converted to a low-frequency one. In 2015, Shoukry et al. proposed an alternative method of sampling active sensors called "PyCRA" (for Physical Challenge-Response Authentication) to detect signal injection attacks [297]. Active sensors "perform some action to evoke

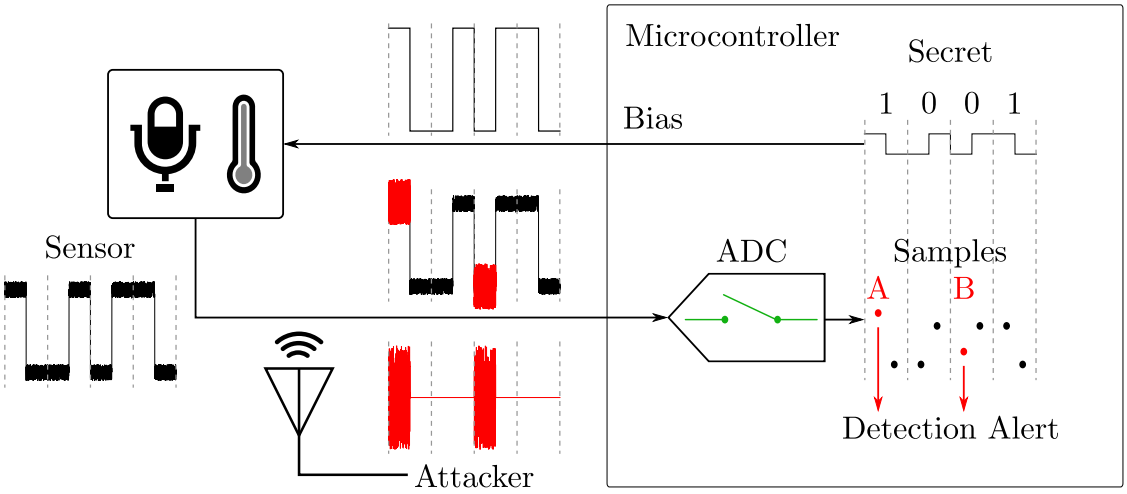


Figure 3.10: High-level overview of the defense mechanism by Zhang and Rasmussen [386]. Oversampling by a factor of $2n$ and selectively turning sensors on and off allows detection of electromagnetic out-of-band attacks: without knowing the secret sequence (1001 here for $n = 4$), the adversary will cause inconsistent (A) or unexpected (B) non-zero samples.

and measure a physical response from some measurable entity", and include, for instance, magnetic encoders measuring angular velocity. Shoukry et al.'s proposal revolves around physical challenges to prove the absence of adversarial transmissions. Specifically, when the actuator is off (*silenced*), there should be no measured quantity unless an attack is taking place. By only shutting down the actuator for a small period of time, PyCRA can detect attackers without compromising the quality of sensor measurements and actuation results. Adversaries cannot stop transmissions in time due to physical and computational delay limits, allowing PyCRA to identify them [297]. It should be noted that Shin et al. have suggested that PyCRA would require high computational overhead in practice, both in terms of the minimum sampling rate needed to hit those physical limits, and for the detectors themselves [295]. Moreover, PyCRA requires active sensors, so it primarily protects against in-band attacks.

However, a similar proposal by Zhang and Rasmussen recently showed how to protect both powered and non-powered passive sensors [386]. The key idea is to use a secret bitstream to selectively turn off the sensor and observe whether the measured signal has been altered by electromagnetic injections, as shown in Figure 3.10. More concretely, for each sensor measurement, $2n$ ADC samples are taken, corresponding to an n-bit secret sequence. Each secret bit is Manchester-encoded, so that a 0-bit is represented as the pair

$(0,1)$, which turns the sensors off for one sample, and then powers them on ("biasing") for another sample. A 1-bit is similarly encoded as the pair $(1,0)$, first powering the sensor on, and then disabling it during the second sample.

When the sensor is turned off, all samples should be close to zero, within some noise- and device-dependent tolerance. Moreover, for fast-enough sampling frequencies and slow-enough sensor signals, the n samples when the sensor is on should be close to each other. As a result, to inject a single measurement successfully, an attacker needs to correctly predict the n secret bits, which only happens with probability 2^{-n} for a randomly chosen bit sequence. By using a switch, non-powered passive sensors can also be adapted to use this approach. Moreover, spikes in the frequency domain allow Zhang and Rasmussen to detect attacker transmissions even for non-constant sensor signals [386]. Overall, by oversampling for each sensor measurement, noise can be distinguished from adversarial signals with probabilistic guarantees.

An alternative approach to prevent attackers from injecting their desired waveforms into a system is to add randomness to the sampling process, especially for ADCs which are only vulnerable for limited carrier frequencies [116]. The effect is essentially one of "having an inaccurate ADC" [336], allowing a moving average to filter out injected periodic signals. This is similar to sampling with a "dynamic sample rate", defeating the side-swing and switching attacks of Tu et al. [337], which were explained in Section 3.4. Out-of-phase sampling has also been proposed as a band-stop filter to reject frequencies near an accelerometer's resonant frequency, thereby removing attacker-injected DC offsets [336].

It should be noted that protecting against signal injection attacks into actuators has not been studied as extensively in literature. However, Muniraj and Farhood recently proposed a detection method based on a watermarking scheme that slightly alters the actuation parameters [224], similar to the proposal of Zhang and Rasmussen [386]. Under attack, the measured response of the system does not match the effect of the watermark, allowing detection. The same paper also suggested pseudo-randomly changing the pulse frequency of PWM signals, making an actuator attack harder to accomplish. Overall, these methods only alter the shape of the waveform that an adversary can inject rather

than solve the root cause of the vulnerability itself. They are therefore not sufficient as countermeasures by themselves.

Sensor Fusion: A few works have suggested that using sensors of different types (*fusion*) or multiple sensors of the same type (*duplication*) with different vulnerable frequency ranges will make injections harder [42, 246, 295, 337, 349]. This is because an adversary would need to mount multiple simultaneous attacks, which potentially interfere destructively. However, in "Sensor CON-Fusion", Nashimoto et al. [232] showed that a fusion algorithm based on Kalman filters could be circumvented (Section 3.4). As a result, better techniques are needed to protect against adversarial injections, instead of simply faulty readings [148, 241]. Tharayil et al. proposed an improved such fusion algorithm, which takes into account the mathematical relations between the underlying physical quantities. This allows them to link measurements by a gyroscope and a magnetometer in a way which can detect adversarial injections without any hardware modifications [327].

Other researchers have proposed that additional sensors be used to measure and counteract attacker signals. These additional components cannot be identical to the vulnerable sensors: as Khazaaleh et al. noted, many MEMS gyroscopes integrate a second, "identical proof mass to perform differential measurements" and "eliminate unwanted vibrations" [159]. However, they still remain vulnerable to ultrasonic attacks.

Instead, "an additional gyroscope [...] that responds only to the resonant frequency" may be able to remove the resonance effect from the main gyroscope [301]. Similarly, microphones might be able to detect (and potentially cancel) resonant frequencies to protect MEMS gyroscopes [349] and HDDs [42]. This approach would be hard in practice, at least for hard drives: the area to be protected would need to cover "the read/write head [completely] as it moves across the disk", and the sound wave to be generated would likely be large, raising many concerns about its implementation [42].

Filtering: Most papers studied highlight the need for better filters to reduce the vulnerable frequency range against conducted [116, 193, 200, 338], electromagnetic [94, 116, 237, 286, 338, 386], acoustic [336, 337], and optical [246] attacks. However, only Foo Kune et al. [94] have performed systematic experiments studying the effects of filtering on the efficacy of out-of-band signal injection attacks. To start with, Foo Kune et al. showed that

adding a low-pass filter in their experiments against Bluetooth headsets allowed audio signals to pass, but attenuated the injected electromagnetic signal by 40 dB. Moreover, they proposed an adaptive filtering mechanism which uses the measured signal and the ambient EMI level to cancel the attacker-injected waveform. Using a Finite Impulse Response (FIR) filter, the algorithm estimates this waveform, and allows quick recovery of the original signal, after an onset period at the beginning of the attack. It should be noted, however, that filters might not be effective against MEMS sensors: Khazaaleh et al. noted that "false readings could not be attenuated by adding a 10 Hz low-pass filter", despite the resonant frequency being in the kHz range [159].

Shielding: Better separation from the environment also improves protection against out-of-band signal injection attacks. For this reason, it has been recommended by most authors investigating attacks on sensors other than microphones. This shielding may come in the form of physical isolation [159, 242, 289, 301, 337], better acoustic dampening materials [42, 336, 337, 349], or radio frequency shielding [94, 116, 153, 200, 237, 262, 286, 295, 338, 386]. For instance, Foo Kune et al. demonstrated a 40 dB attenuation of the injected signal, even when the shielding had "large imperfections" [94]. These openings (e.g., for wires to pass through) can result in "major degradations in the shielding" [324]. Indeed, Selvaraj et al. noted that "while a light sensor can function in a mesh-based Faraday cage, magnetic shielding would prevent light from reaching the sensor" [286]. In addition, Bolton et al. showed that dampening foam "significantly reduced an HDD's susceptibility to write blocking", but "did not attenuate lower frequency signals" [42]. Moreover, the foam led to an increased temperature of 10 °C, which can also result in disk failure. As a result, "it is often necessary to use a combination of shielding and other protective measures" [324].

Anomaly Detection: Instead of trying to prevent signal injection attacks, some works have proposed better software-level processing of sensor signals, primarily for anomaly detection, with or without additional hardware. One such approach is to estimate the ambient level of electromagnetic [94, 153, 338], optical [242] and acoustic [39, 42, 272, 289, 349, 376, 380] emissions. For example, Park et al. noted that saturation attacks can be detected simply "by checking whether the light intensity exceeds the preset maximum

level" [242]. Foo Kune et al. [94] further investigated the use of additional (intentional) antennas or reference conductors to measure the levels of EMI radiation. This estimate can then be used by their adaptive filtering algorithm [94], which was discussed above.

In a similar vein, Tu et al. recommended the addition of a superheterodyne AM receiver to create a tunable EM detector [338]. This detector was shown to be useful in estimating and compensating errors in the measurements. Other detection mechanisms can operate with existing hardware: for example, Khazaaleh et al. noted that "sensing fingers", which are already used to measure displacement in the y-axis, can detect large displacements caused by resonance [159].

The question of how systems should behave once an attack has been detected has been largely side-stepped by many works. However, Bolton et al. introduced an algorithm to augment the hard drive feedback controller and compensate for intentional acoustic interference [42]. The addition of this attenuation controller reduces the position errors of the read and write heads to within the accepted tolerance levels, and allows the HDD to operate in the presence of an attack.

Another way of detecting attacks is to use machine learning classifiers [327]. However, such classifiers can be prone to false positives, and will miss precise waveform injections. As a result, it is often necessary to look for artifacts that would not be present during the normal operation of a sensor, such as harmonics and low- or high-frequency components [39, 116, 272, 376, 380]. This might not always be as straightforward as simply detecting energy at frequencies that are only present due to non-linearities: for sophisticated attackers, defense mechanisms need to exploit the properties of the legitimate signal itself. For example, Roy et al. showed that "voice signals exhibit well-understood patterns of fundamental frequencies", which are not present in attacks and environmental noise. As a result, they can be used to detect acoustic commands generated by ultrasound signals [272]. Similarly, the *absolute refractory period* is hard for an attacker to spoof precisely via EM injections [94]. This period represents the time span after a contraction during which the cardiac tissue will not contract again. As shown by Foo Kune et al. [94], it can be used to distinguish between real and adversarial signals.

Finally, more restrictive processing of sensor data can also help mitigate out-of-band signal injection attacks. For example, safe defaults when the sensor output is deemed as untrustworthy [94, 242] can reduce the effects of successful attacks on health- and safety-critical actions taken by systems. Similarly, less permissive choices in the design of voice interfaces can prevent non-targeted attacks from succeeding. As an example, adding voice authentication and custom keywords can prevent command injections into smartphones [116, 153, 193].

Observation 4: Until more resilient components replace vulnerable ones, defense-in-depth is necessary to protect against signal injections. This can be accomplished through better filtering and shielding to prevent attacks, and through better sampling, fusion, and anomaly detection algorithms to identify them.

Other Defenses: Although the above countermeasures were specifically proposed to protect against out-of-band signal injection attacks, general approaches from related research areas (Section 3.8) are also applicable in this context. For example, before Foo Kune et al. [94] proposed an adversarial EMI detector, Wan et al. [347, 348] introduced a similar design to "increase the immunity of a microcontroller-based system in a complex electromagnetic environment". Moreover, to protect against LiDAR attacks, Shin et al. proposed sensor fusion and redundancy, fail-safe defaults, better shielding (by reducing the receiving angle), and randomized pinging directions and waveforms [294]. Similarly, Davidson et al. proposed sensor fusion and an improved optical flow algorithm to protect against optical in-band sensor spoofing [71]. Moreover, Blue et al. detected (audible) command injections by identifying a frequency band which is produced by electronic speakers but is absent in human speech [39]. In fact, detecting unique features of the sensed property is a common defense mechanism for general sensor manipulation attacks, such as those against Smart Grid power plants [149, 308], or unmanned aircraft systems [224]. However, such approaches require a theoretical system model, and assume an adversary who cannot inject data obeying this model.

Observation 5: Defense mechanisms for in-band attacks, excessive environmental noise, and faulty sensors are often directly applicable to out-of-band signal injection attacks and vice versa.

In a different strand of research, Redouté and Richelli have proposed some guidelines for improving immunity against EMI attacks [263, 264]. These recommendations could be applied in the context of general out-of-band attacks:

1. *Filter induced signals before the non-linear device*. This suggestion is not limited to amplifiers, but can be used in other setups, including power transistors [43]. It has been shown to result in a reduction in EMI-induced offsets of up to 12.5× [210, 263, 346], but may require bulky passive components, adding noise to the circuit.

2. *Linearize the stage generating the DC shift*, for example, by using amplifiers with a wider common mode input range, resulting in better linear behavior [279].

3. *Prevent the accumulation of DC shift*, for instance by addressing the slew rate asymmetry and parasitic capacitances [92, 209, 268].

4. *Compensate and remove the induced offset*, for example, through cross-connected differential pairs [92].

As discussing all possible EMI-resistant amplifier designs is beyond the scope of this thesis (and, by extension, this chapter), the interested reader should consult various comparative works [209, 267, 342] as a starting point. Similarly, one should refer to advances in gyroscopic technologies [22, 287, 288] which do not use MEMS constructions, or reduce sensitivity to random vibrations: as Khazaaleh et al. noted [159], removing the "misalignment between the sensing and driving axes" will make systems more secure.

Observation 6: Accurate and sensitive hardware that is robust to environmental influences is a natural defense mechanism against out-of-band injections.

3.8 Additional Related Work

This section contextualizes out-of-band signal injection attacks by showing their close connections to side-channel leakage and electromagnetic interference. For example, using insights into the resonant frequencies of gyroscopes, Farshteindiker et al. showed that websites could act as covert-channel receivers without requiring user interaction or special permissions, even at very low sampling frequencies of 20 Hz [91]. Block

et al. improved the design by not requiring external equipment for the attack, instead relying on the smartphone's speaker and accelerometer [38]. Matyunin et al. then used the same effect for cross-device tracking using ultrasonic transmissions at or near the resonant frequencies of gyroscopes [205]. Moreover, Michalevsky et al. showed that MEMS gyroscope measurements are sensitive to acoustic signals in their vicinity [208]. In other words, the same source of vulnerability which can be used to destabilize [301] and control [336, 337, 349] gyroscopes and accelerometers can be used for covert-channel communication [38, 91], tracking [205], and speaker identification [15, 16, 208]. Similarly, instead of using microphone non-linearities for command injections [272, 376, 380], Shen et al. [291] and Chen et al. [63] leveraged them to protect users' privacy by jamming nearby recording devices.

The countermeasures proposed in the works above mirror those of Section 3.7, and include anti-aliasing filters, shielding, and sensor fusion.[8] For instance, shielding can protect against optical [170, 195, 373], acoustic [26, 105, 106, 302] and EM side-channel attacks [4, 255, 302]. Filtering such as ferrite choke rings around emanating cables can also effectively reduce their compromising EM emanations [172, 219].

Suggestions to increase noise in side- and regular-channel acoustic [104–106], optical [170, 373], and EM [173] emissions parallel out-of-band defense mechanisms based on reducing the sampling accuracy. For example, decreasing "the fidelity of the input audio" can protect against inaudible voice injection attacks [56]. Similarly, fonts which minimize emissions at high frequencies [173, 317, 318] exploit the human eye sensitivity to "low spatial frequencies" [173]. As EM emanations of video display units ("TEMPEST") [171, 172, 343] mostly convey "the high-frequency part of the video signal" [173], images are transformed in a way that is almost transparent to human viewers, but prevents reconstruction from side-channel listeners. Researchers have likewise shown that adding certain patterns to video frames [383] or flashing LED lights [388] can reduce the fidelity of reconstructed images from camera recordings, while not influencing regular viewers as much. Finally, anomaly detection resembles statistical and machine learning

[8] Shielding cannot be effectively used in devices with communication interfaces such as Wi-Fi cards. Leakage in these mixed-signals devices can propagate further than leakage in pure digital devices [54].

approaches to detect covert channels, and can draw inspiration from seemingly unrelated disciplines, such as timing and storage network covert channels [352].

Observation 7: The sources of vulnerabilities for out-of-band signal injection attacks are often the same as for covert- and side-channel attacks. The same techniques can thus be reused for attacks and defenses across disciplines.

Other research has indicated that devices which are typically used as actuators can effectively act as sensors. LEDs can function as photodiodes [194], while speakers [130] and HDDs [174] can both be converted into microphones. Although all three attacks have so-far required the assistance of malware, further research is required to identify the implications for out-of-band signal injection attacks.

Observation 8: Reuse of off-the-shelf equipment in unconventional setups expands the surface for signal injection attacks exploiting hardware imperfections.

The lines between electromagnetic interference and out-of-band signal injection attacks are also blurred. IEMI is often concerned with loss of functionality and wire destruction [275], or the injection of faults into devices [135, 136, 277]. However, the self-classification of attacks often depends on the research community with which an author is aligned, rather than the end result of the injection. For example, the voice command injection attacks of Kasmi and Lopes Esteves [153, 193] are categorized by their authors as IEMI attacks, despite the relatively low power used and the lack of upsets or destruction of equipment. Similarly, Osuka et al. considered their work to be in the IEMI realm [237], even though they biased the randomness of a TRNG. This fact also partially explains why research on out-of-band attacks against TRNGs has largely ignored attacks against other targets and vice versa.

In general, this mismatch of expectations often results in unexplored avenues of research, as can be seen, for example, in the IEMI attacks on UAVs of Lopes Esteves et al. [192]: although there is a strong inverse correlation between the battery temperature reading and the strength of the electric field, the authors do not further investigate how to precisely control the sensor output. Moreover, as was explained in Section 3.7, research into electromagnetic interference can provide insights into how to build more resilient

hardware, even when the hardware is only tested against "unintentional parasitic signals and does not take a malicious behavior of an attacker into account" [193]. In fact, many IEMI countermeasures are also applicable to out-of-band signal injection attacks, including physical security (e.g., keep-out zones), shielding, shortening of exposed cables, and resonance reduction [256, 278].

Observation 9: The proposed terminology based on the outcome rather than the method of injection can help systematize attack and defense approaches, and reveal previously unexplored connections.

3.9 Future Directions

Despite the extensive research on out-of-band signal injection attacks, there is no common methodology to evaluate how susceptible systems are to them, except for the framework of Chapter 4. This is in contrast to related disciplines, such as side-channel analysis [164], direct power injection and near-field scan immunity [45], fault injection attacks [378], and IEMI attacks [228]. Indeed, although many papers sweep through frequencies to find the resonant ones [94, 336], some do not adopt this terminology [42, 289, 301], and do not specify how wide the frequency steps should be. This proves to be particularly problematic, as some attack windows "are as narrow as a few Hertz" [289].

Recently, Tu et al. [337] provided a more detailed methodology for acoustic injection attacks, which starts with a *profiling* stage. During this phase, single-tone sounds are transmitted, and are swept at an interval of 10 Hz. The devices targeted remain stationary during the profiling stage. Further increments of 1 Hz or smaller can be used near the resonant frequencies to estimate the sampling frequency of the ADC, and account for its drift. The next stage involves *synchronizing* to a frequency which is close to a multiple of the ADC sampling rate. This step is followed by *manipulating* the attack parameters, and *adjusting* them in response to drifts. Although this approach provides some common ground for evaluation, several questions remain unanswered, especially when assessing countermeasures to claim that a system is secure. These questions include: what the frequency range itself should be; what the step should be for wide ranges; what modulation method to use and with what parameters; and whether there are other factors that would

need to be examined during experimentation. For instance, for electromagnetic attacks, the incident angle of the EM field and the distance of attacks can have a profound impact on their success, especially as they relate to generalizing from the near- to the far-field.

Observation 10: A precise experimental procedure which specifies sweep, modulation, and other parameters is needed for out-of-band signal injection attacks.

The question of the maximum feasible attack distance has mostly been of theoretical interest, with practical attacks often limited to a few centimeters. Even though EM attacks should in theory have a longer range than acoustic and optical attacks, the converse appears to be true in the experiments conducted by the works studied in this chapter (Table 3.2). There is also a worrying trend of assuming that more power and more expensive equipment easily translate to longer ranges. For example, Tu et al. [337] claim that with more speakers, gyroscopes can be attacked from an 8× longer distance, but as Roy et al. [272] showed, doing so is not a trivial engineering concern, if the inaudibility of injections is to be maintained.

Similarly, although Foo Kune et al. [94] claim that a 20 dB gain directional antenna and a 1 W source can attack equipment at distances of up to 50 m, these estimates seem optimistic: according to Lopes Esteves and Kasmi [193], a 200 W source is required for a distance of 4 m for remote command injections [153]. What is more, high-powered EM sources have the potential to cause faults in other equipment and be harmful to human life. As a result, determining how to inject precise signals from a distance is particularly challenging. These problems become even more pronounced for magnetic attacks on actuators, which have been limited so far.

Observation 11: Dedicated facilities and test equipment for long-range experimentation are needed.

Note that since many of the systems targeted are safety- and mission-critical, it is reasonable to expect that, in the future, some devices may be required to undergo a certification process. Indeed, a CERT alert warning of MEMS susceptibility to ultrasonic resonance [70] highlights that out-of-band signal injection vulnerabilities are a concern for governments and corporations alike. When the field matures, regulations will pave

the way for an expanding industry around facilities and test equipment for EMC and acoustic immunity against adversarial injections.

In the meantime, as currently published work often leaves experimental details under-specified, reproducibility becomes a significant challenge: for instance, as discussed in Section 3.6, details on the power used were often not readily available, but required searching through datasheets. Moreover, the duration of attacks was also often not specified. Trippel et al. [336] reported that some of their attacks against accelerometers only work for a couple of seconds before the attack fails. Sampling rate drifts thus necessitate manual tuning, or more sophisticated attacking techniques, such as those proposed by Tu et al. [337]. However, without details of the function generator specifications, it would be hard to know whether some of the issues are caused by poor clock accuracy of the generator. This problem is bound to become even more pronounced when using Software-Defined Radio (SDR) and other low-end commodity hardware for attack weaponization.

Minor variations in the construction of devices can also have significant effects on the sensors' behaviors, and will potentially impact the reproducibility of attacks. For instance, Dey et al. [78] showed that otherwise identical accelerometers can be tracked due to slightly different performance characteristics. Overall, the absence of experimental details, coupled with monetary costs and legal requirements associated with using the electromagnetic spectrum, make security research into out-of-band signal injection attacks a challenging space for new researchers to enter.

Observation 12: Reproducibility through common metrics that allow for direct comparison of the effects of out-of-band signal injections in standardized experimental setups are necessary to advance the state-of-the-art.

Besides the defense mechanisms of Section 3.7, to protect future devices from attack, new security-sensitive products must take a fundamentally different approach to trusting the outputs of sensors. In the words of Fu and Xu, there is a need to "shift from component-centric security to system-centric tolerance of untrustworthy components" [98], perhaps taking note of advances in fault-tolerant literature [77, 87, 88, 218, 243, 391]. Fu and Xu also recommend that sensor outputs be "continuously checkable by software for adversarial influence", such as through hidden internal debugging information [98]. They

further highlight the need for interdisciplinary teams and education [98]. Indeed, until new hardware is deployed, many cross-disciplinary solutions will be required to prevent, detect, and mitigate attacks. As out-of-band signal injection attacks become more powerful, collaboration will be necessary to address the multifaceted research influences of the field.

Observation 13: Interdisciplinary research quantifying the effectiveness of countermeasures is needed to inform future hardware and software design choices.

3.10 Summary

People's ever-increasing reliance on sensors and actuators highlights the need for a comprehensive look into electromagnetic (Section 3.2), conducted (Section 3.3), acoustic (Section 3.4), and optical (Section 3.5) out-of-band signal injection attacks. These attacks cause a mismatch between a physical property being measured by a sensor or acted upon by an actuator and its digitized version, and can control or disrupt drones, hard drives, and medical devices, with potentially fatal consequences on human life. In light of the importance of such attacks, this chapter took the first step towards unifying the diverse and expanding research, first through common terminology (Section 3.1), and then through a taxonomy of attacks (Section 3.6) and defenses (Section 3.7). This chapter further revealed inter-disciplinary influences between seemingly disparate topics (Section 3.8), and also made several observations that can inform future research in the area (Section 3.9). Overall, better experimental and reporting procedures are necessary for direct comparisons of the effects of attack and defense mechanisms. Chapter 4 takes a first step in that direction by introducing a framework for evaluating security in the presence of out-of-band signal injection attacks.

All models are wrong, but some are useful.

— George Edward Pelham Box

4

Security Under Signal Injection Attacks

Contents

As explained in the previous chapters, out-of-band signal injection attacks can remotely induce waveforms into the outputs of sensors. As a result, embedded devices are tricked into acting upon values which do not correspond to the true sensor measurements, potentially breaking security guarantees. To address the lack of a unifying framework for evaluating the effects of such adversarial transmissions, this chapter:

1. Proposes a system model which abstracts away from engineering concerns associ-

ated with remote transmissions, such as antenna design (Section 4.1).

2. Defines security in the context of out-of-band signal injection attacks. These definitions address effects ranging from mere disruptions of the sensor readings, to precise waveform injections of attacker-chosen values (Section 4.2).

3. Introduces an algorithm to calculate the security level of a system under the proposed definitions. This algorithm is demonstrated in practice by injecting "OK Google" commands into a smartphone (Section 4.3).

4. Investigates the extent to which commercial Analog-to-Digital Converters (ADCs) are vulnerable to malicious signal injection attacks by testing their demodulation characteristics (Section 4.4).

5. Discusses how the security framework (model and definitions) can be used to inform circuit design choices and interpret defense mechanisms (Section 4.5).

In summary (Section 4.6), this chapter highlights the importance of testing systems against out-of-band signal injection attacks, and proposes a methodology to evaluate the security of real devices.

4.1 System and Adversary Model

Out-of-band signal injection attacks pose new challenges from a threat-modeling perspective, since physical limitations preclude adversaries from arbitrarily and precisely changing sensor measurements. To create a threat model and define security in its context, it is therefore necessary to abstract away from specific circuit designs and engineering restrictions, while also acknowledging the properties of remote transmissions. This can be done by reducing the behavior of an embedded system to two transfer functions. As Section 4.1.1 explains, the first transfer function describes circuit-specific behavior, including how adversarial signals enter the circuit. For instance, for Printed Circuit Boards (PCBs), traces acting as antennas aid electromagnetic attacks (Chapters 2 and 3). The second transfer function described in Section 4.1.1 is ADC-specific. It dictates how the signals which have made it into the circuit are digitized. Section 4.1.2 then

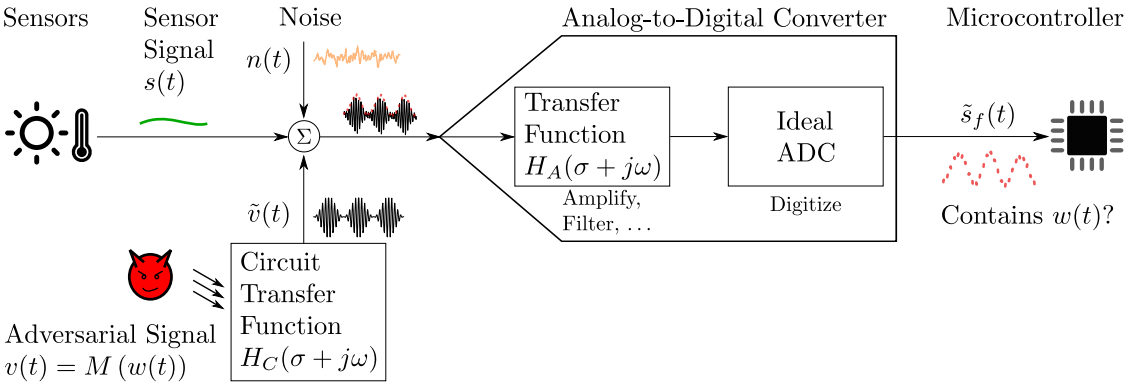


Figure 4.1: System model for out-of-band signal injection attacks: an adversarial signal $v(t)$ enters the circuit and is transformed via a transfer function H_C. It is digitized along with the sensor signal $s(t)$ and the noise $n(t)$ through a transfer function H_A, which is specific to the device's Analog-to-Digital Converter (ADC). In successful attacks, the digitized signal will contain the demodulated version $w(t)$ of the attacker signal $v(t) = M(w(t))$, where M is the modulation function (e.g., amplitude modulation over a high-frequency carrier).

highlights some sources of measurement errors even in the absence of an adversary, while Section 4.1.3 details the attacker capabilities and limitations.

4.1.1 Circuit Abstraction

ADCs are central to the digitization process of converting signals from the analog realm to the digital domain, and the high-level circuit block diagram of Figure 4.1 reflects that. In the absence of an adversary, the ADC digitizes the sensor signal $s(t)$ along with environmental noise $n(t)$, and transfers the digital bits to a microcontroller. The ADC is modeled in two parts: an "ideal" ADC which simply digitizes the signal, and a transfer function H_A. This transfer function describes the internal behavior of the ADC, which includes effects such as filtering and amplification. The digitized version of the signal $\tilde{s}_f(t)$ depends on both this transfer function, and the sampling frequency f of the ADC. An adversarial signal can enter the system (e.g., through the wires connecting the sensor to the ADC) and add to the sensor signal and the noise. This process can be described by a second, circuit-specific transfer function H_C, which transforms the adversarial signal $v(t)$ into $\tilde{v}(t)$. Note that components such as external filters and amplifiers in the signal path between the point of injection and the ADC can be included in either H_A or H_C. They are included in H_A when they also affect the sensor signal $s(t)$, but in H_C when they are specific to the coupling effect. H_C and H_A are discussed in detail below.

Circuit Transfer Function H_C: A transfer function H_C captures the response of the circuit to external signals. It is sensitive to passive and active components on the path to the ADC, as well as their "circuit topology and placement area" [190]. Although it is hard to mathematically model and predict the behavior of circuits in response to different signal transmissions without empirical frequency sweeps, H_C presents a useful abstraction: it effectively separates the behavior of the ADC (which need only be determined once, for instance by the manufacturer) from circuit layout and transmission details.

As explained in Chapters 2 and 3, adversarial signals must typically be transmitted at high frequencies, e.g., due to the inverse relationship between the length of the wire and its resonant frequencies for electromagnetic attacks. In other words, the low-frequency waveform $w(t)$ that the adversary wants to inject into the output of the ADC $\tilde{s}_f(t)$ may need to be modulated over a high-frequency carrier using a modulation function M. The modulated version of the signal is denoted by $v(t) = M(w(t))$. Moreover, H_C can account for distance factors between the adversary and the circuit under test: due to the Friis transmission formula (Equation (2.2)), as distance doubles, transmission power needs to quadruple. This effect can be captured by increasing the attenuation of H_C by 6 dB, while defense mechanisms such as shielding can be addressed similarly. This approach therefore side-steps engineering issues of remote transmissions, and reduces the number of parameters used in the security definitions of Section 4.2.

ADC Transfer Function H_A: As explained in Chapter 2, ADCs contain components which may cause a mismatch between the "true" value at the ADC input and the digitized output. These components include Electrostatic Discharge (ESD) protection diodes and internal amplifiers, which can unintentionally cause Direct Current (DC) shifts. Although the sample-and-hold mechanism should attenuate high-frequency signals, its cutoff frequency is often multiple times the ADC sampling rate (Appendix A). These ADC-specific transformations, modeled through H_A, unintentionally demodulate high-frequency signals which are not attenuated by H_C. They are explored in more detail in Section 4.4 and Appendix A.

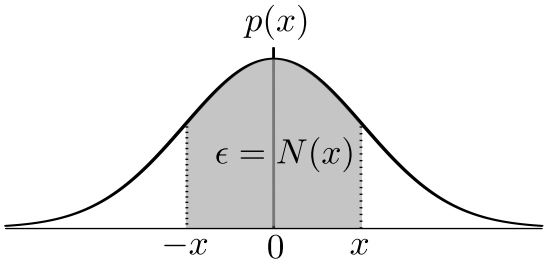


Figure 4.2: Noise probability distribution $p(x)$. The shaded area represents the probability $\epsilon = N(x) = \Pr[|n(t)| \leq x]$.

4.1.2 Common Sampling Errors

The digitization process through ADCs entails quantization and noise errors. These sampling errors can be described by a function $E_s(t)$, which is defined in Equation (4.1). As the equation shows, errors depend on the sensor input into the ADC $s(t)$, the sampling rate f, the discrete output of the ADC $\tilde{s}_f(t)$ as well as the conversion delay τ, which represents the time the ADC takes to complete the digitization:

$$E_s(t) = \begin{cases} \left|\tilde{s}_f(t+\tau) - s(t)\right| & \text{if a conversion starts at } t \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

The first source of errors describes the inherent loss of accuracy in the sampling process: an ADC with a resolution of N bits can only represent 2^N different values. As a result, the true sensor analog value s and the digitized value $\tilde{s}$ differ. The maximum value Q of this *quantization error* is:

$$Q = \frac{V_{max} - V_{min}}{2^{N+1}} \geq |s - \tilde{s}| \tag{4.2}$$

The second source of error comes from environmental noise, which may affect measurements. This noise, denoted by $n(t)$, is assumed to be *white*, i.e., it is independent of the signal being measured, and comes from a zero-mean distribution.[9] The security definitions of Section 4.2 require an estimate of the level of noise in the system, so some relevant notation is introduced here. Assume that $n(t)$ follows a Probability Distribution Function (PDF) $p(x)$, and define $N(x)$ as the probability that the noise is between $-x$

[9]Noise is typically assumed to follow a Gaussian (normal) distribution, but this assumption is not necessary in the definitions that follow.

and x, as shown in Figure 4.2, i.e.,

$$N(x) = \Pr[|n(t)| \leq x] = \int_{-x}^{x} p(u)du \tag{4.3}$$

The inverse of this function can be defined as follows for $0 \leq \epsilon < 1$:

$$N^{-1}(\epsilon) = \inf\{x \geq 0 : N(x) = \epsilon\} \tag{4.4}$$

As shown in Figure 4.2, $N^{-1}(\epsilon)$ represents the smallest $x \geq 0$ such that the probability that the noise magnitude falls within $[-x, x]$ is ϵ. This function is well-defined, and because $N(x)$ is an increasing function, so is $N^{-1}(\epsilon)$.

4.1.3 Adversarial Capabilities

The proposed threat model and definitions can capture a range of attacker goals, from attackers who merely want to disrupt sensor outputs, to those who wish to inject precise waveforms into a system. These notions are defined mathematically in Section 4.2, but this section describes the attacker capabilities based on Figure 4.1: in the model, the adversary can only alter the transmitted adversarial signal $v(t)$, but cannot directly influence the sensor signal $s(t)$, the (residual) noise $n(t)$, or the transfer functions H_A and H_C. The adversary knows H_A, H_C, and the distribution of the noise $n(t)$, although the true sensor signal $s(t)$ might be hidden from the adversary (see Section 4.2.2). The only constraint placed on the adversarial signal is that the attacker is limited to transmissions of signals $v(t)$ whose peak voltage level is bounded by some constant V_{PK}^{Adv}. In other words, $|v(t)| \leq V_{PK}^{Adv}$ for all t. This adversary is called a V_{PK}^{Adv}*-bound adversary*, and all security definitions are against such bounded adversaries.

The model restricts voltage rather than power or distance, as doing so allows adversaries who have access to high-end physical equipment, such as powerful amplifiers and highly-directional antennas. It also reduces the number of parameters used in the security definitions of Section 4.2, since distance and power effects can be compensated directly through V_{PK}^{Adv}, or indirectly through H_C, as discussed in Section 4.1.1.

4.2 Security Definitions

Using the model of Figure 4.1, this section defines security in the presence of out-of-band signal injection attacks. The V_{PK}^{Adv}-bound adversary is only constrained by the voltage budget and is allowed to transmit any waveform $v(t)$, provided that $|v(t)| \leq V_{PK}^{Adv}$ for all t. Whether the adversary succeeds in injecting the target waveform $w(t)$ into the output of the system depends on the transfer functions H_C and H_A. For a given system described by H_A and H_C, there are three types of attacks an adversary can perform:[10]

1. The adversary can disturb the sensor readings, but cannot precisely control the measurement outputs. This attack is called an *existential injection*, while the lack of existential injections can be considered *universal security*.

2. The adversary can inject a target waveform $w(t)$ into the ADC output with high fidelity, performing a *selective injection*. If attacks cannot succeed, the system is *selectively secure* against $w(t)$.

3. The adversary can *universally inject* any waveform $w(t)$. If there is any non-trivial waveform for which injections fail, the system is *existentially secure*.

This section sets out to precisely define the above security notions by accounting for noise and quantization errors (Equation (4.2)). The definitions provided capture the intuition that systems are secure when there are no adversarial transmissions, and are "monotonic" in voltage, i.e., systems are more vulnerable against adversaries with access to higher-powered transmitters. The definitions are also monotonic in noise: in other words, in environments with low noise, even a small disturbance of the output is sufficient to break the security of a system. Section 4.2.1 evaluates whether an adversary can disturb the ADC output away from its correct value sufficiently. Section 4.2.2 then formalizes the notion of selective security against target waveforms $w(t)$. Finally, Section 4.2.3 introduces universal injections by defining what a non-trivial waveform is. The three types of signal injection attacks, the corresponding security properties, and the ensuing ADC errors (injected waveforms) are summarized in Table 4.1.

[10] The terminology chosen was inspired by attacks against signature schemes, where how broken a system is depends on the types of messages an attacker can forge [123].

Security	Injection	ADC Error $E_s(t)$
Universal	Existential	Bounded away from 0
Selective	Selective	Target waveform $w(t)$
Existential	Universal	Every non-trivial $w(t)$

Table 4.1: Correspondence between security properties of a sensor system, adversarial injection attacks, and the resulting Analog-to-Digital Converter (ADC) waveform errors (signals).

4.2.1 Existential Injection and Universal Security

The most primitive signal injection attack is a simple disruption of the sensor readings. There are two axes in which this notion can be evaluated: adversarial voltage and probability of success (success is probabilistic, as noise is a random variable). For a fixed probability of success, the goal is to determine the smallest voltage level for which an attack is successful. For a fixed voltage level, the goal is to find the probability of a successful attack. Alternatively, if both voltage and probability of success are fixed, the goal is to determine if a system is secure against disruptive signal injection attacks.

The definition for universal security is a formalization of the above intuition, calling a system secure when, even in the presence of injections (bounded by adversarial voltage), the true analog sensor value and the ADC digital output do not deviate by more than the quantization error and the noise, with sufficiently high probability. Mathematically:

Definition 2 (Universal Security, Existential Injection) *For* $0 \leq \epsilon < 1$*, and* $V_{PK}^{Adv} \geq 0$*, a system is called* ***universally*** (ϵ, V_{PK}^{Adv})***-secure*** *if*

$$\Pr\left[E_s(t) \geq Q + N^{-1}\left(\frac{\epsilon+1}{2}\right)\right] \leq \frac{\epsilon+1}{2} \tag{4.5}$$

for every adversarial waveform $v(t)$*, with* $|v(t)| \leq V_{PK}^{Adv}$ *for all* t*.* Q *is the quantization error of the system,* N^{-1} *is the noise distribution inverse defined in Equation* (4.4)*, and* E_s *is the sampling error as defined by Equation* (4.1)*. The probability is taken over the duration of the attack, i.e., at each sampling point within the interval* $t_{start} \leq t \leq t_{end}$*. A successful attack is an* ***existential injection****, and the system is simply called universally* ϵ*-secure, when* V_{PK}^{Adv} *is implied.*

The first property of this definition is that, in the absence of injections, the system is universally ϵ-secure for all $0 \leq \epsilon < 1$. Indeed, let $x = N^{-1}\left(\frac{\epsilon+1}{2}\right)$, so that $\Pr[|n(t)| \leq x] = \frac{\epsilon+1}{2}$. Then, in the absence of injections,

$$\Pr\left[E_s(t) \geq Q + N^{-1}\left(\frac{\epsilon+1}{2}\right)\right] = \Pr[|n(t)| \geq x] = 1 - \frac{\epsilon+1}{2} = \frac{1-\epsilon}{2} \leq \frac{\epsilon+1}{2}$$

which holds for all $0 \leq \epsilon < 1$, as desired. This proof is precisely the reason for requiring a noise level and probability of at least 50% in the definition: the proof no longer works if $(\epsilon+1)/2$ is replaced by just ϵ. In other words, mere noise would be classified as an attack by the modified definition.

Voltage: The second property is that a higher adversarial voltage budget can only make a system more vulnerable. Indeed, if a system is universally (ϵ, V_1)-secure, then it is universally (ϵ, V_2)-secure for $V_2 \leq V_1$. For this, it suffices to prove the contrapositive, i.e., that if a system is not universally (ϵ, V_2)-secure, then it is not universally (ϵ, V_1)-secure. For the proof, let $v(t)$ be an adversarial waveform with $|v(t)| \leq V_2$ such that Equation (4.5) does not hold, which exists by the assumption that the system is not universally (ϵ, V_2)-secure. Then, by the transitive property, $|v(t)| \leq V_1$, making $v(t)$ a valid counterexample for universal (ϵ, V_1)-security.

Probability: The third property is probability monotonicity, which shows the existence of a "critical threshold" for ϵ, above which a system is universally secure (for fixed V_{PK}^{Adv}), and below which a system is not universally secure. Indeed, for fixed V_{PK}^{Adv}, if a system is universally (ϵ, V_{PK}^{Adv})-secure, then it is universally $(\epsilon+\delta, V_{PK}^{Adv})$-secure for $0 \leq \delta < 1-\epsilon$, as

$$\Pr\left[E_s(t) \geq Q + N^{-1}\left(\frac{\epsilon+\delta+1}{2}\right)\right] \leq \Pr\left[E_s(t) \geq Q + N^{-1}\left(\frac{\epsilon+1}{2}\right)\right] \leq \frac{\epsilon+1}{2} \leq \frac{\epsilon+\delta+1}{2}$$

because N^{-1} is increasing. The contrapositive is, of course, also true: if a system is not universally secure for a given ϵ, it is also not universally secure for $\epsilon - \delta$ with $0 \leq \delta \leq \epsilon$.

Thresholds: For a given security level ϵ, then, there is a maximum (if any) V_{PK}^{Adv} such that a system is universally (ϵ, V_{PK}^{Adv})-secure, or conversely a minimum (if any) V_{PK}^{Adv} such that a system is not universally (ϵ, V_{PK}^{Adv})-secure. This is the **critical universal voltage level** V_c for the given ϵ. Moreover, for any V_{PK}^{Adv}, there is a unique **critical universal security threshold** ϵ_c such that the system is universally (ϵ, V_{PK}^{Adv})-secure for $\epsilon_c < \epsilon < 1$

and not universally (ϵ, V_{PK}^{Adv})-secure for $0 \leq \epsilon < \epsilon_c$. By convention, $\epsilon_c = 0$ if the system is secure for all ϵ, and $\epsilon_c = 1$ if there is no ϵ for which the system is secure. This critical threshold indicates the security level of a system: the lower ϵ_c is, the better a system is protected against signal injection attacks.

4.2.2 Selective Injection and Selective Security

The second definition captures the notion of security against specific target waveforms $w(t)$: its goal is to find the probability that a V_{PK}^{Adv}-bound adversary can make $w(t)$ appear in the output of the ADC. Conversely, to define security in this context, the digitized signal $\tilde{s}_f(t)$ must differ from the waveform $s(t) + w(t)$ with high probability, even if plenty of noise is allowed. There are two crucial points to notice about the waveform $w(t)$. First, $w(t)$ is not the raw signal $v(t)$ the adversary is transmitting, as this signal undergoes two transformations via H_C and H_A. Instead, $w(t)$ is the signal that the adversary wants the ADC to think that it is seeing, and is usually a demodulated version of $v(t)$ (see Figure 4.1). Second, $w(t)$ does not necessarily cancel out or overpower $s(t)$, because that would require predictive modeling of the sensor signal $s(t)$. However, if the adversary can predict $s(t)$ (e.g., by monitoring the output of the ADC, or by using identical sensors), one can then ask about security against the waveform $w'(t) = w(t) - s(t)$ instead. Given this intuition, selective security is defined as follows:

Definition 3 (Selective Security, Selective Injection) *For $0 \leq \epsilon < 1$, and $V_{PK}^{Adv} \geq 0$, a system is called **selectively** $(\epsilon, w(t), V_{PK}^{Adv})$**-secure** if*

$$\Pr\left[E_{s+w}(t) \geq Q + N^{-1}\left(\frac{(1-\epsilon)+1}{2}\right)\right] > \frac{2-\epsilon}{2} \tag{4.6}$$

for every adversarial waveform $v(t)$, with $|v(t)| \leq V_{PK}^{Adv}$ for all t, where the probability is taken over the duration of the attack. Q is the quantization error of the system, N^{-1} is the noise distribution inverse defined in Equation (4.4)*, and $E_{s+w}(t) = \left|\tilde{s}_f(t+\tau) - s(t) - w(t)\right|$ during sampling periods, and* 0 *otherwise. A successful attack is a **selective injection**, and a system is called selectively ϵ-secure, when V_{PK}^{Adv} and $w(t)$ are clear from context.*

This definition is monotonic in voltage and the probability of success, allowing one to talk about "the" probability of success for a given waveform:

Voltage: A similar argument as for existential injections shows that increasing V_{PK}^{Adv} can only make a secure system insecure, but not vice versa, i.e., that if a system is selectively $(\epsilon, w(t), V_1)$-secure, then it is selectively $(\epsilon, w(t), V_2)$-secure for $V_2 \leq V_1$. The **critical selective voltage level** V_c^w for a given ϵ and $w(t)$ is thus well-defined.

Probability: If a system is selectively ϵ-secure (against a target waveform and voltage budget), then it is selectively $(\epsilon + \delta)$-secure for $0 \leq \delta < 1 - \epsilon$, because

$$\begin{aligned} P &= \Pr\left[E_{s+w}(t) \geq Q + N^{-1}\left(\frac{1-(\epsilon+\delta)+1}{2}\right)\right] \\ &\geq \Pr\left[E_{s+w}(t) \geq Q + N^{-1}\left(\frac{1-\epsilon+1}{2}\right)\right] > \frac{2-\epsilon}{2} \geq \frac{2-(\epsilon+\delta)}{2} \end{aligned}$$

If the system is not selectively ϵ-secure, then it is not selectively $(\epsilon - \delta)$-secure.

For a target $w(t)$ and fixed V_{PK}^{Adv}, then, there is a waveform-specific **critical selective security threshold** ϵ_c^w such that the system is vulnerable for all ϵ^w with $0 \leq \epsilon^w < \epsilon_c^w$ and secure for all ϵ^w with $\epsilon_c^w < \epsilon^w < 1$. By convention, $\epsilon_c^w = 0$ if there is no ϵ for which the system is vulnerable, and $\epsilon_c^w = 1$ if there is no ϵ for which the system is secure.

Threshold Relationship: For a given adversarial waveform $v(t)$, the critical universal threshold of a system ϵ_c is related to the critical selective threshold ϵ_c^0 against the zero waveform $w(t) = 0$ through the equation:

$$\epsilon_c^0 = 1 - \epsilon_c \tag{4.7}$$

Indeed, if a system is not universally ϵ-secure, then $P = \Pr\left[E_s(t) \geq Q + N^{-1}\left(\frac{\epsilon+1}{2}\right)\right] > \frac{\epsilon+1}{2}$, so $\frac{2-(1-\epsilon)}{2} = \frac{\epsilon+1}{2} < P = \Pr\left[E_{s+0}(t) \geq Q + N^{-1}\left(\frac{(1-(1-\epsilon))+1}{2}\right)\right]$, making the system selectively $(1-\epsilon)$-secure for the zero waveform, and vice versa.

4.2.3 Universal Injection and Existential Security

The final notion of security is a weak one, which requires that the adversary cannot inject at least one "representable" waveform into the system, i.e., one which is within the ADC limits. This can be expressed more precisely as follows:

Definition 4 (Representable Waveform) *A waveform $w(t)$ is called **representable** if it is within the ADC voltage levels, and has a maximum frequency component bounded by the Nyquist frequency of the ADC. Mathematically, $V_{min} \leq w(t) \leq V_{max}$ and $f_{max} \leq f_s/2$.*

Using this, one can define security against at least one representable waveform:

Definition 5 (Existential Security, Universal Injection) *For $0 \leq \epsilon < 1$, and $V_{PK}^{Adv} \geq 0$, a system is called **existentially** (ϵ, V_{PK}^{Adv})**-secure** if there exists a representable waveform $w(t)$ for which the system is selectively $(\epsilon, w(t), V_{PK}^{Adv})$-secure. A system is called existentially ϵ-secure when V_{PK}^{Adv} is clear from context. If there is no such $w(t)$, the adversary can perform any **universal injection**.*

As above, voltage and probability are monotonic in the opposite direction.

Voltage: If a system is existentially (ϵ, V_1)-secure, then it is (ϵ, V_2)-secure for $V_2 \leq V_1$. By assumption, there is a representable $w(t)$ such that the system is selectively $(\epsilon, w(t), V_1)$-secure. By the previous section, this system is $(\epsilon, w(t), V_2)$-secure, concluding the proof.

Probability: If a system is existentially (ϵ_1, V)-secure, then it is (ϵ_2, V)-secure for $\epsilon_1 \leq \epsilon_2$. By assumption, there is a representable $w(t)$ such that the system is selectively $(\epsilon_1, w(t), V)$-secure. By the previous section, the system is also $(\epsilon_2, w(t), V)$-secure, as desired.

Thresholds: Extending the definitions of the previous sections, for fixed ϵ, there is a **critical existential voltage level** V_c^{exist} below which a system is existentially ϵ-secure, and above which the system is existentially ϵ-insecure. Similarly, for a fixed adversarial voltage, one can define the **critical existential security threshold** ϵ_c^{exist}, above which the system is existentially secure, and below which the system is insecure.

Designers can adjust the definitions to further restrict target waveforms and existential security counterexamples. For instance, they may wish to check whether an adversary can inject all waveforms which are sufficiently bounded away from 0, periodic waveforms, or waveforms of a specific frequency. The proofs for voltage and probability monotonicity still hold, allowing one to talk about universal security against $\mathcal{S}$-representable waveforms: waveforms which are representable and also in a set $\mathcal{S}$.

Injection	Resulting Signal	Crit. Thres.
Existential	$w(t) \neq 0$	≥ 0.892
Selective	$w(t) = e^{\sin(2\pi f_m t)}$	0.747
Selective	$w(t) = \sin(2\pi f_m t)$	0.562
Universal	"OK Google" commands	≤ 0.562

Table 4.2: The adversary can easily disturb the smartphone output (existential injection), and inject human speech (universal injection). Selective injections of sines are less precise than exponentials of the same frequency.

4.3 Security Evaluation of a Smartphone Microphone

This section illustrates how the security definitions can be used to determine the security level of a commercial off-the-shelf embedded device. Section 4.3.1 first introduces an algorithm to calculate the critical selective security threshold ϵ_c^w against a target waveform $w(t)$. Section 4.3.2 then applies the algorithm to measurements taken from a smartphone microphone. Finally, Section 4.3.3 comments on universal security by showing that the smartphone is vulnerable to complex "OK Google" command injections. The results of the experiments performed in this section are summarized in Table 4.2. Appendix A contains additional experiments for further characterization of the smartphone's ADC.

4.3.1 Algorithm for Selective Security Thresholds

This section introduces an algorithm to calculate the critical selective security threshold ϵ_c^w of a system against a target waveform $w(t)$, using a transmitted signal $v(t)$. The first step in the algorithm (summarized in pseudocode as Algorithm 1) is to determine the noise distribution. To that end, N measurements of the system output $\tilde{s}_f(t)$ are collected during the injection, and one of them is designated as the *reference* signal. $1 \leq k \leq N-2$ of them are then chosen to calculate the noise (*estimation* signals), while the remaining measurements are used to verify the calculations (*validation* signals).

The algorithm first removes any DC offset and re-scales the measurements so that the Root-Mean-Square (RMS) voltages of the signals are the same. The repeated measurements are then phase-aligned, and the distance between the reference signal and the estimation signals is calculated. The average of this distance should be very close

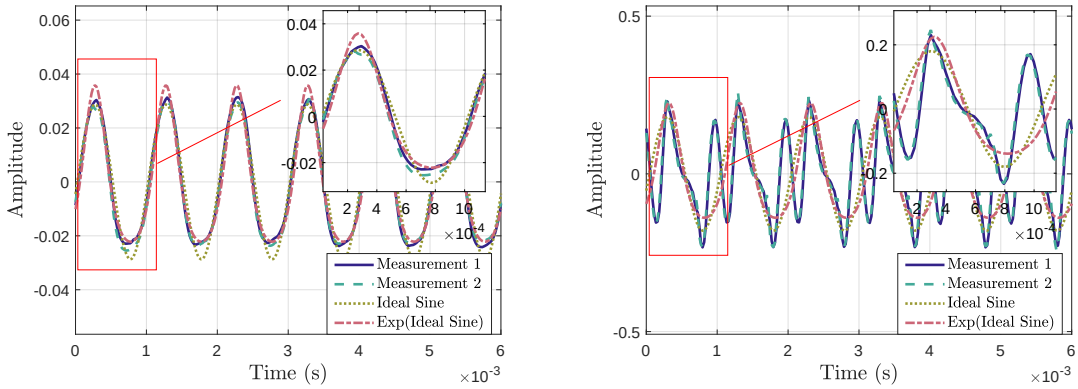
Algorithm 1 Determining the Critical Selective Security Threshold

procedure FINDCRITICALEPSILON($measured$, $ideal$, $sigma$)
  $errors \leftarrow |measured - ideal|/sigma$ ▷ Calculate normalized absolute errors
  $lo \leftarrow 0.5$ ▷ Probabilities need to be between 0.5 and 1
  $hi \leftarrow 1$
  while $lo < hi$ **do**
    $mid \leftarrow (lo + hi)/2$ ▷ mid represents $(2-\epsilon)/2$
    $n_{inv} \leftarrow \text{ppf}((1 + mid)/2)$ ▷ Percentile point function
    $p_{error} \leftarrow \text{length}([x \geq n_{inv} : x \in errors])/\text{length}(errors)$
    if $mid - \delta \leq p_{error} \leq mid$ **then** ▷ Threshold $\delta = 10^{-4}$
      return $2 - 2 * mid$ ▷ Break out if sufficiently close
    else if $p_{error} < mid$ **then**
      $hi \leftarrow mid$
    else
      $lo \leftarrow mid$
procedure COMPARE($\boldsymbol{measurements}$, $ideal$) ▷ Use repeated $\boldsymbol{measurements}$
  $ref \leftarrow \text{detrend}(\boldsymbol{measurements}[0])$ ▷ Pick first series as reference, remove DC
  $\boldsymbol{estimating} \leftarrow \text{align}(\text{scale}(\text{detrend}(\boldsymbol{measurements}[1:]), ref), ref)$
  $\boldsymbol{errors} \leftarrow [(measured - ref)\ \forall measured \in \boldsymbol{estimating}]$
  $\sigma_{noise} \leftarrow \text{stddev}(\boldsymbol{errors})$ ▷ Calculate noise from estimation measurements
  $ideal \leftarrow \text{align}(\text{scale}(\text{detrend}(ideal), ref), ref)$
  return FindCriticalEpsilon(ref, $ideal$, σ_{noise})

to 0, as the signals are generated in the same way. However, the standard deviation σ is non-zero, so the noise can be modeled as following a zero-mean normal distribution $n(t) \sim N(0, \sigma^2)$. The critical threshold between the reference signal and any target *ideal* waveform $w(t)$ can then be found by first detrending, scaling, and aligning the ideal signal to the reference waveform, as with the estimation signals. The errors (distance) between the ideal and the reference signal are then calculated. Finally, a binary search for different values of ϵ reveals the largest ϵ for which Equation (4.6) does not hold: this is the critical threshold ϵ_c^w. To calculate the inverse of the noise, the percentile point function $\text{ppf}(\epsilon)$ is used, as it satisfies $N^{-1}(\epsilon) = \text{ppf}((1 + \epsilon)/2)$. Note that since for a given adversarial transmission $v(t)$, $\epsilon_c = 1 - \epsilon_c^0$ (Equation (4.7) of Section 4.2.2), the same algorithm can be used to calculate a lower bound on the critical universal threshold ϵ_c as well.

4.3.2 Existential and Selective Injections into a Smartphone

This section demonstrates how the proposed algorithm can be used in a realistic setup using a Motorola XT1541 Moto G3 smartphone running Android 6.1. Amplitude-



(a) $V_{RMS}^{Adv} = 0.2\,\text{V}$, $f_c = 200\,\text{MHz}$ **(b)** $V_{RMS}^{Adv} = 0.9\,\text{V}$, $f_c = 25\,\text{MHz}$

Figure 4.3: Clean (a) and Distorted (b) waveforms injected into the smartphone, with ideal sine and exponential sine functions for comparison.

modulated $f_m = 1\,\text{kHz}$ signals are directly injected into the headphone jack of the phone using a Rohde & Schwarz SMC100A/B103 generator. Using the "Audio Recorder" v21.20 application by Sony Mobile Communications, $N = 10$ measurements of 2^{15} sample points per run are collected. The data is recorded at a frequency of $f_s = 44.1\,\text{kHz}$ in a $[-1, 1]$ dimensionless range through the Advanced Audio Coding (AAC) standard. The carrier frequency is initially chosen to be $f_c = 200\,\text{MHz}$ using an output level of $V_{RMS}^{Adv} = V_{PK}^{Adv}/\sqrt{2} = 0.2\,\text{V}$. This injection is demodulated well by the smartphone and has a similarity of over 0.98 compared to a pure 1 kHz tone ("clean" waveform).[11] The second injection uses $f_c = 25\,\text{MHz}$, $V_{RMS}^{Adv} = 0.9\,\text{V}$, and has a similarity of less than 0.55 to the ideal tone ("distorted" waveform). Example measurements of these signals and "ideal" signals (see below) are shown in Figure 4.3.

The algorithm first calculates the noise level using the reference signals. As expected, the error average is very close to zero (usually less than 10^{-6}), while the standard deviation σ is noticeable at around 0.0015. When taking the reference signal as the target signal $w(t)$, the critical selective thresholds are close to 1. In other words, even if the injected waveforms do not correspond to "pure" signals, the adversary can inject them with high fidelity: the system is not selectively secure against them with high probability.

Two waveforms are evaluated as the signal $w(t)$ that the adversary is trying to inject: a pure 1 kHz sine wave, and an exponential of the same sine wave. The averages and

[11] The similarity metric is based on the Pearson Correlation Coefficient (PCC). See Appendix A.

Waveform	Validation	Ideal Sine	$e^{\text{Ideal Sine}}$	$w(t) \neq 0$
Clean	$(0.98, 0.03)$	$(0.56, 0.04)$	$(0.75, 0.06)$	$(0.89, 0.01)$
Distorted	$(0.95, 0.09)$	$(0.31, 0.05)$	$(0.34, 0.05)$	$(0.71, 0.04)$

Table 4.3: Mean and standard deviation (μ, σ) of critical selective thresholds ϵ_c^w for different target signals $w(t)$. Injections using the clean waveform are always more successful than with the distorted waveform. Validation signals are injected with high fidelity, and are better modeled by an exponential rather than a pure sine.

standard deviations for the calculated thresholds over all combinations of k and reference signals are shown in Table 4.3. As one would expect, the thresholds for the distorted waveform are much lower than the values for the clean waveform: the signal is distorted, so it is hard to inject an ideal signal. The exponential function also proves to be a better fit for the demodulated signal and can better explain the harmonics. Table 4.3 also includes the critical universal injection threshold based on the two waveform injections. This threshold is much higher for both waveforms, as injections disturb the ADC output sufficiently, even when the demodulated signal is not ideal.

4.3.3 Universal Injections into a Smartphone

This section demonstrates that the smartphone is vulnerable to the injection of arbitrary commands, which cause the smartphone to behave as if the user initiated an action. Using the setup of Section 4.3.2, a modulated recording of "OK Google, turn on the flashlight" is injected into the microphone port. Two aspects are evaluated: (a) whether the voice command service is activated in response to the "OK Google" portion; and (b) whether the desired action is executed. Measurements are repeated ten times, each time amplitude-modulating the command at a depth of $\mu = 1.0$ with $V_{RMS}^{Adv} = 0.6\,\text{V}$ on 26 carrier frequencies f_c: 25 MHz, 50 MHz, and $100 - 2{,}400$ MHz at a step of 100 MHz. The voice-activation feature ("OK Google") works with 100% success rate (10/10 repetitions) for all frequencies, while the full command is successfully executed for 23 of the 26 frequencies tested (all frequencies except $f_c \in \{1.3, 2.0, 2.4\text{GHz}\}$). Raising the output level to $V_{RMS}^{Adv} = 0.9\,\text{V}$ increases success rate to 25/26 frequencies. Only $f_c = 2.4\,\text{GHz}$ does not result in a full command injection, possibly because the Wi-Fi disconnects in the process.

The above injections are repeated with five further commands to: (1) call a contact; (2) text a contact; (3) set a timer; (4) mute the volume; and (5) turn on airplane mode. The results remain identical, regardless of the actual command to be executed. As a result, all carrier frequencies which are not severely attenuated by H_C (e.g., when coupling to the user's headphones) are vulnerable to injections of complex waveforms such as human speech.

4.4 Qualitative Assessment of ADC Response H_A

As explained in Section 4.1, an adversary trying to inject signals remotely into a system typically needs to transmit modulated signals over high-frequency carriers. As H_C is unique to each circuit and needs to be re-calculated even for minor changes to its components and layout [100], to determine a system's vulnerability, it is important to understand the behavior H_A of the ADC used.

This section does so by exploring the demodulation characteristics of different ADCs. As shown in Figure 4.4, a Rohde & Schwarz SMC100A/B103 signal generator is responsible for creating the modulated signals. These signals are injected directly into the ADC, while additional experiments with an amplifier or with remote transmissions are performed in Appendix A. An Arduino Uno (ATmega328P microcontroller) interfaces with the ADC over the appropriate protocol. A computer is responsible for collecting the measurements from the microcontroller over the Universal Asynchronous Receiver/Transmitter (UART), and for controlling the signal generator over the Virtual Instrument Software Architecture (VISA) interface.

Experiments are conducted with six ADCs from four manufacturers (Texas Instruments, Analog Devices, Atmel, and Xilinx) in different packages: some are part of the silicon in other Integrated Circuits (ICs), while others are standalone surface-mount or through-hole chips. Delta-Sigma ($\Delta\Sigma$), half-flash, and Successive Approximation (SAR) ADCs are tested, with sampling rates f_s ranging from a few Hz to several MHz, and resolutions between 8 and 24 bits. Table 4.4 summarizes these properties along with the $-3\,\mathrm{dB}$ cutoff frequency f_{cut}, calculated using the parameters found in the ADC datasheets. Further details about these ADCs can be found in Appendix A.

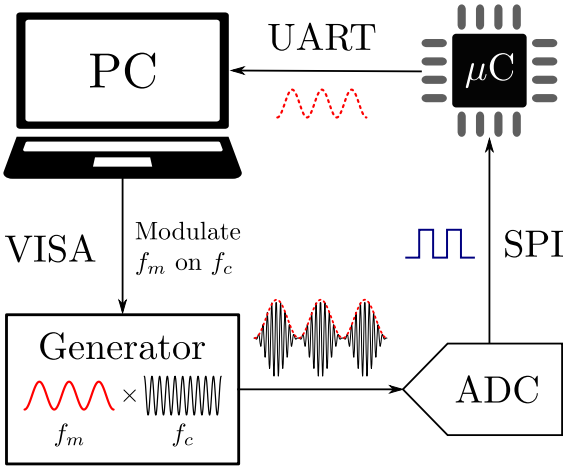


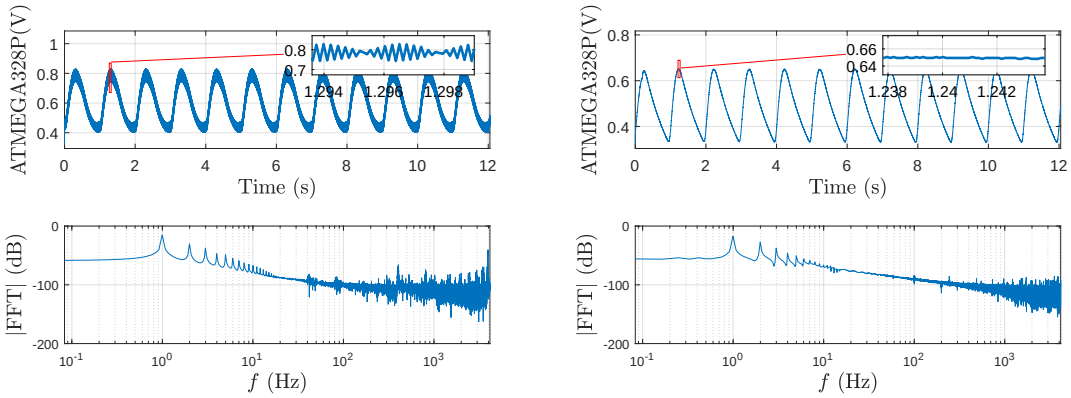
Figure 4.4: To test the demodulation characteristics of an Analog-to-Digital Converter (ADC), amplitude-modulated signals are directly injected into it using a Rohde & Schwarz SMC100A/B103 signal generator. The generator is controlled over the Virtual Instrument Software Architecture (VISA) interface, while measurements are transferred to the computer for analysis via an Arduino microcontroller's Universal Asynchronous Receiver/Transmitter (UART) interface.

ADC	**Protocol**	**Package**	**Type**	**Bits**	**Max f_s**	f_{cut}
TLC549	SPI	DIP	SAR	8	40 kHz	2.7 MHz
ATmega328P	(Internal)	Integrated	SAR	10	76.9 kHz	0.1-11.4 MHz
Artix 7	(Internal)	Integrated	SAR	12	1 MHz	5.3 MHz
AD7276	SPI	TSOT	SAR	12	3 MHz	66.3 MHz
AD7783	SPI	TSSOP	ΔΣ	24	19.79 Hz	[50,60 Hz]
AD7822	Parallel	DIP	Flash	8	2 MHz	128.4 MHz

Table 4.4: The Analog-to-Digital Converters (ADCs) used in the experiments cover a range of different properties. More details about these ADCs are given in Table A.1 of Appendix A.

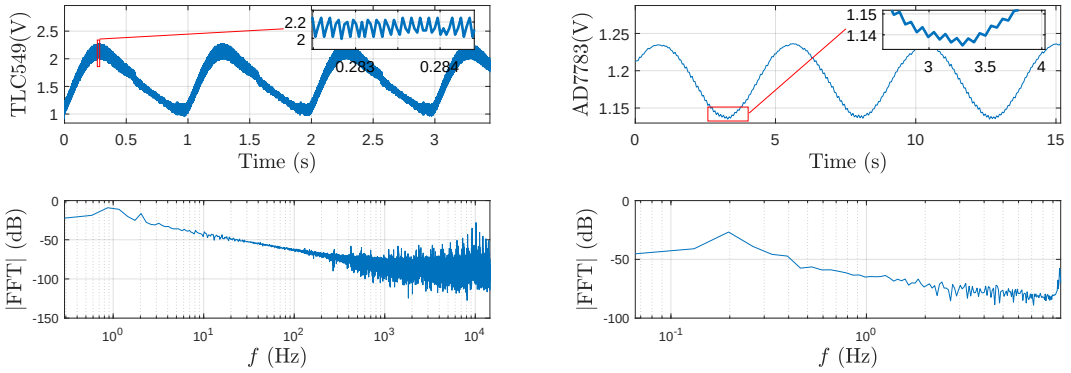
In the experiments of this section, sinusoidal signals of different frequencies f_m are amplitude-modulated on different carrier frequencies f_c, and are then injected into the ADCs. In other words, the intended signal is assumed to be $w(t) = \sin(2\pi f_m t)$, the sensor signal $s(t) = 0$ is considered to be absent, and the goal is to evaluate how "close" $w(t)$ is to the ADC output $\tilde{s}_f(t)$. Some typical results for each ADC are summarized, with more details in Appendix A.

ATmega328P: Figure 4.5 presents two example measurements for the ATmega328P's integrated ADC, both in the time domain and in the frequency domain. The input to the ADC is a $f_m = 1$ Hz signal modulated over different high-frequency carriers. As shown in the frequency domain (bottom of Figure 4.5), the fundamental f_m dominates all



(a) $f_c = 10\,\text{MHz}$ **(b)** $f_c = 80\,\text{MHz}$

Figure 4.5: ATmega328P output for an amplitude-modulated input over different carrier frequencies f_c (power $P = 0\,\text{dBm}$, signal frequency $f_m = 1\,\text{Hz}$, and modulation depth $\mu = 0.5$). The demodulated signal exhibits the correct fundamental frequency, but includes strong harmonics. It also contains a high-frequency component, which is attenuated as f_c increases.

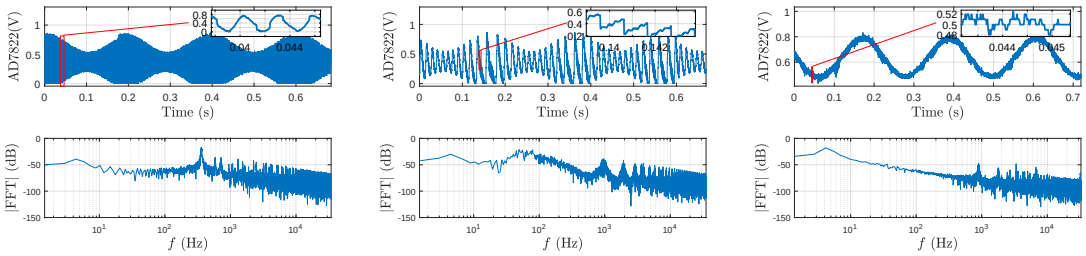


(a) TLC549: $f_c = 80\,\text{MHz}$, $\mu = 0.5$, $f_m = 1\,\text{Hz}$ **(b)** AD7783: $f_c = 40\,\text{MHz}$, $\mu = 1.0$, $f_m = 10\,\text{Hz}$

Figure 4.6: Measurements on (a) TLC549 and (b) AD7783 Analog-to-Digital Converters (ADCs) for an injection power of $P = 5\,\text{dBm}$. Both ADCs demodulate the signal, but contain harmonics and high-frequency components. Moreover, the AD7783 signal is aliased.

other frequencies, so the attacker is able to inject the intended 1 Hz signal into the output of the ADC. However, the output at both carrier frequencies f_c has strong harmonics at $2f_m, 3f_m, \ldots$ Hz, indicating that the resulting signal is not pure. Moreover, there is a residual high-frequency component, which is attenuated as f_c increases. Finally, there is a frequency-dependent DC offset, while the peak-to-peak amplitude of the measured signal decreases as the carrier frequency increases.

TLC549: The TLC549 ADC (Figure 4.6a) also demodulates the injected signal, but still contains harmonics and a small high-frequency component. Its output is more



(a) $f_c = 79.4933\,\text{MHz}$ **(b)** $f_c = 79.4936\,\text{MHz}$ **(c)** $f_c = 79.4937\,\text{MHz}$

Figure 4.7: AD7822 outputs for power $P = -5\,\text{dBm}$, signal frequency $f_m = 5\,\text{Hz}$, and depth $\mu = 0.5$. Signal demodulation requires a fine-tuned carrier frequency f_c.

sawtooth-like rather than purely sinusoidal.

AD7783: As the AD7783 (Figure 4.6b) only has a sampling frequency of $f_s = 19.79\,\text{Hz}$, aliasing occurs when the baseband signal exceeds the Nyquist frequency $f_s/2$. Specifically, for a baseband $f_m = 10\,\text{Hz}$, the frequency dominating the measurements is $2f_m - f_s = 20 - 19.79 = 0.21\,\text{Hz}$, with a high-frequency component of $f_s - f_m = 9.79\,\text{Hz}$.

AD7822, AD7276 & Artix 7: The three remaining ADCs contain strong high-frequency components which dominate the low-frequency signal. Their outputs appear to be amplitude-modulated, but at a carrier frequency which is below the ADC's maximum sampling rate. However, with manual tuning, it is possible to remove this high-frequency component, causing the ADC to demodulate its input. This is shown for the AD7822 Flash ADC in Figure 4.7, where the carrier f_c is tuned in steps of 100 Hz.

Conclusion: The results of the above experiments lead to the following observations:

1. **Generality:** The six ADCs demodulated signal injections at multiple carrier frequencies, matching the theoretical expectations of Section 4.1. As the chips tested cover the major ADC types and a range of resolutions and sampling frequencies, the conclusions drawn should also be valid for other ADCs.

2. **Filtering:** Although all ADCs exhibited Low-Pass Filter (LPF) characteristics, the maximum vulnerable carrier frequency was multiple times the cutoff frequency of the ADC sampling mechanism. This extended the frequency range that an adversary could use for transmissions in attacking the system.

3. **Power:** Too much power in the input of the ADC can saturate or clip the measured signal, while too little power results in zero or noise-like output. As a result, the adversary needs to select the transmission power level carefully.

4. **Carrier Frequency:** Some ADCs were vulnerable at any carrier frequency that is not severely attenuated by the sample-and-hold mechanism. For others, high-frequency components dominated the intended signal frequency f_m. Even then, carefully-chosen carrier frequencies resulted in a demodulated ADC output.

4.5 Discussion

This section discusses how the framework introduced (model and definitions) can inform design choices. To start, choosing the right ADC directly impacts the susceptibility to out-of-band signal injection attacks. As shown in Section 4.4, some ADCs distort the demodulated output, and are thus more resilient to clean sinusoidal injections. Moreover, other ADCs require fine-grained control over the carrier frequency of the injection. As the adversarial signal is transformed through the circuit-specific transfer function H_C, the adversary may not have such control, resulting in a more secure system.

Having chosen the appropriate ADC based on cost, performance, security, or other considerations, a designer needs to assess the impact of H_C. Prior work has shown that even small layout or component changes affect the Electromagnetic Interference (EMI) behavior of a circuit (Chapters 2 and 3). Since the ADC behavior can be independently determined through direct power injections, fewer experiments with remote transmissions are required to evaluate the full circuit behavior and how changes in the circuit's topology influence the system's security.

The selective security definition and algorithm address how to determine the vulnerability of a system against specific waveforms. Universal security, on the other hand, allows designers to directly compare the security of two systems for a fixed adversarial voltage budget through their critical universal security thresholds. Moreover, given a probability (threshold) ϵ, one can calculate the critical universal voltage level, which is the maximum output level for which a system is still universally ϵ-secure.

The smartphone case study showed that the framework can be used in practice with real systems, while the "OK Google" experiments demonstrated that less-than-perfect injections of adversarial waveforms can have the same effect as perfect injections. This is because there is a mismatch between the true noise level of a system and the worst-case noise level that the system expects. In other words, injections worked at all carrier frequencies, even when the demodulated output was noisy or distorted. This is a deliberate, permissive design decision, which allows the adversary to succeed with a range of different and noisy waveforms $w(t)$, despite small amplitudes and DC offsets.

It should be noted that the model and definitions are not restricted to electromagnetic injection attacks. For instance, H_C can account for such imperfections in the sensors themselves (e.g., Micro-Electro-Mechanical System (MEMS) gyroscopes and accelerometers), attenuating injection frequencies which are not close to the resonant frequencies. The system model also makes it easy to evaluate countermeasures and defense mechanisms in its context. For example, shielding increases the attenuation factor of H_C, thereby increasing the power requirements for the adversary (Section 4.1). Alternatively, an LPF before the ADC and/or amplifier changes H_A, and attenuates the high-frequency components which would induce non-linearities. However, as explained in Chapter 3, moving the pre-amplifier, LPF, and ADC into the same IC package does not always fully eliminate the vulnerability to out-of-band signal injection attacks. This is because the channel between the analog sensor and the ADC fundamentally cannot be authenticated.

Overall, as Chapter 3 identified, a combination of improved hardware, filtering, shielding, and sampling will be necessary to protect against out-of-band signal injection attacks by altering the behavior of both H_A and H_C.

4.6 Summary

This chapter introduced a framework containing a system model (Section 4.1) and mathematical definitions (Section 4.2) to understand security in the context of out-of-band signal injection attacks. The system and adversary model abstracts away from specific environments and circuit designs, and presents strong adversaries who are only limited by their transmission power. The framework also makes it easy to discuss and evaluate

countermeasures through the proposed model, and covers different types of out-of-band signal injection attacks. The definitions of security and the algorithm introduced can be used in practice to evaluate commercial off-the-shelf embedded systems, such as smartphones: one can calculate the "critical" security thresholds of a device, which express how successfully an adversary can inject signals into the system (Section 4.3). This chapter also investigated the response of several ADCs to malicious injections, showing that unintentional hardware properties cause their demodulation characteristics (Section 4.4). This chapter finally discussed how the security framework can be used to inform circuit design choices and interpret potential defense mechanisms (Section 4.5). Overall, in response to the emerging out-of-band signal injection threat, this chapter presented a unified reporting methodology, where security can be quantified and compared through the proposed security definitions.

He doth hang the greatest weight upon the smallest wires.

— Francis Bacon

5 FPGA Long-Wire Information Leakage

Contents

The deployment of Field-Programmable Gate Arrays (FPGAs) in safety-critical applications and their proliferation in public cloud infrastructures (Section 2.2) highlights the need for trustworthy FPGA devices. However, the outsourcing of Intellectual Property (IP) cores and prospective multi-tenant setups raise concerns about potentially malicious circuits, which can cause or detect information leakage through covert- or side-channel attacks.

This chapter shows that Vertical Long (`VLONG`) wires influence the delay of nearby long wires: if a `VLONG` carries a logic 1, the delays of nearby `VLONG`s are slightly shorter than when the same `VLONG` carries a logic 0. This difference in delay allows cores sharing the same reconfigurable FPGA fabric to communicate, even when they are not directly connected. Unlike prior work which depends on fast-changing signals [99, 155, 384], the phenomenon of long-wire leakage persists even when the driven value remains constant. In summary, this chapter has the following contributions:

1. It identifies a new system and adversary model that is becoming increasingly relevant for hybrid and multi-tenant FPGA deployments (Section 5.1).

2. It proposes an experimental setup for efficient, on-chip measurement of long-wire leakage without any modifications to the stock prototyping boards (Section 5.2).

3. It demonstrates that the strength of the phenomenon is only influenced by the values carried by the long wire during the period of measurement, and not by the values that precede or follow it (Section 5.3.1).

4. It shows that the extent of the leakage depends on the Hamming Weight (HW) of the transmitted value, and not its switching activity (Section 5.3.2). This differentiates the underlying mechanism from other effects previously reported in the literature (Section 5.3.3).

5. It proves that the relative effect remains constant regardless of the measurement duration, but that longer measurement durations increase the absolute effect, making long-wire leakage more measurable (Section 5.4.1).

6. It conducts experiments on six Xilinx FPGA families and thirteen different boards to characterize the effect of longer overlaps on the strength of the reported phenomenon (Section 5.4.2). `VLONG`s on every board tested leak information about their state, although the strength of the effect varies.

7. It verifies that the leakage is independent of the absolute and relative placements of the transmitter and receiver, and that the direction of propagation also has minimal effect on the measurements (Section 5.5).

8. It validates that only physical separation of the receiver and transmitter can hinder communication, since dynamic activity on the device does not sufficiently counteract the leakage (Section 5.6).

9. It establishes that simultaneous long-wire transmissions are measurable and increase the accuracy and bandwidth of the covert channel (Section 5.7).

10. It creates a practical covert channel with a bandwidth upwards of 6 kbps and 99.9% accuracy, even in the presence of environmental noise (Section 5.8.1).

11. It eavesdrops on signals which are kept constant for as low as 1.3 µs with an accuracy of more than 98.4% when using the long-wire leakage for a side-channel attack (Sections 5.8.2 and 5.8.3).

12. It discusses additional considerations regarding long-wire leakage, including alternative applications and potential defense mechanisms (Section 5.9).

In summary (Section 5.10), this chapter introduces a new source of information leakage on FPGA devices, characterizes it across multiple experimental parameters, and demonstrates how to remotely exploit it in practice.

5.1 System and Adversary Model

As explained in Section 2.2.1, FPGA designs often contain IP cores sourced from third-parties. These may come in pre-placed and pre-routed format (“macros”) to meet timing constraints and reduce compilation time [166, 180–182]. However, as shown in Figure 5.1, some of these cores may contain unwanted functionality. For instance,

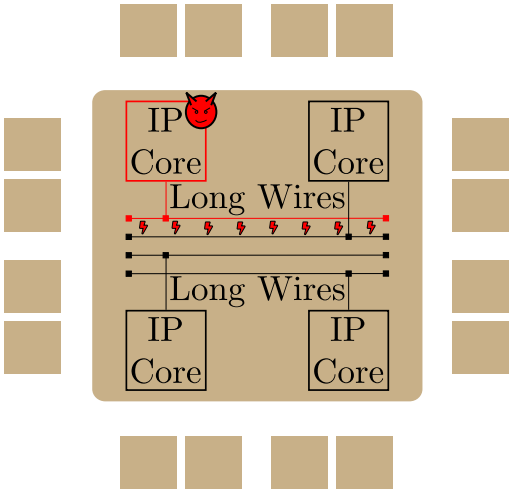


Figure 5.1: High-level system model for multi-tenant attacks: physically-unconnected users or Intellectual Property (IP) cores share common Field-Programmable Gate Array (FPGA) resources (e.g., routing), and may contain malicious functionality.

a covert channel between IP cores of different security guarantees [141, 142, 293] can break separation of privilege. The same is true of multi-tenant setups: a user may be malicious and attempt to infer information about other users despite logical isolation of their designs in hybrid (Section 2.2.1) or cloud (Section 2.2.2) FPGAs.

In order to achieve the goals of covert communication or data exfiltration, the adversary is allowed to insert one or more IP cores into the design, for instance through social engineering, subversion of the compilation tools [141, 169], or by simply renting (part of) an FPGA in multi-tenant clouds. Note that because designs are often verified, for instance to detect Hardware Trojans (HTs) [60, 155, 185, 321, 384], the adversarial cores are assumed to provide legitimate functionality. Indeed, the transmitter and receiver employed have dual use, hiding their malicious functionality in their routing, not their actual combinational and sequential logic. As a result, unlike conventional backdoors, these IP cores would pass timing, netlist, and bitfile verification: since they do not require additional gates, they present a bigger challenge to designers.

Although IP cores are logically isolated (i.e., they cannot be directly connected), adversaries are allowed to define their internal placement and routing. However, adversaries do not have physical access to the board, and can thus not alter the environmental conditions or physically modify the FPGA board in any way. In the experiments of this chapter, no attempt has been made to control for temperature beyond the standard heatsink and fan

already mounted on the FPGA (if any). Moreover, no special voltage regulator is used, and no shielding has been added to the chip or the connected wires. Such modifications reduce noise and improve the stability of measurements [185, 221, 313, 389]. Despite these less-than-ideal conditions, this chapter shows that the leakage remains strong and detectable, even in the presence of power and temperature fluctuations: by using long wires, adversaries can eavesdrop on nearby blocks they do not control to infer their state, or establish covert channels between two co-operating IP cores under their control.

It should be noted that a potential issue with pre-placed and pre-routed IP cores is that they are specific to an FPGA generation (but can be used in different devices within the same family). However, as Section 5.4.2 shows, long-wire leakage persists across many generations of Xilinx chips. As a result, an adversary can provide an IP generation wizard that provides different routing for different families, and dynamically chooses the placement of the cores. In fact, as Section 5.5 shows, the location of the actual logic and wires is not important, so the adversary merely needs to ensure that the transmitter and the receiver use long wires which are adjacent.

Adversaries who only manually route their cores but let the tools pick their placement can still succeed with non-negligible probability. Assume that the FPGA has N long wires in total, that the transmitter and the receiver use T and R long wires respectively, and that the signal can be recovered from w nearby wires. Then, the probability that at least one segment of the transmitter is adjacent to a segment of the receiver is $(R+T-1)\cdot w/N$, assuming the tools place the two cores at random. For most FPGAs used in this chapter, $N \approx 8{,}500$ (equal to the number of Configurable Logic Blocks (CLBs)) and $w = 4$ (i.e., the 2 surrounding wires on each side), so with $R = T = 5$, an adversary has a 0.42% chance of success. Since tools do not pick locations at random or spread the logic, the probability of success is higher in practice. Moreover, the adversary can also increase this probability by accessing relatively unique elements such as Block Random-Access Memory (BRAM), Digital Signal Processing (DSP) blocks, or embedded processors on the FPGA fabric. For example, the same devices have fewer than 150 DSP slices and 300 BRAM blocks. As a result, accessing them reduces the number of possible placements for the attacker's cores.

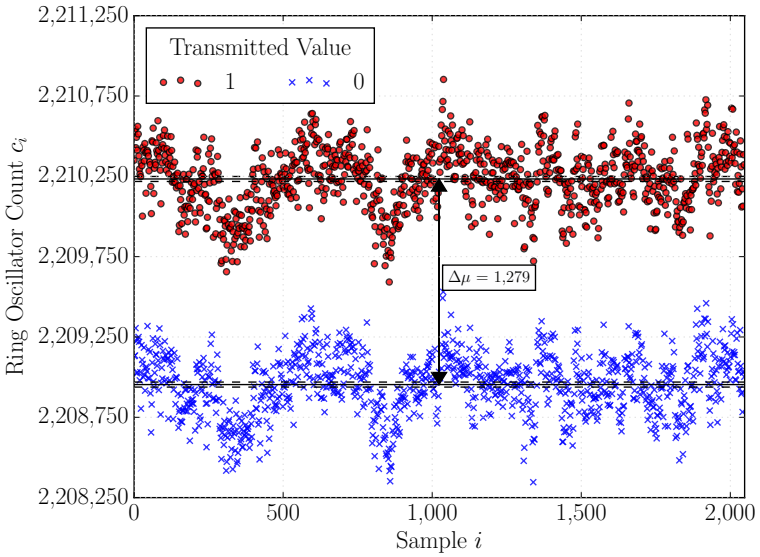


Figure 5.2: Ring oscillator counts and 99% confidence intervals for a transmitter and receiver using five long wires each on a Virtex 5 device. The receiver is able to distinguish between signals using a simple threshold, despite environmental noise.

5.2 Experimental Setup

This section details the experimental setup to characterize long-wire leakage, where the delay of long wires depends on the logic state of nearby wires. To do so, Ring Oscillators (ROs) with `VLONG`s between two of their stages are employed: a difference in the delay of the long wires translates to a change in the delay of the RO frequency. This change can be measured by counting the number of transitions of the RO value within a fixed measurement period. Figure 5.2 exemplifies the dependence of the RO counts on the logic state of the nearby long wires ("transmitters"). The red circles and blue X marks are RO counts when the transmitter wire carries a logic 1 or 0 respectively. The difference between the counts when transmitting 1s and 0s is clear, despite fluctuations due to changes in environmental temperature and power supply voltage.

The high-level experimental setup is depicted in Figure 5.3. It consists of the long-wire transmitter and the ring oscillator receiver (Section 5.2.1) as well as a measurement component (Section 5.2.2), which works independently of any specific communication channel implementation. This component generates the signal to be transmitted, samples the RO counter, and transfers the data to a computer for analysis. Finally, the metric to

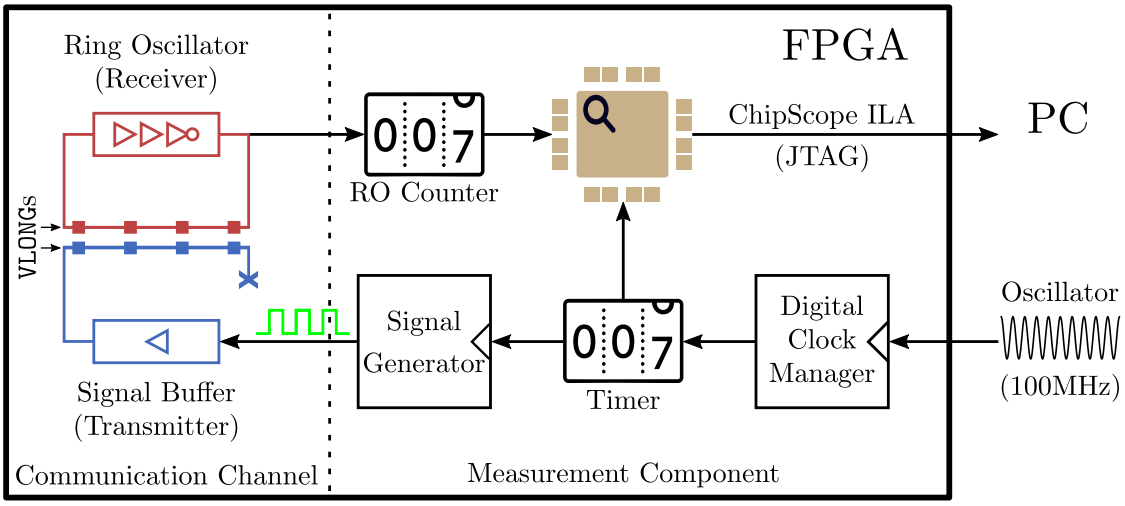


Figure 5.3: Experimental setup. The transmitter and the receiver are logically isolated, and use adjacent long wires to communicate. The measurement component updates the signal to the transmitter every 2^t clock cycles, and measures the signal's effect on the ring oscillator frequency. The ring oscillator counts are transferred to the computer using Xilinx's ChipScope Integrated Logic Analyzer (ILA) core for further analysis and processing.

estimate long-wire leakage is discussed in Section 5.2.3.

The bulk of the experiments in this chapter are conducted on three Virtex 5 XUPV5-LX110T (ML509) evaluation boards. These boards include a heatsink and a fan, but temperature is otherwise not controlled in any way. Similarly, and in accordance with the threat model of Section 5.1, the boards are unmodified and contain their stock voltage regulators. Each experiment is run on every device five times, collecting 2,048 data points per run, with results reported at the 99% confidence level.

5.2.1 Transmitter and Receiver

A minimal *transmitting* circuit is employed to illustrate the information leakage: the transmitter consists of a buffer Lookup Table (LUT) that drives one or more long-wire segments connected end-to-end. Although the term transmitter is used for brevity, as will be demonstrated in Section 5.8, the conclusions drawn are valid whether transmissions are intentional (covert channel) or not (side channel). The *receiving* circuit uses long wires that are adjacent to the transmitter's wire segments as part of a three-stage ring oscillator (Section 2.2.5). The delays of the wires directly influence the frequency of oscillation, which can be estimated by feeding the output of one of the RO stages

to a counter in the measurement component. Although the RO used in this chapter contains one inverter and two buffer stages, alternative RO designs can also detect the long-wire leakage (see Chapter 6).

The receiver and the transmitter are initially fixed on the device, but their location is varied in Section 5.5 to show that it does not influence the strength of the leakage. The number of `VLONG`s used is varied in Section 5.4.2, showing that the effect becomes more pronounced the longer the overlap is.

5.2.2 Measurement Component

The measurement component generates the signals to be transmitted and estimates the RO frequency through a counter. A new trigger event is produced every 2^t clock ticks. At every trigger, the RO counter is read and reset, and a new value is presented to the transmitter. For most experiments, the signal generator simply alternates between 0s and 1s, but other patterns are tested in Section 5.3.

The 100 MHz system clock is driven by a Digital Clock Manager (DCM) to ensure clock quality, although this is not necessary as Chapter 6 shows. In most experiments, $t = 21$ (corresponding to 2^{21} clock ticks, or 21 ms), but t is varied in Section 5.4.1 to explore the accuracy vs. time tradeoffs. The sampled data is transferred to a computer for analysis through Xilinx's ChipScope Integrated Logic Analyzer (ILA) core. No other Input/Output (I/O) is used until the experiments of Section 5.6.

Unlike the communication channel of Section 5.2.1, the measurement component is not hand-placed or hand-routed, due to the large number of experiments performed. Although the measurement logic could influence the RO frequency [206], experiments are repeated on multiple locations, with different transmission patterns, and measurements are averaged over relatively lengthy periods of time. Moreover, in some experiments, the Universal Asynchronous Receiver/Transmitter (UART) replaces the ILA core to ensure that the phenomenon is not caused by the core itself. Thus, any effects of the measurement circuitry should influence the transmissions of both zeros and ones equally. This hypothesis is further supported by the experiments of Section 5.6, which show that the channel is only affected by adjacent long wires.

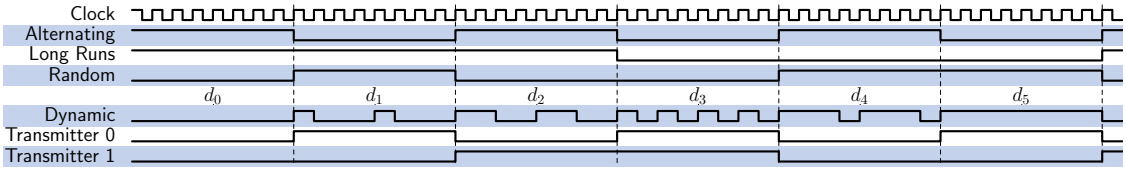


Figure 5.4: Timing diagram for the experimental transmission patterns. Some patterns remain constant within a measurement period (*Alternating, Long Runs, Random*). Others patterns are fast-changing (*Dynamic*). Simultaneous transmissions (*Transmitter 0* and *1*) are also tested.

5.2.3 Evaluation Metric

This section introduces the relevant notation and metric to estimate the strength of the long-wire leakage, with Chapter 6 improving upon it. When a clock of frequency f_{CLK} is sampled every C_{CLK} ticks and a ring oscillator of frequency f_{RO} driving a counter measures C_{RO} ticks, then $f_{RO}/f_{CLK} \approx C_{RO}/C_{CLK}$, with an appropriate quantization error due to the unsynchronized nature of the RO and the system clock. Thus,

$$\frac{f_{RO}^1 - f_{RO}^0}{f_{RO}^1} \approx \frac{C_{RO}^1 - C_{RO}^0}{C_{RO}^1} \tag{5.1}$$

where C_{RO}^i and f_{RO}^i represent the count and respective frequency when the transmitter has value i. As a result, the relative change of frequency can be approximated by using just the measured counts, irrespective of the measurement duration and clock period.

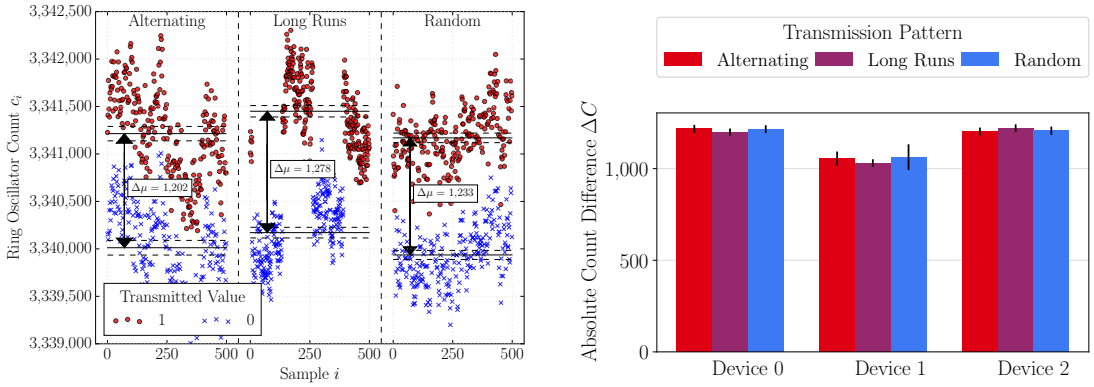
In the basic setup, the transmitter alternates between sending zeros and ones. Let c_i denote the i-th sampled count, so that the pair $p_i = (c_i, c_{i-1})$ always corresponds to different transmitted values, and assume, for the sake of notation clarity, that c_{2i+1} corresponds to a transmission of a logic 1. The quantity

$$\Delta RC_i = \frac{c_{2i+1} - c_{2i}}{c_{2i+1}} \tag{5.2}$$

then indicates the relative frequency change between consecutive transmissions of a zero and a one. Noise can be removed by averaging ΔRC_i over all measurement pairs i, for a metric called the Relative Count Difference (ΔRC).

5.3 Transmitter Patterns

This section investigates the feasibility of a covert channel by showing that only the values carried by the wire during the period of measurement matter, and not the



(a) Example run **(b)** Comparison across devices

Figure 5.5: Measurements using three different static transmission patterns from the timing diagram of Figure 5.4 (*Alternating*, *Long Runs*, and *Random*). (a) shows example absolute Ring Oscillator (RO) measurements for these patterns, while (b) is a comparison of the Relative Count Difference (ΔRC) across devices, with 99% confidence intervals. The magnitude of the effect does not depend on the pattern used, and is similar among different devices.

values that precede or follow it (Section 5.3.1). It also demonstrates that the information leakage gives useful information for side-channel analysis, as RO counts reveal the Hamming Weight (HW) of dynamic signals (Section 5.3.2). Finally, it demonstrates that the phenomenon observed differs from that of prior work, which noted RO slowdowns in response to increased switching activity (Section 5.3.3). The timing properties of patterns tested are shown in Figure 5.4, and are explained in more detail in the respective subsections (simultaneous transmissions are discussed in Section 5.7).

5.3.1 Constant Signals

The default experimental setup uses a slowly alternating signal, where the transmitted value changes every sampling period. This pattern is denoted by *Alternating* in Figure 5.4. This section measures the leakage of other patterns that remain constant within a given measurement period. The first of these greatly slows down the alternation speed of the transmitted signal. This *Long Runs* pattern maintains the same value for 128 consecutive triggers—in essence, testing the effects of long sequences of zeros and ones. The second pattern employs a Linear Feedback Shift Register (LFSR), which produces a pseudo-random pattern of zeros and ones. It is denoted by *Random* in Figure 5.4.

The results of this experiment are shown in Figure 5.5, with a sample of the data in Figure 5.5a, and a comparison across devices in Figure 5.5b. The RO counts remain significantly higher when transmitting a 1 versus a 0, showing that the leakage persists in all three setups. Moreover, the average count difference remains identical, with almost no variability among the patterns, illustrating that the pattern of transmission has no persistent effect on the delay of nearby wires. This fact allows the channel to be used without having to ensure a balanced distribution of transmitted values—a property which is necessary for side-channel attacks.

5.3.2 Dynamic Patterns

This section tests various dynamic patterns to show that the dominating factor in the observed phenomenon is the duration for which the transmitter remains at a logic 1, and *not* the switching activity of the circuit. As a result, even if a signal is not sufficiently long-lived, attackers can still deduce the signal's HW and eavesdrop on signals not under their control. Section 5.8.2 later explains how to use this property to recover secret state such as cryptographic keys through repeated measurements.

The dynamic patterns used are denoted by *Dynamic* in the timing diagram of Figure 5.4. During each sampling period, the transmitter quickly loops through a 4-bit pattern at 100 MHz. The looped pattern is only updated at each new sampling period. For example, for the pattern 1100 (d_2 in Figure 5.4), the transmitter stays high for two 100 MHz clock ticks, then low for two clock ticks, then back to high for two ticks, etc., until the end of the sampling period. The six 4-bit patterns used are: $d_0 = 0000$, $d_1 = 1000$, $d_2 = 1100$, $d_3 = 1010$, $d_4 = 1110$, and $d_5 = 1111$. These patterns respectively have a HW of 0, 25, 50, 50, 75, and 100%, while their switching frequencies are 0, $f = f_{CLK}/4$, f, $2f$, f, and 0 respectively.

Figure 5.6 shows the average RO count C_i for each pattern d_i. The RO frequency increases with the HW, i.e., $C_0 < C_1 < C_2 \approx C_3 < C_4 < C_5$. However, the frequency is otherwise unaffected by the switching transmission activity: the Kolmogorov-Smirnov test suggests that there is no statistically significant difference between the two distributions for d_2 and d_3. Note that the receiver would not able to distinguish between patterns d_2

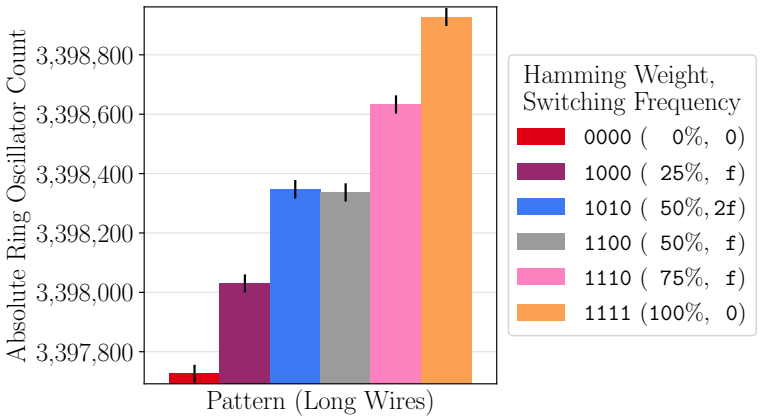


Figure 5.6: Effect of dynamic switching activity using a long-wire transmitter. Counts increase with the Hamming Weight (HW), but not with the switching frequency.

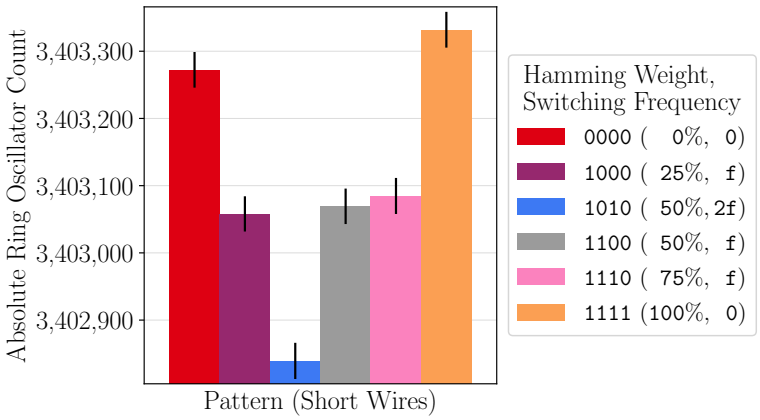


Figure 5.7: Effect of dynamic switching activity without long-wire overlaps. Counts decrease with switching frequency, and are almost unaffected by the Hamming Weight (HW).

and d_3 as a result (or, more generally, any patterns with the same HW), but Section 5.8.2 introduces ways around this limitation.

5.3.3 Local Routing

This section demonstrates that when the transmitter and receiver circuits do not have overlapping long wires, switching activity *decreases* the oscillation frequency of the RO. This reproduces the results reported by prior research on HT detection [155, 384] and also sanity-checks the measurement setup. To test this dependence on the long-wire overlap, the long-wire transmitter is replaced with a buffer of 312 consecutive LUTs packed into 39 CLBs, using only local intra- and inter-CLB routing. The same six dynamic patterns from Section 5.3.2 are then transmitted through the large buffer, with measurements

summarized in Figure 5.7. The ordering of the patterns exactly mirrors their relative switching activity, with the RO counts C_i (corresponding to d_i) decreasing with increased switching activity: $C_3 < C_1 \approx C_2 \approx C_4 < C_0 < C_5$.

The difference between the patterns with the same switching activity d_1, d_2, d_4 is not significant according to the Kolmogorov-Smirnov test, but the count is slightly higher for d_5 compared to d_0, both of which have no switching activity. This suggests that the leakage identified may be present for shorter wires as well, but is considerably weaker, and requires much bigger circuits. Similar results are obtained when testing two transmitting buffers of 328 LUTs each using only local routing. Overall, the results of this section indicate that when the transmitter does not use `VLONG`s that overlap with the receiver, the results of prior work can be reproduced: the observed RO frequency drops in response to large switching activity through multiple redundant buffers.

5.4 Receiver Parameters

This section explores the tradeoffs between the quality of the communication channel and the measurement time (Section 5.4.1). Moreover, it investigates the effect of the length of overlap between the receiver and the transmitter on the strength of the leakage (Section 5.4.2).

5.4.1 Measurement Time

This experiment returns to the alternating pattern of Figure 5.4, but varies the measurement time of 2^t clock cycles by repeatedly quadrupling it. The top of Figure 5.8 shows that the Absolute Count Difference (ΔC) grows linearly with increasing measurement time. Hence, the RO count differences can be amplified proportionally to the duration of the measurement. Moreover, the bottom of Figure 5.8 shows that the Relative Count Differences (ΔRCs) remain approximately constant for measurement periods above 1 ms, in accordance with the theoretical prediction of Equation (5.1). The values for shorter measurement periods are still close, but are far noisier: for short measurement periods, the Absolute Count Differences (ΔCs) are small ($\approx$4 for sampling periods of 82 µs), increasing quantization errors, and making it harder to distinguish between signal

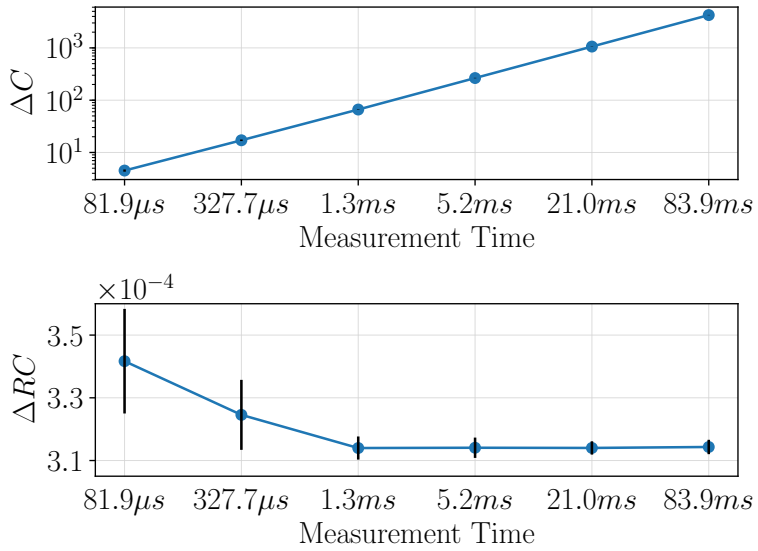


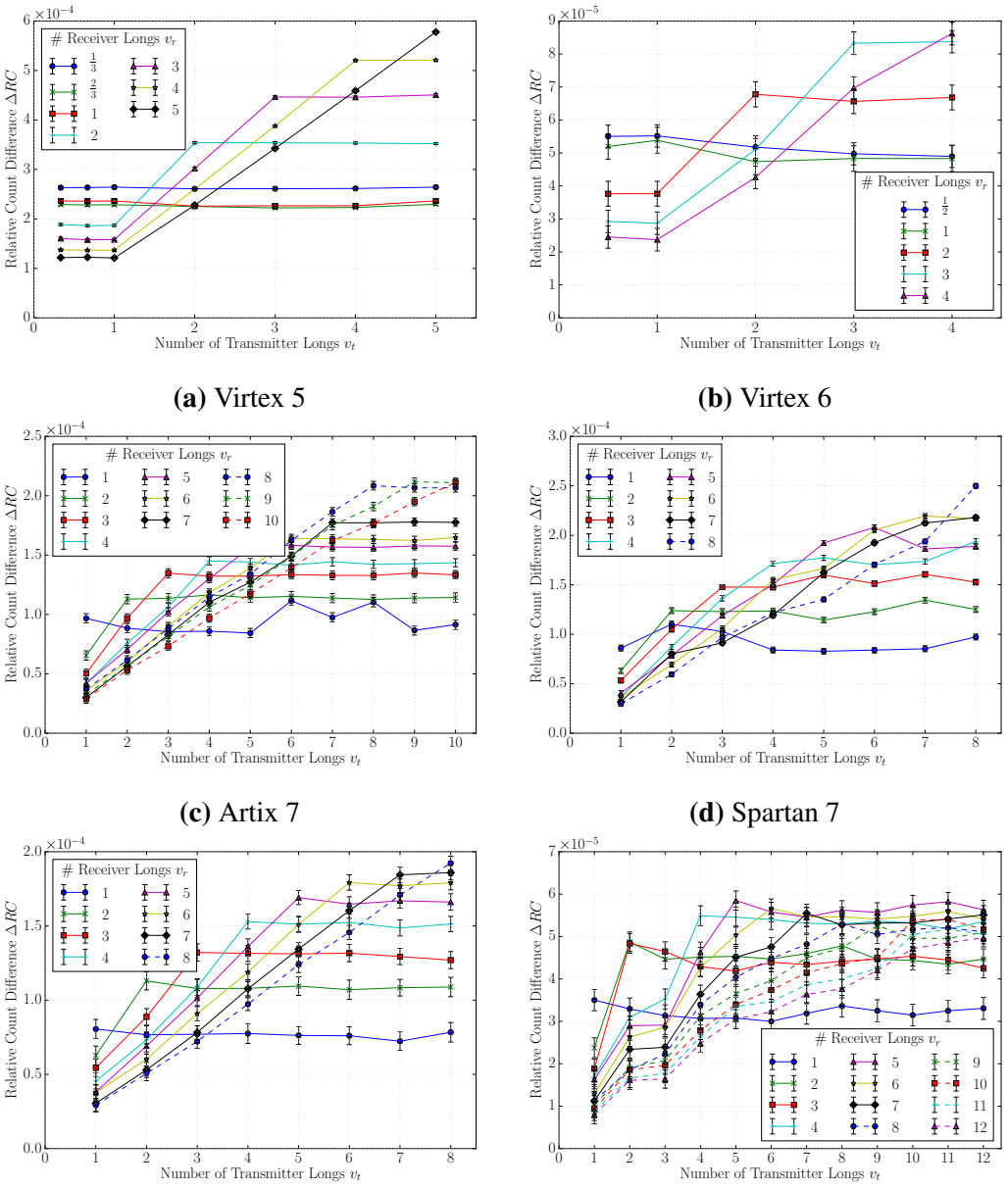
Figure 5.8: Absolute and relative count differences with 99% confidence intervals for various measurement times. For a given transmitter and receiver overlap, the absolute magnitude of the effect as measured using the Absolute Count Difference (ΔC) increases linearly with time. For sufficiently long measurement periods, the Relative Count Difference (ΔRC) remains constant, but short measurement periods increase quantization errors and uncertainty.

and noise. These results indicate that for a given placement of the communication channel, the absolute magnitude of the effect depends solely on measurement time. However, longer measurement periods make it easier to distinguish between signals and noise. An adversary can thus choose the measurement time, trading throughput for lower bit error rate.

5.4.2 Long-Wire Overlap

This section characterizes the effect of varying the number of transmitter and receiver long wires v_t and v_r in six FPGA architectures. Besides the three Virtex 5 devices used so far, the effect is also measured on three ML605s (Virtex 6), two Nexys 4 DDRs and two Basys 3s (Artix 7), as well as an ArtyS7 (Spartan 7), a PYNQ-Z2 (Zynq 7000), and a KC705 (Kintex 7). The relative change in frequency ΔRC is shown for different combinations of v_t and v_r in Figure 5.9, for one device per generation. The same common pattern exists for all devices.

For a given number of long wires v_r used by the ring oscillator, there are three distinct segments for ΔRC as the number of transmitter long wires v_t increases. The first segment



(a) Virtex 5 **(b)** Virtex 6

(c) Artix 7 **(d)** Spartan 7

(e) Zynq 7000 **(f)** Kintex 7

Figure 5.9: Relative Count Difference (ΔRC) with 99% confidence intervals as a function of the transmitter and receiver lengths for different device generations. ΔRC is proportional to the overlap between the transmitter and the receiver.

occurs for transmitters which use only parts of a `VLONG`. Using partial wires is possible because even though `VLONG`s can only be driven from the top or the bottom, they have additional intermediate "taps" which can be used to read the values of the signal they carry (Table 2.1 of Section 2.2.3). In practice, using partial wires does not have an effect on the strength of the phenomenon: ΔRC remains constant for all fractions of a `VLONG`.

This result is to be expected since, electrically, the entire long wire is driven even if the output tap does not take full advantage of its length.

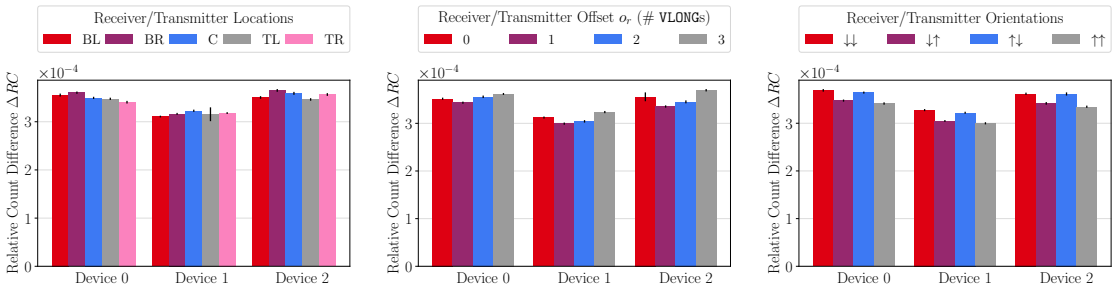
The second segment is the region where $v_t \leq v_r$. Here, ΔRC increases linearly with v_t, suggesting that the phenomenon affects the delay of each long wire equally. The final region consists of $v_t > v_r$, where ΔRC remains constant. The reason for this pattern is that there is no additional overlap between the newly added segments of the transmitter and the receiver.

For a given number of transmitter wires v_t, there are two types of patterns for receivers using v_r long wires. Among receivers with $v_r \geq v_t$, a smaller v_r results in a larger effect. As an example, for $v_t = 3$, the effect for $v_r = 5$ is smaller than it is for $v_r = 3$. This behavior is due to the transmitter affecting only the first v_t out of v_r long-wire segments of the RO. For smaller ROs, these v_t segments represent a larger portion of the number of wires used, and hence of overall delay.

The opposite is true when $v_r \leq v_t$: the larger the RO, the bigger the resulting effect. For instance, for $v_t = 4$, the effect for $v_r = 3$ is larger than the effect for $v_r = 1$. This difference exists because even though the delay of the routing scales linearly, the delay associated with the inverter and buffer LUT stages remains constant. Thus, the routing delay represents a larger fraction of the overall delay (routing delay plus stage delay) for larger ROs. Since this phenomenon only acts on routing delay, larger ROs are affected more than shorter ones. Appendix B contains a comparison across devices with an improved metric (introduced in Section 6.2.2), which calculates the absolute changes in the delays of long wires in pico- and femto-seconds.

5.5 Location Independence

This section validates the location independence of the channel by testing three aspects of the placement of the receiver and the transmitter: the absolute location on the device; the relative offset of the receiver and transmitter; as well as the direction of signal propagation. Figure 5.10 shows the results for all three experiments on the Virtex 5 devices, with 99% confidence intervals. The effect remains approximately constant for each device, regardless of the choice of parameters. Across devices, the



(a) Absolute Location **(b)** Relative Offset o_r **(c)** Direction of Propagation

Figure 5.10: Effect of location on the relative frequency of the receiver ring oscillator for different placements on the device, with 99% confidence intervals. The (a) absolute location, (b) offset, and (c) signal orientation have little influence on the magnitude of the long-wire leakage.

absolute magnitude of the effect varies slightly, but is otherwise almost the same. Any variability across devices is to be expected, since manufacturing variations are known to affect ring oscillator frequencies [132].

Figure 5.10a shows the results when an identical circuit is placed on different locations of the device: the four corners (bottom/top left/right) and the center. Both transmitter and receiver use two `VLONG`s each, and they are adjacent: when the receiver's location is CLB slice (x_r, y_r), the transmitter's location is CLB slice $(x_t, y_t) = (x_r, y_r - 1)$. Within a device, the values are close, and there is no pattern in how the values change between devices. Manufacturing variations within and between devices can thus explain any variability.

The second experiment investigates the effect of the placement of the receiver and the transmitter relative to each other. When the receiver and transmitter have different lengths, it is possible for the two circuits to have the same overlap, but a different starting offset. This relative offset o_r (depicted in Figure 5.11) also has minimal effect on the channel, as confirmed by placing a transmitter made up of five `VLONG`s at a fixed location on the device. The receiver, which uses two `VLONG`s, is placed adjacent to the transmitter, but at an offset of o_r full long wires. This offset needs to correspond to full long-wire lengths due to constraints imposed by the routing architecture of the device. Any other offset would increase the distance d between the transmitter and receiver, which is investigated separately in Section 5.6. Figure 5.10b presents the long-wire leakage for the four possible offset placements, which show approximately the same consistency both within and between devices as those of the previous experiment.

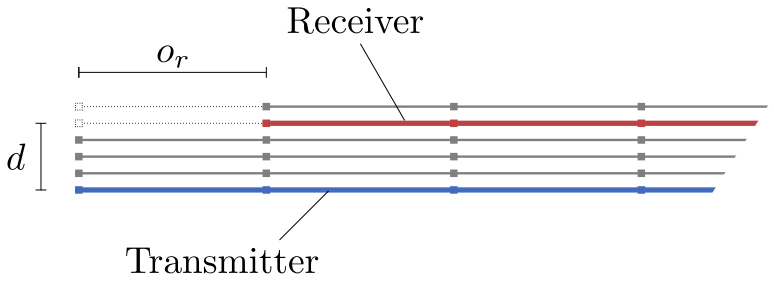


Figure 5.11: Relative placement of the transmitter and receiver long wires, with respect to distance d and receiver offset o_r.

Note that the relative effect of placing the receiver at various offsets forms a consistent pattern across devices. As an example, the effect for the offset $o_r = 3$ is consistently stronger than it is for $o_r = 1$. This pattern can be explained by the FPGA routing layout: as mentioned in Section 2.2.3, the local routing necessary to access the various long-wire segments is different between each test. Because the local routing resources differ, the ratio between the delay incurred by the long-wire segments and the local routing resources changes. While the delay of the long-wire segments is affected by the transmitter, the delay of local routing is not.

The final experiment changes the direction of signal propagation for the transmitter and receiver, and fixes the offset to $o_r = 2$. In the previous experiments, both signals travelled from the bottom of the device to the top. However, in the Virtex 5 architecture, `VLONG` wires are bidirectional (Section 2.2.3), and can thus propagate signals upwards or downwards. Figure 5.10c shows the results for the four different orientations (receiver and transmitter down, receiver down but transmitter up, etc.). ΔRC remains approximately the same for all configurations, although, as with the previous experiment, there is a consistent ordering for the four transmission directions across devices. As in the earlier experiment, this pattern can also be explained by the routing layout. Overall, the results of this section illustrate that only long wires need to be manually specified, while local routing and the placement of registers and LUTs can be left to the compiler tools.

5.6 Resilience To Countermeasures

Although defense mechanisms are discussed in more depth in Section 5.9, this section evaluates how close the transmitter and the receiver need to be for successful

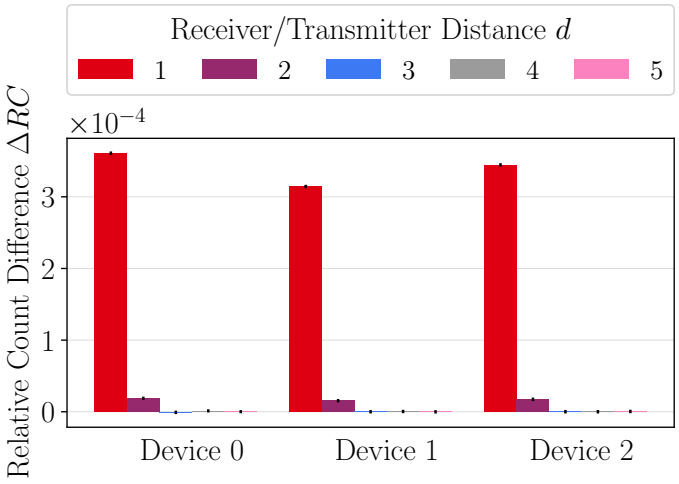


Figure 5.12: Effect of the distance d (defined in Figure 5.11) between the transmitter and the receiver. Long wires leak information up to two wires away.

communication. This is tested by varying the distance d (depicted in Figure 5.11) between the transmitter and the receiver. The results, shown in Figure 5.12, demonstrate that the phenomenon is still measurable when separating the wires by a distance of $d = 2$, but the effect is 20 times weaker. When the wires are farther apart ($d \geq 3$), there is no correlation between the transmitted and received values, i.e., the data comes from the same distribution according to the Kolmogorov-Smirnov test ($p > 0.75$). In other words, any defensive monitoring must be routed within a distance of two to detect a transmission through the channel. Moreover, all four wires adjacent to a signal need to be occupied in order to prevent covert-channel transmissions or side-channel leakage exploitation.

To test whether an active protection mechanism can disrupt the channel through additional dynamic activity on the device, transmissions are attempted in the presence of large, competing circuits which are both in- and out-of-sync with respect to the transmissions. Two large 4096-bit adders are used, adding different parts of a bitstream produced by an LFSR. As a result, both the addends and the sums change every time the LFSR produces a new bit. The bits of each sum are then `XOR`ed together and drive two Light-Emitting Diodes (LEDs) for additional current draw. Experiment are conducted on two Artix 7 Nexys 4 DDR boards, for a transmitter and a receiver using ten `VLONG`s each.

The switching activity of nearby circuits is varied by changing how often the LFSR produces new values. Specifically, the clock driving it is divided by 2^m for $m \in \{1, 7, 15, 20, 24\}$,

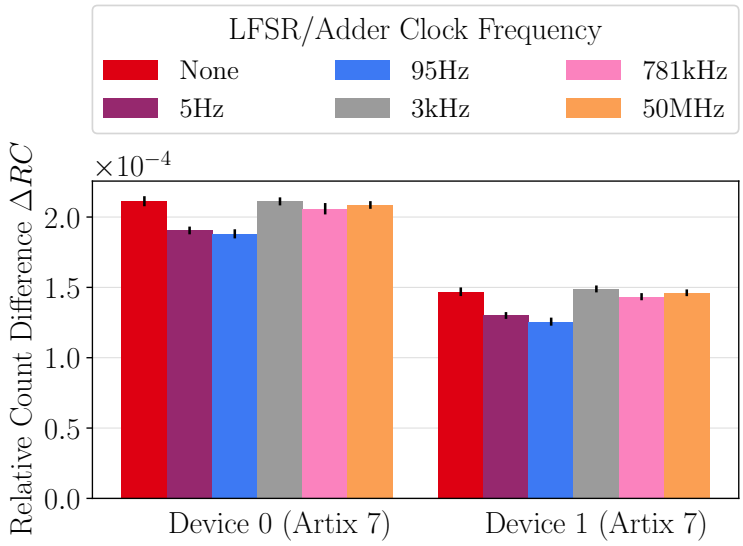


Figure 5.13: Effect of activity induced by adders and Linear Feedback Shift Registers (LFSRs) at different clock frequencies. The additional activity has minimal impact on channel quality.

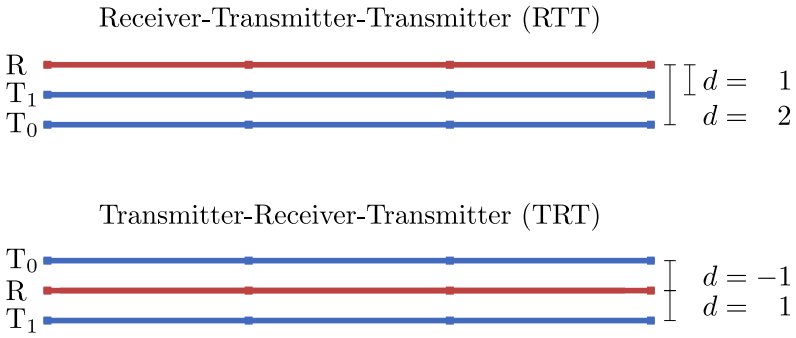
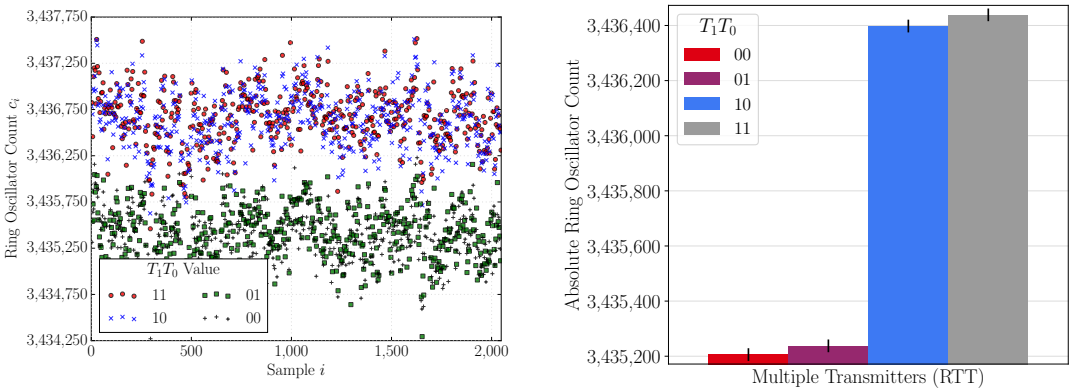


Figure 5.14: Relative placement of multiple transmitters and a single receiver.

producing frequencies between 5 Hz and 50 MHz. The results for the two devices, including the base case of no adders or LFSRs, are summarized in Figure 5.13. Although additional activity cannot disrupt the transmissions, there is some correlation between the frequency of the activity and the corresponding count difference. The resulting change is not sufficient to hinder transmission, but can be used by the adversary to detect the level of activity on the device, a technique already used by Hardware Trojan detectors [155, 384].

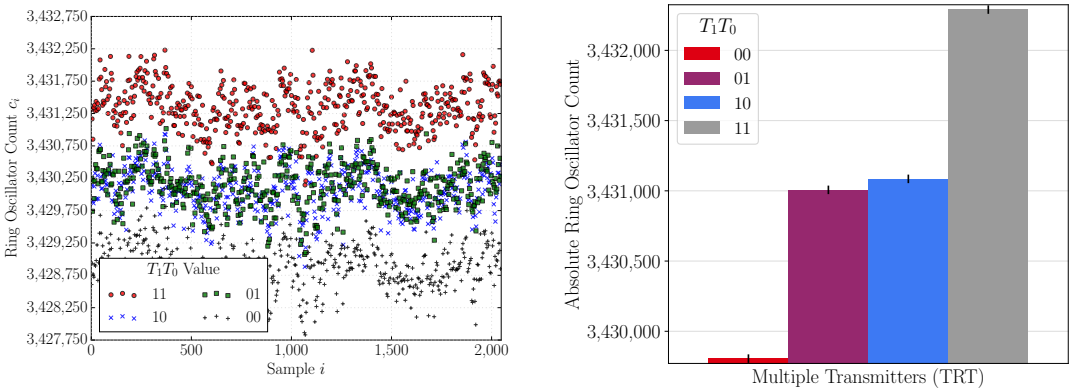
5.7 Simultaneous Transmissions

This section investigates the effect of using multiple transmitters, T_0 and T_1. The first set of experiments returns to the Virtex 5 boards, with the receiver and the transmitters



(a) Raw samples **(b)** Average Counts

Figure 5.15: Samples (a) and averages (b) of the ring oscillator counts during simultaneous transmissions on adjacent long wires (Receiver-Transmitter-Transmitter (RTT) pattern).



(a) Raw samples **(b)** Average Counts

Figure 5.16: Samples (a) and averages (b) of the ring oscillator counts during simultaneous transmissions on adjacent long wires (Transmitter-Receiver-Transmitter (TRT) pattern).

using two long-wire segments each ($v_t = v_r = 2$). As seen in the timing diagram of Figure 5.4 (*Transmitter 0* and *Transmitter 1*), the transmitters are driven independently, and cycle through all 2-bit combinations over multiple sampling periods. Figure 5.14 describes the two different transmitter arrangements.

In the Receiver-Transmitter-Transmitter (RTT) pattern, both transmitters are on the same side of the receiver. T_0 is located at a distance of $d = 2$ to the receiver, while T_1 at a distance of $d = 1$. Figure 5.15 shows that the ring oscillator counts are only affected by the value of the closer transmitting wire, T_1: the data for $T_1 = 0$ and $T_1 = 1$ comes from different distributions with $p < 10^{-145}$ according to the Kolmogorov-Smirnov test, while the data is statistically indistinguishable with regards to T_0 ($p > 0.17$). In other words, the

effects of the closer transmitting wire overpower the influence of the farther transmitter. This localized eavesdropping capability can be exploited to attack even implementations which balance power usage for security (e.g., dual-rail logic [49, 64, 249, 300]): an attacker can infer information about the state of a balanced circuit without the need for physical proximity to the device required by prior work [145, 281, 316].

In the Transmitter-Receiver-Transmitter (TRT) pattern, the receiver is routed between the two transmitters. T_0 is at a distance of $d = -1$ to the receiver, while T_1 at a distance of $d = 1$. Figure 5.16 shows that T_0 and T_1 have roughly equal influence on the receiver RO frequency: the counts are highest when both transmitters are on, lowest when they are both off, and in-between otherwise.

The second set of experiments varies the length of the transmitters in the TRT setup on the Artix 7 boards. The receivers used are the longest possible on each board, namely $v_r = 8$ for the Basys 3 and $v_r = 10$ for the Nexys 4 DDR. The "effective" transmitter length v_t^{eff} can be defined as follows:

$$v_t^{eff} = b_0 v_t^0 + b_1 v_t^1 \tag{5.3}$$

where transmitter i has length v_t^i and carries value b_i. For a fixed ring oscillator and a given v_t^{eff}, one expects that the ΔRC values should remain approximately constant for all possible combinations of $b_i \in \{0, 1\}$ and $0 \leq v_t^i \leq v_t^{eff}$. Indeed, these predictions are validated in Figure 5.17: for a given board, ΔRC is approximately the same regardless of whether only T_0 (*Left*), only T_1 (*Right*), or *Both* are carrying a logic 1.

Unlike the previous experiments where measurements were conducted using Xilinx's ChipScope ILA core, in these experiments, data is transferred over the UART. To even further prove that the measurement setup does not have a significant effect on the strength of the phenomenon, Figure 5.17 also plots the measurements from the experiments of Section 5.4.2. These measurements had been taken using ChipScope in the single-transmitter case (*Single*), and are indistinguishable from the dual-transmitter experiments, as expected. When using ChipScope, neither the single-transmitter Basys 3 experiments with $v_r = 8$ nor the dual-transmitter designs for either board could be routed by Vivado, so they are not included for comparison.

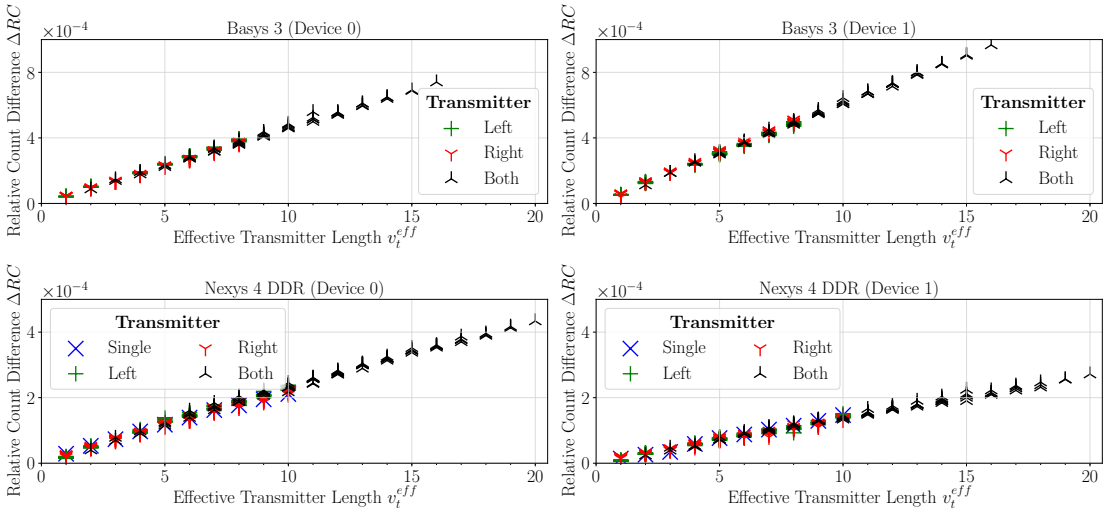


Figure 5.17: Relative Count Difference (ΔRC) as a function of the effective transmitter length v_t^{eff} for the four Artix 7 boards. The strength of the long-wire leakage depends on the total effective length v_t^{eff} and not individual transmitter lengths.

The above results suggest that a covert-channel attacker can use dual-transmitters to increase bandwidth or reduce the likelihood of errors in transmissions. In the first case, choosing transmitters of different lengths allows simultaneous transmission of two independent bits. This effectively doubles bandwidth, since all four combinations result in different ring oscillator frequencies. In the second case, the transmitters are always in sync, and increase the relative effect seen by a fixed ring oscillator by a factor of 2×. The attacker can thus also use a ring oscillator with fewer `VLONG` segments, which is affected more in relative terms as explained in Section 5.4.2.

5.8 Exploiting the Leakage

This section discusses different possibilities for how to exploit the information leakage. In some cases (such as that of Figure 5.2), a threshold is sufficient for distinguishing between 0s and 1s. However, in other setups (such as that of Figure 5.5), this separation might not be as clear: the RO frequency may drift due to random variations in temperature and voltage. To solve this issue, this section details an encoding scheme that enables high-bandwidth covert transmissions (Section 5.8.1). It also explains how to eavesdrop on dynamic signals through repeated measurements (Section 5.8.2), and conducts them in practice (Section 5.8.3).

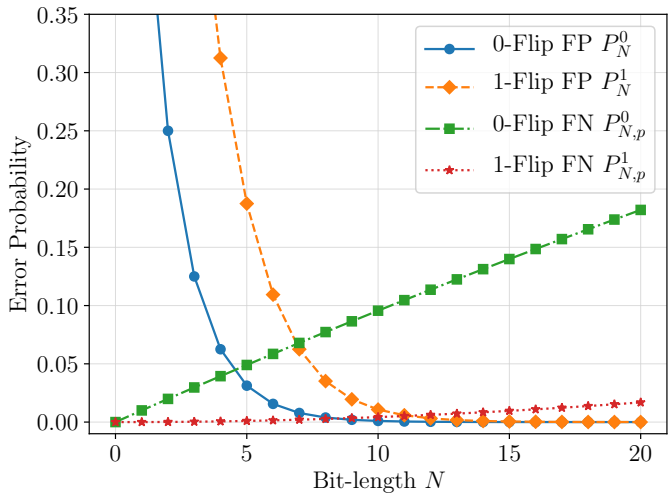


Figure 5.18: Probability of a Start-of-Frame (SoF) error when transmissions are or are not taking place (False Negatives (FNs) and False Positives (FPs) respectively). Errors are calculated with 0 and 1 allowed bit-flips for different SoF lengths N and bit-flip probability $p = 0.01$.

5.8.1 Covert Transmissions

To overcome the hurdle posed by local fluctuations, a Manchester encoding scheme can be used, where zeros are transmitted as the pair $(0,1)$ and ones as the pair $(1,0)$. Since every pair contains each bit once, one can decode the received pair (c_0, c_1) as a 0 if $c_0 < c_1$ and as a 1 otherwise. This scheme enables transmissions in various setups with high accuracies. For example, on the Virtex 5 devices, transmissions using two long wires and lasting 82 µs, or using $\frac{1}{3}$ of a `VLONG` for 21 ms are 99.0–99.9% accurate, without employing any error correction algorithms. Under this encoding scheme, the bandwidth of the channel is up to $1/(2 \cdot 82 \cdot 10^{-6}) = 6.1$ kbps.

To further distinguish between noise and legitimate transmissions, adversaries can employ N-bit Start-of-Frame (SoF) patterns. A longer N makes it harder for noise to accidentally trigger an SoF when no transmissions are taking place. However, it also increases the probability for bit-flip errors during transmissions due to environmental fluctuations and jitter in the ring oscillators. When no transmission is taking place, each measurement is equally likely to be interpreted as a 0 or a 1, since $\Pr[c_0 < c_1] = 1/2$. The false positive probability that noise is interpreted as an SoF when no transmission is taking place is thus $P_N^0 = 2^{-N}$. If the probability of a bit-flip error is p (which is between 0.1% and 1% in the above setup), then an SoF pattern gets corrupted with a false negative

probability of $P^0_{N,p} = 1-(1-p)^N$. These probabilities are shown in Figure 5.18 for different N and $p = 0.01$. Even for $N = 5$, the probability of false positives and negatives is 3% and 5% respectively, which might be too high for practical purposes.

For this reason, adversaries should accept SoF patterns with up to one bit flipped. The number of accepted N-bit patterns thus grows to $N+1$: the correct one, plus one for every possible position where a bit-flip could occur. For example, if the original SoF pattern is `01100`, then `11100`, `00100`, `01000`, `01110`, and `01101` should also be accepted. The false negative rate is therefore $P^1_{N,p} = 1-(1-p)^N - Np(1-p)^{N-1}$, while the false positive rate is $P^1_N = (N+1)/2^N$. Both of these improved error rates are also shown in Figure 5.18. Choosing $N = 11$ allows for an almost equal error rate of approximately 0.5%.

It should be noted that when temperature and voltage fluctuations are not random (e.g., if there is a monotonic change in temperature), the probability that $c_0 < c_1$ when no transmission is taking place is no longer $1/2$. In that case, an all 1s (or all 0s) SoF pattern would be triggered easily due to persistently increasing (respectively decreasing) temperatures. An alternating pattern would prevent this from occurring, while a temperature-aware sensor (e.g., a ring oscillator counter normalized for temperature) would allow the covert channel to operate even in these conditions. Different applications could also choose different N based on their bit-flip probability p and increase the number of allowed bit-flips to improve accuracy.

5.8.2 Signal Exfiltration

Signals which are not under the adversary's control may not remain constant throughout the period of measurement. However, as shown in Section 5.3.2 (Figure 5.6), the delay of the long wire depends only on the proportion of time for which the nearby wire is carrying a logic 1, and *not* its switching frequency. This fact reveals the Hamming Weight (HW) of the transmission during the measurement period. By repeating measurements with a sliding window, an eavesdropping adversary can fully recover nearby dynamic signals such as cryptographic keys with high probability.

Suppose that the adversary wishes to recover an N-bit key $\mathbf{K} = K_0 \dots K_{N-1}$ (or, more generally, any internal secret state, such as an Advanced Encryption Standard (AES)

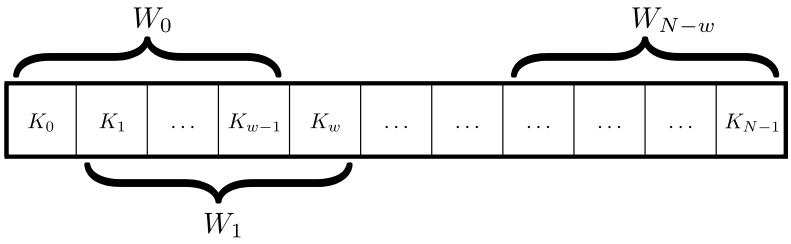


Figure 5.19: A window of width w can determine the relationship between bits K_i and K_{i+w}.

S-Box input bit). Assume that, in one period of measurement, the long wire carries w consecutive bits of the key, with $N = nw$ (Appendix B explains how to remove this assumption). By making repeated measurements of different but overlapping windows as shown in Figure 5.19, the adversary can recover the key with high probability. Specifically, denote the HW (estimated by the RO count) of the first w key bits $K_0 \dots K_{w-1}$ (window W_0) by c_0. Similarly, let the HW of bits $K_1 \dots K_w$ (window W_1) be c_1. Then, if $c_0 \approx c_1$ (within some device-dependent tolerance), one can conclude that $K_0 = K_w$. If $c_0 > c_1$ then $K_0 = 1$ and $K_w = 0$, while if $c_0 < c_1$ then $K_0 = 0$ and $K_w = 1$. By comparing the next count c_2 to c_1, one can determine the values of K_1 and K_{w+1}, and, by repeating this process, one can determine the relationship between K_i and K_{i+w}.

Assuming a randomly generated key, the probability that $K_i = K_j$ for $i \neq j$ is $1/2$. The probability that all of $S_r = (K_r, K_{w+r}, \dots, K_{(n-1)w+r})$ are equal is $1/2^{n-1}$, since there are $n-1$ such pairs. The probability that at least one of the bits in S_r is different than the rest is thus $1 - 1/2^{n-1}$. If at least one bit is different, an adversary can recover all elements of S_r. Repeating this argument for all possible remainders $0 \leq r < w$, the probability of recovering the *entire key* can be calculated as:

$$P = \left(1 - \frac{1}{2^{n-1}}\right)^w \geq 1 - \frac{w}{2^{n-1}} \tag{5.4}$$

by Bernoulli's inequality. Even if it might appear counter intuitive, Equation (5.4) shows that longer keys are easier to recover than short keys. A larger window size w relative to the key length makes recovering the key harder, as there are fewer measurements over the length of the key. For the same reason, a longer key will increase the recovery probability. This means that asymmetric keys (e.g., those used for signature verification) are relatively easy to recover, since they are typically much longer than symmetric keys. In the worst

case (if the entire key consists of a repetition of its first w bits), the proposed approach reduces the guessing space from 2^N to 2^w possibilities (and at most 2 possibilities through multiple window sizes, as explained in Appendix B).

When the long wire contains deterministic values after the last bit of the key has been transmitted, even single window sizes suffice. Specifically, the adversary can consider the w measurements that follow the last key bit as part of the key itself: $K_N, \ldots, K_{N+w-1}$. There are different possibilities to consider after transmitting K_{N-1}:

1. The long wire contains a fixed 0 or 1 bit, i.e., $K_N = \cdots = K_{N+w-1}$. In this case, the adversary can determine K_{N-1} by comparing the count for $K_{N-1} \cdots K_{N+w-2}$ to the count for $K_N \cdots K_{N+w-1}$, and proceed backwards for K_{N-2} and other bits. If at least one of $K_{N-w}, \ldots, K_{N-1}$ is not equal to the fixed value, then the entire key is recovered. Otherwise, the adversary determines that the key consists of all 0s or all 1s. These two cases are easy to distinguish, as the total RO count for all 1s is higher than the total count for all 0s.

2. The code assigns X (don't care) or Z (high-impedance) to the wire. During synthesis, these two possibilities reduce to the fixed-bit case above, as also verified when comparing multiple to single transmitters (Figure 5.17).

3. The long wire contains the last key bit K_{N-1}. This possibility also reduces to Case (1), but the adversary starts recovery at K_{N-2} instead of K_{N-1}.

4. The long wire is used to carry other values (such as a different key), which would not change on repeated measurements. This effectively increases N, and can be combined with the cases above after the second key has finished transmitting.

The only scenario where the adversary cannot fully recover the key is when the long wire is updated with a random value that changes at every clock cycle and between measurements. This suggests a possible leakage countermeasure, and is discussed in Section 5.9.3.

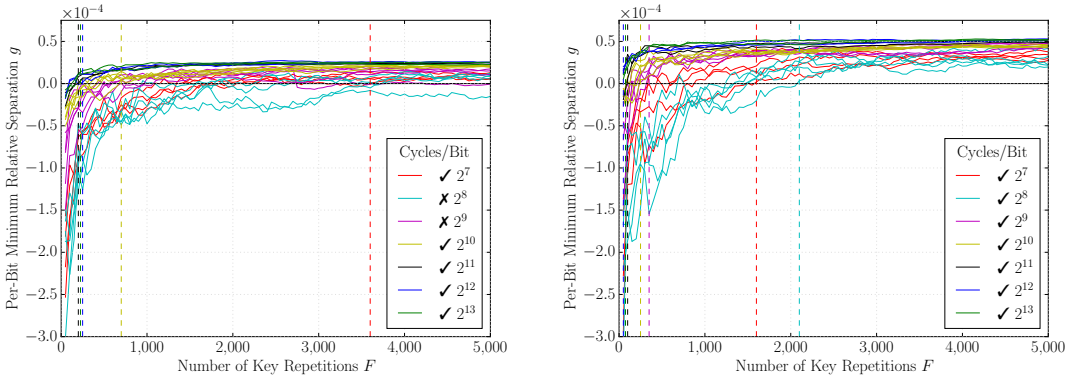
5.8.3 Eavesdropping Attacks

This section implements the single-window attack detailed above on a Basys 3 board. The target is an $N = 32$-bit key transferred over long wires, with a window size of $w = 4$. No knowledge is exploited about the values carried on the long wires after the key has been transmitted. In other words, this attack represents the worst-case scenario for the attacker. Even this less-sophisticated adversary can recover keys in most setups.

The first step is to "calibrate" the setup by measuring the RO frequency for the all-ones and all-zeros keys 1,000 times each. This calibration can be used to set a more precise threshold for how to interpret $c_i \approx c_{i+1}$: let the average all-ones (all-zeros) calibration measurement be C^1 (C^0), with $\Delta = C^1 - C^0$. K_i is then classified to be the same as K_{i+w} if $|c_i - c_{i+1}| < (1 + \alpha) \cdot \Delta$, where $\alpha = 0.5$ is a relaxation threshold to account for noise. Because of the inherent jitter of the RO, $F = 500$ full passes over the key are made, collecting $F \cdot (N - w + 1) = 14{,}500$ RO counts. c_i is then taken to be the average over the F measurements. In this proof-of-concept setup, each full pass corresponds to $w = 4$ repetitions of the key (one for each remainder), but an adversary could equally use w parallel counters and require only F passes instead of $F \cdot w$ ones. Similar to the multi-transmitter experiments of Section 5.7, the UART is used instead of the ILA core.

Five different keys are tested: $\mathbf{K_0}$ = `0`, $\mathbf{K_1}$ = `0xffffffff`, $\mathbf{K_2}$ = `0xdeadbeef`, $\mathbf{K_3}$ = `0xd0e1a3f4`, as well as a randomly chosen key $\mathbf{K_r}$. When the number of long wires used is at least $v \geq 4$ and the number of clock cycles per bit is at least 2^{13} (i.e., bits remain constant for 82 µs), the proposed algorithm correctly recovers all five different keys. Specifically, $\mathbf{K_0}$ and $\mathbf{K_1}$ are reduced to the $2^w = 16$ possibilities of repeated subkeys explained in the previous section, while $\mathbf{K_2}$ is reduced to the two possible values of `X101X110X010X101X011X110X110X111` (the correct value has $X = 1$). $\mathbf{K_3}$ and $\mathbf{K_r}$ are always recovered fully. Reducing the number of `VLONG`s v or the number of clock ticks lowers accuracy, but still allows partial recovery of keys. For example, with only $2^9 = 512$ clock cycles (5.1 µs) and $v = 2$, the algorithm can correctly recover 30 out of the 32 bits of $\mathbf{K_3}$. It also recovers at least 25 bits correctly for all keys when $v \geq 4$.

However, with some additional computation and some more repetitions, the adversary can reduce both the number of long wires and the number of clock cycles needed. In these



(a) For $v = 1$ long wires, some keys cannot be recovered for all measurement periods

(b) For $v = 2$ long wires, keys are recoverable for all measurement periods

Figure 5.20: Per-bit minimum relative separation g for windows of different Hamming Weights (HWs) as a function of the number of key repetitions F. Experiments are conducted for multiple keys and measurement periods. Vertical lines represent the minimum number of repetitions F_{min} needed to distinguish between different HWs, i.e., $g > 0$ for all keys $\mathbf{K}$ and $F \geq F_{min}$.

experiments, signals are kept constant for at least 2^7 clock cycles (1.3 µs or 780 kHz), due to limitations of the Basys 3 boards (insufficient memory and slow UART). In 98.4% of measurements with $F = 5{,}000$ repetitions and $v \geq 1$ long wires, the average count for a window is monotonic in its Hamming Weight (HW). In other words, for any two windows of a key with HWs $h_1 > h_2$, the average counts c_1, c_2 for the two windows obey $c_1 > c_2$, making the key recoverable. Put differently, the key can be recovered if there are constants $0 = H_0, \ldots, H_w$ such that any window of HW h with average RO count c_h satisfies $H_h < c_h < H_{h+1}$. Let K_h denote all subkeys with HW h. Then for two HWs $h_1 > h_2$, the "gap" between them can be defined as follows:

$$\text{gap}(h_1, h_2) = \left(\frac{\min\limits_{k \in K_{h_1}} c_k}{\max\limits_{k \in K_{h_2}} c_k} - 1 \right) \cdot \frac{1}{h_1 - h_2} \tag{5.5}$$

Every window with HW h_1 has a larger average than every window with HW h_2 if and only if $\text{gap}(h_1, h_2) > 0$. Taking the minimum over all HWs, $g = \min\limits_{h_1 > h_2} \text{gap}(h_1, h_2)$ can be interpreted as the minimum relative separation per bit if $g > 0$, while $g \leq 0$ suggests that it is impossible to correctly recover all bits of the key. Figure 5.20 depicts g for different keys $\mathbf{K}$, measurement periods of 2^t clock cycles, and key repetitions F. For both $v = 1$ and $v = 2$ `VLONG`s, 250 measurements suffice to recover keys whose bits are kept

constant for at least 2^{10} cycles. Although 5,000 measurements are not always enough when using a single VLONG, 3,600 measurements can recover keys whose bits change as frequently as every 2^7 cycles (1.3 µs). This number drops to 2,100 measurements for 2 VLONGs, and decreases further as the number of long wires increases. The algorithm can thus cope with higher frequencies by either repeating measurements multiple times or by increasing the overlap between the receiver and the victim long wires.

Having calculated these bounds, the algorithm not only reduces a key of size 2^N to 2^w in the worst-case, but it also indicates what the Hamming Weight of the w bits are. Combined with a window of size $w + 1$ or measurements beyond the end of the key, it fully recovers the entire key, even in the all 0s or all 1s case (Appendix B). This sliding window approach allows adversaries to detect much faster signals compared to simple averaging, which would be more susceptible to environmental variations: as noted in Section 5.4.1, signals which were kept constant for 2^{13} clock cycles (5.1 µs) resulted in an average absolute count difference of only ≈ 4 ticks.

As a final note, a second RO can account for noisy environments. This secondary RO has $v - 1$ long wires that are routed to be adjacent to the first RO but not to the transmitter. Because this secondary RO is only minimally influenced by the transmitter signals, it can be used to estimate environmental conditions such as local voltage and temperature. The calibration measurements can therefore create a linear fit for ring oscillators counts during the transmissions of 0s and 1s, and remove the effect of environmental conditions during the transmission of the unknown key. With only $F = 500$ measurements, $v = 2$ VLONGs, and measurement periods of 2^9 clock cycles, 84% of keys tested could be recovered without any errors. Collecting more data points and applying more sophisticated regression techniques would allow recovery of even faster signals, posing a realistic threat even for low-latency applications.

5.9 Discussion

The discussion of the long-wire leakage phenomenon is structured in three parts: the channel itself (Section 5.9.1), the cause of the leakage (Section 5.9.2), and potential defense mechanisms (Section 5.9.3).

5.9.1 The Channel

The channel characterized in the previous sections does not require any modifications to the device or special tooling, allowing an adversary to distribute it in the form of IP cores. There are no real placement constraints, except to ensure that the receiver and transmitter long wires are adjacent. Moreover, the channel requires very little logic: the entire setup (including the signal generation and measurement component) uses just 71 LUTs and 66 registers, excluding resources to transfer the measurements to the computer for analysis. As an example, the proposed channel would only take up 0.2% of the 33,000 LUTs used in the open-source N200/N210 Universal Software Radio Peripheral (USRP) Software-Defined Radio (SDR) project [89].

The USRP contains code from Ettus Research, Xilinx, Easics NV, and OpenCores (written by different authors) [90], and illustrates that IP from many organizations can become integrated into a single project. Since third-party code is a necessity, and as modern IP blocks can be quite large, the potential for unintended interaction between different cores increases. Attackers can exploit the routing algorithms, which are forced to route through otherwise monolithic black-box IPs due to resource constraints. This enables adversarial logic to communicate covertly or eavesdrop on nearby signals.

As ring oscillators have legitimate benign and useful applications (Section 2.2.5), the transmitters and receivers are of dual-use. Moreover, the source of the leakage identified exists whether transmissions are intentional (covert channel) or not (side channel), is a threat when an adversary controls one or more IP cores, and can bypass local balancing protection mechanisms (Section 5.7). These unintentional long-wire transmissions thus pose new risks for multi-tenant scenarios, including FPGA hybrids and public clouds. Finally, the same phenomenon could also be useful in IP core watermarking [35, 276], or no-contact debugging of stuck signals.

5.9.2 Leakage Cause

So far, this chapter has focused on the novelty and applicability of the phenomenon presented, rather than its cause. Section 5.3 showed that the source of the effect is fundamentally different from that of prior work: long-wire leakage increases with the

HW of transmitted signals, while prior work expects a decrease in the RO frequency in response to an increase in switching activity. Prior to the work of this chapter, Gag et al. had also investigated long-wire delays. Specifically, an RO with a long wire was placed between long wires carrying signals that were synchronized with the RO signal (either equal to it, or opposing it) [99]. It was shown that when the nearby long wires had the same value as the RO wire, the frequency of the RO was higher compared to when the nearby long wires had the opposite value [99]. The work by Gag et al. [99] required the signals to be in sync, and long wires to be directly connected. Moreover, they were driven jointly, and static patterns that are independent of the RO signal were not tested. As a result, the effect identified could not be used directly for covert- or side-channel communication. By contrast, this chapter demonstrated that nearby long wires are influenced even when there is no connection between the transmitter and the receiver, and even when the transmitted value remains constant during the measurement period. These two properties can be exploited in constructing a communication channel—even in Intel devices [253, 260].

As physical layout and process-specific parameters are proprietary, the precise physical cause for this long-wire leakage has not yet been determined for FPGAs. This lack of electrical details for FPGA hardware has been identified by multiple authors in the past [6, 19, 72, 261, 309, 355]. As a result, whether the effect exists due to drive-strength issues, electromagnetic emanations, or some other property of FPGAs remains an open question. However, simulations on 45 nm Application-Specific Integrated Circuits (ASICs) suggest that the effect is caused by "capacitive crosstalk" [161], as also suggested by other literature on FPGA long wires [99, 253, 260].

Overall, the characterization of the channel is valuable even without access to these details: it is always present, and is easily measurable on off-the-shelf devices without special modifications. Moreover, FPGA users cannot alter the electrical behavior of the device, but can only influence how circuits are mapped onto it. As a result, FPGA circuit designers need to be aware of the communication and exfiltration capabilities that this channel introduces.

5.9.3 Defense Mechanisms

Section 5.6 showed that transmissions cannot be detected from a distance $d > 2$, and that spurious activity (in the form of adders and additional current draw) does not eliminate the transmission channel. Hence, defense mechanisms need to prevent vulnerable designs from being loaded onto an FPGA. Since long wires are an integral part of the reconfigurable FPGA fabric, detecting the transmitter is not easy: any long wire can carry sensitive information within or between IP blocks. Routing algorithms thus need to be modified to account for this information leakage by introducing directives which mark signals (or even entire blocks) as sensitive. The tools then need to add "guard wires", by either leaving the four nearby long wires unoccupied, or by occupying the two adjacent long wires with compiler-generated random signals: based on the RTT experiments of Section 5.7, driving all four is superfluous. As mere logical isolation and bitstream protection [335] is insufficient to protect against the attacks of this chapter, physical isolation becomes necessary.

It should be noted that even though excluding routing resources will prevent the leakage from occurring, it is particularly taxing for dense designs: it can make placement and routing more time-consuming, or even lead to timing violations. As an example, Vivado could not route the multi-transmitter designs of Section 5.7 when using ChipScope. Designers using unpatched tools thus need to be aware of this source of leakage, and must manually look for long wires post-routing, explicitly add guard wires, or, more generally, specify placement and routing constraints for both highly-sensitive signals, and untrusted third-party blocks.

When this is not possible, designers should instead introduce randomness to the sensitive values carried on the long wires. For example, one could make the long wire carry a random bit after every secret bit, and ensure that the last secret bit is also followed by random values that change each time the key is used (Section 5.8.2). Although these defense mechanisms do not eliminate the leakage (the adversary can still average over large numbers of measurements and get statistical information about the key), they increase the cost of the attack and force the adversary to use more sophisticated post-processing techniques.

It should be noted that cloud providers and other FPGA stakeholders may attempt to block the RO receiver from operating. However, as Chapter 6 shows, alternative ring oscillator designs bypass currently-deployed countermeasures to exploit long-wire leakage even in a cloud environment. Overall, better defense mechanisms for future FPGA generations are needed at the architectural level, and require a deeper understanding of the cause of this phenomenon.

5.10 Summary

This chapter investigated attacks that can be conducted remotely on FPGAs without any modifications to the underlying hardware (Section 5.1). It demonstrated that a previously-unexplored hardware imperfection of FPGA devices causes the delay of long wires to depend on the logic state of nearby long wires, even when their driven values remain constant. This effect is surprisingly resilient and is measurable within the device by small circuits, without having to account for or isolate environmental noise (Section 5.2). Long-wire leakage reveals the Hamming Weight (HW) of transmissions (Section 5.3), and can be detected for various measurement times (Section 5.4.1), on six different Xilinx FPGA families (Section 5.4.2), and in multiple circuit orientations and locations (Section 5.5). The phenomenon remains quantifiable even in the presence of dynamic activity on the device (Section 5.6), and with multiple simultaneous transmissions (Section 5.7).

This chapter used long-wire leakage to construct an on-chip covert channel with a bandwidth of up to 6 kbps, and an accuracy of 99.9% (Section 5.8.1). It also conducted eavesdropping attacks on signals which are kept constant for as low as 1.3 µs, with an accuracy of more than 98.4% (Sections 5.8.2 and 5.8.3). In other words, the effect identified can break separation of privilege between IP cores of different trust levels, or enable communication between distinct users in multi-tenant setups. Overall, there is a need for software and hardware improvements in FPGA tooling and devices (Section 5.9), which Chapter 6 shows need to go beyond simple physical isolation of user logic.

A small leak will sink a great ship.

— Benjamin Franklin

6 Covert Communication on Cloud FPGAs

Contents

As Chapter 5 demonstrated, physical isolation is a necessary first step towards secure multi-tenant Field-Programmable Gate Array (FPGA) logic. Even then, however, designs might still not be secure against side-channel attacks [167, 197, 282, 387]. One potential limitation of existing attacks is that they target low-end FPGA devices, where physical

isolation is not strong: the transmitting and receiving circuits share the same FPGA die, and, in some cases, even the same clock regions and resources [197, 387].

However, as Section 2.2.4 identified, high-end FPGAs may contain multiple dies incorporated into the same FPGA chip. These distinct dies, called Super Logic Regions (SLRs), could be used to multiplex the FPGA on a per-SLR basis. Although this form of physical isolation per tenant may appear to be stronger and a potential security improvement for virtualized FPGAs in cloud environments, this chapter demonstrates that it is not sufficient to prevent communication between tenants across the SLR dies. Specifically, this chapter introduces the first cross-SLR covert-channel attack, both locally and on the cloud. The channel can be used for potential data exfiltration (e.g., of cryptographic keys or other sensitive information) between users that share the same reconfigurable device.

As some cloud providers such as Amazon Web Services (AWS) prohibit combinatorial loops from user logic [12], this chapter is also concerned with alternative Ring Oscillators (ROs) which can bypass these restrictions. The properties of the proposed ROs are first evaluated by measuring long-wire leakage on the Virtex UltraScale+ FPGAs found on AWS and Huawei Cloud FPGA servers before they are applied to the cross-SLR covert channel. In summary, this chapter furthers the state-of-the-art by:

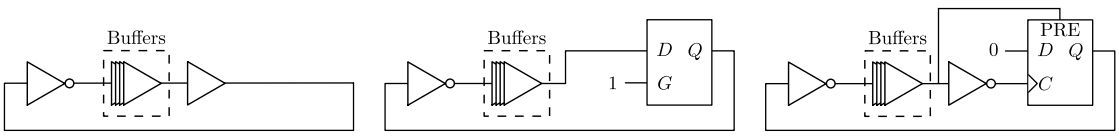
1. Introducing a novel flip-flop-based RO and a latch-based RO which overcome combinatorial loop restrictions (Section 6.1).

2. Proposing a new architectural setup and metric for estimating the strength of long-wire leakage (Section 6.2).

3. Demonstrating that the long-wire leakage phenomenon persists in the Virtex UltraScale+ FPGA family, and measuring femtosecond-scale changes in the delay of the long wires due to this leakage (Section 6.3.1).

4. Identifying intra- and inter-process variations across eleven FPGA boards both locally and on the Amazon and Huawei clouds (Section 6.3.2).

5. Determining that the new RO designs provide almost-identical estimates for the long-wire leakage as the traditional combinatorial loop RO (Section 6.3.3).



(a) LUT-RO **(b)** LD-RO **(c)** FF-RO

Figure 6.1: Three Ring Oscillator (RO) designs with a variable number of intermediate buffer stages, used in local and cloud experiments. Traditional ROs use (a) Lookup Tables (LUTs), but to bypass cloud restrictions, one can use (b) Latches (LDs) or (c) Flip-Flops (FFs) instead.

6. Characterizing the first cross-SLR covert channel both locally and on the cloud across different transmitter and receiver parameters (Section 6.4).

7. Analyzing the channel's bandwidth to show that multi-bit transmissions increase it to 4.6 Mbps with over 97.6% accuracy (Section 6.5).

8. Discussing potential defense mechanisms to mitigate the identified sources of leakage (Section 6.6).

In summary (Section 6.7), this chapter performs the first practical cloud FPGA attacks by introducing new RO designs, improving the long-wire leakage results of Chapter 5, and showing that physical isolation to different SLR dies is not enough for secure multi-tenant FPGAs.

6.1 Ring Oscillator Designs

This section introduces alternative RO designs which overcome restrictions placed by cloud providers such as AWS [12]. Although most ROs use Lookup Tables (LUTs) (Section 2.2.5), some research into alternative ROs has been conducted. For example, ROs replacing one or more stages with an open Latch (LD) have been used for Physical Unclonable Functions (PUFs) [353, 354] and RO-based temperature sensors [258]. Moreover, ROs with Flip-Flops (FFs) have been proposed to characterize FF delays [257, 266], but have only been evaluated in simulations. Recent proof-of-concept work by Sugawara et al. [311] has also shown that LD-based and FF-based ROs can overcome AWS restrictions. However, how these ROs behave relative to traditional ROs was not investigated.

This chapter instead compares three types of ROs. The first design (Figure 6.1a) is the traditional LUT-RO of the previous chapter, with one inverter, one buffer, and zero or

more intermediate buffer stages (Chapter 5 and Sections 6.2 and 6.3 use one intermediate buffer). These stages are implemented using "1-Bit Look-Up Table with General Output" `LUT1` primitives with the `INIT` parameter. The second design (Figure 6.1b) replaces the final buffer with a pass-through latch. More precisely, the "Transparent Latch" `LD` primitive is used with its gate input `G` tied to 1. However, the "Transparent Latch with Clock Enable and Asynchronous Clear" `LDCE` primitive can also be used, setting gate enable `GE` to 1 and clear `CLR` to 0, as suggested by Sugawara et al. [311].

The last design (Figure 6.1c) uses the "D Flip-Flop with Clock Enable and Asynchronous Preset" `FDPE` primitive, with its `D` input tied to 0. The output of the penultimate RO stage is connected to the preset input `PRE`, while its inverted value is connected to the clock port `C`. When `PRE` is high, the output `Q` is also high. On its falling edge, the FF clock transitions from low to high, thereby mirroring `D` to output a 0. The FF thus acts as a buffer. It is worth noting that the proposed RO design differs from the FF-ROs introduced in prior work, which instead depend on delay stages between the clock and clear inputs [257, 266] or the FF output and its clock [311]. The performance of the three RO designs is evaluated in Sections 6.3.3 and 6.4.5 for long-wire and cross-SLR leakage respectively.

6.2 Long-Wire Leakage Setup

This section introduces the architectural design for long-wire leakage measurements on the cloud (Section 6.2.1). It also proposes an alternative metric to estimate the delay difference due to the state of adjacent wires (Section 6.2.2). Throughout the chapter, three types of boards are used: a local VCU118 board, as well as Amazon and Huawei cloud FPGAs. Although they all contain Xilinx Virtex UltraScale+ XCVU9P chips, cloud providers reserve some clock regions for their "shell" interfaces (Section 2.2.2), and provide different clocks to the designs. No attempts are made to improve the clock accuracy via Mixed-Mode Clock Managers (MMCMs) or Phase-Locked Loops (PLLs) for any of the boards, and the boards are not modified in any way. Finally, cloud FPGAs are controlled over the Peripheral Component Interconnect Express (PCIe) interface, whereas the local board over the Universal Asynchronous Receiver/Transmitter (UART). These properties are summarized in Table 6.1.

Property	Local	AWS F1	Huawei FP1
Board	VCU118	Proprietary	Proprietary
XCVU9P Chip	flga2104-2-e	flgb2104-2-i	flgb2104-2-i
Shell Clock Regions	None	X4Y0:X5Y9	X3Y4:X5Y9
Combinatorial Loops?	Yes	No	Yes
Clock Frequency (MHz)	300	125	200
Interface	UART	PCIe	PCIe

Table 6.1: Properties of the boards used in the experiments of this chapter.

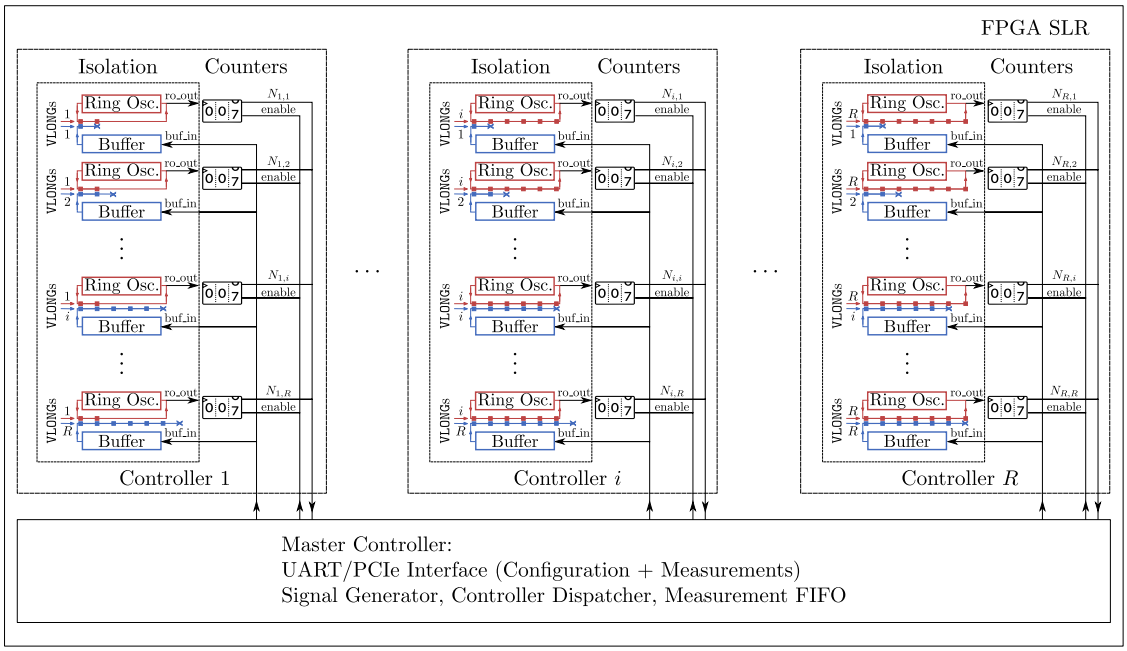


Figure 6.2: The setup uses R controllers, each with R Ring Oscillators (ROs). The ROs in controller i use i Vertical Longs (`VLONG`s) each, while the buffers adjacent to them use 1 to R `VLONG`s. The ROs and buffers of any given controller are isolated from any other logic (including the RO frequency counters), and occupy two clock regions. A master controller is responsible for handling the communication between the computer and the individual sub-controllers. Any logic used is restricted to only one Super Logic Region (SLR).

6.2.1 Architectural Design

As the compilation and programming time for the XCVU9P boards is significantly higher compared to their lower-end counterparts of Chapter 5, multiple long-wire transmitter and receiver combinations are tested at once. Specifically, the setup employs a hierarchical design of R controllers, each of which contains R identical ROs. All ROs in controller i ($1 \leq i \leq R$) use i Vertical Longs (`VLONG`s) each. Each controller also contains R buffers. These buffers use between 1 and R `VLONG`s each, and are adjacent

Parameter	Local	AWS F1	Huawei FP1
# of Boards Tested	1	8	2
# of RO Combinations	$81 = 9^2$	$64 = 8^2$	$36 = 6^2$
Vivado Version	2017.4	2018.2	2017.2

Table 6.2: Parameters of the setup used for long-wire experiments.

to the long wires of the ROs, as shown in Figure 6.2. This setup thus contains R^2 combinations of different long-wire overlaps.

The buffers and ROs of each controller are all placed on separate Configurable Logic Blocks (CLBs) spanning two clock regions, and are routed to use adjacent `VLONG`s. No other logic (including the RO frequency counters) is placed in these regions through the `EXCLUDE_PLACEMENT` and `CONTAIN_ROUTING` constraints. Finally, for a given bitstream, all ring oscillators are of the same type (i.e., LUT-ROs, LD-ROs, or FF-ROs), and all logic is placed on a single SLR. The whole experimental setup (including counters, the UART, etc.) uses less than 0.85% of all XCVU9P resources, with the ROs and buffers taking up at most $405/1{,}182{,}240 = 0.034\%$ of LUTs and $81/2{,}364{,}480 = 0.0034\%$ of registers.

Inter-device process variations are evaluated on the local VCU118 evaluation board, eight FPGAs on an AWS `f1.16xlarge` instance, and two FPGAs on two Huawei Cloud `fp1.2xlarge.11` instances. Due to the shell regions, the number of controllers and ring oscillators R differs between the three setups. These parameters are summarized in Table 6.2, with an example instantiation of the setup on AWS shown in Figure 6.3.

For each setup tested (i.e., for each RO type, on each SLR), three runs of 10,000 measurements are completed, for a total of 30,000 data points collected from each RO per testing configuration. All results are reported at the 99% confidence level. As in Chapter 5, the frequency of each RO is estimated by counting signal transitions during a 2^t clock-cycle period, with $t = 23$ (28-67 ms, depending on clock speed). Additional experiments confirm that the phenomenon persists for $t \in \{13, 15, 17, 19, 21, 25\}$, but gets noisier as t decreases (see Appendix B).

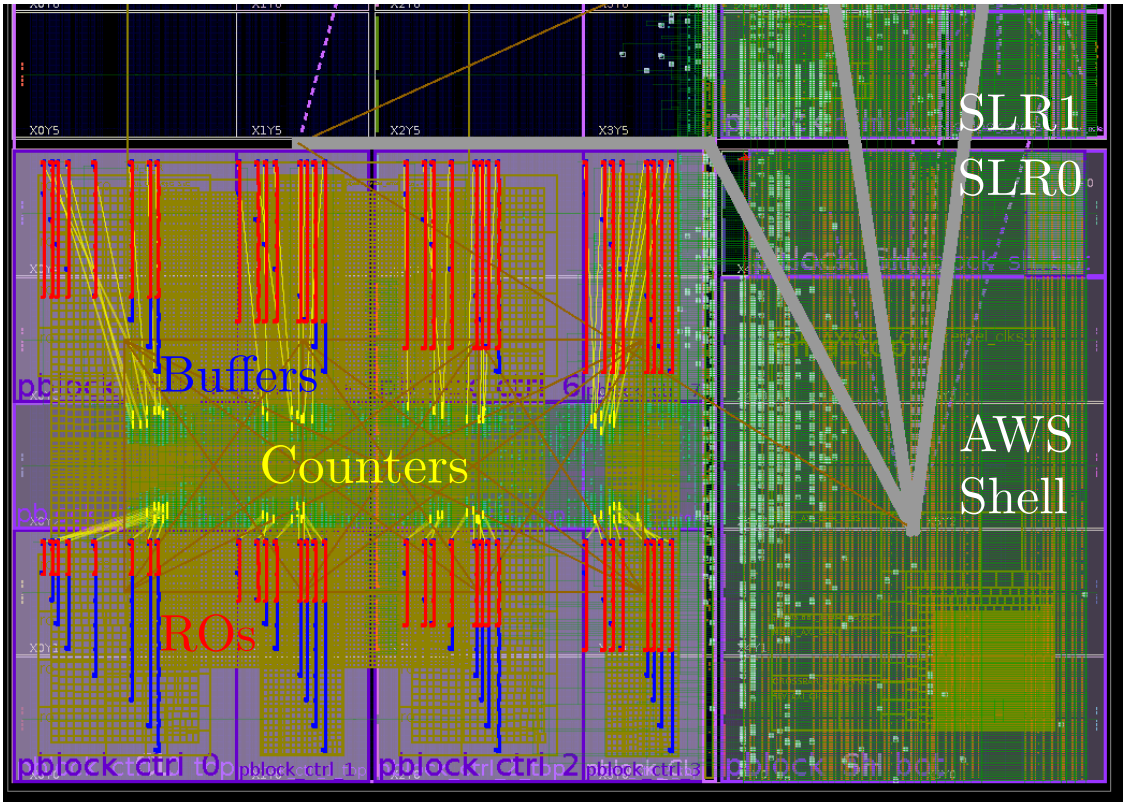


Figure 6.3: Vivado screenshot of the experimental setup for long-wire leakage measurements on an Amazon Web Services (AWS) instance. Ring oscillator long wires are highlighted in red and buffer long wires in blue, with no other logic occupying their clock regions. The counters are placed on separate clock regions, and are shown in yellow. AWS shell logic is also physically isolated, and spans multiple Super Logic Regions (SLRs).

6.2.2 Measurement Metric

Chapter 5 introduced the Relative Count Difference (ΔRC) metric to estimate long-wire leakage. Although ΔRC is independent of the measurement period and the clock frequency, it is sensitive to the absolute RO frequency, which is affected by nearby logic [206] as well as Process, Voltage, and Temperature (PVT) variations [132]. These properties make it hard to compare results across devices and experimental setups.

To overcome this drawback, this section introduces a metric which is independent of the RO frequency, and can estimate the Absolute Delay Difference (Δd_{RO}) of the long wires due to nearby state. The frequency of the ring oscillator f_{RO} is given by:

$$f_{RO} = \frac{1}{2d_{RO}} = f_{CLK} \cdot \frac{C_{RO}}{C_{CLK}} \tag{6.1}$$

where d_{RO} is the delay of the ring oscillator, f_{CLK} is the frequency of the clock, and

C_{RO}, C_{CLK} are the counts driven by the RO and the clock during a measurement period respectively (see Equation (5.1)). As a result, Δd_{RO} can be calculated as follows:[12]

$$\Delta d_{RO} = \frac{1}{2}\left(\frac{1}{f^0_{RO}} - \frac{1}{f^1_{RO}}\right) = \frac{f^1_{RO} - f^0_{RO}}{2 f^0_{RO} f^1_{RO}} \tag{6.2}$$

Assume the RO and buffer use n adjacent `VLONG`s each, and that the overlap of adjacent `VLONG`s is fixed. Then the delay of a signal travelling through the RO is $d_{RO} = n \cdot d_L + d_c$, where d_L is the delay of one long wire, and d_c is an RO-specific constant that accounts for local routing, logic delays, and process variations. As a result:

$$\Delta d_{RO} = d^0_{RO} - d^1_{RO} = n\left(d^0_L - d^1_L\right) = n\Delta d_L \tag{6.3}$$

As Δd_L is small, using $n > 1$ `VLONG`s provides a better estimate of the per-long wire delay:

$$\Delta d_L = \frac{\Delta d_{RO}}{n} = \frac{1}{n} \cdot \frac{C_{CLK}}{2 f_{CLK}} \cdot \frac{C^1_{RO} - C^0_{RO}}{C^0_{RO} C^1_{RO}} \tag{6.4}$$

The above formulas apply to all three RO designs, since the leakage affects the delay of long wires, but not the logic of the ring oscillators themselves. As a result, design differences can be incorporated in the RO-specific constant d_c, which does not influence Δd_{RO} and Δd_L as shown in Equations (6.3) and (6.4). Section 6.3 demonstrates this experimentally by comparing the per-long wire delay difference across eleven different Virtex UltraScale+ boards and all three RO designs.

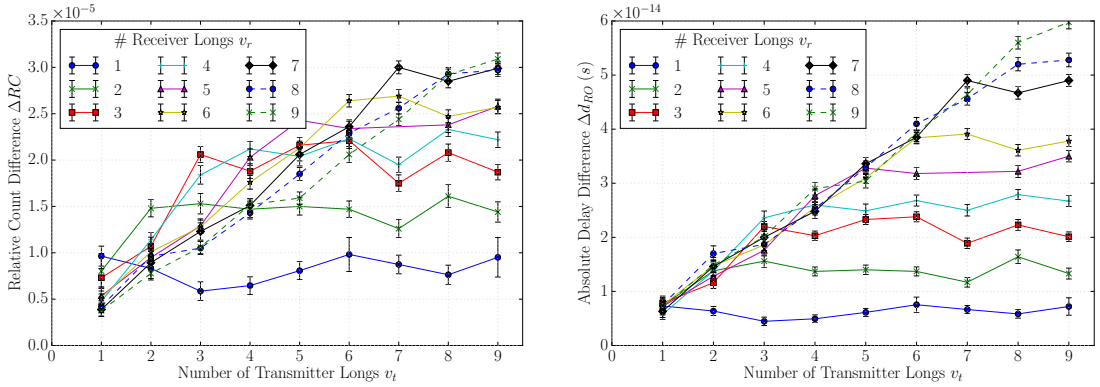
6.3 Ring Oscillator Evaluation

This section first demonstrates that the new delay-based metric is superior to the old relative count metric (Section 6.3.1). It then investigates intra- and inter-device variations in the long-wire leakage of eleven boards with three SLRs each (Section 6.3.2), and finally compares the estimates using the three different RO designs (Section 6.3.3).

6.3.1 Metric Comparison

This section compares the old ΔRC and the new Δd_{RO} metrics for all possible combinations of transmitter and receiver long wires (v_t, v_r), plotting the results in Figure 6.4.

[12] A similar formula focusing on signal transitions was later derived by Provelengios et al. [253]. My derivation precedes that publication, as verified through personal communication with one of its authors.



(a) Relative Count Difference (ΔRC) **(b)** Absolute Delay Difference (Δd_{RO})

Figure 6.4: Relative Count (a) and Absolute Delay (b) differences for different numbers of buffer transmitter and Ring Oscillator (RO) receiver long wires.

Unlike Figure 5.9 of Chapter 5, long-wire leakage measurements using the ΔRC metric are much noisier on the Virtex UltraScale+, as shown in Figure 6.4a. Indeed, although Chapter 5 explained that ΔRC for the pair $(v_t, v_r) = (9,5)$ should be lower than it is for $(v_t, v_r) = (9,6)$, Figure 6.4a suggests the opposite (the pairs $(9,7)$ and $(9,8)$ are similarly inverted).

Meanwhile, the Δd_{RO} metric (Figure 6.4b) can clearly identify the number of `VLONG`s used. Since the new metric measures the absolute delay difference due to adjacent state, it is proportional to the size of the `VLONG` overlap between the buffer and the RO, and is also independent of the clock and RO frequencies.

6.3.2 Inter- and Intra-Device Variations

This section plots the estimate of Δd_L for all SLRs and boards tested using LD-ROs. This is done by first averaging Δd_{RO} for each bitstream over all ring oscillator and buffer combinations, and then using Equation (6.4). The average values over all combinations with 99% confidence intervals are shown in Figure 6.5, which suggests that in all boards and all individual SLRs, `VLONG`s leak information about their state through a change in their delay. This change is in the order of a few femtoseconds, and exists despite the electrical differences in long wires between Virtex UltraScale+ devices and earlier generations (Table 2.1 of Chapter 2). Moreover, for *most* boards, the strength of the leakage is approximately the same for all SLRs, suggesting that SLRs in the same chip

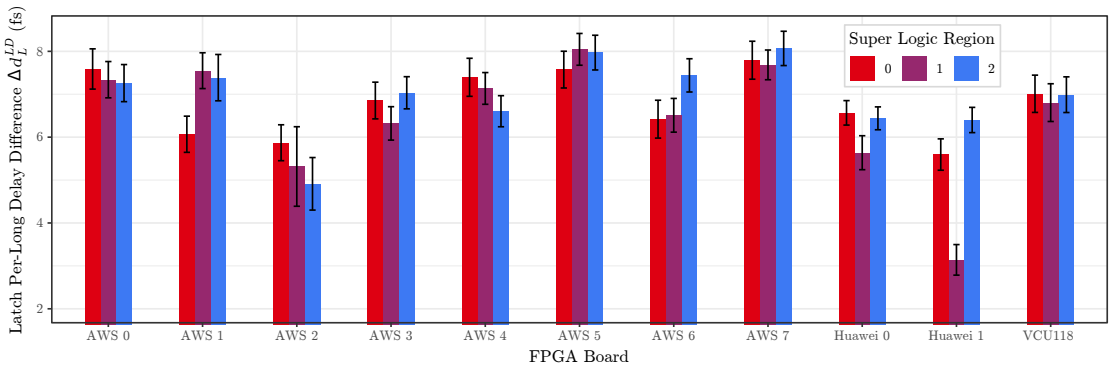


Figure 6.5: Per-long wire delay difference Δd_L^{LD} estimates using latch-based Ring Oscillators (ROs) for all Super Logic Regions (SLRs) and boards tested.

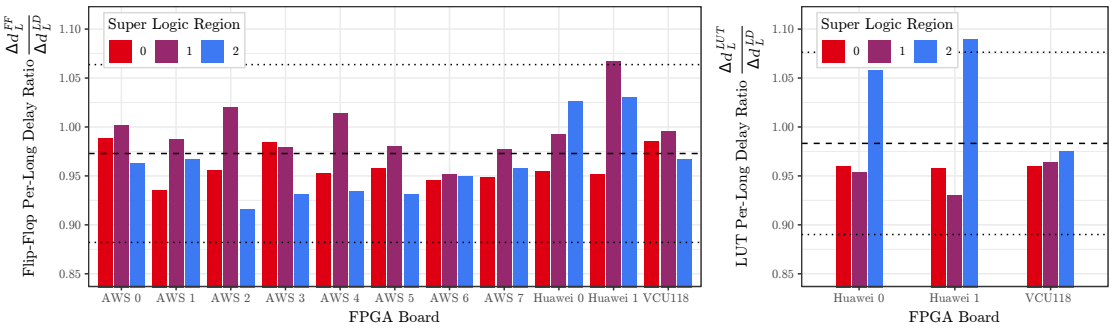


Figure 6.6: Ratio of per-long wire delay difference estimates using flip-flop-based Ring Oscillators (ROs) Δd_L^{FF} (left) and lookup-table-based ROs Δd_L^{LUT} (right) to estimates using latch-based ROs Δd_L^{LD}. The three RO designs estimate Δd_L to within 10% of each other.

might be manufactured together, therefore reducing process variations. However, the inter-SLR variation can sometimes be as large as the inter-chip variation between physically distinct boards (e.g., the AWS 1 and Huawei 1 boards). As a result, different SLRs should be treated as distinct chips with respect to process variations. Finally, within a board there is no consistent pattern in how long-wire leakage varies between SLRs, despite the heavy logic placed by cloud providers in nearby clock regions (SLRs 0 and 1). This suggests that the strength of the leakage is not influenced by nearby logic, allowing adversaries to measure it in the presence of large circuits not under their control.

6.3.3 Leakage Estimate Comparison

This section shows that all three ring oscillators give approximately the same estimate for the difference in the delay of the long wires (averaged over all RO and buffer combinations). Figure 6.6 plots the ratio of the estimate of the per-long wire delay

difference Δd_L using FFs (Δd_L^{FF}) and LUTs (Δd_L^{LUT}, where not prohibited) to the estimate using LDs (Δd_L^{LD}). It shows that there is no consistent pattern for how the estimates using different RO types vary. Even though, on average, LD-ROs result in larger Δd_L estimates compared to the other ROs, the estimates are very close in absolute terms: for example, for SLR 0 of the AWS 0 board, the 99% confidence estimate is $\Delta d_L^{LD} = 7.59 \pm 0.47$fs when using LD-ROs, and $\Delta d_L^{FF} = 7.51 \pm 0.46$fs with FF-ROs. Hence, all three ROs can distinguish nearby state, and estimate femtosecond-scale differences in the delay of `VLONG`s to within 10% of each other, despite environmental noise and process variations.

It should be noted that a metric similar to Equation (6.2) can also estimate delay differences between the different RO types, placing them in the order of 100s of picoseconds. This estimate is within the range of the speed models reported by Vivado, with LUT-ROs being the fastest, and FF-ROs being the slowest.

6.4 Cross-SLR Leakage Characterization

Having established that alternative ROs can bypass currently-deployed cloud countermeasures, this section investigates whether they can be used to create covert channels between logically and physically separated logic on the same FPGA chip. As prior work has shown that physically-isolating circuits within the same die can still result in information leaks (Section 2.2.6), this section rather places communicating circuits on separate SLR dies. The motivation for doing this partly lies in the fact that, at least for monolithic FPGAs, the strength of intra-die voltage-based leaks (i.e., the magnitude of the voltage drop) decreases with increasing distance between the transmitter and the receiver [252]. As a result, even though the power rails between different SLRs may be shared, it is reasonable to wonder whether cross-SLR communication remains possible, when logic is placed much farther apart compared to intra-SLR logic.

This system model, shown in Figure 6.7, extends that of Chapter 5 (Figure 5.1), which did not require physical isolation of malicious logic. As before, since cloud providers allow attackers to place and route logic within the confines of their dedicated regions, custom placement and routing is also allowed in this chapter. However, as explained in Section 6.6, this is not crucial for the success of the covert channel. Moreover, adversaries

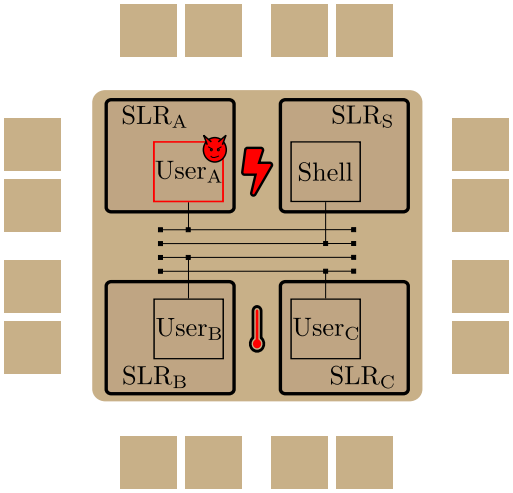


Figure 6.7: System model for multi-tenant devices with strong physical isolation. Malicious and benign user designs share the reconfigurable fabric, but are logically and physically isolated to separate Super Logic Regions (SLRs). Adversarial logic cores can act as side-channel receivers and transmitters by influencing voltage and/or temperature.

are restricted to using well-defined interfaces, such as those provided by a cloud shell, and do not have physical access to the FPGA hardware. Consequently, adversaries cannot directly read temperature and voltage conditions provided by system monitors (Section 2.2.2), but may attempt to infer or influence them indirectly.

It should be noted that as current cloud providers allocate FPGAs on a per-user basis, the attacks of this chapter do not affect cloud users in practice. However, they show that before cloud providers can implement multi-tenant FPGAs, architectural improvements in the hardware layer are necessary: even separation of tenants by SLRs on different dies is not immune to covert-channel attacks. Before Section 6.5 can calculate the covert-channel bandwidth, however, this section must first evaluate the leakage across different experimental setups, which are described in Section 6.4.1. Section 6.4.2 then adapts the metric introduced in Section 6.2.2 to these new setups. Finally Sections 6.4.3, 6.4.4, and 6.4.5 respectively characterize the information leakage across different transmitter sizes, SLR locations, and types of transmitting and receiving ROs.

6.4.1 Experimental Setup

In order to cause measurable voltage drops across SLR dies, a large number of ROs is used for transmissions. Specifically, T independent transmitters, each containing N_T

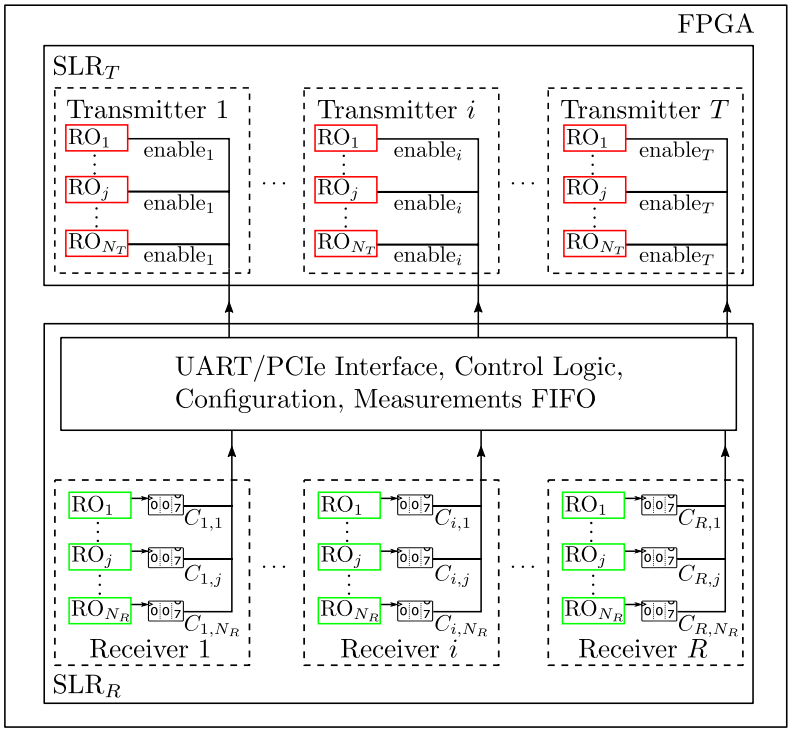


Figure 6.8: Block diagram of the experimental setup for communication between Super Logic Regions (SLRs). The $R \cdot N_R$ Ring Oscillator (RO) receivers and the $T \cdot N_T$ RO transmitters are logically and physically isolated onto separate SLRs.

Parameter	**Local**	**AWS F1**	**Huawei FP1**
# of Boards Tested	1	2	2
# of Receivers, R	5	5	5
# of ROs per Receiver, N_R	5	5	5
# of Transmitters, T	8	12	12
Vivado Version	2017.4	2018.3	2017.2

Table 6.3: Fixed parameters for experiments between Super Logic Regions (SLRs).

ROs, are placed on separate clock regions of the same SLR, SLR_T. ROs are also used as receivers: R receivers, each with N_R ROs, are placed on different clock regions of SLR SLR_R. Their frequencies are estimated using counters, which are placed along with other control logic on separate clock regions of SLR_R through the `EXCLUDE_PLACEMENT` and `CONTAIN_ROUTING` constraints. Figure 6.8 shows a diagram of the experimental setup.

Experiments are conducted on a local VCU118 evaluation board and on two FPGAs on each of the Huawei and AWS clouds. The placement and routing of transmitter ROs within their assigned clock regions is left to the manufacturer tools, while only the

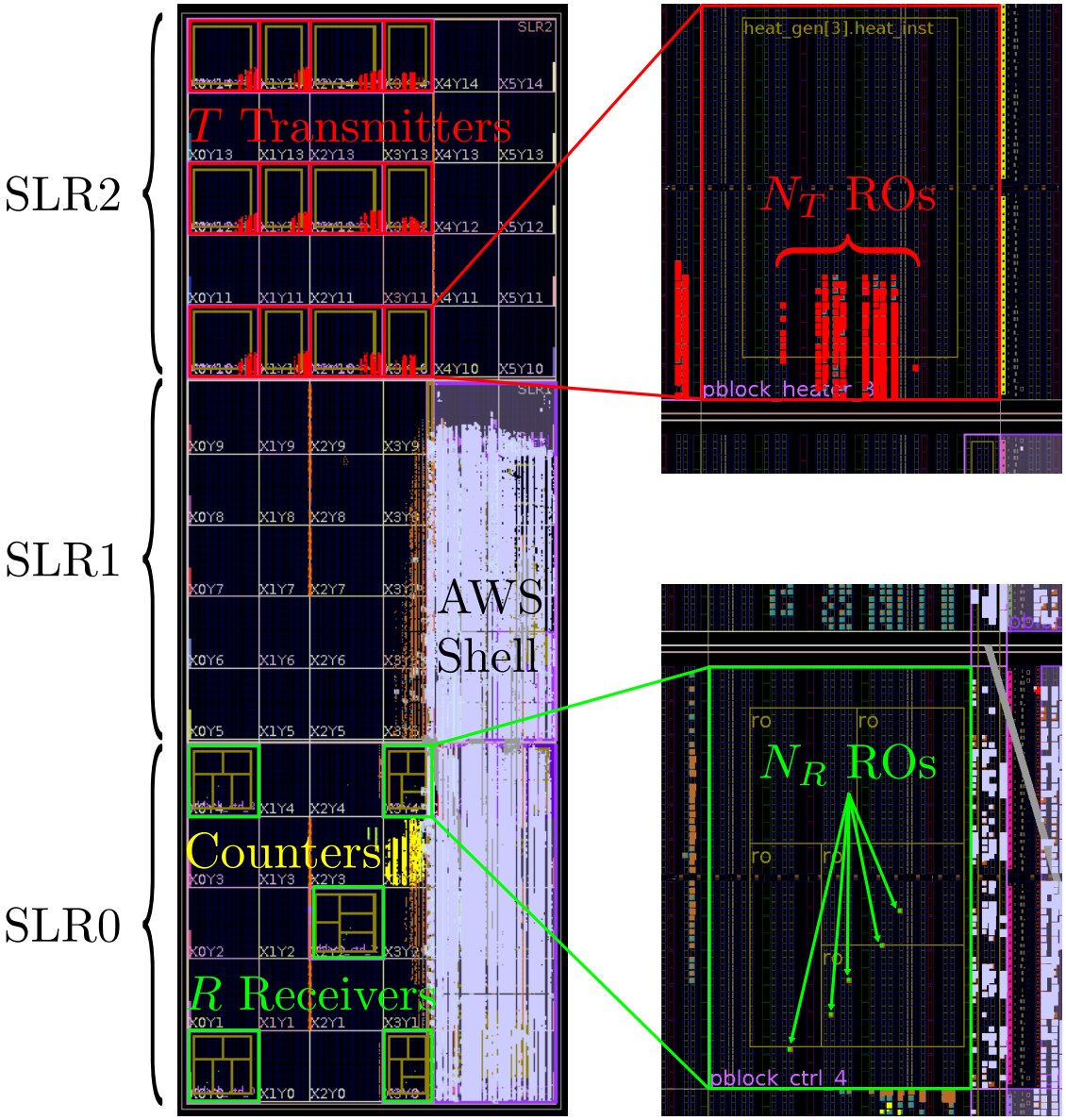


Figure 6.9: Vivado screenshot of one instantiation of the experimental setup on Amazon Web Services (AWS). The receivers (green) and counters (yellow) are on Super Logic Region (SLR) 0, while the transmitters (red) on SLR 2. Shell logic (grey) spans SLRs 0 and 1, and interfaces (brown) with the control logic.

placement of receiver ROs is fixed to identify the effect of distance on the accuracy of the communication channel. Moreover, as the cloud providers reserve some clock regions for their shell, the number and placement of the receivers and transmitters is not identical for the three setups—a fact which does not influence the quality of the communication channel, as shown in subsequent sections. These fixed properties of the experimental setup are summarized in Table 6.3, while Figure 6.9 shows an example instantiation of

Parameter	Value
RO Type	LD
# of RO Buffer Stages	2
# of ROs per Transmitter, N_T	500
Transmitter SLR, SLR_T	0
Receiver SLR, SLR_R	1
Measurement Cycles	2^7

Table 6.4: Default parameters for experiments between Super Logic Regions (SLRs).

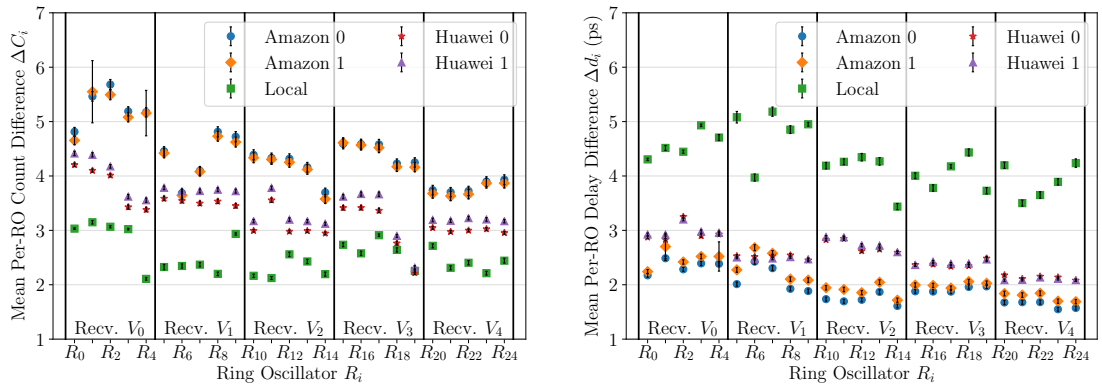
the measurement architecture on AWS for one of the experiments of Section 6.4.4.

By default, receiver and transmitter ROs use latches (LD-ROs) with two additional intermediate buffer stages for more stable measurements. The transmitters and the receivers are placed on SLRs 0 and 1 respectively. $N_T = 500$ ROs are instantiated per transmitter. Moreover, the number of RO signal transitions are counted during a 2^t clock-cycle period with $t = 7$, corresponding to 0.4-1.0 µs, depending on clock speed. These parameters are summarized in Table 6.4, and are varied in subsequent sections. For each setup, five runs of 2,048 measurements each are completed, thus collecting 10,240 data points from each RO per testing configuration. All results are reported at the 99% confidence level.

6.4.2 Measurement Metrics

To quantify the leakage, one can simply compare the average RO count for a receiver R_i when the transmitters are disabled (C_i^0) and when they are enabled (C_i^1). The differences $\Delta C_i = C_i^0 - C_i^1$ across all $R \cdot N_R = 25$ ROs per FPGA board are plotted in Figure 6.10a for the default setup of Table 6.4. Since $\Delta C_i > 0$ for all i (i.e., for all ROs on all FPGAs), the receiver can easily distinguish between 0 and 1 transmissions, with fewer than $5/127{,}875 = 0.004\%$ misclassifications per board (the encoding scheme and classification algorithm are described in Section 6.5). Moreover, Figure 6.10a illustrates that for a given RO R_i, ΔC_i is close for identical boards, with small shifts accounting for process variations.

However, as above, this metric does not allow for meaningful comparisons within or across FPGA boards due to its sensitivity to the clock and RO frequencies. The absolute



(a) Average count difference ΔC_i

(b) Average delay difference Δd_i

Figure 6.10: Average count (a) and delay (b) differences in the default setup across the $R \cdot N_R = 25$ Ring Oscillators (ROs) per board, with 99% confidence intervals.

delay difference Δd_i of each RO can instead be estimated by adjusting Equation (6.4):

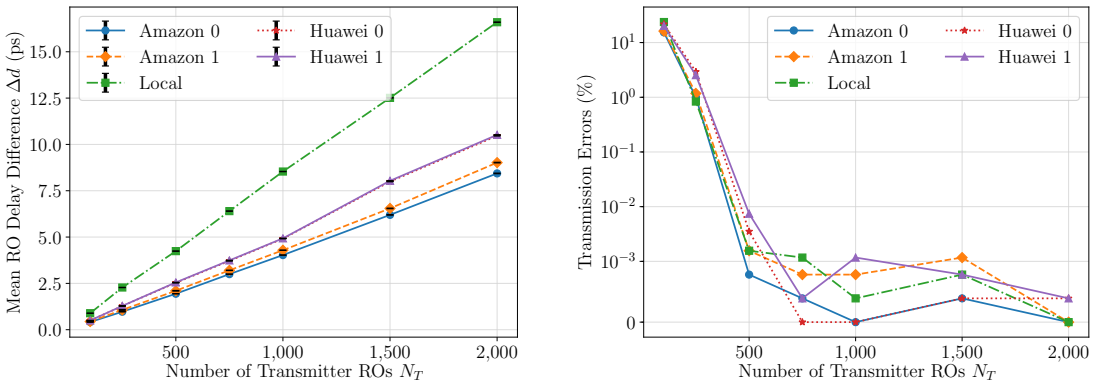
$$\Delta d_i = \frac{2^t}{2 f_{CLK}} \cdot \frac{C_i^0 - C_i^1}{C_i^0 C_i^1} \tag{6.5}$$

where f_{CLK} is the clock frequency and 2^t the measurement period.

Figure 6.10b plots the absolute delay differences Δd_i in picoseconds for the 25 ROs and five boards using Equation (6.5). For a given board, Δd_i is generally close for all ROs, and there is, on average, less variation in Δd_i within a receiver V_j compared to the variation between receivers. Moreover, the average delay Δd^j of the five ROs in receiver V_j mostly follows the distance of the receiver to the transmitters, i.e., $\Delta d^{\{0,1\}} > \Delta d^{\{2\}} > \Delta d^{\{3,4\}}$, with V_2 being an exception in the Huawei boards. Finally, boards with a faster clock frequency and a smaller shell are affected more. These effects might be attributed to increased switching activity and out-of-sync competing logic respectively. However, it is not possible to conclusively determine why the three FPGA setups behave differently, as the strength of the leakage also depends on the FPGA board voltage regulator.

6.4.3 Transmitter Sizes

This section tests the effect of increasing the size of the transmitting circuits on the strength of the cross-SLR leakage. Figure 6.11a plots the delay difference Δd averaged over all 25 ROs per board for different numbers of transmitting ROs N_T and all five FPGA boards tested. As one would expect, more transmitting ROs result in larger voltage drops,



(a) Average delay differences Δd

(b) Transmission errors

Figure 6.11: Average delay differences Δd (a) and number of transmission errors (b) for different transmitter sizes N_T on the five boards tested.

and therefore larger changes in the RO frequency. Although the local VCU118 board was designed with fewer transmitters ($T = 8$ due to differences in the communication logic over the UART instead of $T = 12$ for the cloud FPGAs over PCIe), the effect is stronger, and is consistent with the preliminary results of Section 6.4.2. It should be noted that although the average is taken over all ROs, other statistics can also be used. For example, the median, the sum, or even a fixed choice of RO, all result in similar graphs.

Figure 6.11b further plots the percentage of bit-flip errors (Section 6.5) for various transmitter sizes. It therefore demonstrates the tradeoffs between the amount of transmitter logic (area) and accuracy. A transmitter size of $N_T = 100$ results in correct classifications over 75% of the time, while increasing the number of transmitters to $N_T = 250$ and $N_T \geq 500$ results in accuracies of over 97% and 99.9% respectively.

6.4.4 Transmitter and Receiver Locations

To ensure that the covert channel is present on all locations on the FPGA, Figure 6.12 varies the SLRs on which the receivers and transmitters are placed (SLR_R and SLR_T respectively). For all SLR combinations, the leakage remains measurable, with $\Delta d > 0$. Figure 6.12 also confirms that when the transmitters and the receivers are two SLRs apart, Δd is smaller than when they are only one SLR apart. The other four placements result in similar Δd, except for $(SLR_R, SLR_T) = (2, 1)$ in the cloud, potentially due to the dynamic activity of the shell. Accuracy remains over 99.9% for all setups.

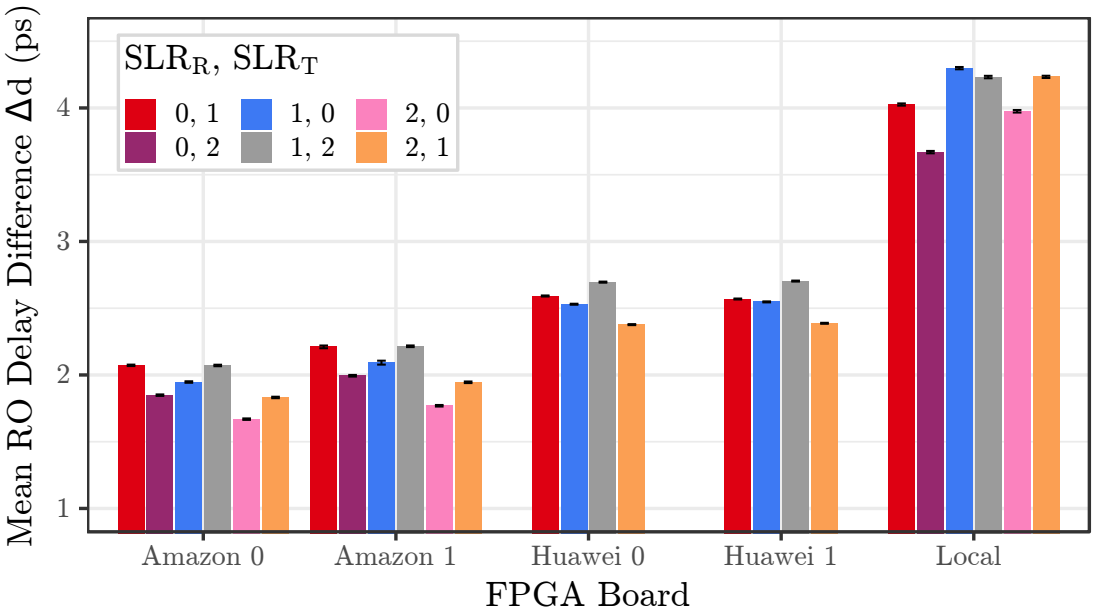


Figure 6.12: Average delay differences Δd for different receiver (SLR_R) and transmitter (SLR_T) Super Logic Regions (SLRs), with 99% confidence intervals. As placement and routing failed on SLRs 0 and 2 of the Huawei boards, they are not included in the figure.

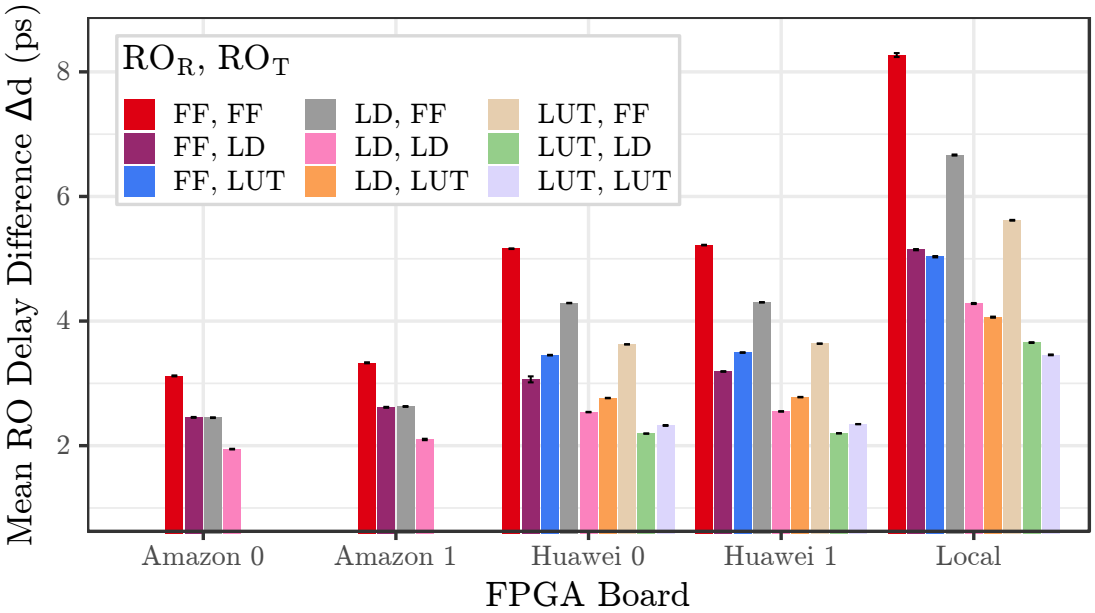
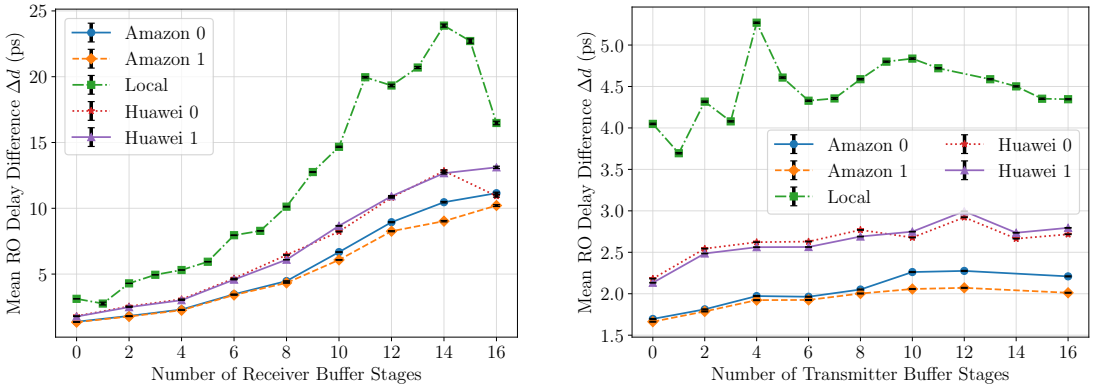


Figure 6.13: Average delay differences Δd for different receiver (RO_R) and transmitter (RO_T) types with 99% confidence intervals.

6.4.5 Ring Oscillator Properties

The next set of experiments investigates the effect of the receiver and transmitter RO types, with Figure 6.13 showing the change in delay for all nine such combinations



(a) Receiver **(b)** Transmitter

Figure 6.14: Average delay differences Δd for different intermediate buffer stages in the (a) receiver and (b) transmitter Ring Oscillators (ROs), with 99% confidence intervals.

(four on Amazon, as LUT-ROs are prohibited [12]). First of all, Figure 6.13 indicates that all three types of ROs are effective as both transmitters and receivers, since $\Delta d > 0$. Moreover, some consistent patterns emerge. For example, for a fixed receiver type, the FF-RO transmitter results in a larger Δd, as it has more stages. Similarly, for a fixed transmitter type, receiver FF-ROs are affected more than LD-ROs or LUT-ROs, as FF-ROs have more stages which are influenced by the voltage drop. By contrast, the absolute count difference ΔC follows the opposite pattern, as FF-ROs are the slowest, due to their extra inverter stage. Accuracy remains over 99.9% for all setups.

The effect of additional stages is further highlighted in Figures 6.14a and 6.14b, which vary the number of intermediate buffers in the receiver and transmitter ROs respectively. Figure 6.14a shows that more receiver buffer stages result in higher Δd. However, because the RO frequency decreases, so does ΔC, resulting in more errors: with nine stages, the error jumps from a fraction of a percent to over 2% as $\Delta C < 1$, and exceeds 26% with fifteen stages. Moreover, at sixteen intermediate buffer stages, ROs can no longer fit in a single CLB, even when using dual output `LUT6_2` lookup primitives.

Increasing the number of transmitter stages (Figure 6.14b) generally increases Δd but at a decreasing rate. This is due to a tradeoff between the amount of logic activated on a transmission of 1 (which increases), and the switching frequency of the logic (which decreases). The errors for ≥ 1 transmitter stages remain consistently below 0.01%.

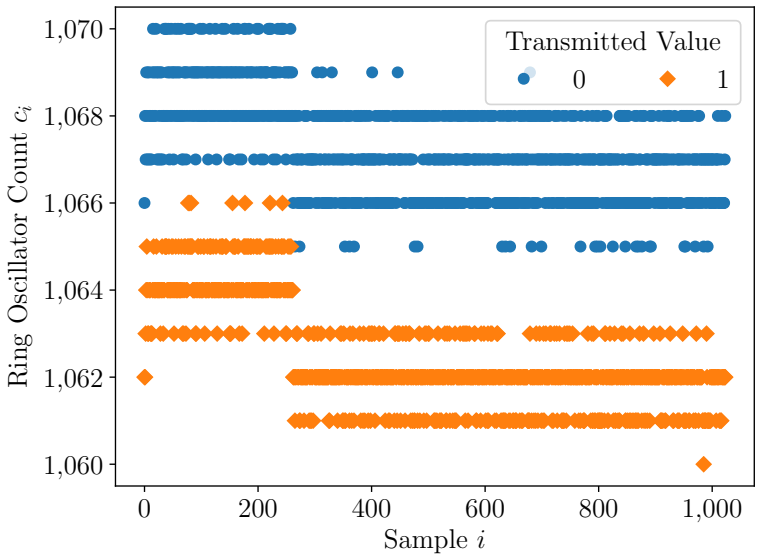


Figure 6.15: Raw ring oscillator counts from an experiment on Amazon servers. A simple threshold cannot always account for environmental conditions.

6.5 Bandwidth Analysis

With Section 6.4 having characterized cross-SLR information leakage, this section estimates the bandwidth of the ensuing covert channel. Section 6.5.1 first discusses the encoding scheme and resulting bandwidth in the basic use-case. Section 6.5.2 then examines how to increase bandwidth through multi-bit transmissions.

6.5.1 Encoding Scheme

In some setups (e.g., when using the local VCU118 board), a simple threshold is sufficient to reach accuracies of almost 100%. However, these thresholds vary per RO, require calibration, and are sensitive to environmental conditions. This is shown in Figure 6.15, which plots raw RO counts from an AWS experiment. To account for manufacturing variations as well as temperature and voltage fluctuations, a Manchester encoding scheme can again be used: a 0-bit is thus encoded as the pair $(0, 1)$, i.e., the transmitters are disabled for one measurement period, and then enabled for the next one, with a 1-bit reversing this order. Although this effectively halves the bandwidth relative to a simple threshold, it allows for on-chip classification of data by comparing successive measurements c_0 and c_1. Unlike Chapter 5, however, if $c_0 > c_1$, the bit is classified as

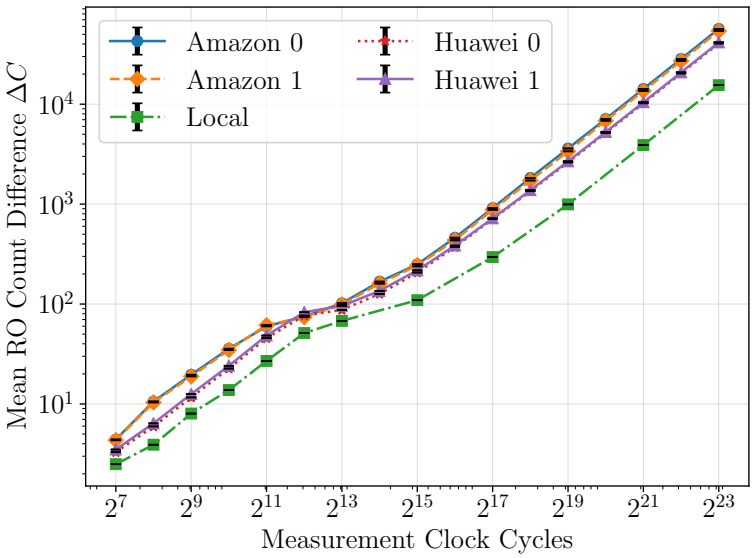


Figure 6.16: Average count differences ΔC using different measurement periods 2^t, with 99% confidence intervals for all five boards locally and on the cloud.

a 0, and as a 1 if $c_0 < c_1$ (equality can be considered an error). This is because while long-wire leakage increases RO frequencies, cross-SLR transmissions result in voltage drops, and hence slower ROs. The bandwidth b_t of the encoding scheme is:

$$b_t = \frac{f_{CLK}}{2^{t+1}} \tag{6.6}$$

where f_{CLK} is the clock frequency, and 2^t the measurement period. In the default setup, $t = 7$, so b_t is over 1.17 Mbps for the local VCU118 board, 781 kbps for the Huawei boards, and 488 kbps for the AWS boards, with over 99.9% accuracy.

Increasing the measurement time of 2^t clock cycles reduces bandwidth, but increases ΔC linearly, as shown in Figure 6.16. However, larger count differences can reduce accuracy: errors increase to about 1.1% for $t \geq 15$ on the VCU118 board, as more prolonged environmental fluctuations result in bit-flips.

It should be noted that Manchester encoding is often used for its self-clocking properties in covert-channel attacks [357] or protocols such as `10BASE-T` Ethernet [143]. However, in this thesis, rising and falling edges are detected through differences in RO frequencies. As a result, absent an external synchronization method, the receiver must sweep through the possible clock phases (linearly or with binary search): the largest (average) RO count difference ΔC corresponds to a synchronized receiver and transmitter.

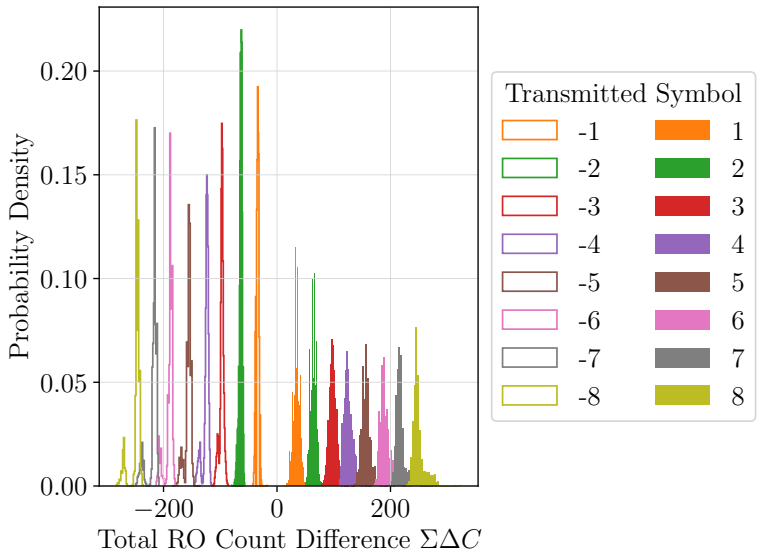


Figure 6.17: Histogram of the count differences sum $\Sigma\Delta C$ for different symbols. Symbols $\pm i$ correspond to i transmitters, enabled during the first or second measurement periods.

6.5.2 Multi-Bit Transmissions

Although previously all T transmitters were simultaneously controlled, this section shows that the bandwidth can be improved by selectively enabling only some of them. Specifically, with T transmitters, the bandwidth can be increased by a factor of $1+\log_2 T$ compared to the simple Manchester encoding scheme. The $2T$ symbols $\pm 1, \ldots, \pm T$ can be encoded in two measurement periods as follows: to transmit symbol $1 \leq i \leq T$, all transmitters are disabled during the first measurement period, and then i transmitters are turned on during the second measurement period. To transmit symbol $-T \leq i \leq -1$, the process is reversed, first enabling i transmitters, and then disabling them.

However, as the count difference ΔC for an individual RO is small, the sum of all such count differences $\Sigma\Delta C$ is considered instead. This sum is shown in Figure 6.17 for the VCU118 board, where the number of ROs per transmitter is increased to $N_T = 2{,}000$, for a total of $T \cdot N_T = 16{,}000$ ROs on the SLR. Denote the $\Sigma\Delta C$ measurements for i enabled transmitters by the set S_i, and let p_i^α be the α percentile of S_i. Then use $t_i^\alpha = (p_{i-1}^{100-\alpha} + p_i^\alpha)/2$ as the lower threshold for symbol i (with $1 \leq i \leq T$), classifying a measurement $s > 0$ as symbol i if $t_i^\alpha \leq s < t_{i+1}^\alpha$ (the $s < 0$ case is analogous for $-T \leq i \leq -1$). Using this classification scheme with $\alpha \in \{1, \ldots, 20\}$, all $2T$ symbols can be recovered

over 96% of the time, reaching a bandwidth of $b_t^T = \log_2(2T) \cdot f_{CLK}/2^{t+1} = 4.6\,\text{Mbps}$. The maximum accuracy of 97.6% occurs for $\alpha = 8$, while $\alpha = 0$, corresponding to minima and maxima thresholds, can correctly recover about 80% of transmissions.

6.6 Countermeasures

There are four types of defense mechanisms that can be deployed to protect against the types of attacks introduced in this section: placement and routing restrictions; design rule checks; balanced power usage; and runtime protections.

Placement & Routing Restrictions: Chapter 5 (and the first half of this chapter) showed that physical isolation is a necessary prerequisite for secure multi-tenant FPGAs, despite "large costs in terms of frequency and routing congestion" [377]. However, isolation within a die is not a sufficient protection mechanism on its own [167, 197, 282, 387]. This chapter has further shown for the first time that isolation across SLRs is also not enough to protect multi-tenant FPGA designs. The cross-SLR channel of this chapter did not fundamentally depend on custom placement and routing: only placement directives were used to measure the effect of distance on the ensuing channel. As a result, defense mechanisms that depend on randomizing or disallowing user-defined layouts are also ineffective.

Design Rule Checks (DRCs): An alternative approach is to prevent receiver circuits from being instantiated in cloud FPGAs, such as disallowing combinatorial loops [121, 122, 167, 332, 387]. However, as this chapter showed, alternative RO designs can bypass such checks. Banning latches [122, 168, 311] can prevent LD-ROs and Time-to-Digital Converters (TDCs) from being deployed, although alternative RO and TDC designs may overcome such restrictions. One example is the FF-RO of this section, which instead uses local clocks to drive flip-flops and can thus also be detected. Overall, checks prohibiting latches and ensuring that register clocks are derived from cloud shells can raise the bar for attackers, until alternative malicious designs emerge. Some specific DRC warnings that appear in the designs of this section can thus be promoted to errors:

- `LUTLP-2` detects combinatorial loops, but is only an error on AWS.
- `PLHOLDVIO-2` warns about "non-optimal connections" in LUT-driven clocks.

- `PDRC-153` appears in "gated clock nets" when not using the clock enable `CE` pin.

Balanced Power Usage: Another defense mechanism is to prevent transmitter circuits from being created in cloud FPGAs. Although ROs were used in this chapter, other designs with large dynamic power consumption (e.g., programmable interconnects [390]) could also be employed. Consequently, malicious senders can likely still find circuit designs that modulate dynamic power draw despite cloud FPGA DRCs. In fact, as Chapter 7 shows, Central Processing Unit (CPU) and Graphics Processing Unit (GPU) activity can be used in lieu of FPGA transmissions for cross-board communications.

Runtime Protections: To prevent damage to the physical hardware, runtime monitors for temperature and power usage are necessary. Although AWS gates clocks should a maximum power threshold be reached [10], ROs could still cause damage to the device, as they cannot be stopped. This fact necessitates more aggressive protection mechanisms, such as clearing the FPGA. However, voltage fluctuations can come from legitimate circuits, so any such mechanism must ensure that there are low false positives.

Overall, future FPGAs are in need of architectural improvements, which will likely come with heavy performance and energy overheads. Independent power supplies to different tenants (i.e., SLR dies) might be a good start, but this may still not be enough: cross-FPGA channels are possible even with independent board voltage regulators (Chapter 7).

6.7 Summary

This chapter introduced novel RO designs which can be instantiated on cloud FPGAs, bypassing currently-deployed countermeasures (Section 6.1). These ROs can measure femtosecond-scale changes in the delays of Xilinx Virtex UltraScale+ long wires both locally and on the Huawei and Amazon clouds (Sections 6.2 and 6.3). Moreover, the ROs can be used to characterize cross-SLR leakage across multiple parameters, such as the locations, types, and sizes of the source transmitters and sink receivers (Section 6.4). The ensuing cross-SLR covert channel has a bandwidth of up to 4.6 Mbps that is over 97.6% accurate due to multi-bit transmissions (Section 6.5). Overall, a lack of robust and general countermeasures (Section 6.6) highlights the need for hardware-level architectural changes if secure multi-tenant FPGAs are to become a reality in the future.

The greater the power, the more dangerous the abuse.

— Edmund Burke

7 Cross-Board Covert-Channel Attacks

Contents

The previous two chapters investigated Field-Programmable Gate Array (FPGA) security in the context of multi-tenant occupancy of the reconfigurable fabric, revealing

several shortcomings of the underlying hardware that prevent shared clouds from becoming a reality in the near future. However, it is also important to investigate whether attacks are possible in single-tenant scenarios. One potential source of leakage that persists across chip boundaries is data-dependent power draw and associated voltage fluctuations. Although power and voltage are easy to measure and exploit using external equipment [103, 131, 147, 164, 274, 284, 305, 310],[13] doing so is not possible in cloud environments without physical access to the underlying hardware.

Schellenberg et al. recently showed that cross-FPGA attacks are possible using Time-to-Digital Converters (TDCs) [283]. However, the chips targeted were located on the same FPGA board, and hence shared the same voltage regulator. This made them much easier to influence directly, due to the lack of additional intermediate components between their Power Distribution Networks (PDNs). Moreover, the boards used were explicitly "designed for external side-channel analysis research" [283], so the threat model employed did not quite capture the reality of cloud environments.

By contrast, this chapter investigates a new class of remote, cross-FPGA attacks between separate boards, mirroring cloud environments with co-located FPGAs in the same server rack. Specifically, this chapter shows how powering two independent, off-the-shelf boards by the same Power Supply Unit (PSU) can leak information between them, breaking isolation and separation of privilege. The same source of leakage can also be used to detect high levels of activity by Central Processing Units (CPUs) and Graphics Processing Units (GPUs) if the PSU also powers a computer. This chapter therefore:

1. Identifies a new threat model, where shared PSUs are a source of vulnerability, even for unprivileged FPGA designs without access to voltage or temperature system monitors (Section 7.1).

2. Introduces a novel measurement setup that deploys Ring Oscillators (ROs) on the sink FPGA to stress its voltage regulator (Section 7.2).

3. Proposes a classification metric that uses the stressor ROs to reliably detect external voltage fluctuations (Section 7.3).

[13] Recent work has shown that key recovery attacks are possible remotely by measuring Analog-to-Digital Converter (ADC) noise [118, 235], but these attacks remain limited to single-chip systems.

4. Creates the first remote covert-channel attack between FPGAs on distinct physical boards that are dedicated on a per-user basis, reaching accuracies of up to 100% (Section 7.4). The cross-FPGA covert channel is demonstrated between off-the-shelf, unmodified, Xilinx-designed Artix 7 and Kintex 7 boards in either direction of communication, across various architectural choices (e.g., number, type, and size of the ROs), and with different bandwidth-accuracy tradeoffs.

5. Establishes the first CPU-to-FPGA and GPU-to-FPGA covert channels that detect high loads of activity on the respective processors (Section 7.5).

6. Discusses the practicality of the proposed attacks, as well as potential countermeasures to reduce the impact of the leakage (Section 7.6).

In summary (Section 7.7), the FPGA-to-FPGA, CPU-to-FPGA, and GPU-to-FPGA attacks of this chapter show that there is a fundamental need to re-think the security of hardware, even for monolithic, single-tenant designs in shared infrastructures.

7.1 System and Adversary Model

Although Chapters 5 and 6 were concerned with multi-tenant FPGA attacks without and with physical isolation respectively, this chapter focuses on remote attacks against platforms where the entire logic is allocated to a single user. Design logic cannot access voltage or thermal system monitors present on the FPGA fabric, as these are inaccessible in a cloud environment due to the presence of the shell (Section 2.2.2). This stronger threat model necessitates that any effect caused by a side- or covert-channel transmitter be measurable across extensive physical separation (as opposed to logic on the same FPGA chip), and with multiple intermediate components (passive capacitors, inductors, voltage regulators, etc.) on the path between the source and sink FPGA boards.

This chapter specifically investigates remote voltage-based attacks, where a shared PSU provides an indirect connection between CPUs, GPUs, and FPGAs. The high-level system model of this (leaky, as it turns out) connection is shown in Figure 7.1.

This chapter makes no assumptions regarding how the FPGAs are connected to the computer: FPGAs may use the Peripheral Component Interconnect Express (PCIe)

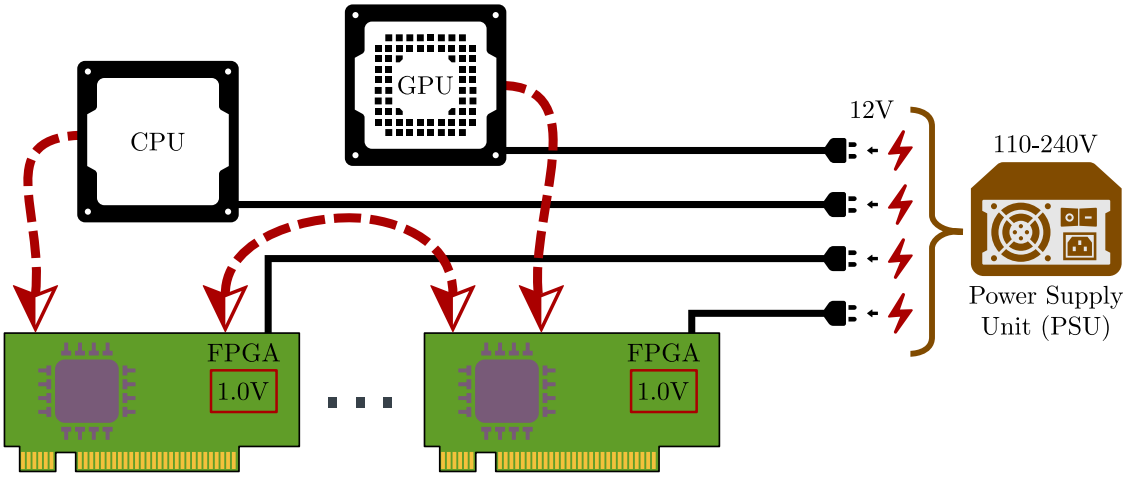


Figure 7.1: System model for voltage-based covert channels from Field-Programmable Gate Arrays (FPGAs), Central Processing Units (CPUs), and Graphics Processing Units (GPUs) to other FPGAs in co-located environments. The devices are powered through the same Power Supply Unit (PSU), but do not share logic, and do not have access to any system monitors for measuring voltage or temperature changes.

interface, the Universal Asynchronous Receiver/Transmitter (UART) standard, or they might not even be (logically) connected to the computer at all. In other words, the only assumption is that of a shared PSU between the two communicating parties, operating under normal conditions (i.e., without being overloaded).

Within an FPGA, (potentially adversarial) users are allowed to place and route the designs of their choice, respecting any restrictions imposed by potential cloud-provided shells. One of the key contributions of this chapter is therefore the ability to communicate across unmodified devices, without external equipment or access to internal voltage monitors, which are off-limits to unprivileged FPGA designs.

7.2 Experimental Setup

This section details the experimental setup, with Section 7.2.1 first delving into the architectural design of the FPGA transmission and reception circuitry. Section 7.2.2 then describes the hardware properties of the FPGA boards used, while Section 7.2.3 expands on the computer PSUs, CPUs, and GPUs, which are effectively turned into covert-channel transmitters. Section 7.2.4 finally discusses the process followed for data collection.

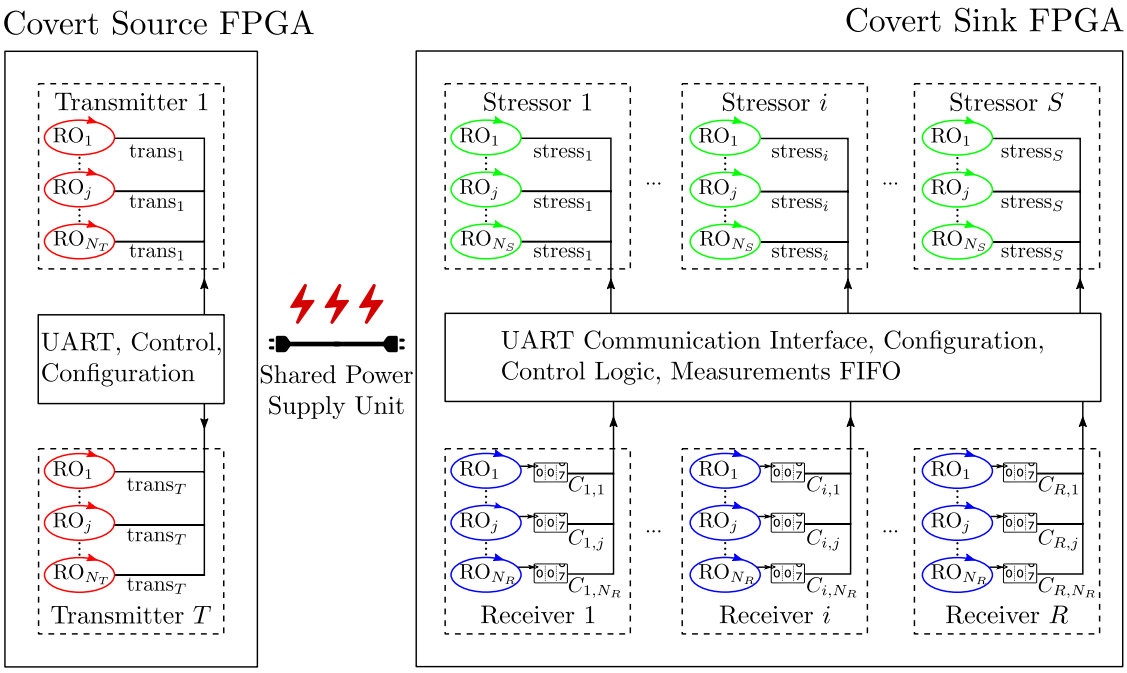


Figure 7.2: Experimental setup block diagram. The covert source (left) uses $T \cdot N_T$ Ring Oscillators (ROs), while the sink (right) has $R \cdot N_R$ measurement ROs and $S \cdot N_S$ stressor ROs. The source and sink boards are powered by the same power supply unit.

7.2.1 Architectural FPGA Design

This section gives a high-level overview of the covert-channel source and sink FPGA designs, noting that the ROs used have one inverter and three buffer stages, and are implemented using Lookup Tables (LUTs), i.e., are LUT-ROs (Figure 6.1). The additional buffer stage was added after preliminary experiments showed that it made measurements more stable. Alternative types of ROs are evaluated in Section 7.4.5.

Channel Source: To cause detectable changes on the sink, the source FPGA employs ring oscillators organized as T *transmitters*, which can be controlled independently. These transmitters are placed on separate clock regions to make power consumption more evenly spread throughout the FPGA. They contain N_T ROs each, for a total of $T \cdot N_T$ ROs, as shown in the left part of Figure 7.2.

Channel Sink: To receive transmissions, the sink employs R *receivers*, placed on separate clock regions of the sink FPGA, and each containing N_R ROs. The RO frequency can be estimated by counting the number of RO signal transitions in a fixed measurement interval of 2^t clock cycles through counters placed outside of the RO clock regions.

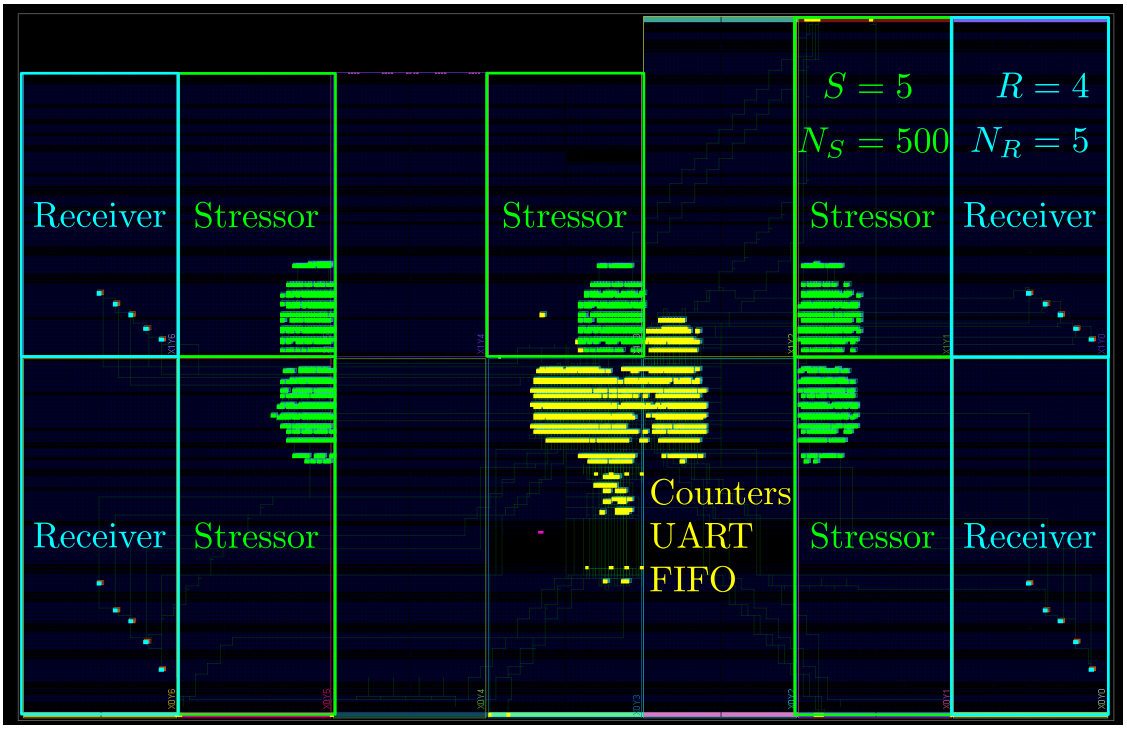


Figure 7.3: Vivado screenshot (rotated) of the sink architecture on a Kintex 7 board, with receiver Ring Oscillators (ROs) in blue, stressor ROs in green, and other logic (counters, Universal Asynchronous Receiver/Transmitter (UART), and First In, First Out (FIFO) interfaces) in yellow.

However, this setup on its own is not sufficient to decode covert transmissions, due to inherent noise in the power supply and environmental fluctuations. Instead, it is necessary to introduce additional circuitry on the sink FPGA which stresses the board's voltage regulator, making it harder to maintain a constant voltage. This fact allows the sink FPGA to sense voltage changes induced by the source FPGA, or even by CPU and GPU activity, as later presented in Section 7.5. Specifically, the sink design includes S *stressors*, each with N_S ROs. As with the source transmitters, these S stressors are placed on separate clock regions, and can also be controlled independently. The block diagram for the sink design is shown in the right part of Figure 7.2, while Figure 7.3 shows a concrete instantiation of the sink architecture on a Kintex 7 board. Section 7.3 further demonstrates the need for stressor ROs.

7.2.2 FPGA Boards

The experiments of this chapter use Xilinx Kintex 7 KC705 and Artix 7 AC701 boards. The 28 nm chips these devices contain are similar, but the Kintex 7 is larger and more

Property/Parameter	Artix 7	Kintex 7
Board	AC701	KC705
Part Number	XC7A200T	XC7K325T
Slices	33,650	50,950
Clock Regions	2×5	2×7
Core Voltage, `VCCINT`	1.0 V	1.0 V
Voltage Regulator	LMZ31710	PTD08A020W
Clock Frequency (MHz)	200	200
# of Boards Tested	2	2
# of Transmitters, T	10	14
# of Stressors, S	5	5
# of Receivers, R	4	4
# of ROs per Receiver, N_R	5	5

Table 7.1: Board properties and fixed compile-time choices for the source and sink designs.

performant, while the Artix 7 is optimized for low power [358, 359]. Both FPGAs have a 200 MHz oscillator and operate at a core `VCCINT` voltage of 1.0 V, but the boards use different regulators to convert the 12 V PSU output to 1.0 V [360, 363].

The source FPGA designs place a transmitter on each clock region of the FPGA. As the Artix 7 board has 10 clock regions, while the Kintex 7 has 14, the number of transmitters on these devices is $T = 10$ and $T = 14$ respectively. The sink FPGAs contain $R = 4$ receivers in the corners of the chip, each with $N_R = 5$ ROs. Sink FPGAs also contain $S = 5$ stressors, one of which is placed in the center of the device, while the remaining four next to the receiver clock regions (Figure 7.3 shows an example with $N_S = 500$). Although not shown to be significant in the experiments of this chapter, these early architectural choices were made to ensure that the power draw was approximately equally spread across the PDN of the FPGA fabric.

These decisions and other FPGA properties are summarized in Table 7.1. More compile-time and run-time parameters, such as the measurement period and the number of source transmitters ROs N_T and sink stressor ROs N_S, are varied in Section 7.4.

7.2.3 Power Supply Units & Computer Transmitters

To verify that the covert channel is not due to faulty design in a line of specific power supply units, this chapter tests communication on two PSUs made by different manu-

Property	Computer A	Computer B
PSU Brand	Corsair	Dell
Power Rating	850 W	1,300 W
80 Plus Certification	Gold	Gold
Motherboard	SuperMicro X8DAL-i	Dell Precision T7600
Xeon CPU Model	E5645	E5-2609
# of CPU Cores	6 @ 2.4 GHz	4 @ 2.4 GHz
# of Threads	12	4
# of CPUs	2	1
GeForce GPU	ZOTAC GT 430	EVGA GTX 750 Ti
GPU Memory	1 GB GDDR3	2 GB GDDR5
# of CUDA Cores	96 @ 0.7 GHz	640 @ 1.0 GHz

Table 7.2: Hardware properties of the two computers used, with their corresponding Power Supply Units (PSUs), Central Processing Units (CPUs), and Graphics Processing Units (GPUs).

facturers (Corsair and Dell), rated for different loads (850 W and 1,300 W respectively), and both with a Gold 80 Plus Certification (which guarantees 90% efficiency at 50% load). These PSUs are integrated in two computers, the first of which contains two Xeon E5645 CPUs for a total of 24 threads, while the second a single Xeon E5-2609 with 4 threads. They also contain Nvidia GeForce GPUs, with 96 and 640 Compute Unified Device Architecture (CUDA) cores respectively. The CPU and GPU cores are used as the covert-channel sources in Section 7.5. The properties of the computer hardware used are summarized in Table 7.2.

7.2.4 Data Collection and Encoding

For the data collection process, the computers attached to the PSUs were used normally during experimentation, including running and installing other software. To ensure leakage is not due to temperature, for most experiments, the FPGAs were placed outside the computer case and away from computer fans, which may affect measurements by turning on or off based on the computer temperature. FPGAs were similarly placed next to each other horizontally (as opposed to being stacked vertically), further minimizing cross-FPGA temperature effects. In addition, to ensure that the information leakage is not caused by other voltage effects, the FPGAs were not connected to the computer over

PCIe, which would likely increase the potential for leakage. However, as Section 7.4.5 shows, the covert channel operates with similar accuracy, even when the FPGAs are connected to the computer over PCIe and are enclosed in the computer tower without accounting for temperature variations. Finally, to verify that the leakage is not caused by the UART interface, one computer often takes the measurements, when the other one is powering the source and sink boards through its PSU.

As there is inherent noise in the measurements, (a) the absolute RO frequency is not well-suited for comparison, and (b) the RO counts need to be averaged over repeated measurements to produce meaningful results. To address both concerns, Manchester encoding is used, where to send a 1, the source transmitters are enabled for one measurement period and disabled for the next (a 0 is similarly encoded by first disabling transmitters during the first measurement period and enabling them in the second period). These measurement periods are $M \cdot 2^t$ clock cycles long, averaging M RO counts collected by ROs every 2^t clock cycles (see Section 7.3). The bandwidth can thus be calculated as:

$$b = \frac{f_c}{2 \cdot 2^t \cdot M} \tag{7.1}$$

where $f_c = 200\,\text{MHz}$ is the FPGA clock frequency.

Most experiments transmit the 20-bit number `0xf3ed1`, which is Manchester-encoded in 40 bits to be communicated across the covert channel. This number was chosen as it has relatively long runs of ones and zeros, but more and longer patterns are evaluated in Section 7.4.5. Moreover, to ensure that perfect synchronization is not needed between the source and the sink, for each of the 40 periods, four sets of M measurements are taken, where M is in the order of a few hundred (see Table 7.3 and Section 7.4.4). The four sets of repetitions create $4^2 = 16$ Manchester-encoded pairs per bit to be transferred, for a total of $16 \times 20 = 320$ pairs to estimate the covert-channel accuracy.

7.3 Classification Metric

This section introduces a novel methodology to detect changes in the power supply voltage through “stressor” ROs. Section 7.3.1 first motivates why the naive approach

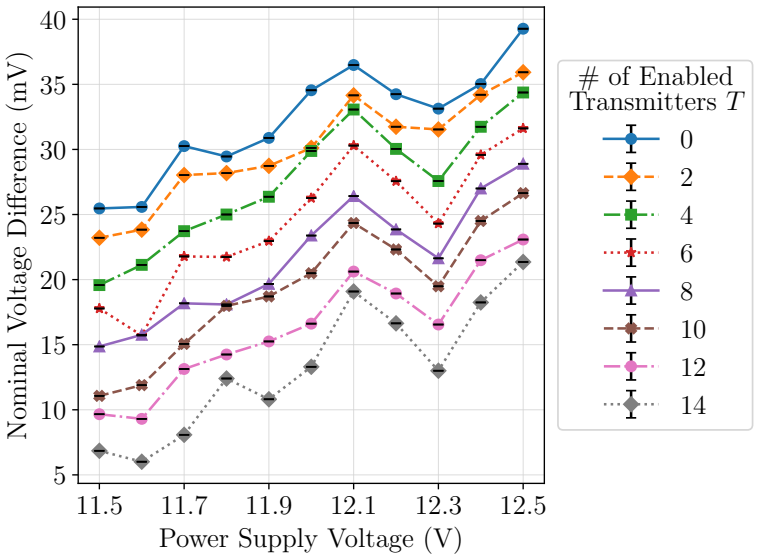


Figure 7.4: Voltage as set by the power supply and measured by the oscilloscope for various numbers of enabled transmitters T on the KC705-2 source, with 99% confidence intervals.

of using the absolute ring oscillator counts is insufficient for the classification of transmissions in this scenario. Section 7.3.2 then introduces the new stressor-based metric, while Section 7.3.3 finally explains why the proposed technique works.

7.3.1 Motivation

Broadly speaking, when the transmitters are activated on the source FPGA (and similarly on the source CPU or GPU), there is a voltage drop that is visible not just at the board regulator, but also at the 12 V rail input of the source FPGA. Indeed, Figure 7.4 demonstrates this for a Kintex 7 source across multiple input voltages and different numbers of enabled transmitters T, with N_T = 1,000. The board is powered by a Keithley 2231A power supply. Simultaneously, a Tektronix MDO3104 Mixed Domain Oscilloscope with TPP1000 1 GHz passive probes measures the voltage at the power rail of the board, taking 10,000 data points. Figure 7.4 indicates that at any voltage level provided by the power supply (11.5-12.5 V), as the number of enabled source transmitters T increases, the voltage measured by the oscilloscope decreases. For example, at 12.5 V, the oscilloscope measures 12.539 V when no transmitters are enabled, but only 12.521 V when 14 transmitters are enabled, for a voltage drop of approximately 18 mV. At 11.5 V,

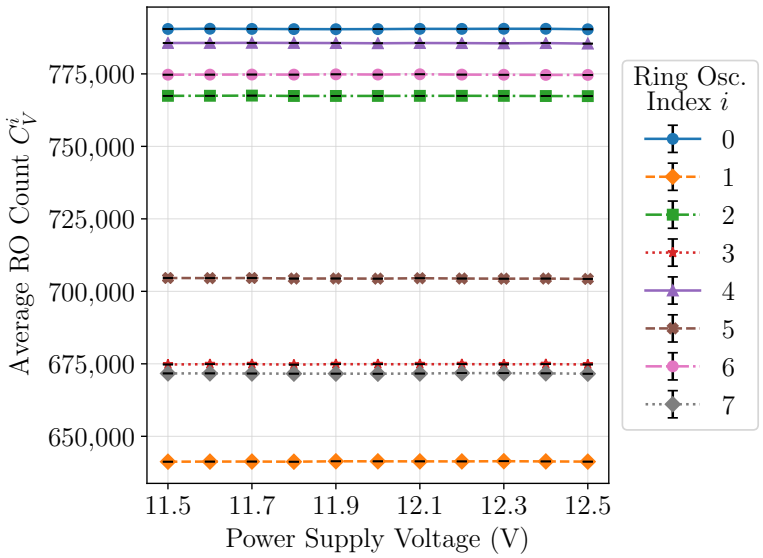


Figure 7.5: The average Ring Oscillator (RO) counts C_V^i (at 99% confidence) on the AC701-1 sink remain approximately the same for different power supply voltages V and all eight ROs R_i.

the measured voltage similarly drops from 11.525 V to 11.507 V.

For a ring oscillator i, let its average count be C_V^i when the voltage provided by the power supply is $11.5\,\text{V} \leq V \leq 12.5\,\text{V}$. Although one would expect the RO frequency to increase with higher supply voltages [132], i.e., $C_{V_1}^i > C_{V_2}^i$ whenever $V_1 > V_2$, this is not the case in practice, as the voltage regulator stabilizes `VCCINT` to around 1.0 V. Instead, Figure 7.5 suggests that the RO counts remain approximately the same for all eight ring oscillators and voltages V tested on an Artix 7 sink for measurement periods of 1.3 ms. As a result, the absolute RO frequency cannot be used to decode cross-FPGA covert-channel transmissions.

7.3.2 Description

The issues identified above can be solved by introducing ROs to "stress" the voltage regulator and make external changes in the power supply voltage measurable. For any bit transmission (say the i-th one), the sink FPGA takes M measurements as follows:

1. For the first measurement period, it disables all stressor ROs, and lets the receiver ROs run for 2^t clock cycles, producing counts $\mathbf{C}_0^i = (C_0^0, \ldots, C_0^{R \cdot N_R - 1})$.

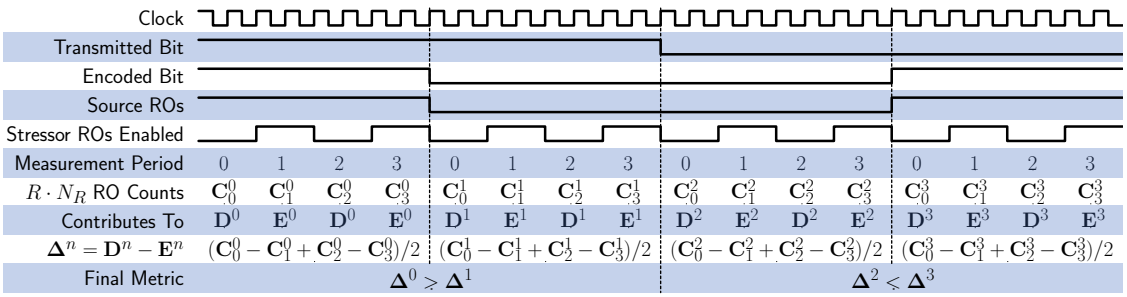


Figure 7.6: Timing diagram for a Manchester-encoded transmission of the two bits `10`, with $M = 4$ measurement periods. Half of the Ring Oscillators (ROs) counts are taken when the stressors are enabled (**E**), and the other $M/2 = 2$ counts when they are disabled (**D**) to compute $\mathbf{\Delta} = \mathbf{D} - \mathbf{E}$. The receiver uses the sign (positive or negative) of the difference $\mathbf{\Delta}^{2n} - \mathbf{\Delta}^{2n+1}$ between the two parts of the encoded transmission of the n-th bit to determine its decoded value. For example, $(\mathbf{C}_0^0 - \mathbf{C}_1^0 + \mathbf{C}_2^0 - \mathbf{C}_3^0)/2 = \mathbf{\Delta}^0 > \mathbf{\Delta}^1 = (\mathbf{C}_0^1 - \mathbf{C}_1^1 + \mathbf{C}_2^1 - \mathbf{C}_3^1)/2$, so the first bit is decoded as a 1. Similarly, $\mathbf{\Delta}^2 < \mathbf{\Delta}^3$, so the second bit is decoded as a 0.

2. In the second measurement period, it enables all (or some, see Sections 7.3.3 and 7.4.4) stressor ROs, and estimates the RO frequencies through their counts, $\mathbf{C}_1^i$.

3. In the third measurement period, it disables all stressor ROs, re-enables them in the fourth period, and so forth.

This procedure produces $M/2$ measurements $\mathbf{C}_0^i, \mathbf{C}_2^i, \ldots$ corresponding to disabled stressors, and $M/2$ measurements $\mathbf{C}_1^i, \mathbf{C}_3^i, \ldots$ corresponding to enabled stressors, as also shown in the timing diagram of Figure 7.6. Figure 7.6 represents Manchester-encoded transmissions of the two bits `10`, averaging over $M = 4$ measurements and only repeating transmissions once (actual measurements have $M = 500$, with four repetitions). The average of each set per RO is then used to calculate the disabled-stressor average $\mathbf{D}^i = 2/M \cdot \sum_{k=0}^{M/2-1} \mathbf{C}_{2k}^i$ and the enabled-stressor average $\mathbf{E}^i = 2/M \cdot \sum_{k=0}^{M/2-1} \mathbf{C}_{2k+1}^i$. The value $\mathbf{\Delta}^i = \mathbf{D}^i - \mathbf{E}^i$ is the final metric for the recovery of the transmitted bit.

Specifically, assume that the sink FPGA is trying to recover the n-th bit, corresponding to transmissions $2n$ and $2n+1$. Since each bit b is Manchester-encoded as the pair $(b, 1-b)$, each transmission pair contains a 1-bit and a 0-bit. This allows the sink board to compare the $R \cdot N_R$ counts of $\mathbf{\Delta}^{2n}$ and $\mathbf{\Delta}^{2n+1}$. If the majority of the RO differences in the first set of measurements is bigger than the corresponding differences in the second set of measurements (i.e., $\Delta^{2n} > \Delta^{2n+1}$ for most ROs), it classifies the n-th bit as a 1, while if the majority is smaller ($\Delta^{2n} < \Delta^{2n+1}$ for most ROs), it classifies the n-th bit as a 0.

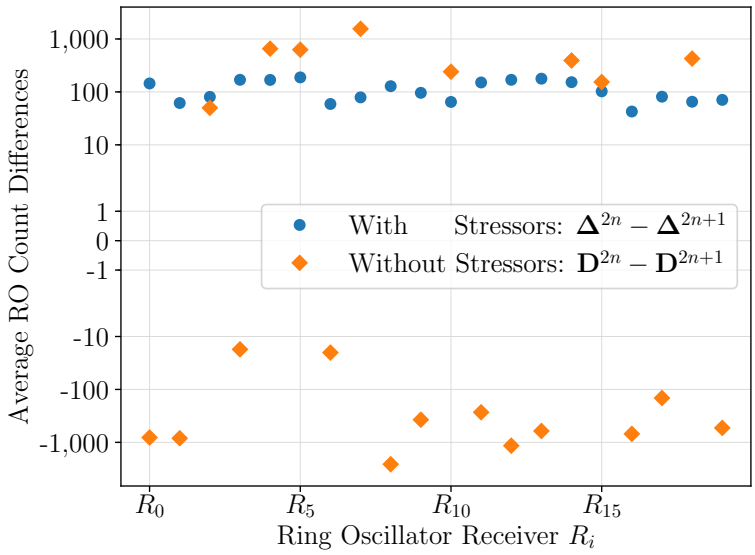


Figure 7.7: All Ring Oscillator (RO) count differences with stressors $\mathbf{\Delta}^{2n} - \mathbf{\Delta}^{2n+1}$ (blue circles) are positive, correctly decoding a transmission of 1. However, the naive metric without stressors $\mathbf{D}^{2n} - \mathbf{D}^{2n+1}$ (orange diamonds) behaves randomly, with only about half positive values.

Figure 7.7 demonstrates the need for this more complicated procedure in practice for a transmission of a Manchester-encoded 1-bit. Specifically, it compares the new metric with stressor ROs $\mathbf{\Delta}^{2n} - \mathbf{\Delta}^{2n+1}$ against the naive bit-recovery metric $\mathbf{D}^{2n} - \mathbf{D}^{2n+1}$ for all twenty receiver ROs. As Figure 7.7 (blue circles) shows, $\mathbf{\Delta}^{2n} - \mathbf{\Delta}^{2n+1} > 0$ for all receiver ROs $R_0, R_1, \ldots$, so the novel metric correctly recovers this bit transmission. However, the $\mathbf{D}^{2n} - \mathbf{D}^{2n+1}$ values with stressors disabled (orange diamonds) behave randomly. Indeed, in the experiment from which these measurements originated (between the two Kintex 7 boards, with $N_S = 500$, $N_T = 1{,}000$, and measurement periods of 10 ms), the $\mathbf{\Delta}$ metric successfully recovered over 98% of transmissions, compared to 53% using the naive method without the stressors. Section 7.3.3 further expands on why the new technique makes for a good approach in detecting cross-board transmissions.

7.3.3 Explanation

This section tests the receiving circuit on its own to characterize its behavior. Figure 7.8 first plots the average metric Δ_V^i for the eight ring oscillators R_i of Figure 7.5 across the same power supply voltages $11.5\,\text{V} \leq V \leq 12.5\,\text{V}$, but enabling 1,000 stressor ROs on the board. As expected, for all ROs, $\Delta_{V_1}^i < \Delta_{V_2}^i$ whenever $V_1 > V_2$: when there is

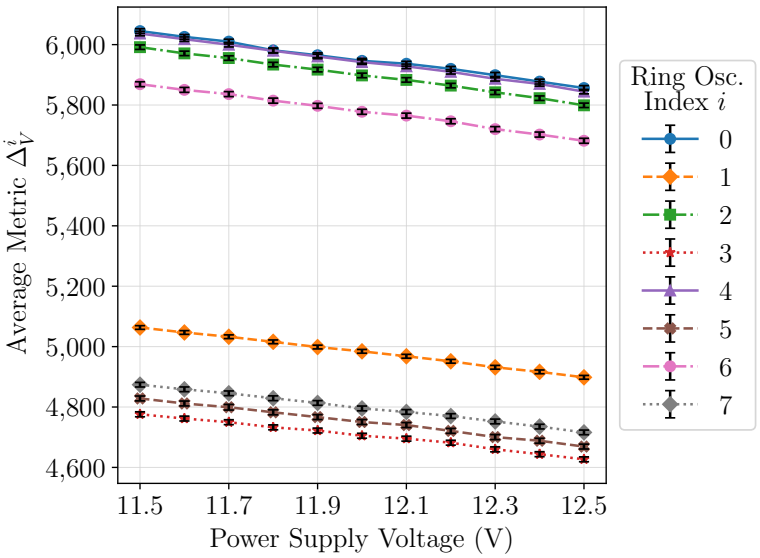


Figure 7.8: The average metric Δ_V^i (with 99% confidence intervals) on the AC701-1 sink decreases with higher power supply voltages V for all eight ring oscillators R_i.

an external voltage drop (e.g., when the source FPGA enables the transmitter ROs), the $\boldsymbol{\Delta}$ metric increases compared to when there are no external transmissions.

Given this monotonic relationship of $\boldsymbol{\Delta}$ as a function of voltage, the second aspect to investigate is how $\boldsymbol{\Delta}$ behaves across different measurement times of 2^t clock cycles and numbers of enabled stressors S. Specifically, Figure 7.9 calculates the average value of the $\boldsymbol{\Delta}$ metric over all 20 receiver ROs on an Artix 7 sink at two voltage levels (11.5 V and 12.5 V) and plots their differences, leading to several observations.

First of all, the average difference $\Delta = \Delta_{11.5} - \Delta_{12.5}$ is close to zero for time periods up to 41 µs, indicating that prolonged measurement times are necessary to distinguish between transmissions of zero and one, which in practice result in much smaller voltage drops of $\approx 20\,\text{mV}$. Moreover, until 2.6 ms, $\Delta > 0$ for all choices of how many stressors S to enable simultaneously, with fewer stressors resulting in a larger effect. However, for even larger time periods, $\Delta < 0$, with more stressors resulting in a bigger effect in magnitude. Consequently, the choice of how many stressors to enable and for how long is intricately linked with the accuracy of the covert channel. In addition, it helps explain why in some experimental setups (e.g., the KC705-1 receiver on PSU-B of Table 7.3), the recovered pattern is flipped, i.e., a 0-bit is identified as a 1-bit and vice-versa.

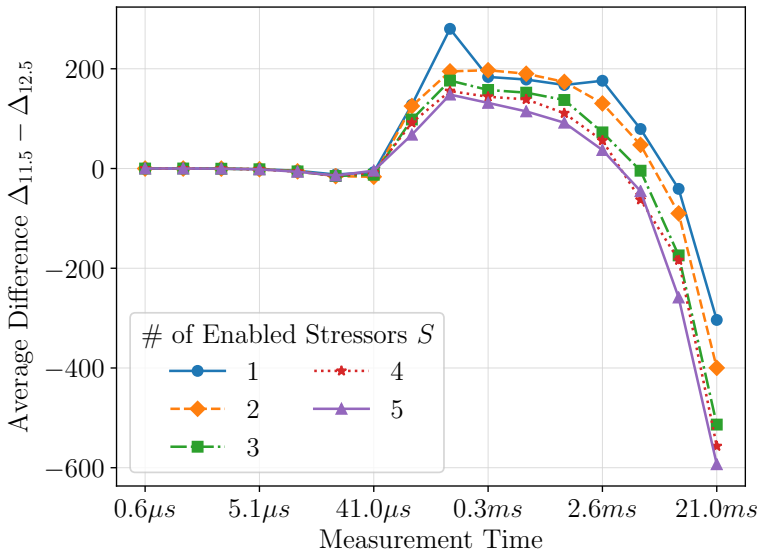


Figure 7.9: Difference between the average Δ metric as measured at 11.5 V and 12.5 V for various measurement times and numbers of stressors enabled on the AC701-1 sink.

7.4 Cross-FPGA Communication

This section explores FPGA-to-FPGA covert communication, first presenting a summary of results with default experimental parameters in Section 7.4.1. The numbers of source transmitter and sink stressor ROs are then respectively varied in Sections 7.4.2 and 7.4.3. Bandwidth-accuracy tradeoffs are investigated in Section 7.4.4, while the performance of the covert channel across different patterns, types of ROs, and measurement setups is evaluated in Section 7.4.5.

7.4.1 Overview of Results

This section provides an overview of the cross-FPGA results. The values for the default experimental parameters used in these experiments and the corresponding covert-channel bandwidths are summarized in Table 7.3. These values were chosen based on exploratory testing, as they represent a good tradeoff between accuracy and bandwidth. However, in some cases, better accuracy can be achieved at the cost of bandwidth, or the same accuracy can be maintained despite increasing the bandwidth (see Section 7.4.4).

The results of measurements across all 12 combinations of source and sink FPGAs on both PSUs are summarized in Table 7.4. As the table shows, covert communication

Parameter	Artix 7	Kintex 7	Section
# of Transmitter ROs, N_T	1,000	1,000	7.4.2
# of Enabled Transmitters	10	14	7.4.2
Transmitted Pattern	`0xf3ed1`	`0xf3ed1`	7.4.5
Transmitter Types	LUT-ROs	LUT-ROs	7.4.5
# of Stressor ROs, N_S	500	500	7.4.3
# of Enabled Stressors	1	5	7.4.4
Stressor & Receiver Types	LUT-ROs	LUT-ROs	7.4.5
# of Measurements, M	500	500	7.4.4
Measurement Cycles	2^{15}	2^{21}	7.4.4
Bandwidth b (bps)	6.1	0.1	7.4.4

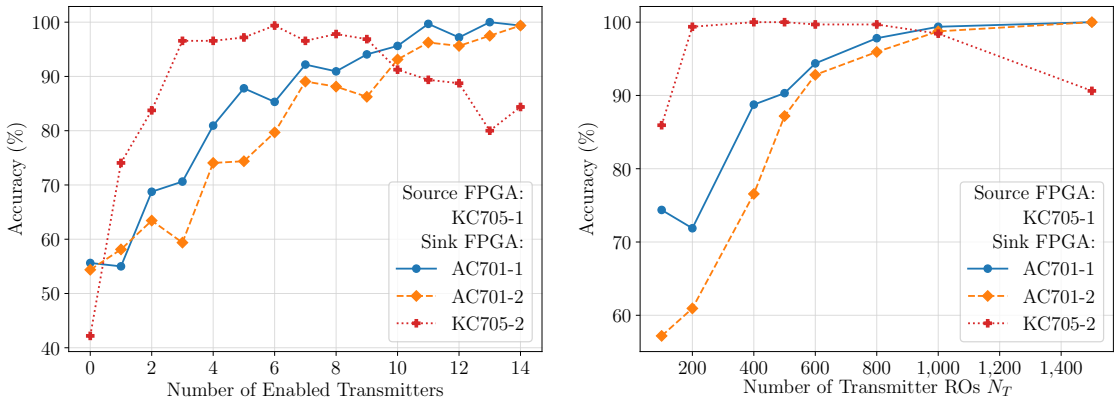
Table 7.3: Default values for accuracy- and bandwidth-related parameters, and the chapter sections in which they are varied. Bandwidth is calculated using Equation (7.1).

PSU	↓ T → R	AC701-1	AC701-2	KC705-1	KC705-2
A	**AC701-1**	-	79%	92%	100%
A	**AC701-2**	99%	-	93%	100%
A	**KC705-1**	100%	86%	-	100%
A	**KC705-2**	100%	98%	99%	-
B	**AC701-1**	-	100%	†98%	100%
B	**AC701-2**	100%	-	†99%	100%
B	**KC705-1**	100%	95%	-	100%
B	**KC705-2**	100%	100%	†98%	-

Table 7.4: Accuracy of the covert channels between transmitter (T) and receiver (R) Field-Programmable Gate Arrays (FPGAs) on the two Power Supply Units (PSUs) A and B, using the default experimental parameters. † signifies that the recovered bit-pattern is flipped.

is possible with high accuracy between any two boards, in either direction, and on both PSUs. The table also suggests that the behavior is not the same for otherwise-identical boards. This is likely due to both process variations internal to the FPGA chip (which affect RO measurements), and because of different component tolerances. As an example, the AC701-2 board is a worse sink than the AC701-1 board, while the KC705-1 board is a worse source than the KC705-2 board.

Moreover, the Kintex 7 boards are generally better sources than the Artix 7 boards due to the higher count of transmitters they contain ($T = 14$ as opposed to $T = 10$). As Section 7.4.2 shows, more transmitters tend to improve the quality of the covert channel.



(a) Transmitters **(b)** Transmitter ROs

Figure 7.10: Increasing the number of (a) simultaneously-enabled transmitters and (b) transmitter Ring Oscillators (ROs) on the KC705-1 source board generally increases the accuracy of the cross-board covert channel, except for the KC705-2 sink past a certain threshold.

Finally, although the information leakage remains strong with both PSUs, the accuracy of the recovered data on the 1,300 W PSU-B is usually higher than the accuracy on the 850 W PSU-A. Although one might expect the higher-rated PSU to produce more stable output under sudden changes in the load, this appears to not be the case.

7.4.2 Transmitter ROs

This section evaluates how changing the effective size of the transmitting circuit in the source FPGA impacts the accuracy of the covert channel. This is done in two ways. First, since each of the T transmitters (with N_T ROs each) can be controlled independently (Figure 7.2), Figure 7.10a varies the number of simultaneously-enabled transmitters on the KC705-1 source board, plotting the results across the three possible sink boards. Second, for the same board configuration, Figure 7.10b changes the number of transmitter ROs N_T, while enabling all T transmitters at the same time. Both experiments show that increasing the number of effective transmitter ROs $T \cdot N_T$ generally increases the accuracy of the covert channel. This is because the ensuing voltage drops are more pronounced, and can thus be more easily detected by the receiving boards. However, for the KC705-2 sink, too much activity on the transmitter can decrease the accuracy of the channel. This is because although the magnitude of the voltage drop increases in isolation (Figure 7.4), the stressor ROs are also causing a voltage drop that can overshadow that of the source FPGA.

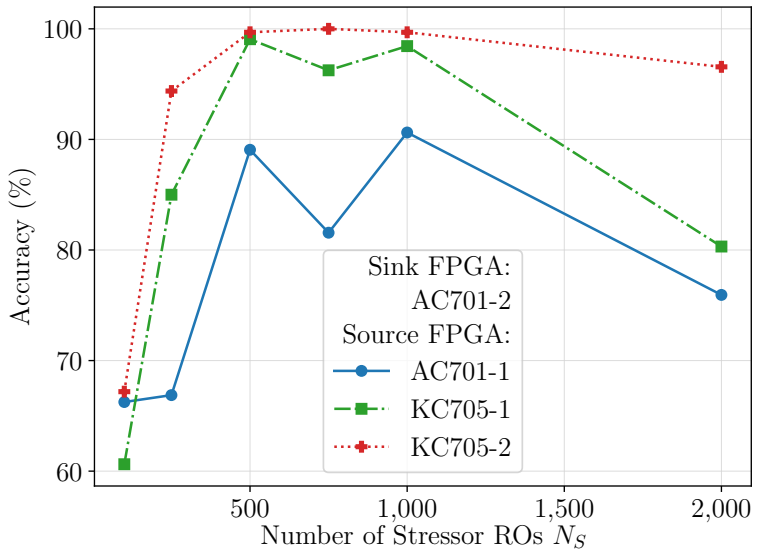


Figure 7.11: Increasing the number of stressor Ring Oscillators (ROs) N_S on the AC701-2 sink board can decrease accuracy, as the additional activity can overshadow external transmissions.

7.4.3 Stressor ROs

Although the architectural design fixes the number of stressors at $S = 5$ (Table 7.1), this section evaluates the effect of changing the number of stressor ROs N_S on the sink AC701-2 board. The accuracy of the covert channel across these experiments is plotted in Figure 7.11. Consistently with Figure 7.9, although stressor ROs are necessary to detect covert transmissions, further increasing N_S can have the opposite effect: the voltage drop caused by the stressors overpowers any effect caused by the source transmissions, and starts pushing the average difference from positive to negative.

7.4.4 Bandwidth-Accuracy Tradeoffs

This section investigates bandwidth-accuracy tradeoffs by first varying the number of measurements M over which the RO counts are averaged. It tests the AC701-1 and AC701-2 boards as sinks, and plots the results from all other possible FPGA sources in Figure 7.12. In general, increasing the number of measurements increases the accuracy of the covert channel, but at a cost of lower bandwidth. $M = 500$ represents a good tradeoff between accuracy and bandwidth (over 90% accuracy at 6.1 bps for the Artix 7 boards), but $M \geq 1000$ results in higher accuracy at half the bandwidth.

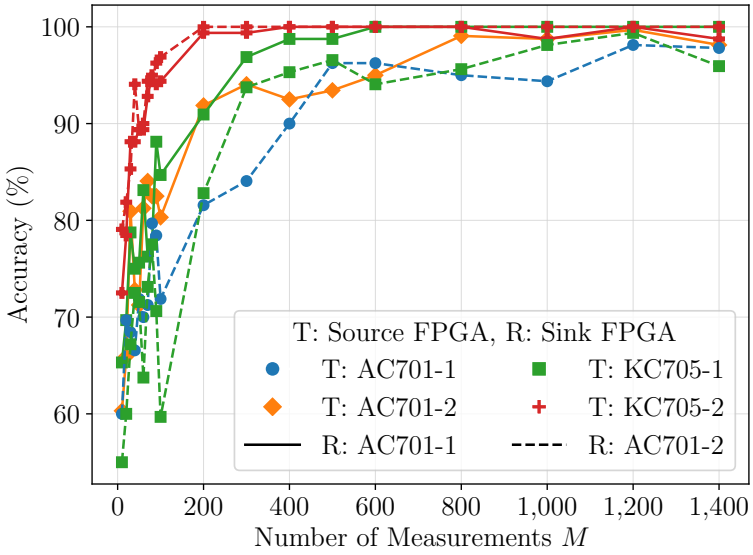
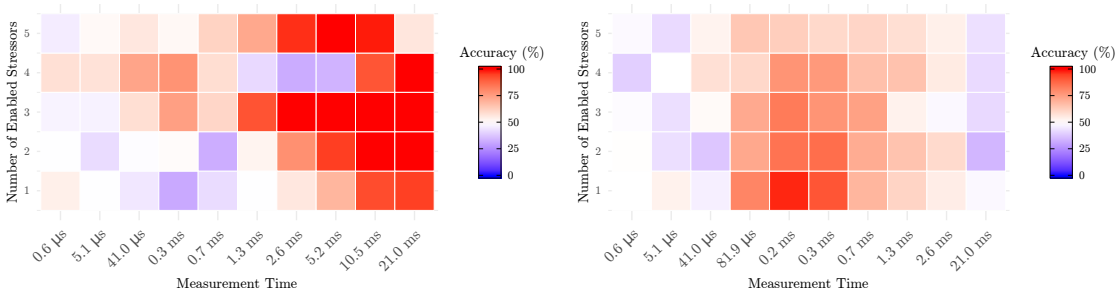


Figure 7.12: Increasing the number of measurements M improves accuracy to any AC701 sink R, from any other Artix 7 and Kintex 7 source T.



(a) Kintex 7 **(b)** Artix 7

Figure 7.13: Accuracy for different measurement times and numbers of enabled stressors on (a) the KC705-1 and (b) the AC701-1 sinks.

The second set of experiments varies both the number of clock cycles 2^t for which each RO is counting, and the number of enabled stressors on the sink FPGA, using the AC701-2 board as the covert-channel source. The results for the KC705-1 and AC701-1 sinks are shown in Figures 7.13a and 7.13b respectively and indicate that the parameters for the receivers need to be carefully tuned for different types of boards. For example, the Artix 7 board necessitates that fewer stressors be driven, which is consistent with the results of Sections 7.3.3 and 7.4.3. On the other hand, the KC705-1 sink remains accurate across a wider range of enabled stressors, but requires longer measurement periods for acceptably low error rates.

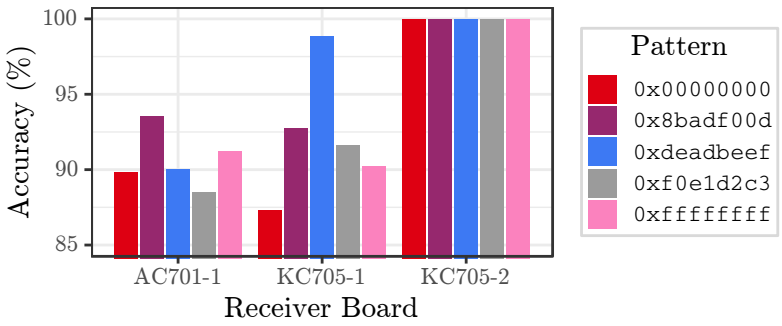


Figure 7.14: The accuracy of the covert channel with the AC701-2 source remains similar across the transmission of five different 32-bit patterns.

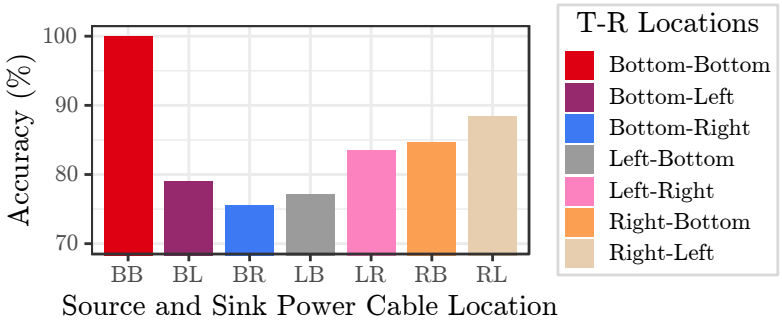


Figure 7.15: The accuracy of communication between the transmitter (T) and the receiver (R) Kintex 7 boards depends on how they are connected to the Power Supply Unit (PSU).

7.4.5 Other Parameters

This section finally tests the accuracy of the covert channel when varying the patterns transmitted, changing the cable layout, and using different types of ROs.

Transmitted Patterns: Although most experiments in the previous sections transmitted a 20-bit number (`0xf3ed1`), five additional 32-bit patterns (64 encoded bits) were tested from the AC701-2 board to the other three sink FPGAs. Figure 7.14 indicates that the covert channel remains similarly accurate for these longer patterns with different Hamming Weights (HWs) and runs of zeros and ones: the properties of the transmitted values cannot fundamentally alter the accuracy of the covert channel.

Measurement Layouts: In the majority of the previous experiments, the source and sink FPGA boards were connected to the same PSU output through a Corsair peripheral cable with four Molex connectors. This cable was attached to one of the "bottom" 6-pin outputs of the PSU. However, other types of connections are possible, including a 12-pin output of the PSU splitting into two 6-pin PCIe cables, denoted by "left"

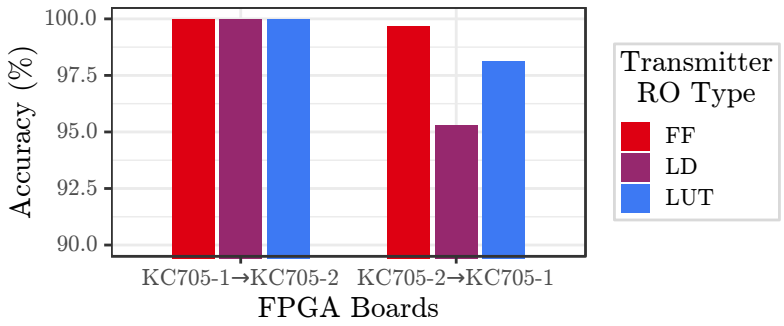


Figure 7.16: The accuracy between the two Kintex 7 boards is consistently high for all types of transmitter (source) Ring Oscillators (ROs) tested.

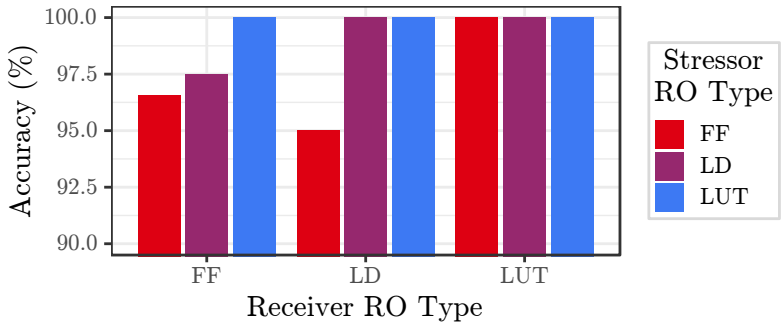


Figure 7.17: The accuracy of transmissions from the KC705-1 source to the KC705-2 sink using different types of receiver and stressor Ring Oscillators (ROs) remains similarly high.

and "right". This set of experiments therefore verifies that communication from the KC705-1 board to the KC705-2 board remains possible across these alternative cable setups. $M = 1{,}000$ measurements are taken of ROs counting for 2^{21} clock cycles, with all five stressors enabled. The results are summarized in Figure 7.15, which demonstrates that the information leakage persists in all setups tested. This is perhaps to be expected, since the PSU uses a "dedicated single +12 V rail" [69], but the results further indicate that there are differences among the various ports. Specifically, the covert channel is most accurate between FPGA boards on the same cable (as they are at exactly the same electric potential difference) and least accurate between the single location on the bottom of the PSU and either of the dual outputs. Finally, it should be noted that the recovered pattern is flipped in all setups, except when sharing the cable on the bottom output.

Ring Oscillator Types: The final set of tests for cross-FPGA communication uses the alternative ROs from Chapter 6 on the Kintex 7 boards. It also tests the channel in less ideal and more noisy environments. Specifically, both boards are connected to

Computer A over PCIe, and are enclosed in the computer tower to avoid isolating thermal effects. The setup otherwise uses the default experimental parameters of Table 7.3. Figure 7.16 first shows that for all three types of transmitter ROs, the accuracy of the cross-KC705 channel remains above 95%, despite potential noise introduced by thermal conditions and the shared PCIe buses. Similarly, Figure 7.17 shows that accuracy remains above 95% when using these alternative ROs for stressors and receivers on a KC705 sink. Although in many cases bits are again flipped, blocking combinatorial loops and introducing environmental noise cannot prevent the cross-FPGA channel from operating.

7.5 Additional Covert Channels

This section explores CPU-to-FPGA and GPU-to-FPGA covert channels in Sections 7.5.1 and 7.5.2 respectively.

7.5.1 CPU Transmissions

The CPU-to-FPGA communication channel replaces the power draw of the FPGA source with heavy CPU loads. This is done through the open-source `stress` program, which is available on Debian-based Linux distribution package managers [350]. The number of threads that `stress` uses is varied from 0 (i.e., no transmissions, corresponding to random measurements), up to the number of threads available on each computer, i.e., 24 on the CPU attached to PSU-A, and 4 on the CPU attached to PSU-B.

The measurement process and classification metric remain the same as for the cross-FPGA channels, except for an additional delay of three seconds after the `stress` program has started to ensure full utilization of the cores. Three seconds are also added after killing the process to guarantee that the usage has returned to normal. Moreover, when testing with PSU-A, and to increase accuracy, the measurement period for the KC705 receivers is reduced to $2^t = 2^{18}$ clock cycles (1.3 ms) from 2^{21} (10 ms), while the number of stressors is lowered to 4 instead of 5 (the default parameters are used on PSU-B, but with M = 1,200 measurements for the AC701 boards). This increases the bandwidth of the covert channel by a factor of 8× to 0.8 bps compared to the cross-FPGA channel.

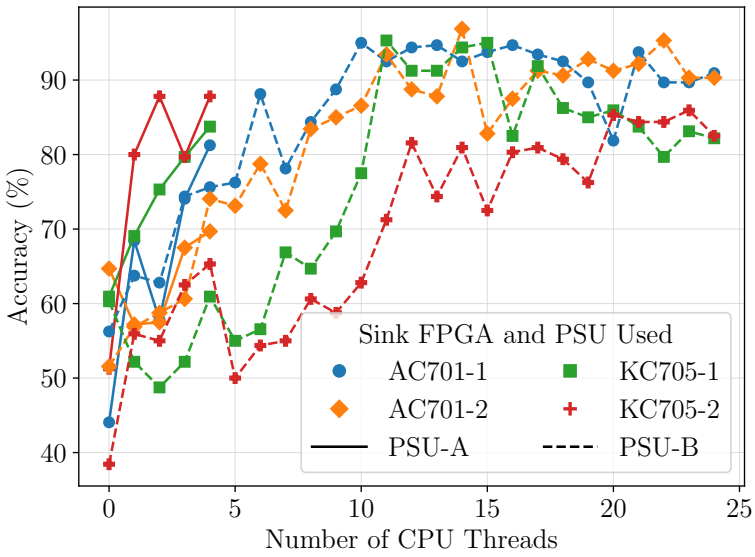


Figure 7.18: Accuracy of transmissions from a Central Processing Unit (CPU) to the four Field-Programmable Gate Arrays (FPGAs) on both Power Supply Units (PSUs) for different numbers of CPU threads. As PSU-A powers a 4-core CPU, no more than 4 threads can be dispatched.

The results for the two PSUs are shown in Figure 7.18, which suggests three main conclusions. First of all, there is a critical CPU activity threshold which is necessary to make the covert channel possible. On PSU-A, this requires about 4 threads for the AC701 boards, and 7 threads for the KC705 boards. Moreover, increasing the number of threads does not always make the covert channel more accurate. For example, increasing the number of CPU threads from 0 to 10 increases accuracy, but the accuracy generally plateaus between 10 and 17 CPU threads, and then decreases, perhaps due to hyper-threading. Finally, for a similar number of threads used, the accuracy on PSU-B is often higher compared to that for PSU-A. This parallels the cross-FPGA results of Section 7.4, and indicates that PSU-B is generally more prone to covert communication. The maximum accuracy achieved, the number of CPU threads used, and other experimental parameters are summarized in Table 7.5.

7.5.2 GPU Transmissions

The process for testing GPU-to-FPGA transmissions is similar to that of CPU-to-FPGA transmissions. The GPUs are stressed with the open-source `gpu_burn` [333] program, which uses Nvidia's CUDA platform to fully utilize the GPU cores. As the

PSU	Parameter	AC701-1	AC701-2	KC705-1	KC705-2
A	Accuracy	95%	97%	95%	86%
A	Bandwidth (bps)	6.1	6.1	0.8	0.8
B	Accuracy	81%	70%	†84%	88%
B	Bandwidth (bps)	2.5	2.5	0.1	0.1
A	# of Threads	10	14	11	23
A	# of Enabled Stressors	1	1	4	4
A	# of Measurements	500	500	500	500
A	Measurement Cycles	2^{15}	2^{15}	2^{18}	2^{18}
B	# of Threads	4	4	4	4
B	# of Enabled Stressors	1	1	5	5
B	# of Measurements	1,200	1,200	500	500
B	Measurement Cycles	2^{15}	2^{15}	2^{21}	2^{21}

Table 7.5: Maximum accuracy of transmissions from a Central Processing Unit (CPU) source to the four Field-Programmable Gate Arrays (FPGAs) on the two Power Supply Units (PSUs), along with the parameters for which it is achieved. † signifies that the recovered bit-pattern is flipped.

Property	GPU-A	GPU-B
Architecture	Fermi	Kepler
Node Size (nm)	40	28
Driver Version	390.87	418.67
CUDA Version	8.0	10.1
Compiler Flag	`compute_20`	`compute_50`

Table 7.6: Properties of Graphics Processing Unit (GPU) testing with `gpu_burn`.

two GPUs use different architectures, `gpu_burn` is compiled and run against different Nvidia drivers and CUDA versions. These differences are summarized in Table 7.6. Moreover, experiments return to the default measurement period of $2^t = 2^{21}$ cycles for the Kintex 7 boards, but the number of measurements for all boards is increased to $M = 1{,}500$, reducing bandwidth by a factor of 3×. These parameters and the corresponding results are summarized in Table 7.7. As in the CPU case, three seconds of delay are added after starting or terminating the `gpu_burn` program to allow usage to return to normal levels.

Figure 7.19 plots the results of these experiments for the four boards on both GPUs, showing that it is always possible to create a communication channel in all eight setups. As expected, since there are fewer GPU cores attached to PSU-A, the covert channel is weaker, but the accuracy is over 95% for three boards when using the GPU attached to

PSU	Parameter	AC701-1	AC701-2	KC705-1	KC705-2
A	Accuracy	76%	70%	94%	89%
B	Accuracy	97%	87%	96%	†100%
A&B	Bandwidth (bps)	2.0	2.0	0.03	0.03
A&B	# of Enabled Stressors	1	1	5	5
A&B	# of Measurements	1,500	1,500	1,500	1,500
A&B	Measurement Cycles	2^{15}	2^{15}	2^{21}	2^{21}

Table 7.7: Maximum accuracy of transmissions from a Graphics Processing Unit (GPU) source to the four Field-Programmable Gate Arrays (FPGAs) on the two Power Supply Units (PSUs), along with the parameters for which it is achieved. † signifies that the recovered bit-pattern is flipped.

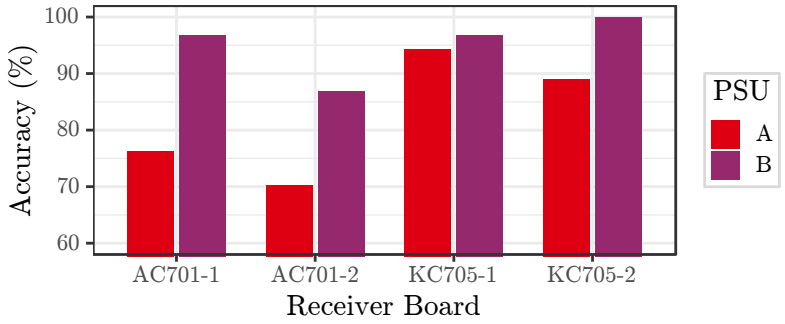


Figure 7.19: Accuracy of Graphics Processing Unit (GPU) transmissions to the four FPGA sink boards using the two Power Supply Units (PSUs).

PSU-B, which is larger. Moreover, the AC701 boards are worse sinks than the KC705 boards. Although this pattern is not entirely identical across the FPGA-to-FPGA, CPU-to-FPGA, and GPU-to-FPGA channels, it broadly remains consistent.

7.6 Discussion

This section discusses how practical the cross-board covert channels are (Section 7.6.1), and proposes countermeasures to mitigate their impact (Section 7.6.2).

7.6.1 Practicality of Attacks

Two aspects of how practical the communication scheme is are evaluated in this section. The first is how costly transmissions are in terms of resources used on the FPGA boards. The amount of logic instantiated is moderate, but not negligible. On the transmitting end, $G \cdot T \cdot N_T$ LUTs are used, where $G = 4$ is the number of ring oscillator

Algorithm	Key Size	AC701	KC705
AES	256	0.7 min	44.7 min
ECDSA	521	1.4 min	91.1 min
RSA	1,024	2.8 min	179.0 min

Table 7.8: Time to leak cryptographic keys of different sizes to the Artix 7 and Kintex 7 boards.

stages. In particular, the source design (including the UART interface and other logic) utilizes 16.6% of LUT resources on the Artix 7 FPGA chip. Similarly, the sink design uses $G \cdot (R \cdot N_R + S \cdot N_S)$ LUTs for the receiver and stressor ROs, and $L \cdot R \cdot N_R$ registers for counting, where $L = 32$ is the length of the counters. Only 7.8% of the Artix 7 resources are used in this case—a number which can be reduced to 3.4%, as the AC701 boards only enable one stressor for higher accuracy.

The second aspect is the channel capacity, which was shown to be up to 6.1 bps. This is much higher than the capacity of cross-device thermal attacks, which can transmit under 15 bits in an hour [129, 332], but lower than cross-CPU thermal attacks [31, 204], which have a capacity of up to 300 bps. By contrast, power attacks within CPUs can transfer between 20 and 120 bits per second [7, 157], and have a theoretical capacity of up to 2 Mbps [213]. These types of attacks are related to Dynamic Voltage and Frequency Scaling (DVFS) on modern processors, which regulates the voltage and frequency of CPUs in accordance with usage demands, and can be further exploited to cause faults in computations [319]. The higher capacity of thermal and power channels within Integrated Circuits (ICs) is to be expected due to the shared PDN and physical proximity: as Chapter 6 showed, the covert channel between Super Logic Regions (SLRs) of the same FPGA chip has a bandwidth of up to 4.1 Mbps.

Although the Kintex 7 boards were shown to be better sinks (often with 0% error rate), the Artix 7 boards were faster by a factor of 7.6× (6.1 bps vs 0.8 bps). This difference is significant in practice: Table 7.8 shows how long it would take to transmit keys for different popular cryptographic algorithms. Even assuming that the channel is not noisy, it would take almost 45 minutes to transfer a 256-bit AES key to a KC705 board and three hours to transfer a 1024-bit RSA key. However, the AC701 board would need less than three minutes to transfer the same RSA key, despite the potential drop in accuracy.

To increase accuracy, one can either tweak the parameters of the source and sink FPGA designs (including the number of measurements M over which the RO counts are averaged), or instead change the communication scheme itself. For example, a 3-repetition code decreases bandwidth by a factor of 3, but also lowers the error rate e to $3e^2 - 2e^3$: a 10% error rate is reduced to under 3%. The channel capacity is $1 - H(e) = 1 + e\log_2 e + (1-e)\log_2(1-e)$, and for smaller bit-flip probabilities, other error correcting codes such as Hamming and Golay codes can be used to improve accuracy.

7.6.2 Defense Mechanisms

This section discusses potential software and hardware defenses against voltage-based covert-channel attacks. To start with, some countermeasures might revolve around preventing intentional transmissions from the covert-channel source. However, doing so would be hard without huge sacrifices in terms of power and performance. Although this chapter used ring oscillators to cause fluctuations in the voltage of FPGAs sharing the same PSU, other switching activity can also result in voltage over- and under-shoots. For example, prior work has shown that switching large sets of programmable interconnect points [390] or flip-flops [119, 120] can cause voltage fluctuations outside of the allowed operating voltage range for an FPGA device. Moreover, the CPU-to-FPGA and GPU-to-FPGA channels demonstrate that the problem is not FPGA-specific, but can be found in other types of activities which result in large power draws. Consequently, unless power is equalized among all possible devices and algorithm implementations, some leakage which can differentiate between levels of activity will persist.

To prevent side-channel attacks from being possible, designers may remove the power-draw or voltage-level dependence on the data being processed and increase the noise level. Although several masking and hiding techniques have been proposed, leakage on FPGAs persists due to variations in placement and routing [73]. Consequently, a better approach would be to prevent the leakage from being measurable on the FPGA sinks.

Although some cloud providers prevent LUT-ROs from being instantiated in their infrastructures [12], novel receiver designs (e.g., the ROs of Chapter 6 or TDCs [282, 283]) can bypass them: designing effective countermeasures against side- and covert-channel

receivers is an arms race. Hence, defense-in-depth dictates the need for run-time solutions in addition to any preventive approaches. One feature of the covert channel demonstrated is the high switching activity on the receiver. As a result, built-in voltage monitors could be used by cloud providers to detect abnormal fluctuations—with the caveat that legitimate circuits may also cause similar patterns, and that, at least on the AC701 boards, the number of enabled stressor ROs was small ($N_S = 500$). In fact, proposals to "detect the insertion of power measurement circuits onto a device's power rail" [183] are similar, though doing so without false positives remains a challenge.

Finally, better hardware (at a higher cost) can also help hide the useful signal under the noise floor. For example, independent, fully separate power supplies for different boards would require that the leakage be detectable even over the Alternating Current (AC) power line and through two different rectifiers to Direct Current (DC). Alternatively, improved voltage regulators with better transient responses, better isolation of power circuits within the same PSU, as well as more filters and smoothing capacitors can also reduce the signal available to an attacker.

7.7 Summary

This chapter introduced covert channels between distinct, single-tenant FPGA boards that are merely powered by the same PSU. This strong threat model (Section 7.1) required a novel measurement setup (Section 7.2) and classification metric (Section 7.3), both of which depend on stressor ROs in the sink FPGA to strain its voltage regulator. The FPGA-to-FPGA information leakage was present in off-the-shelf Artix 7 and Kintex 7 boards in either direction of communication, across various architectural choices, and often with 100% accuracy (Section 7.4). Moreover, this chapter showed that high loads of computer activity can be used to create CPU-to-FPGA and GPU-to-FPGA covert channels, with similarly high transmission accuracies (Section 7.5). Despite some possible mitigating factors (Section 7.6), the presence of leakage even in dedicated FPGAs further highlights the need for better architectural designs not only of the FPGA chips themselves, but also of the boards on which they are deployed.

The faucet leak, and learn to leave them so.

— Marya Mannes (Sec)

8 Conclusion

Contents

This thesis demonstrated the effects of unintentional hardware properties on the integrity and confidentiality of information processed by embedded devices. It specifically focused on remote attacks without physical access to the device in question, and made several contributions to the state-of-the-art in out-of-band signal injections and Field-Programmable Gate Array (FPGA) covert communication with on-chip receivers (Section 8.1). In summary, this thesis has shown that in order to make future systems more secure, hardware limitations need to be acknowledged and integrated into the threat and system model of embedded devices (Section 8.2).

8.1 Summary of Contributions

This thesis started by identifying gaps in the literature of out-of-band signal injection attacks. These injections can remotely impact the integrity of sensor measurements processed by embedded devices. Specifically, Chapter 3 contained a comprehensive survey of related work, portraying a chronological and thematic evolution of out-of-

band signal injections. It created a taxonomy of the terminology used, the sources of vulnerability, as well as proposed countermeasures, and placed out-of-band research in the context of side-channel leakage and electromagnetic interference attacks. In doing so, it highlighted open research problems and identified several insights into how to overcome these challenges in the future.

Chapter 4 addressed one of these challenges, namely the lack of a unifying framework through which to evaluate the effects of out-of-band signal injection attacks. The proposed framework included a system model and associated mathematical definitions for security, which addressed effects ranging from mere disruptions of sensor readings to precise waveform injections of attacker-chosen values. The framework was evaluated in practice through an algorithm which calculated the "security level" of a real, off-the-shelf system. Chapter 4 further investigated the unintentional demodulation properties of Analog-to-Digital Converters (ADCs), which are crucial in out-of-band attacks. Finally, it also showed that the proposed model could be used to inform circuit design choices and evaluate defense mechanisms in its context.

Having systematically mapped the space of integrity attacks on embedded devices, this thesis then evaluated whether on-chip circuits can measure side-channel leakage. Chapter 5, in particular, demonstrated that "long" wires in FPGA chips leak information about their state in a way which can be measured using Ring Oscillators (ROs). This long-wire leakage was characterized across six families of Xilinx FPGAs, and the effect was shown to be independent of the device used, the location and orientation of the transmitter and receiver, and the pattern of transmission. Chapter 5 further exploited this leakage for fast covert- and side-channel attacks with high accuracy, even in the presence of dynamic activity and simultaneous transmissions.

Chapter 6 then turned its attention to multi-tenant attacks on cloud FPGAs. To bypass countermeasures currently-deployed by cloud providers, novel RO structures had to be developed. These ROs were shown to be capable of estimating femtosecond-scale changes in the delays of long wires on a seventh family of Xilinx FPGAs. Chapter 6 further investigated whether physical isolation of user circuits onto separate dies, called Super Logic Regions (SLRs), of the same FPGA could act as a potential countermeasure.

However, it showed that cross-SLR covert-channel attacks were possible on two cloud providers across multiple sizes, locations, and types of the receivers and the transmitters. In other words, Chapter 6 highlighted that, despite some mitigating factors, current FPGA architectures are inherently unsuited for multi-tenant cloud setups.

Finally, Chapter 7 showed for the first time that it is possible to implement remote cross-device attacks in dedicated, single-tenant FPGAs. Specifically, it demonstrated that voltage fluctuations in Power Supply Units (PSUs) leak information that can be exploited by FPGA, Central Processing Unit (CPU), and Graphics Processing Unit (GPU) transmitters for cross-board covert channels, even for unprivileged FPGA sink designs without access to voltage or temperature system monitors. This was accomplished through a novel classification metric and measurement setup, which stressed the receiver FPGA in order to estimate modulated activity on the transmitter.

Overall, this thesis, in its systematic approach towards out-of-band signal injection attacks, highlighted previously-unexplored connections, experimental gaps, and challenges for future research, addressing some of them through a novel framework to quantify security in the presence of imperfect hardware and integrity attacks on the sensor inputs to embedded devices. This thesis further demonstrated how to remotely break the confidentiality of data processed by FPGA devices through three new sources of information leakage that can be used for covert- and side-channel attacks in local and cloud environments. Although better software can alleviate some of these issues, fundamental improvements at the hardware layer are necessary for secure embedded devices in the future.

8.2 Future Outlook & Parting Thoughts

Out-of-band signal injections on sensors and remote attacks on FPGAs are nascent research areas. Yet, due to the increasing relevance of embedded systems in people's daily lives, they have attracted the interest of several academic communities, and are starting to become acknowledged even by industrial and governmental entities. Indeed, as adversaries do not require physical access to the devices they target, these remote

attacks pose new challenges for embedded systems security, and put the safety of the people relying on embedded devices in danger.

In investigating remote attacks on both sensors and FPGAs, this thesis demonstrated that unintentional hardware properties break the confidentiality and integrity of embedded systems. Perhaps even more importantly, it identified a fundamental need to rethink assumptions and threat models, and improve experimental methodologies and reporting procedures. It further showed that solutions to these problems require interdisciplinary collaboration between security, circuit design, and manufacturing experts, among others.

Thankfully, it is still early enough to act. With the move towards multi-die chips and shrinking node sizes, the multi-tenant threat model is becoming more accepted by the security and FPGA communities, and may soon become a practicable reality. But until then, academics and manufacturers need to come together to pinpoint the causes of existing imperfections that remain elusive without details of the internal hardware design of the chips.

In addition, although existing proof-of-concept attacks are good ways to demonstrate novel sources of vulnerability, how to turn them into robust, undetectable integrity or confidentiality attacks has not yet been extensively explored. Using dedicated experimental facilities for long-range out-of-band signal injection attacks will thus be key in evaluating their unintentional effects (e.g., upsets in digital equipment for electromagnetic attacks or audible byproducts for acoustic ones). Moreover, additional encoding schemes, regression techniques, triggering mechanisms, and logic-hiding approaches could be investigated in the future when "weaponizing" attacks for FPGAs, along with alternative applications of the leakage, e.g., in Intellectual Property (IP) core watermarking.

Overall, this thesis highlighted the need to peek at the "layer below" in order to achieve a more secure future, and took a first step towards that future by exposing its damaging, leaky abstractions.

I guess there are never enough books.

— John Ernst Steinbeck Jr.

Bibliography

[1] C. Acar and A. Shkel. *MEMS Vibratory Gyroscopes: Structural Approaches to Improve Robustness*. Springer, 2nd edition, 2009.

[2] R. Adler. A study of locking phenomena in oscillators. *Proceedings of the IRE (JRPROC)*, 34(6):351–357, Jun 1946.

[3] A. Agne, H. Hangmann, M. Happe, M. Platzner, and C. Plessl. Seven recipes for setting your FPGA on fire—A cookbook on heat generators. *Microprocessors and Microsystems*, 38(8):911–919, Nov 2014.

[4] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM side-channel(s). In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2002.

[5] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan detection using IC fingerprinting. In *IEEE Symposium on Security and Privacy (S&P)*, 2007.

[6] W. K. Al-Assadi and S. Kakarla. A BIST technique for crosstalk noise detection in FPGAs. In *IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems (DFT)*, 2008.

[7] M. Alagappan, J. Rajendran, M. Doroslovački, and G. Venkataramani. DFS covert channels on multi-core platforms. In *IEEE/IFIP International Conference on VLSI and System-on-Chip (VLSI-SoC)*, 2017.

[8] M. M. Alam, S. Tajik, F. Ganji, M. Tehranipoor, and D. Forte. RAM-Jam: Remote temperature and voltage fault attack on FPGAs using memory collisions. In *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2019.

[9] Alibaba Cloud. Compute optimized and FPGA-equipped instance type families. `https://www.alibabacloud.com/help/doc-detail/108504.htm`. Created: 2019-04-09. Accessed: 2019-10-01.

[10] Amazon Web Services. AFI power. `https://github.com/aws/aws-fpga/blob/master/hdk/docs/afi%5Fpower.md`. Created: 2019-01-25. Accessed: 2019-10-01.

[11] Amazon Web Services. Amazon EC2 F1 instance partners. `https://aws.amazon.com/ec2/instance-types/f1/partners/`. Created: Unknown. Accessed: 2019-10-01.

[12] Amazon Web Services. AWS EC2 FPGA HDK+SDK errata. `https://github.com/aws/aws-fpga/blob/master/ERRATA.md`. Created: 2019-09-20. Accessed: 2019-10-01.

[13] Amazon Web Services. Developer preview—-EC2 instances (F1) with programmable hardware. `https://aws.amazon.com/blogs/aws/developer-preview-ec2-instances-f1-with-programmable-hardware/`. Created: 2016-11-30. Accessed: 2019-10-01.

[14] N. A. Anagnostopoulos, T. Arul, Y. Fan, C. Hatzfeld, A. Schaller, W. Xiong, M. Jain, M. U. Saleem, J. Lotichius, S. Gabmeyer, J. Szefer, and S. Katzenbeisser. Intrinsic run-time Row Hammer PUFs: Leveraging the Row Hammer effect for run-time cryptography and improved security. *Cryptography*, 2(3):1–45, Jun 2018.

[15] S. A. Anand and N. Saxena. Speechless: Analyzing the threat to speech privacy from smartphone motion sensors. In *IEEE Symposium on Security and Privacy (S&P)*, 2018.

[16] S. A. Anand, C. Wang, J. Liu, N. Saxena, and Y. Chen. Spearphone: A speech privacy exploit via accelerometer-sensed reverberations from smartphone loudspeakers. arXiv 1907.05972, 2019. http://arxiv.org/abs/1907.05972.

[17] AnandTech. Apple's MacBook Pro with 128GB SSD: Performance and battery life investigated. https://www.anandtech.com/show/2504/2. Created: 2008-04-15. Accessed: 2019-10-01.

[18] AnandTech. Intel shows Xeon scalable gold 6138P with integrated FPGA, shipping to vendors. https://www.anandtech.com/show/12773/intel-shows-xeon-scalable-gold-6138p-with-integrated-fpga-shipping-to-vendors. Created: 2018-05-17. Accessed: 2019-10-01.

[19] J. H. Anderson and F. N. Najm. Interconnect capacitance estimation for FPGAs. In *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2004.

[20] A. Antonopoulos, C. Kapatsori, and Y. Makris. Security and trust in the analog/mixed-signal/RF domain: A survey and a perspective. In *IEEE European Test Symposium (ETS)*, 2017.

[21] A. Antonopoulos, C. Kapatsori, and Y. Makris. Trusted analog/mixed- signal/RF ICs: A survey and a perspective. *IEEE Design & Test (D&T)*, 34(6):63–76, Dec 2017.

[22] M. N. Armenise, C. Ciminelli, F. Dell'Olio, and V. M. N. Passaro. *Advances in Gyroscope Technologies*. Springer, 1st edition, 2011.

[23] Ars Technica. Sounds bad: Researchers demonstrate "sonic gun" threat against smart devices. https://arstechnica.com/gadgets/2017/07/sounds-bad-researchers-demonstrate-sonic-gun-threat-against-smart-devices. Created: 2017-07-28. Accessed: 2019-10-01.

[24] A. Ayed, T. Dubois, J.-L. Levant, and G. Duchamp. Failure mechanism study and immunity modeling of an embedded analog-to-digital converter based on immunity measurements. *Microelectronics Reliability*, 55(10):2067–2071, Sep 2015.

[25] A. Ayed, T. Dubois, J.-L. Levant, and G. Duchamp. Immunity measurement and modeling of an ADC embedded in a microcontroller using RFIP technique. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 57(5):955–962, Oct 2015.

[26] M. Backes, M. Dürmuth, S. Gerling, M. Pinkal, and C. Sporleder. Acoustic side-channel attacks on printers. In *USENIX Security Symposium*, 2010.

[27] Baidu Cloud. FPGA cloud compute. https://cloud.baidu.com/product/fpga.html. Created: Unknown. Accessed: 2019-10-01.

[28] J. Balasch, B. Gierlichs, and I. M. R. Verbauwhede. Electromagnetic circuit fingerprints for Hardware Trojan detection. In *IEEE International Symposium on Electromagnetic Compatibility (EMC)*, 2015.

[29] V. Barbaro, P. Bartolini, G. Calcagnini, F. Censi, B. Beard, P. S. Ruggera, and D. M. Witters. On the mechanisms of interference between mobile phones and pacemakers: Parasitic demodulation of GSM signal by the sensing amplifier. *Physics in Medicine & Biology*, 48(11):1661–1671, Jun 2003.

[30] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache. Fault injection attacks on cryptographic devices: Theory, practice and countermeasures. *Proceedings of the IEEE (JPROC)*, 100(11):3056–3076, Nov 2012.

[31] D. B. Bartolini, P. Miedl, and L. Thiele. On the capacity of thermal covert channels in multicores. In *European Conference on Computer Systems (EuroSys)*, 2016.

[32] C. E. Baum, T. K. Liu, and F. M. Tesche. On the analysis of general multiconductor transmission-line networks. *Interaction Note*, 350(6):467–547, Nov 1978.

[33] P. Bayon, L. Bossuet, A. Aubert, and V. Fischer. Fault model of electromagnetic attacks targeting ring oscillator-based true random number generators. *Journal of Cryptographic Engineering (JCEN)*, 6(1):61–74, Apr 2016.

[34] P. Bayon, L. Bossuet, A. Aubert, V. Fischer, F. Poucheret, B. Robisson, and P. Maurine. Contactless electromagnetic active attack on ring oscillator based true random number generator. In *International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE)*, 2012.

[35] G. T. Becker, M. Kasper, A. Moradi, and C. Paar. Side-channel based watermarks for integrated circuits. In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2010.

[36] J. Benesty, J. Chen, and Y. Huang. On the importance of the Pearson correlation coefficient in noise reduction. *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, 16(4):757–765, May 2008.

[37] J. Benford, J. A. Swegle, and E. Schamiloglu. *High Power Microwaves*. CRC Press, 3rd edition, 2016.

[38] K. Block, S. Narain, and G. Noubir. An autonomic and permissionless Android covert channel. In *ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, 2017.

[39] L. Blue, L. Vargas, and P. Traynor. Hello, is it me you're looking for? Differentiating between human and electronic speakers for voice interface security. In *ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, 2018.

[40] N. Bochard, F. Bernard, V. Fischer, and B. Valtchanov. True-randomness and pseudo-randomness in ring oscillator-based true random number generators. *International Journal of Reconfigurable Computing*, pages 1–13, Dec 2010.

[41] E. Boemo and S. López-Buedo. Thermal monitoring on FPGAs using ring-oscillators. In *International Workshop on Field-Programmable Logic and Applications (FPL)*, 1997.

[42] C. Bolton, S. Rampazzi, C. Li, A. Kwong, W. Xu, and K. Fu. Blue note: How intentional acoustic interference damages availability and integrity in hard disk drives and operating systems. In *IEEE Symposium on Security and Privacy (S&P)*, 2018.

[43] C. Bona and F. Fiori. A new filtering technique that makes power transistors immune to EMI. *IEEE Transactions on Power Electronics (TPEL)*, 26(10):2946–2955, Oct 2011.

[44] A. Boyer, S. Ben Dhia, and E. Sicard. Modelling of a direct power injection aggression on a 16 bit microcontroller input buffer. In *IEEE International Workshop on Electromagnetic Compatibility of Integrated Circuits (EMC Compo)*, 2007.

[45] A. Boyer, S. Ben Dhia, and E. Sicard. Modelling of a mixed signal processor susceptibility to near-field aggression. In *IEEE International Symposium on Electromagnetic Compatibility (EMC)*, 2007.

[46] A. Boyer, B. Vrignon, and M. Cavarroc. Modeling magnetic near-field injection at silicon die level. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 58(1):257–269, Feb 2016.

[47] F. Brauer, F. Sabath, and J. L. ter Haseborg. Susceptibility of IT network systems to interferences by HPEM. In *IEEE International Symposium on Electromagnetic Compatibility (EMC)*, 2009.

[48] British Broadcasting Corporation. Dewsbury driver who used speed camera jammer jailed. https://www.bbc.co.uk/news/uk-england-leeds-47202419. Created: 2019-02-11. Accessed: 2019-10-01.

[49] M. Bucci, L. Giancane, R. Luzzi, and A. Trifiletti. Three-phase dual-rail pre-charge logic. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2006.

[50] S. Buchovecká and J. Hlaváč. Frequency injection attack on a random number generator. In *IEEE International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, 2013.

[51] T. Buczkowski, D. Janusek, H. Zavala-Fernandez, M. Skrok, M. Kania, and A. Liebert. Influence of mobile phones on the quality of ECG signal acquired by medical devices. *Measurement Science Review*, 13(5):231–236, Nov 2013.

[52] S. Byma, J. G. Steffan, H. Bannazadeh, A. L. Garcia, and P. Chow. FPGAs in the cloud: Booting virtualized hardware accelerators with OpenStack. In *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2014.

[53] M. G. Bäckström and K. G. Lovstrand. Susceptibility of electronic systems to high-power microwaves: Summary of test experience. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 46(3):396–403, Aug 2004.

[54] G. Camurati, S. Poeplau, M. Muench, T. Hayes, and A. Francillon. Screaming channels: When electromagnetic side channels meet radio transceivers. In *ACM Conference on Computer and Communications Security (CCS)*, 2018.

[55] Y. Cao, V. Rožić, B. Yang, J. Balasch, and I. M. R. Verbauwhede. Exploring active manipulation attacks on the TERO random number generator. In *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2016.

[56] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. Hidden voice commands. In *USENIX Security Symposium*, 2016.

[57] B. Carrara and C. Adams. Out-of-band covert channels—A survey. *ACM Computing Surveys (CSUR)*, 49(2):1–36, Jun 2016.

[58] S. T. Castro, R. N. Dean, G. Roth, G. T. Flowers, and B. E. Grantham. Influence of acoustic noise on the dynamic performance of MEMS gyroscopes. In *International Mechanical Engineering Congress and Exposition (IMECE)*, 2007.

[59] F. Censi, G. Calcagnini, M. Triventi, E. Mattei, and P. Bartolini. Interference between mobile phones and pacemakers: A look inside. *Annali dell'Istituto Superiore di Sanità*, 43(3):254–259, Sep 2007.

[60] R. S. Chakraborty, I. Saha, A. Palchaudhuri, and G. K. Naik. Hardware Trojan insertion by direct modification of FPGA configuration bitstream. *IEEE Design & Test (D&T)*, 30(2):45–54, Apr 2013.

[61] R. Chauhan, R. M. Gerdes, and K. Heaslip. Demonstration of a false-data injection attack against an FMCW radar. In *Embedded Security in Cars (ESCAR)*, 2014.

[62] F. Chen, Y. Shan, Y. Zhang, Y. Wang, H. Franke, X. Chang, and K. Wang. Enabling FPGAs in the cloud. In *ACM International Conference on Computing Frontiers (CF)*, 2014.

[63] Y. Chen, H. Li, S. Nagels, Z. Li, P. Lopes, B. Y. Zhao, and H. Zheng. Understanding the effectiveness of ultrasonic microphone jammer. arXiv 1904.08490, 2019. http://arxiv.org/abs/1904.08490.

[64] Z. Chen and Y. Zhou. Dual-rail random switching logic: A countermeasure to reduce side channel leakage. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2006.

[65] A. Cheng, S. Nazarian, D. D. Spragg, K. Bilchick, H. Tandri, L. Mark, H. Halperin, H. Calkins, R. D. Berger, and C. A. Henrikson. Effects of surgical and endoscopic electrocautery on modern-day permanent pacemaker and implantable cardioverter-defibrillator systems. *Pacing and Clinical Electrophysiology*, 31(3):344–350, Mar 2008.

[66] A. Cherkaoui, V. Fischer, A. Aubert, and L. Fesquet. A self-timed ring based true random number generator. In *IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, 2013.

[67] Y.-K. Choi, J. Cong, Z. Fang, Y. Hao, G. Reinman, and P. Wei. In-depth analysis on microarchitectures of modern heterogeneous CPU-FPGA platforms. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 12(1):1–20, Apr 2019.

[68] L. Cohan, F. M. Kusumoto, and N. F. Goldschlager. Environmental effects on cardiac pacing systems. In F. M. Kusumoto and N. F. Goldschlager, editors, *Cardiac Pacing for the Clinician*, pages 595–618. Springer, 2nd edition, 2008.

[69] Corsair Components, Inc. Professional series Gold AX850 – 80 PLUS Gold certified fully-modular power supply. `https://www.corsair.com/p/CMPSU-850AX`. Created: Unknown. Accessed: 2019-10-01.

[70] Cybersecurity and Infrastructure Security Agency. ICS-ALERT-17-073-01A: MEMS accelerometer hardware design flaws (update A). `https://www.us-cert.gov/ics/alerts/ICS-ALERT-17-073-01A`. Created: 2017-04-11. Accessed: 2019-10-01.

[71] D. Davidson, H. Wu, R. Jellinek, T. Ristenpart, and V. Singh. Controlling UAVs with sensor input spoofing attacks. In *USENIX Workshop on Offensive Technologies (WOOT)*, 2016.

[72] T. De Cnudde, B. Bilgin, B. Gierlichs, V. Nikov, S. Nikova, and V. Rijmen. Does coupling affect the security of masked implementations? In *International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE)*, 2017.

[73] T. De Cnudde, M. Ender, and A. Moradi. Hardware masking, revisited. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2018(2):123–148, May 2018.

[74] R. N. Dean, S. T. Castro, G. T. Flowers, G. Roth, A. Ahmed, A. S. Hodel, B. E. Grantham, D. A. Bittle, and J. P. Brunsch. A characterization of the performance of a MEMS gyroscope in acoustically harsh environments. *IEEE Transactions on Industrial Electronics (TIE)*, 58(7):2591–2596, Jul 2011.

[75] R. N. Dean, G. T. Flowers, A. S. Hodel, G. Roth, S. T. Castro, R. Zhou, A. Moreira, A. Ahmed, R. Rifki, B. E. Grantham, D. A. Bittle, and J. P. Brunsch. On the degradation of MEMS gyroscope performance in the presence of high power acoustic noise. In *IEEE International Symposium on Industrial Electronics (ISIE)*, 2007.

[76] J. Delsing, J. Ekman, J. Johansson, S. Sundberg, M. G. Bäckström, and T. Nilsson. Susceptibility of sensor networks to intentional electromagnetic interference. In *International Zürich Symposium on Electromagnetic Compatibility (EMCZUR)*, 2006.

[77] C. Deshpande, B. Yuce, L. Nazhandali, and P. Schaumont. Employing dual-complementary flip-flops to detect EMFI attacks. In *IEEE Asian Hardware-Oriented Security and Trust Symposium (AsianHOST)*, 2017.

[78] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi. AccelPrint: Imperfections of accelerometers make smartphones trackable. In *Network and Distributed System Security Symposium (NDSS)*, 2014.

[79] P. Di Lena and L. Margara. Optimal global alignment of signals by maximization of Pearson correlation. *Information Processing Letters*, 110(16):679–686, Jul 2010.

[80] Digilent, Inc. Arty S7: Spartan-7 FPGA board for hobbyists and makers. https://store.digilentinc.com/arty-s7-spartan-7-fpga-board-for-hobbyists-and-makers/. Created: Unknown. Accessed: 2019-10-01.

[81] Digilent, Inc. Basys 3 Artix-7 FPGA trainer board: Recommended for introductory users. https://store.digilentinc.com/basys-3-artix-7-fpga-trainer-board-recommended-for-introductory-users/. Created: Unknown. Accessed: 2019-10-01.

[82] Digilent, Inc. Nexys A7: FPGA trainer board recommended for ECE curriculum. https://store.digilentinc.com/nexys-a7-fpga-trainer-board-recommended-for-ece-curriculum/. Created: Unknown. Accessed: 2019-10-01.

[83] Digilent, Inc. Virtex-5 OpenSPARC FPGA development board: ML509 (RETIRED). https://store.digilentinc.com/virtex-5-opensparc-fpga-development-board-ml509-retired/. Created: Unknown. Accessed: 2019-10-01.

[84] S. Driessen, A. Napp, K. Schmiedchen, T. Kraus, and D. Stunder. Electromagnetic interference in cardiac electronic implants caused by novel electrical appliances emitting electromagnetic fields in the intermediate frequency range: A systematic review. *EP Europace*, 21(2):219–229, Feb 2019.

[85] T. Dutta and A. R. Barnard. Performance of hard disk drives in high noise environments. *Noise Control Engineering Journal*, 65(5):386–395, Sep 2017.

[86] K. Eguro and R. Venkatesan. FPGAs for trusted cloud computing. In *International Conference on Field Programmable Logic and Applications (FPL)*, 2012.

[87] D. El-Baze, J.-B. Rigaud, and P. Maurine. An embedded digital sensor against EM and BB fault injection. In *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2016.

[88] D. El-Baze, J.-B. Rigaud, and P. Maurine. A fully-digital EM pulse detector. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2016.

[89] Ettus Research. N200/N210. https://kb.ettus.com/N200/N210. Created: 2019-08-21. Accessed: 2019-10-01.

[90] Ettus Research. The USRP hardware driver FPGA repository. https://github.com/EttusResearch/fpga. Created: 2019-09-03. Accessed: 2019-10-01.

[91] B. Farshteindiker, N. Hasidim, A. Grosz, and Y. Oren. How to phone home with someone else's phone: Information exfiltration using intentional sound noise on gyroscopic sensors. In *USENIX Workshop on Offensive Technologies (WOOT)*, 2016.

[92] F. Fiori. Design of an operational amplifier input stage immune to EMI. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 49(4):834–839, Nov 2007.

[93] D. Firestone, A. Putnam, S. Mundkur, D. Chiou, A. Dabagh, M. Andrewartha, H. Angepat, V. Bhanu, A. Caulfield, E. Chung, et al. Azure accelerated networking: SmartNICs in the public cloud. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2018.

[94] D. Foo Kune, J. Backes, S. S. Clark, D. Kramer, M. Reynolds, K. Fu, Y. Kim, and W. Xu. Ghost talk: Mitigating EMI signal injection attacks against analog sensors. In *IEEE Symposium on Security and Privacy (S&P)*, 2013.

[95] Forbes. Want to ruin someone's Oculus Rift fun? Fire this sonic gun at their head. https://www.forbes.com/sites/thomasbrewster/2017/07/11/alibaba-researchers-attack-facebook-vr-with-soundwaves. Created: 2017-07-11. Accessed: 2019-10-01.

[96] Fox News. Sonic weapon knocks drones right out of the sky. `https://www.foxnews.com/tech/sonic-weapon-knocks-drones-right-out-of-the-sky`. Created: 2017-11-26. Accessed: 2019-10-01.

[97] H. T. Friis. A note on a simple transmission formula. *Proceedings of the IRE (JRPROC)*, 34(5):254–256, May 1946.

[98] K. Fu and W. Xu. Risks of trusting the physics of sensors. *Communications of the ACM (CACM)*, 61(2):20–23, Feb 2018.

[99] M. Gag, T. Wegner, A. Waschki, and D. Timmermann. Temperature and on-chip crosstalk measurement using ring oscillators in FPGA. In *IEEE International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, 2012.

[100] J. Gago, J. Balcells, D. González, M. Lamich, J. Mon, and A. Santolaria. EMI susceptibility model of signal conditioning circuits based on operational amplifiers. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 49(4):849–859, Nov 2007.

[101] X.-L. Gao, C.-Y. Tian, L.-Y. Lao, Y.-H. Chen, and Y.-Y. Chen. Improved direct power injection model of 16-bit microcontroller for electromagnetic immunity prediction. *Journal of Central South University of Technology*, 18(6):2031–2035, Dec 2011.

[102] C. H. Gebotys. *Security in Embedded Devices*. Springer, 1st edition, 2010.

[103] D. Genkin, I. Pipman, and E. Tromer. Get your hands off my laptop: Physical side-channel key-extraction attacks on PCs. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2014.

[104] D. Genkin, R. Schuster, M. Pattani, and E. Tromer. Synesthesia: Detecting screen content via remote acoustic side channels. In *IEEE Symposium on Security and Privacy (S&P)*, 2019.

[105] D. Genkin, A. Shamir, and E. Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. In *International Cryptology Conference (CRYPTO)*, 2014.

[106] D. Genkin, A. Shamir, and E. Tromer. Acoustic cryptanalysis. *Journal of Cryptology*, 30(2):392–443, Apr 2017.

[107] H. Ghadamabadi, J. J. Whalen, R. Coslick, C. Hung, T. Johnson, W. Sitzman, and J. Stevens. Comparison of demodulation RFI in inverting operational amplifier circuits of the same gain with different input and feedback resistor values. In *IEEE International Symposium on Electromagnetic Compatibility (EMC)*, 1990.

[108] I. Giechaskiel, C. Cremers, and K. B. Rasmussen. On Bitcoin security in the presence of broken cryptographic primitives. In *European Symposium on Research in Computer Security (ESORICS)*, 2016.

[109] I. Giechaskiel, C. Cremers, and K. B. Rasmussen. When the "crypto" in cryptocurrencies breaks: Bitcoin security under broken primitives. *IEEE Security & Privacy*, 16(4):46–56, Aug 2018.

[110] I. Giechaskiel, K. Eguro, and K. B. Rasmussen. Leakier wires: Exploiting FPGA long wires for covert- and side-channel attacks. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 12(3):1–29, Sep 2019.

[111] I. Giechaskiel and K. B. Rasmussen. Taxonomy and challenges of out-of-band signal injection attacks and defenses. *IEEE Communications Surveys & Tutorials (COMST)*, 22(1):645–670, Mar 2020.

[112] I. Giechaskiel, K. B. Rasmussen, and K. Eguro. Leaky wires: Information leakage and covert communication between FPGA long wires. In *ACM ASIA Conference on Computer and Communications Security (ASIACCS)*, 2018.

[113] I. Giechaskiel, K. B. Rasmussen, and J. Szefer. Measuring long wire leakage with ring oscillators in cloud FPGAs. In *International Conference on Field-Programmable Logic and Applications (FPL)*, 2019.

[114] I. Giechaskiel, K. B. Rasmussen, and J. Szefer. Reading between the dies: Cross-SLR covert channels on multi-tenant cloud FPGAs. In *IEEE International Conference on Computer Design (ICCD)*, 2019.

[115] I. Giechaskiel, K. B. Rasmussen, and J. Szefer. C^{3}APSULe: Cross-FPGA covert-channel attacks through power supply unit leakage. In *IEEE Symposium on Security and Privacy (S&P)*, 2020.

[116] I. Giechaskiel, Y. Zhang, and K. B. Rasmussen. A framework for evaluating security in the presence of signal injection attacks. In *European Symposium on Research in Computer Security (ESORICS)*, 2019.

[117] D. V. Giri and F. M. Tesche. Classification of intentional electromagnetic environments (IEME). *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 46(3):322–328, Aug 2004.

[118] D. R. E. Gnad, J. Krautter, and M. B. Tahoori. Leaky noise: New side-channel attack vectors in mixed-signal IoT devices. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2019(3):305–339, May 2019.

[119] D. R. E. Gnad, F. Oboril, S. Kiamehr, and M. B. Tahoori. Analysis of transient voltage fluctuations in FPGAs. In *International Conference on Field-Programmable Technology (FPT)*, 2016.

[120] D. R. E. Gnad, F. Oboril, S. Kiamehr, and M. B. Tahoori. An experimental evaluation and analysis of transient voltage fluctuations in FPGAs. *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*, 26(10):1817–1830, Oct 2018.

[121] D. R. E. Gnad, F. Oboril, and M. B. Tahoori. Voltage drop-based fault attacks on FPGAs using valid bitstreams. In *International Conference on Field Programmable Logic and Applications (FPL)*, 2017.

[122] D. R. E. Gnad, S. Rapp, J. Krautter, and M. B. Tahoori. Checking for electrical level security threats in bitstreams for multi-tenant FPGAs. In *International Conference on Field-Programmable Technology (FPT)*, 2018.

[123] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[124] S. Gollakota, H. Hasssanieh, B. Ransford, D. Katabi, and K. Fu. They can hear your heartbeats: Non-invasive security for implantable medical devices. In *ACM Special Interest Group on Data Communication Conference (SIGCOMM)*, 2011.

[125] L. B. Gravelle and P. F. Wilson. EMI/EMC in printed circuit boards—A literature review. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 34(2):109–116, May 1992.

[126] A. Gregerson, A. Farmahini-Farahani, B. Buchli, S. Naumov, M. Bachtis, K. Compton, M. Schulte, W. H. Smith, and S. Dasu. FPGA design analysis of the clustering algorithm for the CERN Large Hadron Collider. In *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2009.

[127] B. Gregg. Unusual disk latency. `http://www.brendangregg.com/blog/2008-12-31/unusual-disk-latency.html`. Created: 2008-12-31. Accessed: 2019-10-01.

[128] S.-J. Guo, L.-S. Wu, M. Tang, and J.-F. Mao. Analysis of illuminated bent microstrip line based on Baum-Liu-Tesche (BLT) equation. In *IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*, 2015.

[129] M. Guri, M. Monitz, Y. Mirski, and Y. Elovici. BitWhisper: Covert signaling channel between air-gapped computers using thermal manipulations. In *IEEE Computer Security Foundations Symposium (CSF)*, 2015.

[130] M. Guri, Y. Solewicz, A. Daidakulov, and Y. Elovici. SPEAKE(a)R: Turn speakers to microphones for fun and profit. In *USENIX Workshop on Offensive Technologies (WOOT)*, 2017.

[131] M. Guri, B. Zadov, D. Bykhovsky, and Y. Elovici. PowerHammer: Exfiltrating data from air-gapped computers through power lines. arXiv 1804.04014, 2018. `http://arxiv.org/abs/1804.04014`.

[132] A. Hajimiri, S. Limotyrakis, and T. H. Lee. Jitter and phase noise in ring oscillators. *IEEE Journal of Solid-State Circuits (JSSC)*, 34(6):790–804, Jun 1999.

[133] D. Haperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *IEEE Symposium on Security and Privacy (S&P)*, 2008.

[134] D. M. Harris and S. L. Harris. *Digital Design and Computer Architecture*. Morgan Kaufmann, 2nd edition, 2012.

[135] Y.-I. Hayashi, S. Gomisawa, Y. Li, N. Homma, K. Sakiyama, T. Aoki, and K. Ohta. Intentional electromagnetic interference for fault analysis on AES block cipher IC. In *IEEE International Workshop on Electromagnetic Compatibility of Integrated Circuits (EMC Compo)*, 2011.

[136] Y.-I. Hayashi, N. Homma, T. Sugawara, T. Mizuki, T. Aoki, and H. Sone. Non-invasive EMI-based fault injection attack against cryptographic modules. In *IEEE International Symposium on Electromagnetic Compatibility (EMC)*, 2011.

[137] D. L. Hayes, P. J. Wang, D. W. Reynolds, N. A. M. Estes, J. L. Griffith, R. A. Steffens, G. L. Carlo, G. K. Findlay, and C. M. Johnson. Interference with cardiac pacemakers by cellular telephones. *New England Journal of Medicine (NEJM)*, 336(21):1473–1479, May 1997.

[138] C. Herder, M.-D. M. Yu, F. Koushanfar, and S. Devadas. Physical unclonable functions and applications: A tutorial. *Proceedings of the IEEE (JPROC)*, 102(8):1126–1141, Aug 2014.

[139] R. Hesselbarth, F. Wilde, C. Gu, and N. Hanley. Large scale RO PUF analysis over slice type, evaluation time and temperature on 28nm Xilinx FPGAs. In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2018.

[140] R. Hoad, N. J. Carter, D. Herke, and S. P. Watkins. Trends in EM susceptibility of IT equipment. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 46(3):390–395, Aug 2004.

[141] T. Huffmire, B. Brotherton, T. Sherwood, R. Kastner, T. Levin, T. D. Nguyen, and C. Irvine. Managing security in FPGA-based embedded systems. *IEEE Design & Test of Computers (D&T)*, 25(6):590–598, Nov 2008.

[142] T. Iakymchuk, M. Nikodem, and K. Kępa. Temperature-based covert channel in FPGA systems. In *International Workshop on Reconfigurable Communication-Centric Systems-on-Chip (ReCoSoC)*, 2011.

[143] IEEE Standard for Information Technology. Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. *IEEE Standards*, IEEE Std. 802.3–2002, Mar 2002.

[144] iFixit. MacBook Pro 15" unibody early 2011 teardown. `https://www.ifixit.com/Teardown/MacBook+Pro+15-Inch+Unibody+Early+2011+Teardown/4990`. Created: 2011-02-24. Accessed: 2019-10-01.

[145] V. Immler, R. Specht, and F. Unterstein. Your rails cannot hide from localized EM: How dual-rail logic fails on FPGAs. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2017.

[146] W. Inrich, L. Batz, R. Müller, and R. Tobisch. Electromagnetic interference of pacemakers by mobile phones. *Pacing and Clinical Electrophysiology (PACE)*, 19(10):1431–1446, Jun 2006.

[147] M. A. Islam and S. Ren. Ohm's law in data centers: A voltage side channel for timing power attacks. In *ACM Conference on Computer and Communications Security (CCS)*, 2018.

[148] R. Ivanov, M. Pajic, and I. Lee. Attack-resilient sensor fusion for safety-critical cyber-physical systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 15(1):1–24, Feb 2016.

[149] J. Jiang and Y. Qian. Defense mechanisms against data injection attacks in smart grid networks. *IEEE Communications Magazine*, 55(10):76–82, Oct 2017.

[150] R. Jin and K. Zeng. Physical layer key agreement under signal injection attacks. In *IEEE Conference on Communications and Network Security (CNS)*, 2015.

[151] D. Karaklajić, J.-M. Schmidt, and I. M. R. Verbauwhede. Hardware designer's guide to fault attacks. *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*, 21(12):2295–2306, Dec 2013.

[152] R. Karri, J. Rajendran, and K. Rosenfeld. Trojan taxonomy. In M. Tehranipoor and C. Wang, editors, *Introduction to Hardware Security and Trust*, pages 325–338. Springer, 1st edition, 2012.

[153] C. Kasmi and J. Lopes Esteves. IEMI threats for information security: Remote command injection on modern smartphones. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 57(6):1752–1755, Dec 2015.

[154] S. Katzenbeisser, Ü. Kocabaş, V. Rožić, A.-R. Sadeghi, I. M. R. Verbauwhede, and C. Wachsmann. PUFs: Myth, fact or busted? A security evaluation of physically unclonable functions (PUFs) cast in silicon. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2012.

[155] S. Kelly, X. Zhang, M. Tehranipoor, and A. Ferraiuolo. Detecting Hardware Trojans using on-chip sensors in an ASIC design. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 31(1):11–26, Feb 2015.

[156] S. Kennedy, M. R. Yuce, and J.-M. Redouté. Susceptibility of flash ADCs to electromagnetic interference. *Microelectronics Reliability*, 81:218–225, Feb 2018.

[157] S. K. Khatamifard, L. Wang, A. Das, S. Köse, and U. R. Karpuzcu. POWERT channels: A novel class of covert communication exploiting power management vulnerabilities. In *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2019.

[158] A. Khawaja, J. Landgraf, R. Prakash, M. Wei, E. Schkufza, and C. J. Rossbach. Sharing, protection, and compatibility for reconfigurable fabric with AmorphOS. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2018.

[159] S. Khazaaleh, G. Korres, M. Eid, M. Rasras, and M. F. Daqaq. Vulnerability of MEMS gyroscopes to targeted acoustic attacks. *IEEE Access*, 7:89534–89543, Jul 2019.

[160] J. S. Kim, M. Patel, H. Hassan, and O. Mutlu. The DRAM latency PUF: Quickly evaluating physical unclonable functions by exploiting the latency-reliability tradeoff in modern commodity DRAM devices. In *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2018.

[161] C. Kison, O. M. Awad, M. Fyrbiak, and C. Paar. Security implications of intentional capacitive crosstalk. *IEEE Transactions on Information Forensics and Security (TIFS)*, 14(12):3246–3258, Dec 2019.

[162] O. Knodel, P. R. Genssler, F. Erxleben, and R. G. Spallek. FPGAs and the cloud—An endless tale of virtualization, elasticity and efficiency. *International Journal on Advances in Systems and Measurements*, 11(3):230–249, Dec 2018.

[163] O. Knodel, P. R. Genssler, and R. G. Spallek. Virtualizing reconfigurable hardware to provide scalability in cloud architectures. In *International Conference on Advances in Circuits, Electronics and Micro-electronics (CENICS)*, 2017.

[164] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi. Introduction to differential power analysis. *Journal of Cryptographic Engineering (JCEN)*, 1(1):5–27, Apr 2011.

[165] P. Kohlbrenner and K. Gaj. An embedded true random number generator for FPGAs. In *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*, 2004.

[166] S. Korf, D. Cozzi, M. Koester, J. Hagemeyer, M. Porrmann, U. Rückert, and M. D. Santambrogio. Automatic HDL-based generation of homogeneous hard macros for FPGAs. In *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2011.

[167] J. Krautter, D. R. E. Gnad, and M. B. Tahoori. FPGAhammer: Remote voltage fault attacks on shared FPGAs, suitable for DFA on AES. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2018(3):44–68, Sep 2018.

[168] J. Krautter, D. R. E. Gnad, and M. B. Tahoori. Mitigating electrical-level attacks towards secure multi-tenant FPGAs in the cloud. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 12(3):1–26, Sep 2019.

[169] C. Krieg, C. Wolf, and A. Jantsch. Malicious LUT: A stealthy FPGA Trojan injected and triggered by the design flow. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016.

[170] M. G. Kuhn. Optical time-domain eavesdropping risks of CRT displays. In *IEEE Symposium on Security and Privacy (S&P)*, 2002.

[171] M. G. Kuhn. Electromagnetic eavesdropping risks of flat-panel displays. In *International Workshop on Privacy Enhancing Technologies (PET)*, 2004.

[172] M. G. Kuhn. Compromising emanations of LCD TV sets. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 55(3):564–570, Jun 2013.

[173] M. G. Kuhn and R. J. Anderson. Soft Tempest: Hidden data transmission using electromagnetic emanations. In *International Workshop on Information Hiding (IH)*, 1998.

[174] A. Kwong, W. Xu, and K. Fu. Hard drive of hearing: Disks that eavesdrop with a synthesized microphone. In *IEEE Symposium on Security and Privacy (S&P)*, 2019.

[175] F. Laermer. Mechanical microsensors. In J. G. Korvink and O. Paul, editors, *MEMS: A Practical Guide to Design, Analysis, and Applications*, pages 523–566. Springer, 1st edition, 2006.

[176] F. Lafon, F. De Daran, M. Ramdani, R. Perdriau, and M. Drissi. Immunity modeling of integrated circuits: An industrial case. *IEICE Transactions on Communications*, E93.B(7):1723–1730, Jul 2010.

[177] J. L. Lagos and F. Fiori. Worst-case induced disturbances in digital and analog interchip interconnects by an external electromagnetic plane wave—Part I: Modeling and algorithm. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 53(1):178–184, Feb 2011.

[178] J. L. Lagos and F. Fiori. Worst-case induced disturbances in digital and analog interchip interconnects by an external electromagnetic plane wave—Part II: Analysis and validation. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 53(2):491–500, May 2011.

[179] S. Lakshminarayana, J. S. Karachiwala, S.-Y. Chang, G. Revadigar, S. L. S. Kumar, D. K. Y. Yau, and Y.-C. Hu. Signal jamming attacks against communication-based train control: Attack impact and countermeasure. In *ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, 2018.

[180] C. Lavin, B. Nelson, and B. Hutchings. Impact of hard macro size on FPGA clock rate and place/route time. In *International Conference on Field Programmable Logic and Applications (FPL)*, 2013.

[181] C. Lavin, M. Padilla, S. Ghosh, B. Nelson, B. Hutchings, and M. Wirthlin. Using hard macros to reduce FPGA compilation time. In *International Conference on Field Programmable Logic and Applications (FPL)*, 2010.

[182] C. Lavin, M. Padilla, J. Lamprecht, P. Lundrigan, B. Nelson, and B. Hutchings. HMFlow: Accelerating FPGA compilation with hard macros for rapid prototyping. In *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2011.

[183] A. Le Masle and W. Luk. Detecting power attacks on reconfigurable hardware. In *International Conference on Field Programmable Logic and Applications (FPL)*, 2012.

[184] C. Leber, B. Geib, and H. Litz. High frequency trading acceleration using FPGAs. In *International Conference on Field Programmable Logic and Applications (FPL)*, 2011.

[185] M. Lecomte, J. J. A. Fournier, and P. Maurine. Thoroughly analyzing the use of ring oscillators for on-chip Hardware Trojan detection. In *International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, 2015.

[186] S. Lee, K. Fu, T. Kohno, B. Ransford, and W. H. Maisel. Clinically significant magnetic interference of implanted cardiac devices by portable headphones. *Heart Rhythm*, 6(10):1432–1436, Oct 2009.

[187] M. Leone. Radiated susceptibility on the Printed-Circuit-Board level: Simulation and measurement. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 47(3):471–478, Aug 2015.

[188] M. Leone and H. L. Singer. On the coupling of an external electromagnetic field to a printed circuit board trace. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 41(4):418–424, Nov 1999.

[189] C. Li, A. Raghunathan, and N. K. Jha. Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. In *IEEE International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2011.

[190] A. Lissner, E. Hoene, B. Stube, and S. Guttowski. Predicting the influence of placement of passive components on EMI behaviour. In *European Conference on Power Electronics and Applications*, 2007.

[191] Y. Liu, P. Ning, and M. K. Reiter. False data injection attacks against state estimation in electric power grids. In *ACM Conference on Computer and Communications Security (CCS)*, 2009.

[192] J. Lopes Esteves, E. Cottais, and C. Kasmi. Unlocking the access to the effects induced by IEMI on a civilian UAV. In *International Symposium and Exhibition on Electromagnetic Compatibility (EMC Europe)*, 2018.

[193] J. Lopes Esteves and C. Kasmi. Remote and silent voice command injection on a smartphone through conducted IEMI: Threats of smart IEMI for information security. Technical Report System Design & Assessment Note 48, Wireless Security Lab, French Network and Information Security Agency (ANSSI), Apr 2018. `http://ece-research.unm.edu/summa/notes/SDAN/SDAN0048.pdf`.

[194] J. Loughry. ("Oops! Had the silly thing in reverse")—Optical injection attacks in through LED status indicators. In *International Symposium and Exhibition on Electromagnetic Compatibility (EMC Europe)*, 2019.

[195] J. Loughry and D. A. Umphress. Information leakage from optical emanations. *ACM Transactions on Information and System Security (TISSEC)*, 5(3):262–289, Aug 2002.

[196] R. Maes and I. M. R. Verbauwhede. Physically unclonable functions: A study on the state of the art and future research directions. In A.-R. Sadeghi and D. Naccache, editors, *Towards Hardware-Intrinsic Security: Foundations and Practice*, pages 3–37. Springer, 1st edition, 2010.

[197] D. Mahmoud and M. Stojilović. Timing violation induced faults in multi-tenant FPGAs. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019.

[198] A. Maiti, J. Casarona, L. McHale, and P. Schaumont. A large scale characterization of RO-PUF. In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2010.

[199] A. Maiti and P. Schaumont. Improved ring oscillator PUF: An FPGA-friendly secure primitive. *Journal of Cryptology*, 24(2):375–397, Apr 2011.

[200] A. T. Markettos and S. W. Moore. The frequency injection attack on ring-oscillator-based true random number generators. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2009.

[201] H. Martín, T. Korak, E. San Millán, and M. Hutter. Fault attacks on STRNGs: Impact of glitches, temperature, and underpowering on randomness. *IEEE Transactions on Information Forensics and Security (TIFS)*, 10(2):266–277, Feb 2015.

[202] H. Martín, P. Martin-Holgado, P. Peris-Lopez, Y. Morilla, and L. Entrena. On the entropy of oscillator-based true random number generators under ionizing radiation. *Entropy*, 20(7):1–11, Jul 2018.

[203] P. Marwedel. *Embedded System Design: Embedded Systems, Foundations of Cyber-Physical Systems, and the Internet of Things*. Springer, 3rd edition, 2018.

[204] R. J. Masti, D. Rai, A. Ranganathan, C. Müller, L. Thiele, and S. Capkun. Thermal covert channels on multi-core platforms. In *USENIX Security Symposium*, 2015.

[205] N. Matyunin, J. Szefer, and S. Katzenbeisser. Zero-permission acoustic cross-device tracking. In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2018.

[206] D. Merli, F. Stumpf, and C. Eckert. Improving the quality of ring oscillator PUFs on FPGAs. In *Workshop on Embedded Systems Security (WESS)*, 2010.

[207] B. Mesgarzadeh and A. Alvandpour. A study of injection locking in ring oscillators. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2005.

[208] Y. Michalevsky, D. Boneh, and G. Nakibly. Gyrophone: Recognizing speech from gyroscope signals. In *USENIX Security Symposium*, 2014.

[209] F. Michel and M. Steyaert. Comparison of high impedance input topologies with low EMI susceptibility. *Analog Integrated Circuits and Signal Processing*, 65(2):299–309, Nov 2010.

[210] F. Michel and M. Steyaert. Differential input topologies with immunity to electromagnetic interference. In *European Conference on Solid-State Circuits (ESSCIRC)*, 2011.

[211] Microsemi Corporation. RISC-V CPUs. `https://www.microsemi.com/product-directory/mi-v-embedded-ecosystem/4406-risc-v-cpus`. Created: Unknown. Accessed: 2019-10-01.

[212] Microsoft Azure. What are field-programmable gate arrays (FPGA) and how to deploy. `https://docs.microsoft.com/en-us/azure/machine-learning/service/how-to-deploy-fpga-web-service`. Created: 2019-07-25. Accessed: 2019-10-01.

[213] P. Miedl, X. He, M. Meyer, D. B. Bartolini, and L. Thiele. Frequency scaling as a security threat on multicore systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 37(11):2497–2508, Nov 2018.

[214] F. M. Mims, III. Bidirectional optoisolator puts two LEDs nose to nose. *Electronics*, 52(10):127–127, May 1979.

[215] F. M. Mims, III. Sun photometer with light-emitting diodes as spectrally selective detectors. *Applied Optics*, 31(33):6965–6967, Nov 1992.

[216] J. Misiri, F. Kusumoto, and N. Goldschlager. Electromagnetic interference and implanted cardiac devices: The medical environment (part II). *Clinical Cardiology*, 35(6):321–328, Jun 2012.

[217] J. Misiri, F. Kusumoto, and N. Goldschlager. Electromagnetic interference and implanted cardiac devices: The nonmedical environment (part I). *Clinical Cardiology*, 35(5):276–280, May 2012.

[218] N. Miura, Z. Najm, W. He, S. Bhasin, X. T. Ngo, M. Nagata, and J.-L. Danger. PLL to the rescue: A novel EM fault countermeasure. In *Design Automation Conference (DAC)*, 2016.

[219] J. V. Monaco. SoK: Keylogging side channels. In *IEEE Symposium on Security and Privacy (S&P)*, 2018.

[220] M. I. Montrose. *EMC and the Printed Circuit Board: Design, Theory, and Layout Made Simple*. Wiley, 1st edition, 1999.

[221] A. Moradi. Side-channel leakage through static power. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2014.

[222] A. Moradi, A. Barenghi, T. Kasper, and C. Paar. On the vulnerability of FPGA bitstream encryption against power analysis attacks: Extracting keys from Xilinx Virtex-II FPGAs. In *ACM Conference on Computer and Communications Security (CCS)*, 2011.

[223] A. Moradi, K. Markus, and C. Paar. Black-box side-channel attacks highlight the importance of countermeasures: An analysis of the Xilinix Virtex-4 and Virtex-5 bitstream encryption mechanism. In *Cryptographers' Track at the RSA Conference (CT-RSA)*, 2012.

[224] D. Muniraj and M. Farhood. Detection and mitigation of actuator attacks on small unmanned aircraft systems. *Control Engineering Practice*, 83:188–202, Feb 2019.

[225] B. Murmann. ADC performance survey 1997-2019. `http://web.stanford.edu/~murmann/adcsurvey.html`. Created: 2019-08-02. Accessed: 2019-10-01.

[226] D. Månsson, T. Nilsson, R. Thottappillil, and M. G. Bäckström. Propagation of UWB transients in low-voltage installation power cables. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 49(3):585–592, Aug 2007.

[227] D. Månsson, R. Thottappillil, and M. G. Bäckström. Propagation of UWB transients in low-voltage power installation networks. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 50(3):619–629, Aug 2008.

[228] D. Månsson, R. Thottappillil, and M. G. Bäckström. Methodology for classifying facilities with respect to intentional EMI. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 51(1):46–52, Feb 2009.

[229] D. Månsson, R. Thottappillil, M. G. Bäckström, and O. Lundén. Vulnerability of European rail traffic management system to radiated intentional EMI. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 50(1):101–109, Feb 2008.

[230] D. Månsson, R. Thottappillil, T. Nilsson, O. Lundén, and M. G. Bäckström. Susceptibility of civilian GPS receivers to electromagnetic radiation. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 50(2):434–437, May 2008.

[231] S. Narasimhan and S. Bhunia. Hardware Trojan detection. In M. Tehranipoor and C. Wang, editors, *Introduction to Hardware Security and Trust*, pages 339–364. Springer, 1st edition, 2012.

[232] S. Nashimoto, D. Suzuki, T. Sugawara, and K. Sakiyama. Sensor CON-Fusion: Defeating Kalman filter in signal injection attack. In *ACM ASIA Conference on Computer and Communications Security (ASIACCS)*, 2018.

[233] National Public Radio. Can you hear it? Sonic devices play high-pitched noises to repel teens. `https://www.npr.org/2019/07/10/739908153/can-you-hear-it-sonic-devices-play-high-pitched-noises-to-repel-teens`. Created: 2019-07-10. Accessed: 2019-10-01.

[234] Nimbix, Inc. Xilinx Alveo accelerator cards. `https://www.nimbix.net/alveo/`. Created: Unknown. Accessed: 2019-10-01.

[235] C. O'Flynn and A. Dewar. On-device power analysis across hardware security domains: Stop hitting yourself. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2019(4):126–153, Aug 2019.

[236] L. Orosa, Y. Wang, I. Puddu, M. Sadrosadati, K. Razavi, J. Gómez-Luna, H. Hassan, N. Mansouri-Ghiasi, A. Tavakkol, M. Patel, J. Kim, V. Seshadri, U. Kang, S. Ghose, R. Azevedo, and O. Mutlu. Dataplant: In-DRAM security mechanisms for low-cost devices. arXiv 1902.07344, 2019. `http://arxiv.org/abs/1902.07344`.

[237] S. Osuka, D. Fujimoto, Y.-I. Hayashi, N. Homma, A. Beckers, J. Balasch, B. Gierlichs, and I. M. R. Verbauwhede. EM information security threats against RO-based TRNGs: The frequency injection attack based on IEMI and EM information leakage. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 61(4):1122–1128, Aug 2019.

[238] J. Ouyang, S. Lin, S. Jiang, Z. Hou, Y. Wang, and Y. Wang. SDF: Software-defined flash for web-scale internet storage systems. In *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2014.

[239] L. Palíšek and L. Suchý. High Power Microwave effects on computer networks. In *International Symposium and Exhibition on Electromagnetic Compatibility (EMC Europe)*, 2011.

[240] Y. V. Parfenov, L. N. Zdoukhov, W. A. Radasky, and M. Ianoz. Conducted IEMI threats for commercial buildings. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 46(3):404–411, Aug 2004.

[241] J. Park, R. Ivanov, J. Weimer, M. Pajic, and I. Lee. Sensor attack detection in the presence of transient faults. In *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2013.

[242] Y. Park, Y. Son, H. Shin, D. Kim, and Y. Kim. This ain't your dose: Sensor spoofing attack on medical infusion pump. In *USENIX Workshop on Offensive Technologies (WOOT)*, 2016.

[243] S. Patranabis and D. Mukhopadhyay. Classical countermeasures against differential fault analysis. In S. Patranabis and D. Mukhopadhyay, editors, *Fault Tolerant Architectures for Cryptography and Hardware Security*, pages 171–182. Springer, 1st edition, 2018.

[244] M. J. M. Pelgrom. *Analog-to-Digital Conversion*. Springer, 3rd edition, 2017.

[245] J. Petit and S. E. Shladover. Potential cyberattacks on automated vehicles. *IEEE Transactions on Intelligent Transportation Systems (TITS)*, 16(2):546–556, Apr 2015.

[246] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl. Remote attacks on automated vehicles sensors: Experiments on camera and LiDAR. Black Hat Europe, 2015.

[247] S. L. Pinski and R. G. Trohman. Interference in implanted cardiac devices, part I. *Pacing and Clinical Electrophysiology (PACE)*, 25(9):1367–1381, Sep 2002.

[248] S. L. Pinski and R. G. Trohman. Interference in implanted cardiac devices, part II. *Pacing and Clinical Electrophysiology (PACE)*, 25(10):1496–1509, Oct 2002.

[249] T. Popp and S. Mangard. Masked dual-rail pre-charge logic: DPA-resistance without routing constraints. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2005.

[250] C. Pouant, F. Torrès, A. Reineix, P. Hoffmann, J. Raoult, and L. Chusseau. Modeling and analysis of large-signal RFI effects in MOS transistors. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 61(1):111–120, Feb 2019.

[251] D. Pozar. *Microwave Engineering*. Wiley, 4th edition, 2011.

[252] G. Provelengios, D. Holcomb, and R. Tessier. Characterizing power distribution attacks in multi-user FPGA environments. In *International Conference on Field-Programmable Logic and Applications (FPL)*, 2019.

[253] G. Provelengios, C. Ramesh, S. B. Patil, K. Eguro, R. Tessier, and D. Holcomb. Characterization of long wire data leakage in deep submicron FPGAs. In *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*, 2019.

[254] A. Putnam, A. M. Caulfield, E. S. Chung, D. Chiou, K. Constantinides, J. Demme, H. Esmaeilzadeh, J. Fowers, G. P. Gopal, J. Gray, et al. A reconfigurable fabric for accelerating large-scale datacenter services. In *ACM/IEEE International Symposium on Computer Architecuture (ISCA)*, 2014.

[255] J.-J. Quisquater and D. Samyde. ElectroMagnetic analysis (EMA): Measures and countermeasures for smart cards. In *International Conference on Research in Smart Cards: Smart Card Programming and Security (E-smart)*, 2001.

[256] W. A. Radasky, C. E. Baum, and M. W. Wik. Introduction to the special issue on high-power electromagnetics (HPEM) and intentional electromagnetic interference (IEMI). *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 46(3):314–321, Aug 2004.

[257] T. Ragheb and A. Marshall. Calibration of propagation delay of flip-flops. In *IEEE International SOC Conference (SOCC)*, 2012.

[258] N. Rahmanikia, A. Amiri, H. Noori, and F. Mehdipour. Performance evaluation metrics for ring-oscillator-based temperature sensors on FPGAs: A quality factor. *Integration: The VLSI Journal*, 57:81–100, Mar 2017.

[259] M. Ramdani, E. Sicard, A. Boyer, S. Ben Dhia, J. J. Whalen, T. H. Hubing, M. Coenen, and O. Wada. The electromagnetic compatibility of integrated circuits—Past, present, and future. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 51(1):78–100, Feb 2009.

[260] C. Ramesh, S. B. Patil, S. N. Dhanuskodi, G. Provelengios, S. Pillement, D. Holcomb, and R. Tessier. FPGA side channel attacks without physical access. In *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2018.

[261] Y. Ran and M. Marek-Sadowska. Crosstalk noise in FPGAs. In *Design Automation Conference (DAC)*, 2003.

[262] K. B. Rasmussen, C. Castelluccia, T. S. Heydt-Benjamin, and S. Capkun. Proximity-based access control for implantable medical devices. In *ACM Conference on Computer and Communications Security (CCS)*, 2009.

[263] J.-M. Redouté and A. Richelli. A fundamental approach to EMI resistant folded cascode operational amplifier design. In *International Symposium and Exhibition on Electromagnetic Compatibility (EMC Europe)*, 2013.

[264] J.-M. Redouté and A. Richelli. A methodological approach to EMI resistant analog integrated circuit design. *IEEE Electromagnetic Compatibility Magazine*, 4(2):92–100, Aug 2015.

[265] J.-M. Redouté and M. Steyaert. *EMC of Analog Integrated Circuits*. Springer, 1st edition, 2010.

[266] R. P. Ribas, A. I. Reis, and A. Ivanov. Performance and functional test of flip-flops using ring oscillator structure. In *IEEE International Design and Test Workshop (IDT)*, 2011.

[267] A. Richelli. EMI susceptibility issue in analog front-end for sensor applications. *Journal of Sensors*, 2016:1–9, 2016.

[268] A. Richelli, L. Colalongo, M. Quarantelli, and Z. M. Kovács-Vajna. Robust design of low EMI susceptibility CMOS OpAmp. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 46(2):291–298, May 2004.

[269] M. Rostami, F. Koushanfar, and R. Karri. A primer on hardware security: Models, methods, and metrics. *Proceedings of the IEEE (JPROC)*, 102(8):1283–1295, Aug 2014.

[270] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar. Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. In *USENIX Security Symposium*, 2010.

[271] N. Roy, H. Hassanieh, and R. R. Choudhury. BackDoor: Making microphones hear inaudible sounds. In *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2017.

[272] N. Roy, S. Shen, H. Hassanieh, and R. R. Choudhury. Inaudible voice commands: The long-range attack and defense. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2018.

[273] M. Rushanan, A. D. Rubin, D. Foo Kune, and C. M. Swanson. SoK: Security and privacy in implantable medical devices and body area network. In *IEEE Symposium on Security and Privacy (S&P)*, 2014.

[274] S. Saab, A. Leiserson, and M. Tunstall. Key extraction from the primary side of a switched-mode power supply. In *IEEE Asian Hardware-Oriented Security and Trust Symposium (AsianHOST)*, 2016.

[275] F. Sabath. Classification of electromagnetic effects at system level. In *International Symposium and Exhibition on Electromagnetic Compatibility (EMC Europe)*, 2008.

[276] P. Samarin, K. Lemke-Rust, and C. Paar. IP core protection using voltage-controlled side-channel receivers. In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2016.

[277] L. Sauvage, J.-L. Danger, S. Guilley, N. Homma, and Y.-I. Hayashi. Advanced analysis of faults injected through conducted intentional electromagnetic interferences. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 55(3):589–596, Jun 2013.

[278] E. Savage and W. Radasky. Overview of the threat of IEMI (intentional electromagnetic interference). In *IEEE International Symposium on Electromagnetic Compatibility (EMC)*, 2012.

[279] S. Sbaraini, A. Richelli, and Z. M. Kovács-Vajna. EMI susceptibility in bulk-driven Miller OpAmp. *Electronics Letters*, 46(16):1111–1113, Aug 2010.

[280] A. Schaller, W. Xiong, N. A. Anagnostopoulos, M. U. Saleem, S. Gabmeyer, B. Škorić, S. Katzenbeisser, and J. Szefer. Decay-based DRAM PUFs in commodity devices. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 16(3):462–475, May 2019.

[281] P. Schaumont and K. Tiri. Masking and dual-rail logic don't add up. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2007.

[282] F. Schellenberg, D. R. E. Gnad, A. Moradi, and M. B. Tahoori. An inside job: Remote power analysis attacks on FPGAs. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018.

[283] F. Schellenberg, D. R. E. Gnad, A. Moradi, and M. B. Tahoori. Remote inter-chip power analysis side-channel attacks at board-level. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2018.

[284] J.-M. Schmidt, T. Plos, M. Kirschbaum, M. Hutter, M. Medwed, and C. Herbst. Side-channel leakage across borders. In *International Conference on Smart Card Research and Advanced Applications (CARDIS)*, 2010.

[285] S. J. Seidman, R. Brockman, B. M. Lewis, J. Guag, M. J. Shein, W. J. Clement, J. Kippola, D. Digby, C. Barber, and D. Huntwork. In vitro tests reveal sample radiofrequency identification readers inducing clinically significant electromagnetic interference to implantable pacemakers and implantable cardioverter-defibrillators. *Heart Rhythm*, 7(1):99–107, Jan 2010.

[286] J. Selvaraj, G. Y. Dayanıklı, N. P. Gaunkar, D. Ware, R. M. Gerdes, and M. Mina. Electromagnetic induction attacks against embedded systems. In *ACM ASIA Conference on Computer and Communications Security (ASIACCS)*, 2018.

[287] D. E. Serrano, R. Lipka, D. Younkin, P. Hrudey, J. Tovera, A. Rahafrooz, M. F. Zaman, S. Nagpal, I. Jafri, and F. Ayazi. Environmentally-robust high-performance tri-axial bulk acoustic wave gyroscopes. In *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2016.

[288] D. E. Serrano, M. F. Zaman, A. Rahafrooz, P. Hrudey, R. Lipka, D. Younkin, S. Nagpal, I. Jafri, and F. Ayazi. Substrate-decoupled, bulk-acoustic wave gyroscopes: Design and evaluation of next-generation environmentally robust devices. *Microsystems & Nanoengineering*, 2:1–10, Dec 2016.

[289] M. Shahrad, A. Mosenia, L. Song, M. Chiang, D. Wentzlaff, and P. Mittal. Acoustic denial of service attacks on HDDs. In *Workshop on Attacks and Solutions in Hardware Security (ASHES)*, 2018.

[290] H. Shall, Z. Riah, and M. Kadi. A novel approach for modeling near-field coupling with PCB traces. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 56(5):1194–1201, Oct 2014.

[291] H. Shen, W. Zhang, H. Fang, Z. Ma, and N. Yu. JamSys: Coverage optimization of a microphone jamming system based on ultrasounds. *IEEE Access*, 7:67483–67496, May 2019.

[292] L. L. Shen, I. Ahmed, and V. Betz. Fast voltage transients on FPGAs: Impact and mitigation strategies. In *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2019.

[293] D. M. Shila, V. Venugopalan, and C. D. Patterson. Unraveling the security puzzle: A distributed framework to build trust in FPGAs. In *International Conference on Network and System Security (NSS)*, 2015.

[294] H. Shin, D. Kim, Y. Kwon, and Y. Kim. Illusion and dazzle: Adversarial optical channel exploits against LiDARs for automotive applications. In *Conference on Cryptographic Hardware and Embedded Systems (CHES)*, 2017.

[295] H. Shin, Y. Son, Y. Park, Y. Kwon, and Y. Kim. Sampling race: Bypassing timing-based analog active sensor spoofing detection on analog-digital systems. In *USENIX Workshop on Offensive Technologies (WOOT)*, 2016.

[296] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava. Non-invasive spoofing attacks for anti-lock braking systems. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2013.

[297] Y. Shoukry, P. Martin, Y. Yona, S. Diggavi, and M. Srivastava. PyCRA: Physical challenge-response authentication for active sensors under spoofing attacks. In *ACM Conference on Computer and Communications Security (CCS)*, 2015.

[298] M. Šimka and P. Komenského. Active non-invasive attack on true random number generator. In *PhD Student Conference and Scientific and Technical Competition of Students of FEI TU Košice, Košice, Slovakia*, 2006.

[299] Sixth Tone. The gadget that boosts your step count while you nap. `https://www.sixthtone.com/news/1002530/the-gadget-that-boosts-your-step-count-while-you-nap-`. Created: 2018-06-28. Accessed: 2019-10-01.

[300] D. Sokolov, J. Murphy, A. Bystrov, and A. Yakovlev. Improving the security of dual-rail circuits. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2004.

[301] Y. Son, H. Shin, D. Kim, Y. Park, J. Noh, K. Choi, J. Choi, and Y. Kim. Rocking drones with intentional sound noise on gyroscopic sensors. In *USENIX Security Symposium*, 2015.

[302] C. Song, F. Lin, Z. Ba, K. Ren, C. Zhou, and W. Xu. My smartphone knows what you print: Exploring smartphone-based side-channel attacks against 3D printers. In *ACM Conference on Computer and Communications Security (CCS)*, 2016.

[303] L. Song and P. Mittal. POSTER: Inaudible voice commands. In *ACM Conference on Computer and Communications Security (CCS)*, 2017.

[304] M. Soucarros, C. Canovas-Dumas, J. Clédière, P. Elbaz-Vincent, and D. Réal. Influence of the temperature on true random number generators. In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2011.

[305] R. Spolaor, L. Abudahi, V. Moonsamy, M. Conti, and R. Poovendran. No free charge theorem: A covert channel via USB charging cable on mobile devices. In *International Conference on Applied Cryptography and Network Security (ACNS)*, 2017.

[306] J. Spolsky. The law of leaky abstractions. `https://www.joelonsoftware.com/2002/11/11/the-law-of-leaky-abstractions/`. Created: 2002-11-11. Accessed: 2019-10-01.

[307] R. Spreitzer, V. Moonsamy, T. Korak, and S. Mangard. Systematic classification of side-channel attacks: A case study for mobile devices. *IEEE Communications Surveys & Tutorials (COMST)*, 20(1):465–488, Feb 2018.

[308] S. Sridhar and G. Manimaran. Model-based attack detection and mitigation for automatic generation control. *IEEE Transactions on Smart Grid (TSG)*, 5(2):580–591, Mar 2014.

[309] C. Su, Y.-T. Chen, M.-J. Huang, G.-N. Chen, and C.-L. Lee. All digital built-in delay and crosstalk measurement for on-chip buses. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2000.

[310] Y. Su, D. Genkin, D. Ranasinghe, and Y. Yarom. USB snooping made easy: Crosstalk leakage attacks on USB hubs. In *USENIX Security Symposium*, 2017.

[311] T. Sugawara, K. Sakiyama, S. Nashimoto, D. Suzuki, and T. Nagatsuka. Oscillator without a combinatorial loop and its threat to FPGA in data centre. *Electronics Letters*, 55(11):640–642, Jun 2019.

[312] G. E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *Design Automation Conference (DAC)*, 2007.

[313] J. Sun, R. Bittner, and K. Eguro. FPGA side-channel receivers. In *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*, 2011.

[314] B. Sunar, W. J. Martin, and D. R. Stinson. A provably secure true random number generator with built-in tolerance to active attacks. *IEEE Transactions on Computers (TC)*, 56(1):109–119, Jan 2007.

[315] Y.-H. Sutu and J. J. Whalen. Statistics for demodulation RFI in operational amplifiers. In *IEEE International Symposium on Electromagnetic Compatibility (EMC)*, 1983.

[316] D. Suzuki and M. Saeki. Security evaluation of DPA countermeasures using dual-rail pre-charge logic style. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2006.

[317] H. Tanaka. Information leakage via electromagnetic emanations and evaluation of Tempest countermeasures. In *International Conference on Information Systems Security (ICISS)*, 2007.

[318] H. Tanaka, O. Takizawa, and A. Yamamura. Evaluation and improvement of the Tempest fonts. In *International Workshop on Information Security Applications (WISA)*, 2005.

[319] A. Tang, S. Sethumadhavan, and S. Stolfo. CLKSCREW: Exposing the perils of security-oblivious energy management. In *USENIX Security Symposium*, 2017.

[320] TechInsights. Apple iPhone 7 teardown. `https://www.techinsights.com/blog/apple-iphone-7-teardown`. Created: 2016-09-15. Accessed: 2019-10-01.

[321] M. Tehranipoor and F. Koushanfar. A survey of Hardware Trojan taxonomy and detection. *IEEE Design & Test of Computers (D&T)*, 27(1):10–25, Jan 2010.

[322] Tencent Cloud. Instance types. `https://intl.cloud.tencent.com/document/product/213/11518`. Created: 2019-09-26. Accessed: 2019-10-01.

[323] F. M. Tesche. Development and use of the BLT equation in the time domain as applied to a coaxial cable. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 49(1):3–11, Feb 2007.

[324] F. M. Tesche, M. V. Ianoz, and T. Karlsson. *EMC Analysis Methods and Computational Models*. Wiley, 1st edition, 1996.

[325] F. M. Tesche, J. M. Keen, and C. M. Butler. Example of the use of the BLT equation for EM field propagation and coupling calculations. *URSI Radio Science Bulletin*, 2005(312):32–47, Mar 2005.

[326] Texas Advanced Computing Center. TACC to launch new Catapult system to researchers worldwide. `https://www.tacc.utexas.edu/-/tacc-to-launch-new-catapult-system-to-researchers-worldwide`. Created: 2015-11-12. Accessed: 2019-10-01.

[327] K. S. Tharayil, B. Farshteindiker, S. Eyal, N. Hasidim, R. Hershkovitz, S. Houri, I. Yoffe (Iofedov), M. Oren, and Y. Oren. Sensor defense in-software (SDI): Practical software based detection of spoofing attacks on position sensors. arXiv 1905.04691, 2019. http://arxiv.org/abs/1905.04691.

[328] The Inquirer. Sonic attacks can bork hard disks and crash Windows and Linux. https://www.theinquirer.net/inquirer/news/3033287/sonic-and-ultrasonic-attacks-can-crash-hard-disks-and-windows-and-linux. Created: 2018-05-31. Accessed: 2019-10-01.

[329] The New York Times. It's possible to hack a phone with sound waves, researchers show. https://www.nytimes.com/2017/03/14/technology/phone-hacking-sound-waves.html. Created: 2017-03-14. Accessed: 2019-10-01.

[330] The Register. Boffins Rickroll smartphone by tickling its accelerometer. https://www.theregister.co.uk/2017/03/15/boffins%5Frickroll%5Fsmartphone%5Fby%5Ftickling%5Fits%5Faccelerometer. Created: 2017-03-15. Accessed: 2019-10-01.

[331] The Wall Street Journal. U.S. downed Iranian drone with new technology. https://www.wsj.com/articles/u-s-downed-iranian-drone-with-new-technology-11563579400. Created: 2019-07-19. Accessed: 2019-10-01.

[332] S. Tian and J. Szefer. Temporal thermal covert channels in cloud FPGAs. In *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*, 2019.

[333] V. Timonen. Multi-GPU CUDA stress test. http://wili.cc/blog/gpu-burn.html. Created: 2016-11-30. Accessed: 2019-10-01.

[334] S. Trimberger and S. McNeil. Security of FPGAs in data centers. In *IEEE International Verification and Security Workshop (IVSW)*, 2017.

[335] S. Trimberger and J. J. Moore. FPGA security: Motivations, features, and applications. *Proceedings of the IEEE (JPROC)*, 102(8):1248–1265, Aug 2014.

[336] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu. WALNUT: Waging doubt on the integrity of MEMS accelerometers with acoustic injection attacks. In *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017.

[337] Y. Tu, Z. Lin, I. Lee, and X. Hei. Injected and delivered: Fabricating implicit control over actuation systems by spoofing inertial sensors. In *USENIX Security Symposium*, 2018.

[338] Y. Tu, S. Rampazzi, B. Hao, A. Rodriguez, K. Fu, and X. Hei. Trick or heat? Manipulating critical temperature-based control systems using rectification attacks. In *ACM Conference on Computer and Communications Security (CCS)*, 2019.

[339] TUL Corporation. Introducing TUL PYNQ-Z2. http://www.tul.com.tw/download/PYNQ-Z2%5FPA%5Fv3.pdf. Created: Unknown. Accessed: 2019-10-01.

[340] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields. Cocaine noodles: Exploiting the gap between human and machine speech recognition. In *USENIX Workshop on Offensive Technologies (WOOT)*, 2015.

[341] A. Vaishnav, K. D. Pham, and D. Koch. A survey on FPGA virtualization. In *International Conference on Field Programmable Logic and Applications (FPL)*, 2018.

[342] M. J. van der Horst, W. A. Serdijn, and A. C. Linnenbank. *EMI-Resilient Amplifier Circuits*. Springer, 1st edition, 2014.

[343] W. van Eck. Electromagnetic radiation from video display units: An eavesdropping risk? *Computers & Security*, 4(4):269–286, Dec 1985.

[344] M. Varchola and M. Drutarovsky. New high entropy element for FPGA based true random number generators. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2010.

[345] G. P. Veropoulos, P. J. Papakanellos, and C. Vlachos. A probabilistic approach for the susceptibility assessment of a straight PCB trace excited by random plane-wave fields. *IEEE Transactions on Electromagnetic Compatibility (TEMC)*, 60(1):258–265, Feb 2018.

[346] C. Walravens, S. van Winchel, J.-M. Redouté, and M. Steyaert. Efficient reduction of electromagnetic interference effects in operational amplifiers. *Electronics Letters*, 43(2):84–85, Jan 2007.

[347] F. Wan, F. Duval, H. Cao, X. Savatier, A. Louis, and B. Mazari. Increase of immunity of microcontroller to conducted continuous-wave interference by detection method. *Electronics Letters*, 46(16):1113–1114, Aug 2010.

[348] F. Wan, F. Duval, X. Savatier, A. Louis, and B. Mazari. Electromagnetic interference detection method to increase the immunity of a microcontroller-based system in a complex electromagnetic environment. *IET Science, Measurement & Technology*, 6(4):254–260, Jul 2012.

[349] Z. Wang, K. Wang, B. Yang, S. Li, and A. Pan. Sonic gun to smart devices: Your devices lose control under ultrasound/sound. Black Hat USA, 2017.

[350] A. P. Waterland. Stress. `https://web.archive.org/web/20190502184531/https://people.seas.harvard.edu/~apw/stress/`. Created: 2014-07-18. Accessed: 2019-10-01.

[351] J. Weerasinghe, F. Abel, C. Hagleitner, and A. Herkersdorf. Enabling FPGAs in hyperscale data centers. In *IEEE International Conference on Ubiquitous Intelligence and Computing, Autonomic and Trusted Computing, Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, 2015.

[352] S. Wendzel, S. Zander, B. Fechner, and C. Herdin. Pattern-based survey and categorization of network covert channel techniques. *ACM Computing Surveys (CSUR)*, 47(3):1–27, Apr 2015.

[353] A. Wild, G. T. Becker, and T. Güneysu. On the problems of realizing reliable and efficient ring oscillator PUFs on FPGAs. In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2016.

[354] A. Wild, G. T. Becker, and T. Güneysu. A fair and comprehensive large-scale analysis of oscillation-based PUFs for FPGAs. In *International Conference on Field Programmable Logic and Applications (FPL)*, 2017.

[355] S. J. E. Wilton. A crosstalk-aware timing-driven router for FPGAs. In *ACM/SIGDA International Symposium Field Programmable Gate Arrays (FPGA)*, 2001.

[356] T. Wolfgramm, A. Manicke, and H. G. Krauthäuser. Field coupling to nonlinear circuits in resonating structures. In *IEEE International Symposium on Electromagnetic Compatibility (EMC)*, 2015.

[357] Z. Wu, Z. Xu, and H. Wang. Whispers in the hyper-space: High-bandwidth and reliable covert channel attacks inside the cloud. *IEEE/ACM Transactions on Networking (TNET)*, 23(2):603–615, Apr 2015.

[358] Xilinx, Inc. 7 series FPGAs data sheet: Overview (DS180). `https://www.xilinx.com/support/documentation/data%5Fsheets/ds180%5F7Series%5FOverview.pdf`. Created: 2018-02-27. Accessed: 2019-10-01.

[359] Xilinx, Inc. 7 series product brief. `https://www.xilinx.com/publications/prod%5Fmktg/7-Series-Product-Brief.pdf`. Created: Unknown. Accessed: 2019-10-01.

[360] Xilinx, Inc. AC701 evaluation board for the Artix-7 FPGA (UG952). https://www.xilinx.com/support/documentation/boards%5Fand%5Fkits/ac701/ug952-ac701-a7-eval-bd.pdf. Created: 2019-08-06. Accessed: 2019-10-01.

[361] Xilinx, Inc. Accelerated computing partners. https://www.xilinx.com/products/boards-and-kits/alveo/where-to-buy.html. Created: Unknown. Accessed: 2019-10-01.

[362] Xilinx, Inc. Getting started with the Xilinx Virtex-6 FPGA ML605 evaluation kit (UG533). https://www.xilinx.com/support/documentation/boards%5Fand%5Fkits/ug533.pdf. Created: 2011-10-21. Accessed: 2019-10-01.

[363] Xilinx, Inc. KC705 evaluation board for the Kintex-7 FPGA (UG810). https://www.xilinx.com/support/documentation/boards%5Fand%5Fkits/kc705/ug810%5FKC705%5FEval%5FBd.pdf. Created: 2019-02-04. Accessed: 2019-10-01.

[364] Xilinx, Inc. Large FPGA methodology guide, including Stacked Silicon Interconnect (SSI) technology (UG872). https://www.xilinx.com/support/documentation/sw%5Fmanuals/xilinx14%5F7/ug872%5Flargefpga.pdf. Created: 2012-10-16. Accessed: 2019-10-01.

[365] Xilinx, Inc. UltraScale architecture and product data sheet: Overview (DS890). https://www.xilinx.com/support/documentation/data%5Fsheets/ds890-ultrascale-overview.pdf. Created: 2019-08-21. Accessed: 2019-10-01.

[366] Xilinx, Inc. VCU118 evaluation board (UG1224). https://www.xilinx.com/support/documentation/boards%5Fand%5Fkits/vcu118/ug1224-vcu118-eval-bd.pdf. Created: 2018-10-17. Accessed: 2019-10-01.

[367] Xilinx, Inc. Virtex-5 family overview (DS100). https://www.xilinx.com/support/documentation/data%5Fsheets/ds100.pdf. Created: 2015-08-21. Accessed: 2019-10-01.

[368] Xilinx, Inc. Virtex-6 family overview (DS150). https://www.xilinx.com/support/documentation/data%5Fsheets/ds150.pdf. Created: 2015-08-20. Accessed: 2019-10-01.

[369] Xilinx, Inc. Xilinx powers Huawei FPGA accelerated cloud server. https://www.xilinx.com/news/press/2017/xilinx-powers-huawei-fpga-accelerated-cloud-server.html. Created: 2017-09-06. Accessed: 2019-10-01.

[370] Xilinx, Inc. Zynq-7000 SoC data sheet: Overview (DS190). https://www.xilinx.com/support/documentation/data%5Fsheets/ds190-Zynq-7000-Overview.pdf. Created: 2018-07-02. Accessed: 2019-10-01.

[371] W. Xiong, A. Schaller, N. A. Anagnostopoulos, M. U. Saleem, S. Gabmeyer, S. Katzenbeisser, and J. Szefer. Run-time accessible DRAM PUFs in commodity devices. In *Conference on Cryptographic Hardware and Embedded Systems (CHES)*, 2016.

[372] W. Xu, C. Yan, W. Jia, X. Ji, and J. Liu. Analyzing and enhancing the security of ultrasonic sensors for autonomous vehicles. *IEEE Internet of Things Journal (IoT-J)*, 5(6):5015–5029, Dec 2018.

[373] Y. Xu, J.-M. Frahm, and F. Monrose. Watching the watchers: Automatically inferring TV content from outdoor light effusions. In *ACM Conference on Computer and Communications Security (CCS)*, 2014.

[374] C. Yan, K. Fu, and W. Xu. On Cuba, diplomats, ultrasound, and intermodulation distortion. *Computers in Biology and Medicine*, 104:250–266, Jan 2019.

[375] C. Yan, W. Xu, and J. Liu. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. DEF CON, 2016.

[376] C. Yan, G. Zhang, X. Ji, T. Zhang, T. Zhang, and W. Xu. The feasibility of injecting inaudible voice commands to voice assistants. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, Mar 2019.

[377] S. Yazdanshenas and V. Betz. Interconnect solutions for virtualized field-programmable gate arrays. *IEEE Access*, 6:10497–10507, Feb 2018.

[378] B. Yuce, P. Schaumont, and M. Witteman. Fault attacks on secure embedded software: Threats, design, and evaluation. *Journal of Hardware and Systems Security (HASS)*, 2(2):111–130, Jun 2018.

[379] K. C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, and Y. Yang. All your GPS are belong to us: Towards stealthy manipulation of road navigation systems. In *USENIX Security Symposium*, 2018.

[380] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu. DolphinAttack: Inaudible voice commands. In *ACM Conference on Computer and Communications Security (CCS)*, 2017.

[381] J. Zhang and G. Qu. Recent attacks and defenses on FPGA-based systems. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 12(3):1–24, Sep 2019.

[382] J. Zhang, G. Qu, Y.-Q. Lv, and Q. Zhou. A survey on silicon PUFs and recent advances in ring oscillator PUFs. *Journal of Computer Science and Technology*, 29(4):664–678, Jul 2014.

[383] L. Zhang, C. Bo, J. Hou, X.-Y. Li, Y. Wang, K. Liu, and Y. Liu. Kaleido: You can watch it but cannot record it. In *International Conference on Mobile Computing and Networking (MobiCom)*, 2015.

[384] X. Zhang and M. Tehranipoor. RON: An on-chip ring oscillator network for Hardware Trojan detection. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2011.

[385] X. Zhang and M. Tehranipoor. Design of on-chip lightweight sensors for effective detection of recycled ICs. *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*, 22(5):1016–1029, May 2014.

[386] Y. Zhang and K. B. Rasmussen. Detection of electromagnetic interference attacks on sensor systems. In *IEEE Symposium on Security and Privacy (S&P)*, 2020.

[387] M. Zhao and G. E. Suh. FPGA-based remote power side-channel attacks. In *IEEE Symposium on Security and Privacy (S&P)*, 2018.

[388] S. Zhu, C. Zhang, and X. Zhang. Automating visual privacy protection using a smart LED. In *International Conference on Mobile Computing and Networking (MobiCom)*, 2017.

[389] K. M. Zick and J. P. Hayes. Low-cost sensing with ring oscillator arrays for healthier reconfigurable systems. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 5(1):1–26, Mar 2012.

[390] K. M. Zick, M. Srivastav, W. Zhang, and M. French. Sensing nanosecond-scale voltage attacks and natural transients in FPGAs. In *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*, 2013.

[391] L. Zussa, A. Dehbaoui, K. Tobich, J.-M. Dutertre, P. Maurine, L. Guillaume-Sage, J. Clédière, and A. Tria. Efficiency of a glitch detector against electromagnetic fault injection. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2014.

Appendices

The greatest value of a picture is when it forces us to notice what we never expected to see.

— John Wilder Tukey

A Additional Experiments with ADCs

Contents

This appendix contains more measurements on the demodulation properties of Analog-to-Digital Converters (ADCs), extending the results of Chapter 4. Section A.1 precisely defines the similarity metric mentioned in Chapter 4, and validates the experimental setup. Section A.2 and A.3 then conduct further characterization experiments of the smartphone microphone and ATmega328P ADC respectively. Finally, Section A.4 contains additional examples of the demodulation characteristics of the remaining ADCs.

A.1 Similarity Metric and Setup Validation

The experiments of Section 4.3.2 required an independent metric to evaluate how "similar" two signals are as a way of further validating the security definitions of Chapter 4. The metric proposed for this task is based on the Pearson Correlation Coefficient (PCC),

which is commonly found in signal-alignment and optimization applications [36, 79]. It is defined as the covariance of two variables divided by the product of their standard deviations:

$$\rho(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}} \tag{A.1}$$

It is a suitable metric because it removes the mean value of the signals (Direct Current (DC) shift), as well as the effects of scaling (related to transmission power). In other words, $\rho(X, aX+b) = 1$ for a variable X and scalars a, b. However, PCC is sensitive to signal alignment. To overcome this issue, the phase (time) offset between two signals s_a, s_b can be found using cross-correlation. Specifically, the signals are aligned when the cross-correlation coefficient is maximized:

$$\operatorname{lag}(s_a, s_b) = \arg\max_n \left((s_a \star s_b)[n]\right) \tag{A.2}$$

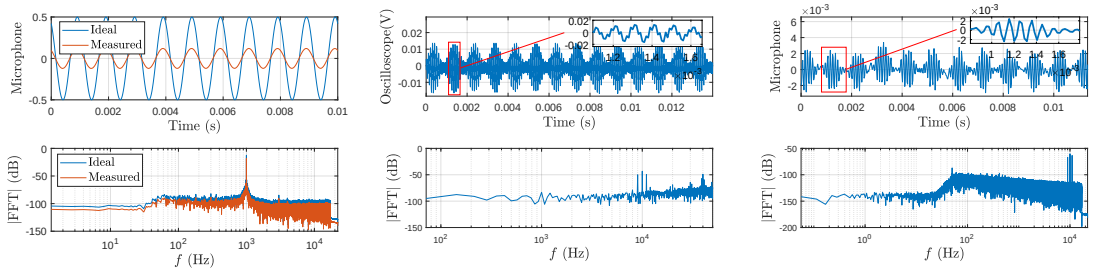
Using Equations (A.1) and (A.2), the similarity metric between the measured signal $\tilde{s}_f(t)$ and the ideal signal $w(t)$ can be defined as follows:

$$\text{similarity}\left(\tilde{s}_f, w\right) = \rho\left(\tilde{s}_f, w^{lag}\right) \tag{A.3}$$

To sanity-check this metric and the experimental setup, an unmodulated 20 mV $f_m = 1$ kHz signal is generated. Figure A.1a shows this waveform as measured by the smartphone of Section 4.3.2 along with an ideal 1 kHz signal. Even though the amplitudes are different, the frequency responses of the measured and the ideal signal are almost identical, with the two signals having a similarity of 0.9991 according to Equation (A.3).

Figures A.1b and A.1c additionally show the same $f_m = 1$ kHz signal modulated on a carrier frequency of $f_c = 10$ kHz at a depth of $\mu = 1.0$. This carrier frequency was chosen as it is within the Nyquist range of the smartphone ADC (sampling frequency $f_s = 44.1$ kHz). Figure A.1b contains measurements taken by a Rigol DS2302A oscilloscope with a timescale division of 500 µs, while Figure A.1c uses the smartphone microphone.

Unlike the examples of Chapter 4, the measurements shown in Figure A.1 do not exhibit harmonics, but rather high-frequency components at f_c and $f_c \pm f_m$, as expected. Consequently, the demodulation characteristics are due to non-linearities in amplifiers and ADCs when used outside of their intended range, instead of the experimental setup.



(a) f_m (Ideal & Microphone) **(b)** $f_m \times f_c$ (Oscilloscope) **(c)** $f_m \times f_c$ (Microphone)

Figure A.1: Measurements for unmodulated and modulated injections with an oscilloscope and a smartphone microphone, with Root-Mean-Square (RMS) voltage $V_{RMS}^{Adv} = 20\,\text{mV}$, signal frequency $f_m = 1\,\text{kHz}$, carrier frequency $f_c = 10\,\text{kHz}$, and modulation depth $\mu = 1.0$.

A.2 Smartphone Microphone Properties

This section characterizes the smartphone microphone through the direct injection methodology of Chapter 4. An $f_m = 1\,\text{kHz}$ tone is amplitude-modulated with a depth of $\mu = 1.0$ on the following carrier frequencies f_c: 25 MHz, 50 MHz, and $0.1 - 2.4\,\text{GHz}$ at a step of 100 MHz. The Root-Mean-Square (RMS) output level $V_{RMS}^{Adv} = V_{PK}^{Adv}/\sqrt{2}$ of the signal generator is also varied between $0.2 - 0.9\,\text{V}$ at a step of 100 mV.

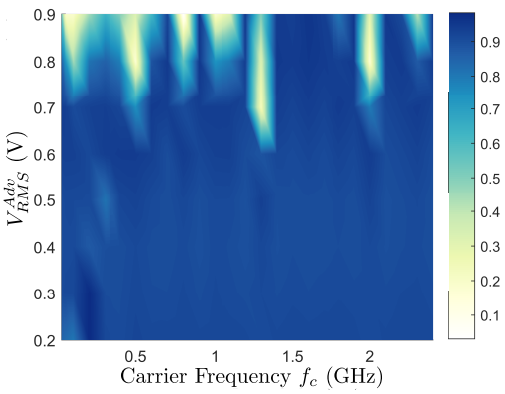
The similarity of the measured to the ideal signal for various carrier frequencies f_c and output levels V_{RMS}^{Adv} is shown in Figure A.2a. It is consistently high for all frequencies when $0.2\,\text{V} \leq V_{RMS}^{Adv} \leq 0.6\,\text{V}$, while higher voltage levels lead to more pronounced harmonics and clipping, reducing the similarity. The results are consistent across measurements: the 99% confidence interval of the similarity is always below ±0.0005, except for the (f_c, V_{RMS}^{Adv}) pairs $(300\,\text{MHz}, 0.5\,\text{V})$ and $(2.4\,\text{GHz}, 0.9\,\text{V})$, where it reaches 0.035.

According to work by Sutu and Whalen on amplifier properties [315], the measured RMS voltage level V_{RMS}, the input voltage V_{RMS}^{Adv}, the modulation depth μ, the signal frequency f_m, and the carrier frequency f_c satisfy the following relationship:

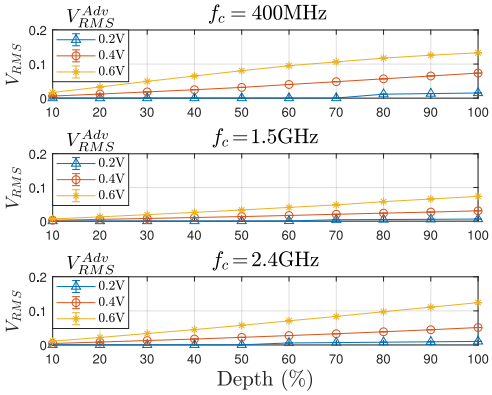
$$V_{RMS} = \frac{\sqrt{2}}{2}\mu\left(V_{RMS}^{Adv}\right)^2 |H_2(f_c, -(f_c - f_m))| \tag{A.4}$$

where H_2 is a second-order transfer function. Figure A.2 mostly confirms this equation for the microphone and ADC subsystem of the smartphone.

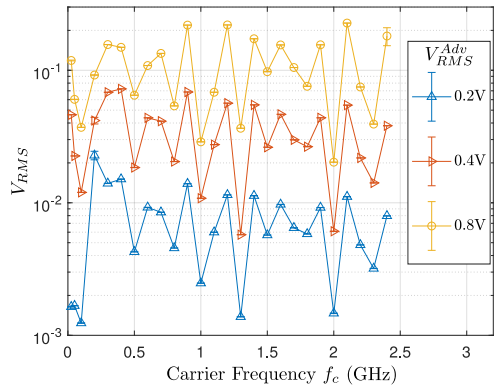
Specifically, fixing μ and f_m in Equation (A.4) suggests that $V_{RMS}^{V_1}/V_{RMS}^{V_2} = (V_1/V_2)^2$ across all carrier frequencies f_c. Indeed, Figure A.2b verifies that the received RMS



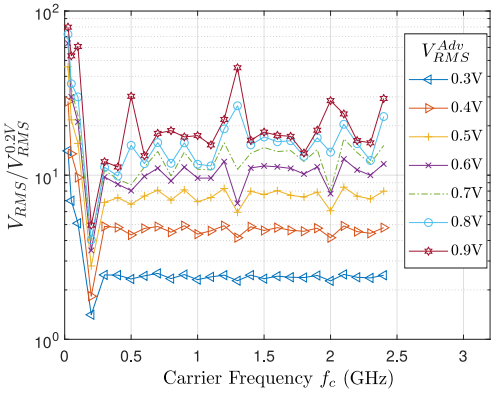
(a) Similarity to the ideal signal



(b) V_{RMS} for different μ



(c) V_{RMS} for different V_{RMS}^{Adv}



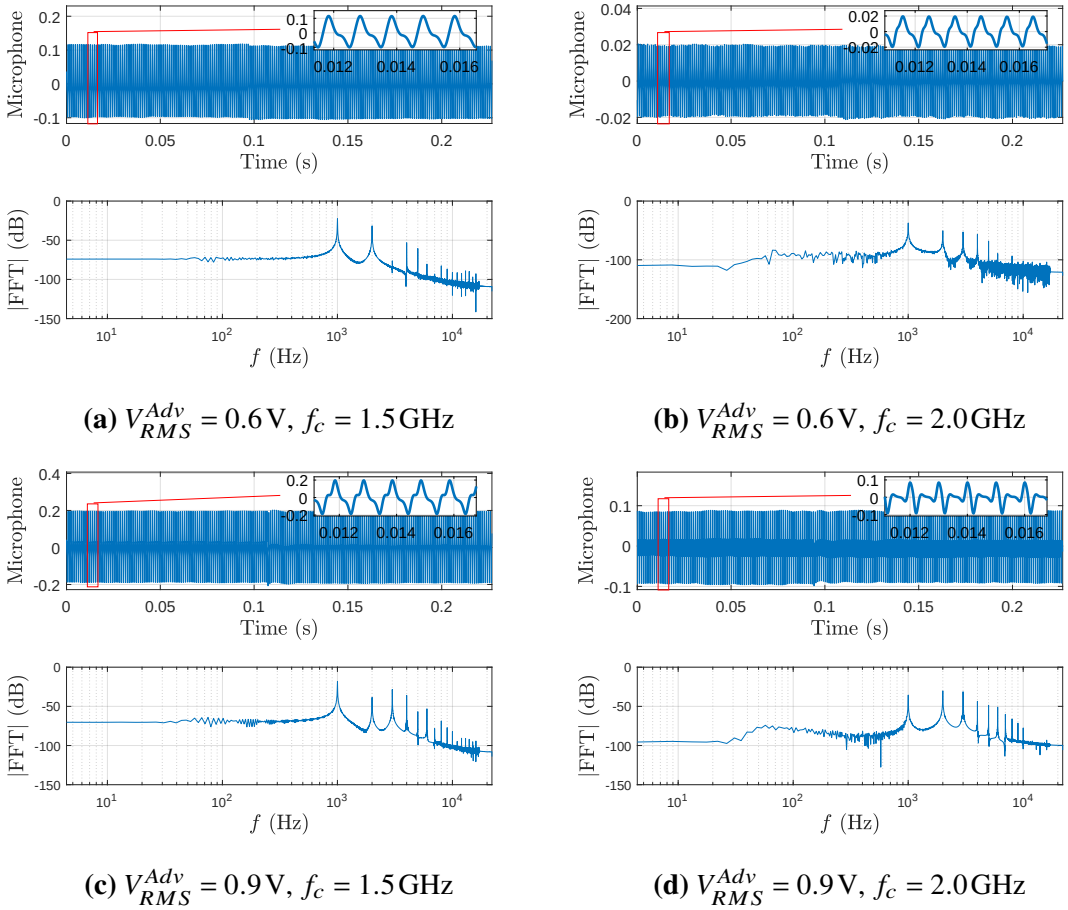
(d) V_{RMS} relative to $V_{RMS}^{Adv} = 0.2\,\mathrm{V}$

Figure A.2: Results of amplitude-modulated $f_m = 1\,\mathrm{kHz}$ injections into the smartphone microphone for different carrier frequencies f_c, depths μ, and output voltage levels V_{RMS}^{Adv}. (a) shows the similarity of the measured output compared to the ideal signal. (b) and (c) illustrate the received Root-Mean-Square (RMS) voltages V_{RMS} for different modulation depths μ and V_{RMS}^{Adv} respectively. Finally, (d) shows V_{RMS} relative to $V_{RMS}^{Adv} = 0.2\,\mathrm{V}$.

voltage V_{RMS} is linear in the modulation depth μ for fixed $V_{RMS}^{Adv} \in \{0.4, 0.6\}\mathrm{V}$ with $R^2 > 0.97$. For 0.2 V, the relationship becomes linear after $\mu \approx 0.8$, as the measured V_{RMS} is approximately 0 below that.

Figure A.2c illustrates that the transfer function for the V_{RMS} response is frequency-dependent. Finally, Figure A.2d shows V_{RMS} relative to $V_{RMS}^{Adv} = 0.2\,\mathrm{V}$. For $0.3\,\mathrm{V} \leq V_{RMS} \leq 0.6\,\mathrm{V}$ and $f_c \geq 100\,\mathrm{MHz}$, there is a constant relationship between the carrier frequency and $V_{RMS}/V_{RMS}^{0.2\mathrm{V}}$, as predicted by Equation (A.4).

Figure A.3 shows example microphone outputs for different carrier frequencies f_c and output voltages V_{RMS}^{Adv}. They all exhibit high harmonics, but the similarity compared to the ideal signal for Figures A.3a–A.3c is still over 0.9. However, the injection of Figure A.3d



(a) $V_{RMS}^{Adv} = 0.6\,\text{V}$, $f_c = 1.5\,\text{GHz}$

(b) $V_{RMS}^{Adv} = 0.6\,\text{V}$, $f_c = 2.0\,\text{GHz}$

(c) $V_{RMS}^{Adv} = 0.9\,\text{V}$, $f_c = 1.5\,\text{GHz}$

(d) $V_{RMS}^{Adv} = 0.9\,\text{V}$, $f_c = 2.0\,\text{GHz}$

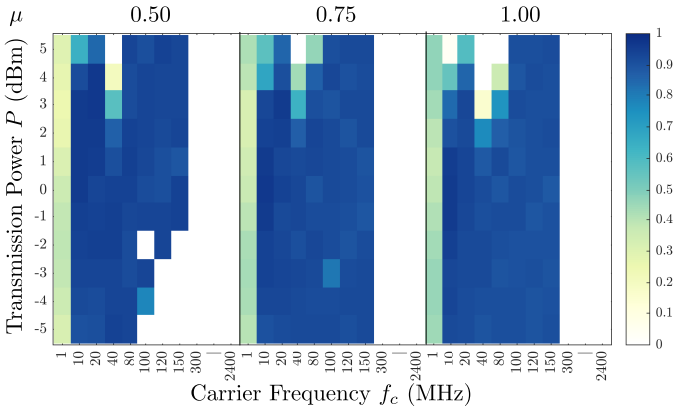
Figure A.3: Microphone output for signal frequency $f_m = 1\,\text{kHz}$ and modulation depth $\mu = 1.0$.

contains more pronounced distortions, and the similarity drops to less than 0.3.

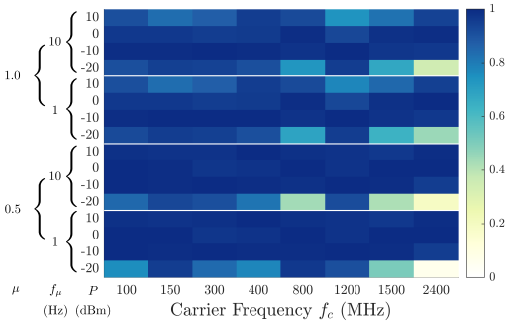
Overall, the results of this section show that, for ADC injections, a higher-order transfer function may be needed to more accurately predict the ADC output, both in terms of its RMS voltage, and in terms of the harmonics it produces.

A.3 ATmega328P Characterization

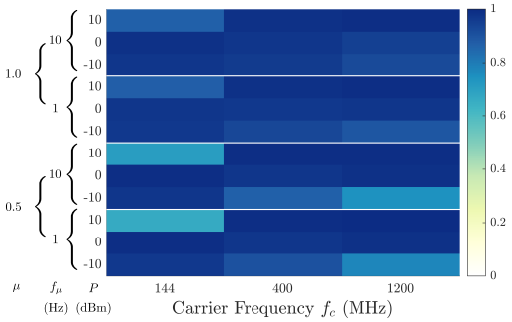
This section contains detailed results for injections into the ATmega328P ADC in three different arrangements: (a) the ADC on its own; (b) the ADC with an amplifier; and (c) the ADC with an amplifier and an antenna. The experimental results support the theoretical model of Chapter 4 and show that, despite its low-pass filtering behavior, the ADC demodulates signals carried at frequencies multiple times the cut-off frequency of



(a) ADC only



(b) ADC + Amplifier



(c) ADC + Amplifier + Antenna

Figure A.4: Similarity metric for injections into the ATmega328P for different transmission powers P, modulation depths μ, and carrier frequencies f_c. The amplifier increases the vulnerable frequencies to the GHz range, allowing remote attacks.

the sample-and-hold mechanism. Moreover, external amplifier non-linearities increase the vulnerable frequency band into the GHz range.

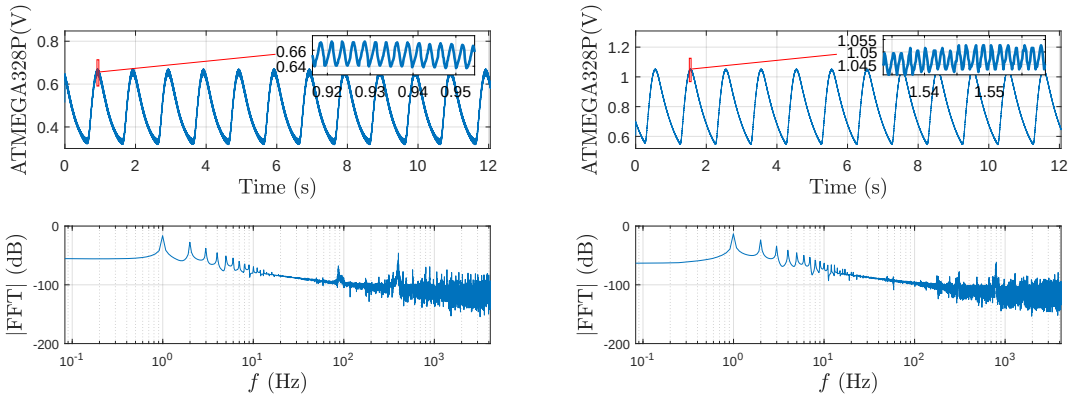
ATmega328P Only: The first experiment targets the ATmega328P directly, without using any additional components. The similarity of the demodulated signal to the ideal signal is calculated for injections at different powers P, modulation depths μ, carrier frequencies f_c, and a signal frequency of $f_m = 1\,\text{Hz}$. As Figure A.4a illustrates, the similarity for $f_c = 1\,\text{MHz}$ is always low due to aliasing. However, similarity is high for f_c between $10 - 150\,\text{MHz}$, but signals are severely attenuated for $f_c \geq 300\,\text{MHz}$. Small modulation depths and powers do not result in demodulated outputs, while too much power causes the ADC to be saturated. This leads to partial clipping of the signal, or induces a DC offset which is beyond the range of the ADC. Overall, the adversary has a range of choices for

P and μ, and can use carrier frequencies which are multiple times the cutoff frequency of the ADC, provided these are not attenuated by the circuit-specific transfer function H_C.

ATmega328P + Amplifier: A low-cost, off-the-shelf wideband Low-Noise Amplifier (LNA) is added before the input to the ADC to change the transfer function H_A. The amplifier works between $1 - 2{,}000\,\text{MHz}$ and can perform a maximum amplification of 32 dB. It can output at most 10 dBm, and has a noise figure of approximately 2 dB. Sinusoidals of $f_m = 1\,\text{Hz}$ and $f_m = 10\,\text{Hz}$ are modulated at depths of $\mu \in \{0.5, 1.0\}$ on carrier frequencies f_c from 100 MHz to 2.4 GHz at transmission powers P between $-20\,\text{dBm}$ and 10 dBm. As can be seen in Figure A.4b, the similarity is high across all frequencies, provided the transmission power is above a minimum threshold.

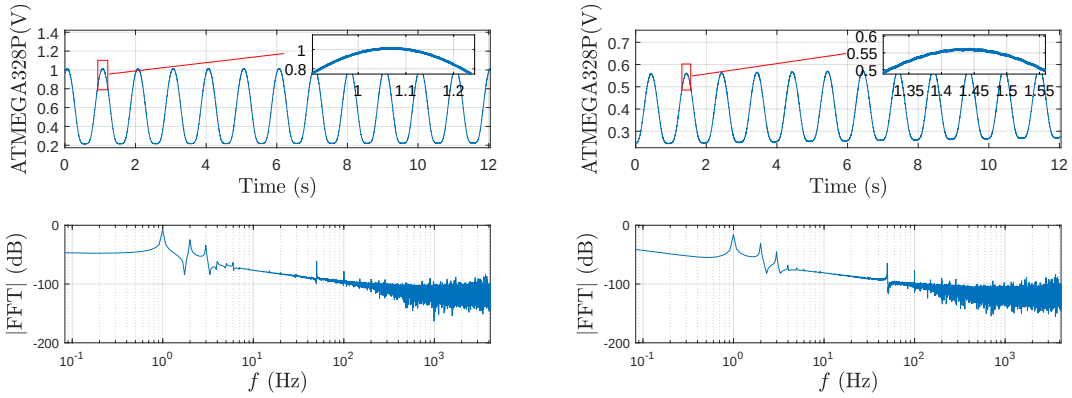
The amplifier thus both reduces the power requirements for the adversary, and increases the vulnerable frequencies to the GHz range. This allows an attacker to target systems with short wires between the ADC and the sensor with a lower power budget: short wires are not a sufficient defense against electromagnetic out-of-band signal injection attacks. Moreover, it should be noted that an adversary gains an advantage by not obeying the amplifier constraints: abusing the amplifier by transmitting higher-powered signals or by driving frequency signals outside of the intended range still results in recognizable output. In other words, although the signal may be distorted, non-linearities produce outputs within the range of the ADC.

ATmega328P + Amplifier + Antenna: The final set of experiments changes the circuit-specific transfer function H_C by using a transmitting antenna at the signal generator output and a receiving antenna at the amplifier input. The antennas used are Ettus Research omnidirectional VERT400 antennas, which are resonant at 144 MHz, 400 MHz, and 1.2 GHz. The antennas are placed in parallel at a distance of 5 cm to one another, and the results for sine signals of $f_m = 1\,\text{Hz}$ and $f_m = 10\,\text{Hz}$ are presented in Figure A.4c. Although the minimum power required for successful injections is higher due to transmission losses, the system remains vulnerable for all three frequencies due to the amplifier and ADC non-linearities. In other words, results are reproducible across multiple setups, whether through remote transmissions, or through direct injections with an identity transfer function $H_C(\sigma + j\omega) = \mathcal{L}\{\tilde{v}(t)\}/\mathcal{L}\{v(t)\} = 1$.



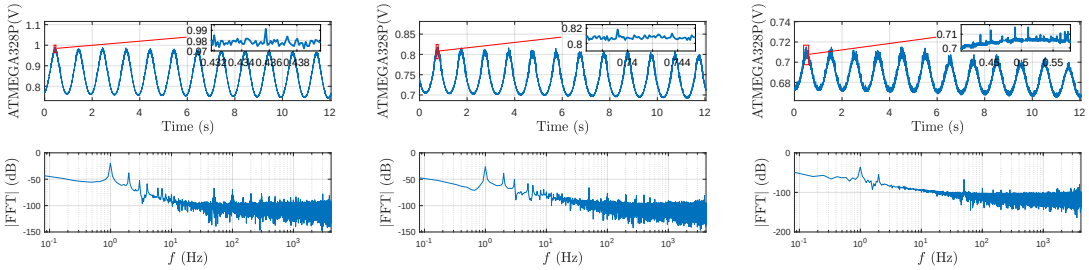
(a) $f_c = 20\,\text{MHz}$ **(b)** $f_c = 40\,\text{MHz}$

Figure A.5: ATmega328P-only output for power $P = 0\,\text{dBm}$, signal frequency $f_m = 1\,\text{Hz}$, and modulation depth $\mu = 0.5$.



(a) $f_c = 1.5\,\text{GHz}$ **(b)** $f_c = 2.4\,\text{GHz}$

Figure A.6: ATmega328P with amplifier output for power $P = -5\,\text{dBm}$, signal frequency $f_m = 1\,\text{Hz}$, and modulation depth $\mu = 1.0$.



(a) $f_c = 144\,\text{MHz}$ **(b)** $f_c = 400\,\text{MHz}$ **(c)** $f_c = 1.2\,\text{GHz}$

Figure A.7: ATmega328P with amplifier and antenna output for power $P = 0\,\text{dBm}$, signal frequency $f_m = 1\,\text{Hz}$, and modulation depth $\mu = 0.5$.

Figures A.5–A.7 show example outputs from the internal ATmega328P ADC for different carrier frequencies f_c, powers P, and modulation depths μ, with f_m fixed at 1 Hz.

ADC	Manufacturer	Eff. f_s	R	C
TLC549	Texas Instruments	29 kHz	1 kΩ	60 pF
ATmega328P	Atmel	8.3 kHz	1-100 kΩ	14 pF
Artix 7	Xilinx	198 kHz	10 kΩ	3 pF
AD7276	Analog Devices	35 kHz	75 Ω	32 pF
AD7783	Analog Devices	19.71 Hz	N/A	N/A
AD7822	Analog Devices	84 kHz	310 Ω	4 pF

Table A.1: Further properties of the Analog-to-Digital Converters (ADCs) used in the experiments. Also see Table 4.4 in Chapter 4.

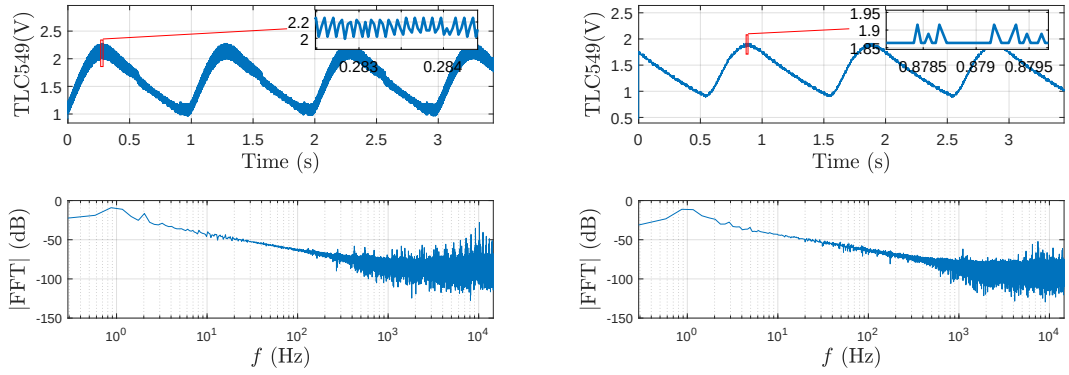
Figure A.5 first shows the results for the ADC on its own, and complements Figure 4.5 of Chapter 4. Although harmonics of the fundamental persist, the high-frequency component becomes less pronounced as f_c increases.

Figure A.6 then shows output from the ATmega328P for two different carrier frequencies f_c when connected to an amplifier. The ADC no longer behaves like a low-pass filter due to non-linearities, while harmonics of the fundamental remain strong. Finally, Figure A.7 shows output from the ATmega328P for remote injections using the VERT400 antenna with the amplifier. As in the amplifier case, carrier frequencies in the GHz range are still demodulated, and harmonics (and some high-frequency components) persist.

A.4 Further ADC Demodulation Examples

This section contains additional examples of injections into the different ADCs, more of whose properties are specified in Table A.1. These properties include the series resistance R and capacitance C of the ADC sample-and-hold circuit as noted in their respective datasheets, which were used to calculate the ADC cutoff frequency (Table 4.4). Although the AD7783 datasheet does not give R, C parameters for its input, it includes notch filters to reject 50 and 60 Hz signals for Alternating Current (AC) mains hum suppression. The ADCs (except for the Artix 7) are controlled with an Arduino, with measurements transferred over the Universal Asynchronous Receiver/Transmitter (UART) interface. As a result, the effective sampling rate is lower than the maximum sampling rate. It is therefore reported separately as "Eff. f_s".

A.4.1 TLC549

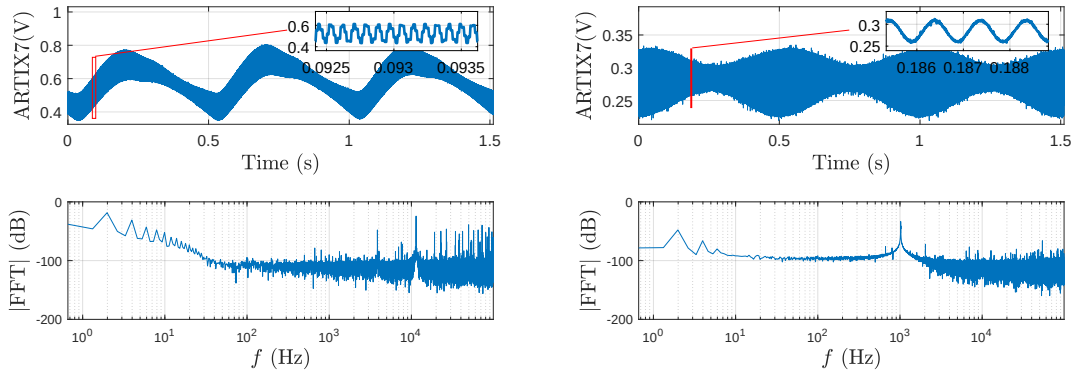


(a) $f_c = 20\,\text{MHz}$ **(b)** $f_c = 120\,\text{MHz}$

Figure A.8: TLC549 output for power $P = 5\,\text{dBm}$, signal frequency $f_m = 1\,\text{Hz}$, and depth $\mu = 0.5$.

Figure A.8 shows example outputs from the TLC549 ADC for two carrier frequencies f_c. Harmonics of the fundamental are not pronounced, as the resolution is only 8 bits.

A.4.2 Artix 7

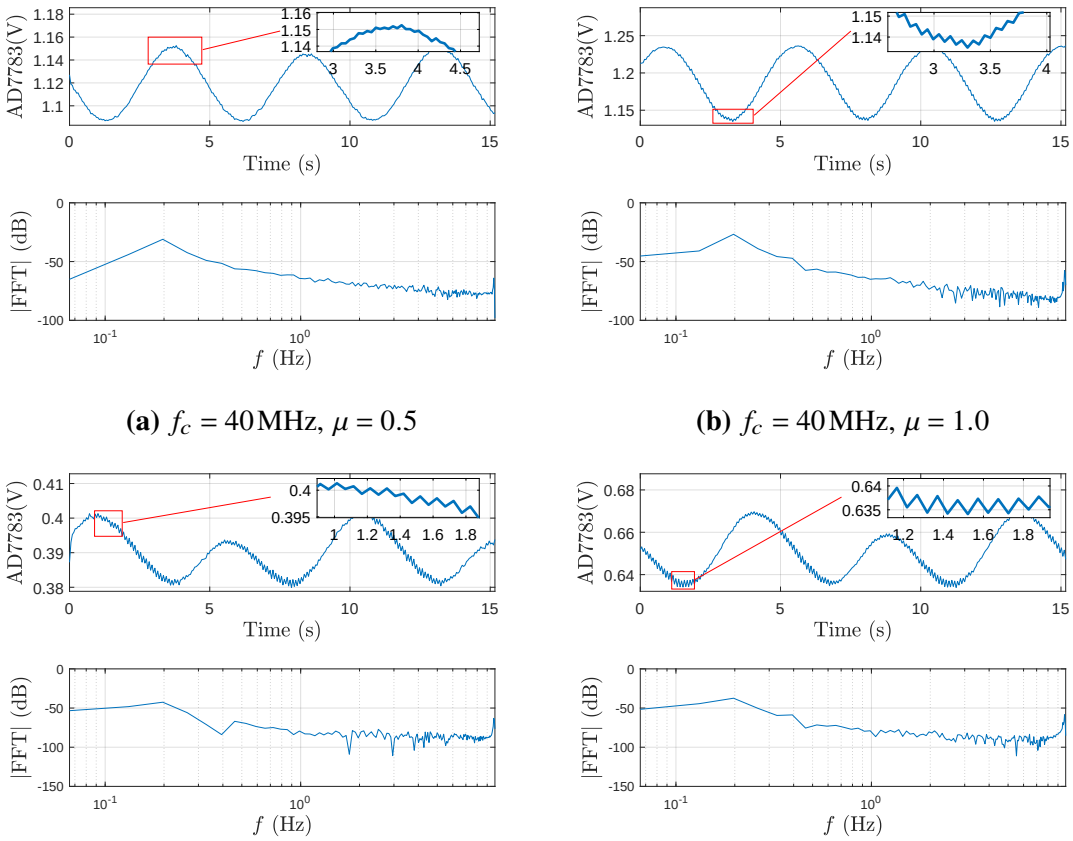


(a) $f_c = 80\,\text{MHz}$ **(b)** $f_c = 100\,\text{MHz}$

Figure A.9: Artix 7 output for power $P = 10\,\text{dBm}$, signal frequency $f_m = 1\,\text{Hz}$, and depth $\mu = 0.5$.

Figure A.9 shows example outputs from the Artix 7 Xilinx Analog-to-Digital Converter (XADC) for two carrier frequencies f_c. The output is more sawtooth-like, and contains high-frequency components which dominate the target signal. As a result, injections require more fine-grained control over the carrier frequency.

A.4.3 AD7783



(a) $f_c = 40\,\text{MHz}$, $\mu = 0.5$

(b) $f_c = 40\,\text{MHz}$, $\mu = 1.0$

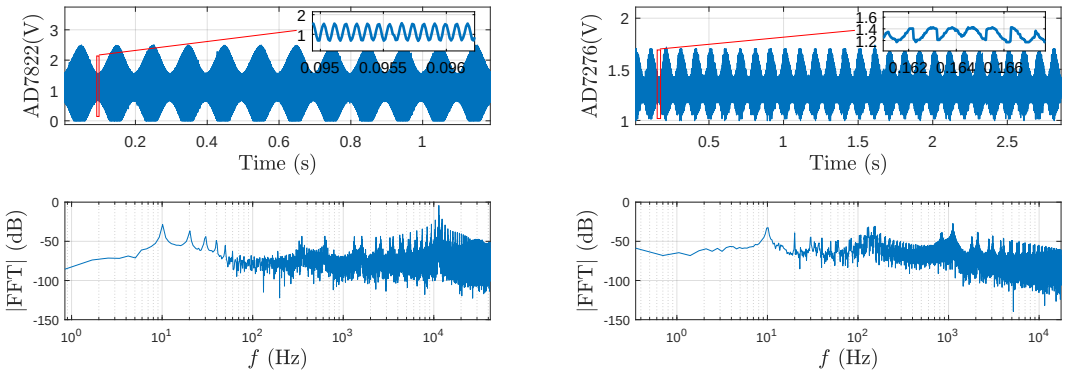
(c) $f_c = 80\,\text{MHz}$, $\mu = 0.5$

(d) $f_c = 80\,\text{MHz}$, $\mu = 1.0$

Figure A.10: AD7783 output for power $P = 5\,\text{dBm}$ and signal frequency $f_m = 10\,\text{Hz}$.

Figure A.10 shows the output from the (slow) AD7783 Delta-Sigma ($\Delta\Sigma$) ADC for different carrier frequencies and modulation depths. As $f_m = 10\,\text{Hz}$ is above the Nyquist frequency, aliasing occurs. The strongest frequency present is $2f_m - f_s = 0.21\,\text{Hz}$, while the high-frequency component is $f_s - f_m = 9.79\,\text{Hz}$.

A.4.4 AD7822 & AD7276



(a) AD7822: $f_c = 10\,\text{MHz}$ **(b)** AD7276: $f_c = 40\,\text{MHz}$

Figure A.11: AD7822 (a) and AD7276 (b) output for $P = -1\,\text{dBm}$, $\mu = 0.5$, and $f_m = 10\,\text{Hz}$.

Figure A.11a and Figure A.11b show example measurements for the AD7822 and the AD7276 ADCs respectively. As for the Artix 7, high-frequency components dominate the output, and hence require manual tuning to get a demodulated, low-frequency output.

And I know the wire. And I know my limits.

— Philippe Petit

B

Medium, Long, and Super-Long Wires

Contents

This appendix complements the experiments of Chapters 5 and 6 on long wires in Field-Programmable Gate Arrays (FPGAs). Specifically, it first explains how to remove an assumption made in Chapter 5 regarding side-channel exfiltration of keys using long-wire leakage (Section B.1). It then presents the effect of varying the measurement time in estimating long-wire leakage in the Virtex UltraScale+ family (Section B.2). It continues by depicting some example measurements of the leakage (Section B.3), and directly comparing the behavior of different device generations (Section B.4). Finally, it summarizes some preliminary experiments regarding leakage on alternative types of routing resources, namely "medium" (Section B.5) and "super-long" (Section B.6) wires.

B.1 Generalized Signal Exfiltration

Section 5.8.2 explained how to conduct side-channel attacks through a windowing approach, reducing the guessing space from 2^N to 2^w possibilities in the worst case. However, it assumed that the key size N is a multiple of the window size w. This section removes this assumption, and further explains how to always fully recover keys by varying the window size. Assume that $N = nw + m$, with $0 \leq m < w$. Then, the probability p_r that the bits in $S_r = \left(K_r, K_{w+r}, K_{2w+r}, \ldots\right)$ are the same is:

$$p_r = \begin{cases} 2^{-n} & \text{for } 0 \leq r < m \\ 2^{-n+1} & \text{for } m \leq r < w \end{cases} \tag{B.1}$$

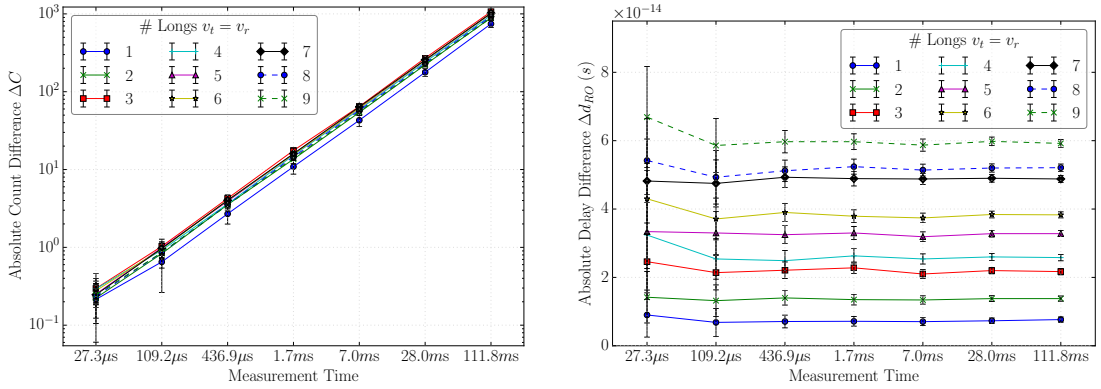
since $|S_r|$ is $n+1$ and n respectively. For $N \geq 2w$, Equation (5.4) becomes:

$$P = \left(1 - \frac{1}{2^n}\right)^m \left(1 - \frac{1}{2^{n-1}}\right)^{w-m} \tag{B.2}$$

The lower bound on N is necessary to recover the first w bits of the key, because in order to have elements in S_r, it must be the case that $r + w < N$ for each r with $0 \leq r \leq w - 1$.

Suppose now that the original measurements could not recover the bits in S_r because they were all identical. By repeating measurements with a window of size $w + 1$, the algorithm either recovers all bits in the sequence $S'_r = \left(K_r, K_{w+1+r}, K_{2(w+1)+r}, \ldots\right)$ or shows that they too are identical. In the first case, the algorithm recovers K_r, and hence S_r since all its bits are identical. If instead all bits in S'_r are also identical, the entire key consists of a single repeated bit (i.e., all ones or all zeros). This is because $K_r = K_{w+1+r} = K_{r+1 \pmod w}$, and $K_r = K_{2(w+1)+r} = K_{r+2 \pmod w}$, etc. (one might have to vary r to cover all the residues $\bmod w$). The key is thus recovered with probability 1 if there are at least two different bits in it, or it is determined that the key consists of all 0s or all 1s. Distinguishing between the last two cases is easy, as the total Ring Oscillator (RO) count is higher when the key consists solely of ones compared to when it consists only of zeros.

As a final note, a window of size w needs $N - w + 1$ measurements in w runs if using one counter (with run r responsible for S_r), or a single run if using w counters in parallel. Thus, using both window sizes, and to fully determine all the bits of a key, one needs to take $2N - 2w + 1$ measurements over just $2w + 1$ runs. In other words, the key only needs to be repeated $2w + 1$ times to be fully leaked even in a single-counter setup.



(a) Absolute Count Difference (ΔC) **(b)** Absolute Delay Difference (Δd_{RO})

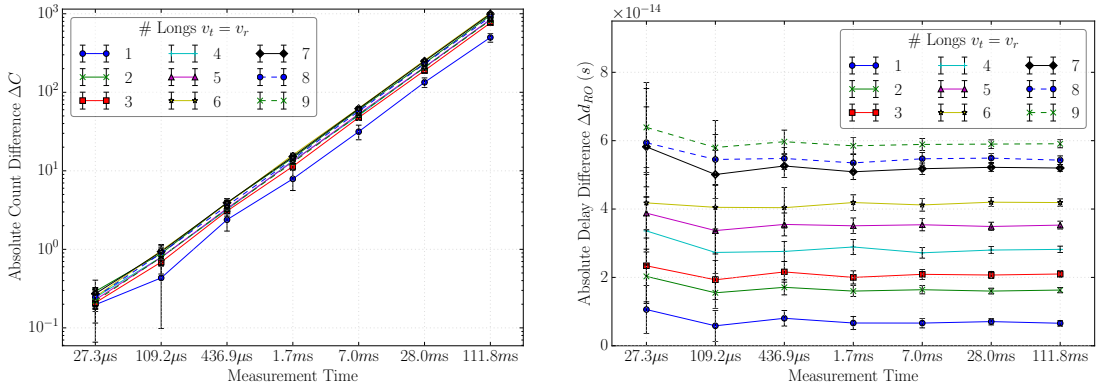
Figure B.1: Absolute Count (a) and Absolute Delay (b) Differences for various measurement times with Ring Oscillators (ROs) using Lookup Tables (LUTs).

B.2 Measurement Time Experiments

Section 5.4.1 demonstrated that on Virtex 5 devices, the Absolute Count Difference (ΔC) grows linearly with increasing measurement time, while the Relative Count Difference (ΔRC) stays approximately the same. However, measurements for shorter time periods are far noisier. This section repeats these experiments on the VCU118 Virtex UltraScale+ device, and shows that these patterns still hold for all three RO types and the new Absolute Delay Difference (Δd_{RO}) metric.

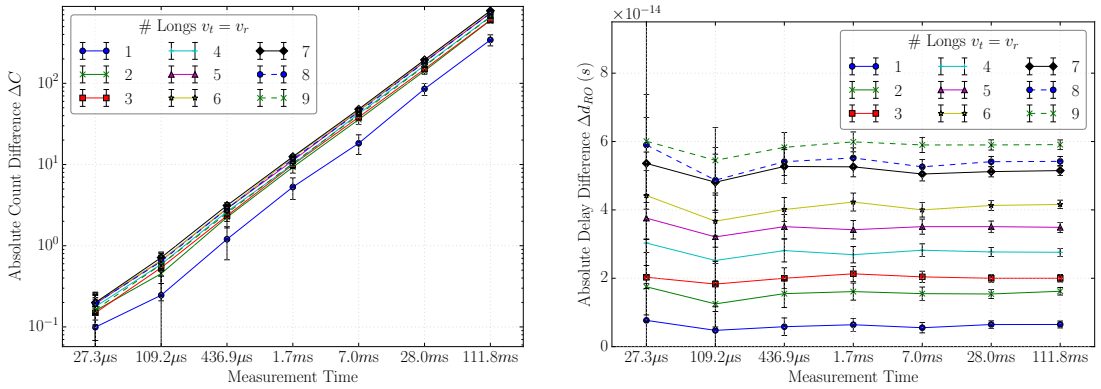
Figure B.1 plots these metrics for ROs using Lookup Tables (LUTs) for all setups where the transmitter and the receiver use an equal number of wires $v = v_t = v_r$. Figure B.1a in particular presents ΔC, with the pattern of linear growth and less noisy results for larger measurement periods remaining the same. However, it is worth highlighting that the differences between different v cannot easily indicate which setup results in more pronounced leakage. Figure B.1b, which depicts Δd_{RO} instead, makes the pattern abundantly clear: larger overlaps result in larger differences in the delay of the RO routing.

For completeness, Figures B.2 and B.3 plot the same metrics for latch-based and register-based ROs respectively. As expected, the patterns are nearly identical, but are somewhat noisier than the LUT-based measurements.



(a) Absolute Count Difference (ΔC) **(b)** Absolute Delay Difference (Δd_{RO})

Figure B.2: Absolute Count (a) and Absolute Delay (b) Differences for various measurement times with Ring Oscillators (ROs) using Latches (LDs).



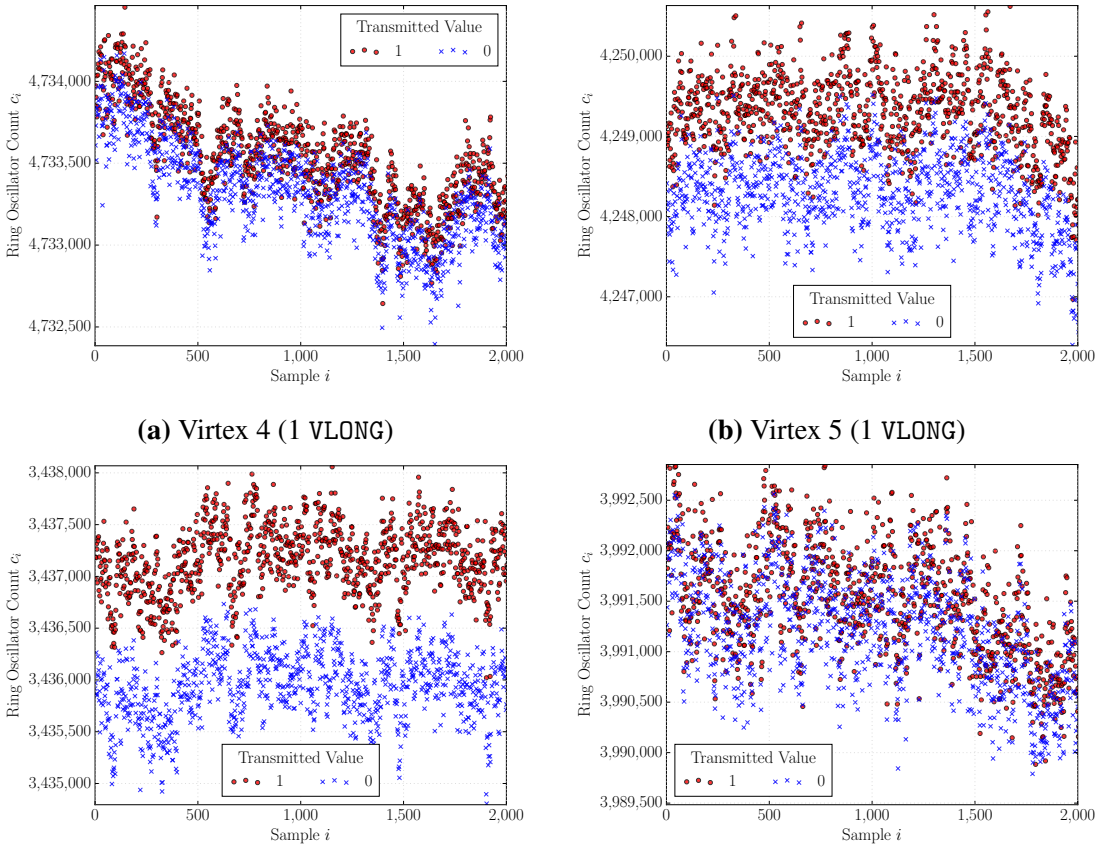
(a) Absolute Count Difference (ΔC) **(b)** Absolute Delay Difference (Δd_{RO})

Figure B.3: Absolute Count (a) and Absolute Delay (b) Differences for various measurement times with Ring Oscillators (ROs) using Flip-Flops (FFs).

B.3 Examples of Long-Wire Leakage

Chapter 5 explained that Manchester encoding is necessary to counteract the effects of environmental fluctuations and noise. Moreover, it demonstrated differences in the strength of the leakage measured through the Relative Count Difference (ΔRC), while Chapter 6 did so using the Absolute Delay Difference (Δd_{RO}). However, it is worth looking at some of the raw data to more clearly see the effects across different device architectures in practice.

Figure B.4 does so for three generations of Virtex FPGAs. In particular, it includes measurements on a Virtex 4 board (Figure B.4a), which was not discussed earlier in this



(a) Virtex 4 (1 `VLONG`)

(b) Virtex 5 (1 `VLONG`)

(c) Virtex 5 (2 `VLONG`s)

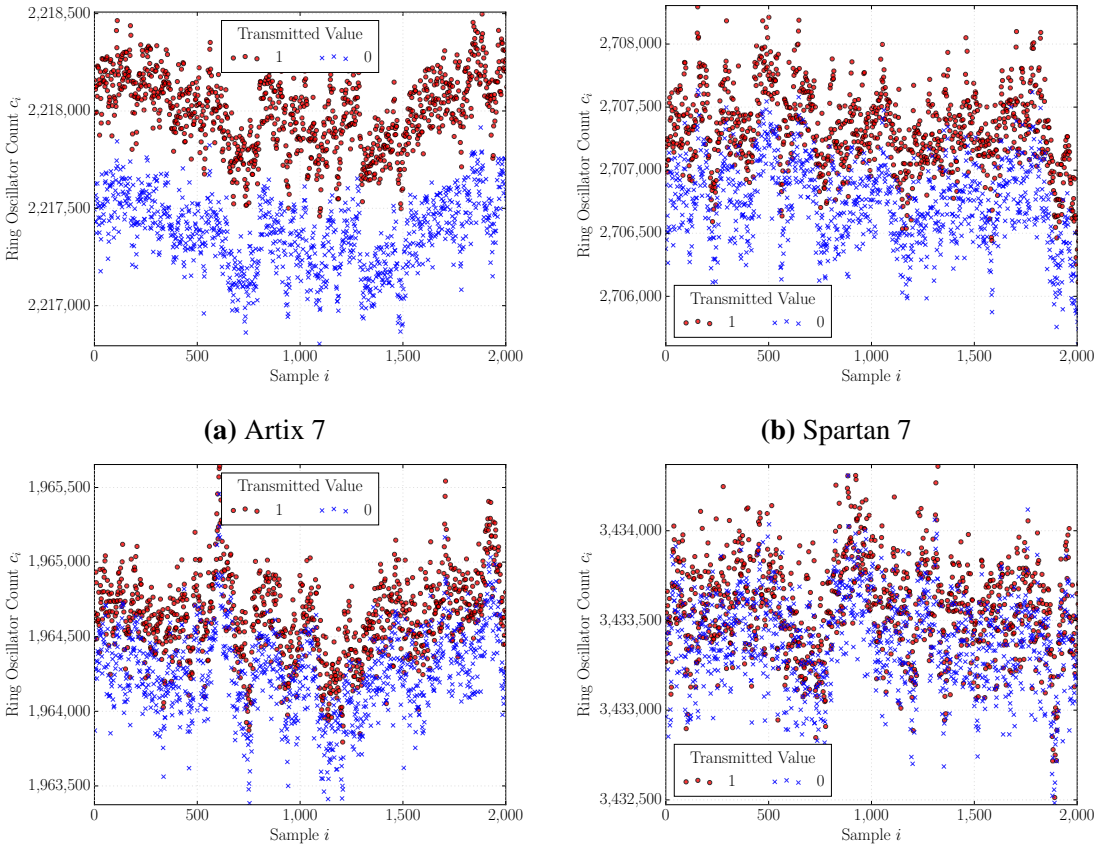
(d) Virtex 6 (4 `VLONG`s)

Figure B.4: Examples of long-wire leakage on (a) a Virtex 4; (b) and (c) a Virtex 5; and (d) a Virtex 6 with different amounts of chained Vertical Longs (`VLONG`s).

thesis, as it is a 90 nm design from 2004 with limited resources. Specifically, it is not tall enough to chain more than a couple of Vertical Longs (`VLONG`s), but even an overlap of one long wire ($v_t = v_r = 1$) leaks information about the long-wire state.

Figures B.4b and B.4c complement Figure 5.2 of Chapter 5, which showed raw measurements for the Virtex 5 family with $v_t = v_r = 5$. Although the leakage is clear even with one `VLONG` (Figure B.4b), it is only starting to become distinguishable with a simple threshold for $v_t = v_r = 2$ (Figure B.4c). On the other hand, leakage on the Virtex 6 (Figure B.4d) is much noisier, even when using four `VLONG`s.

Figure B.5 then presents some example measurements from Series 7 devices, all with $v_t = v_r = 7$. The leakage on the Basys 3 Artix 7 FPGA (Figure B.5a) is most pronounced, but still sensitive to local fluctuations. The behaviors of the Arty S7 Spartan 7 (Figure B.5b) and the PYNQ-Z2 Zynq 7000 (Figure B.5c) are almost identical. Finally, the



(a) Artix 7

(b) Spartan 7

(c) Zynq 7000

(d) Kintex 7

Figure B.5: Examples of long-wire leakage on Series 7 devices with $v_t = v_r = 7$.

measurements on the KC705 Kintex 7 device (Figure B.5d) are much noisier and require averaging over multiple runs (for side-channel attacks), or the use of encoding schemes to remove the effects of temperature and voltage variations (for covert-channel attacks).

The RO counts on the Virtex UltraScale+ (Figure B.6) are almost equally noisy (though still distinguishable locally), even when using $v_t = v_r = 9$ `VLONG`s per transmitter and receiver. This comparison with the Kintex 7 FPGA is perhaps unfair, as the technology node size has shrunk, and the properties of the long wires differ between the two generations, as noted in Table 2.1 of Section 2.2.3. However, some comparisons among different families are useful, and they are therefore the topic of Sections B.4 and B.5.

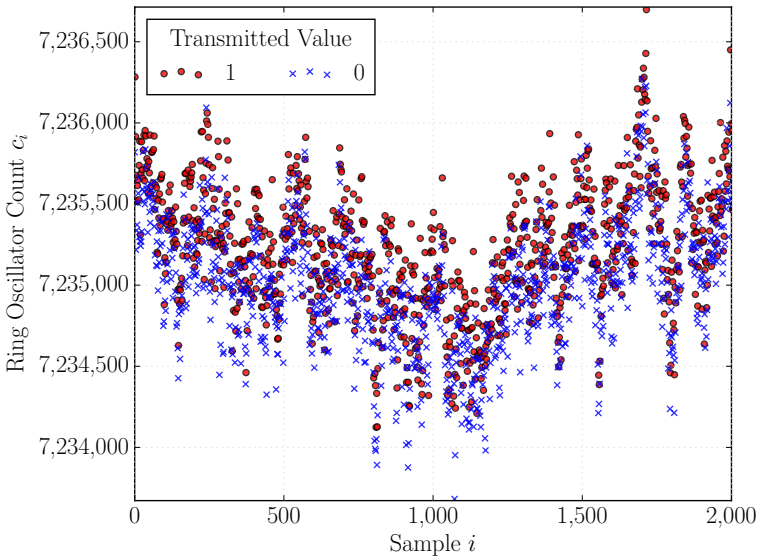


Figure B.6: Example of long-wire leakage on a Virtex UltraScale+ device with $v_t = v_r = 9$.

B.4 Device Comparison

The trends for the long-wire leakage of all individual boards tested with an equal number of transmitter and receiver wires $v_t = v_r$ are shown in Figure B.7 (Virtex UltraScale+ results are instead presented in Section B.5). As Figure B.7 shows, all thirteen boards are susceptible to long-wire information leakage. Moreover, this leakage is most pronounced in the Virtex 5 family, while the Virtex 6 family is least vulnerable. Although most Series 7 devices lie somewhere in-between, specific boards (the Kintex 7 KC705 and one of the Artix 7 Nexys 4 DDR boards) come closer to the Virtex 6 ones. Overall, there is extensive variability even within device families and otherwise identical boards, highlighting that process variations contribute extensively to the strength of the leakage.

B.5 Medium-Wire Leakage

Besides the long wires which span 18 Configurable Logic Blocks (CLBs) and have an intermediate tap (Table 2.1 of Section 2.2.3), Series 7 devices also have wires which span only 12 CLBs, without an intermediate tap. Although Vivado still classifies them as long wires, to avoid confusion this section refers to them as “medium” wires. It should be noted that although these wires more closely resemble the Virtex UltraScale+ long

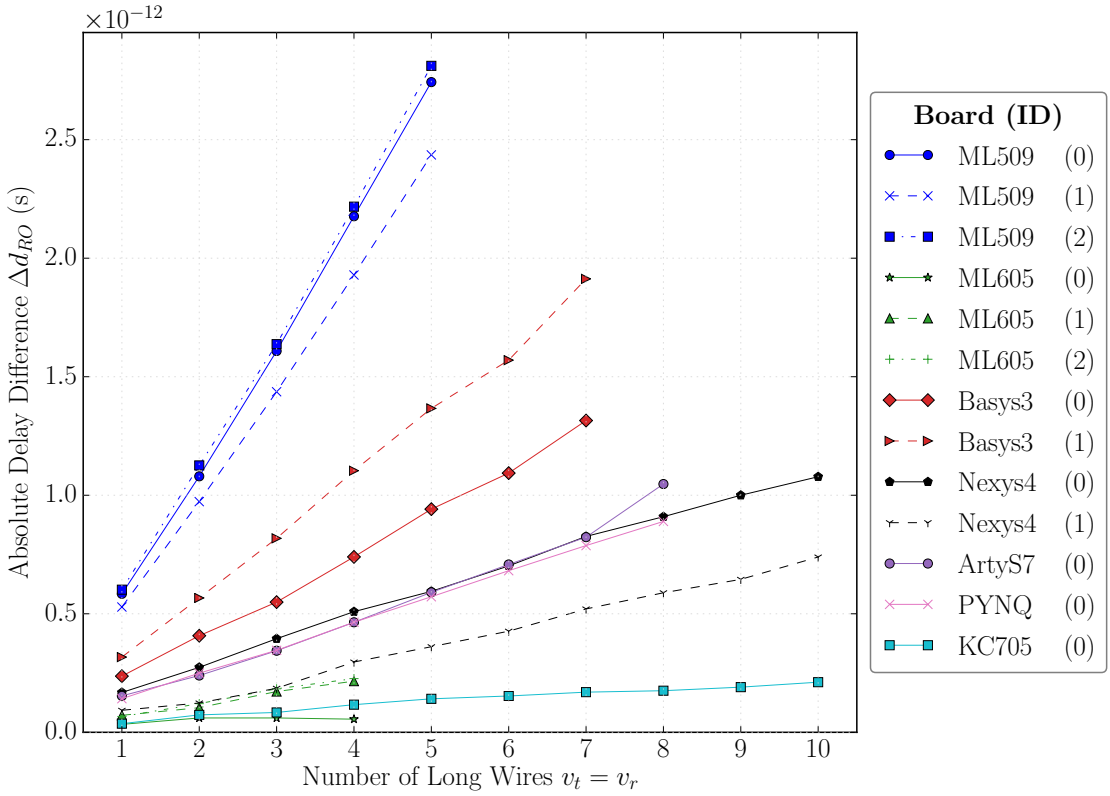


Figure B.7: Absolute Delay Difference (Δd_{RO}) for all boards tested within different Xilinx generations and for different numbers of transmitter and receiver long wires $v_t = v_r$. Results from identical devices are shown in the same color.

wires, they are not identical, as they are still bidirectional, with two medium wires per CLB instead of a channel of eight in each direction.

Figure B.8 shows the trends in medium-wire leakage for the Series 7 boards and some of the Virtex UltraScale+ boards tested, again with an equal number of transmitter and receiver wires $v_t = v_r$. As before, medium wires leak information about their state across all setups, with longer overlaps resulting in larger differences in the delay of the ROs. Perhaps more importantly, the patterns of leakage between devices are consistent whether using long or medium wires. For example, the Basys 3 boards are always leakier than the Arty S7 one. Moreover, medium-wire leakage is less pronounced than long-wire leakage in the same devices, likely not only because of their shorter length, but also because of their lack of intermediate taps.

Much like their Intel counterparts [253], how leakage varies among device generations is not clear for either type of wire, and process variations can also confuse any discernable

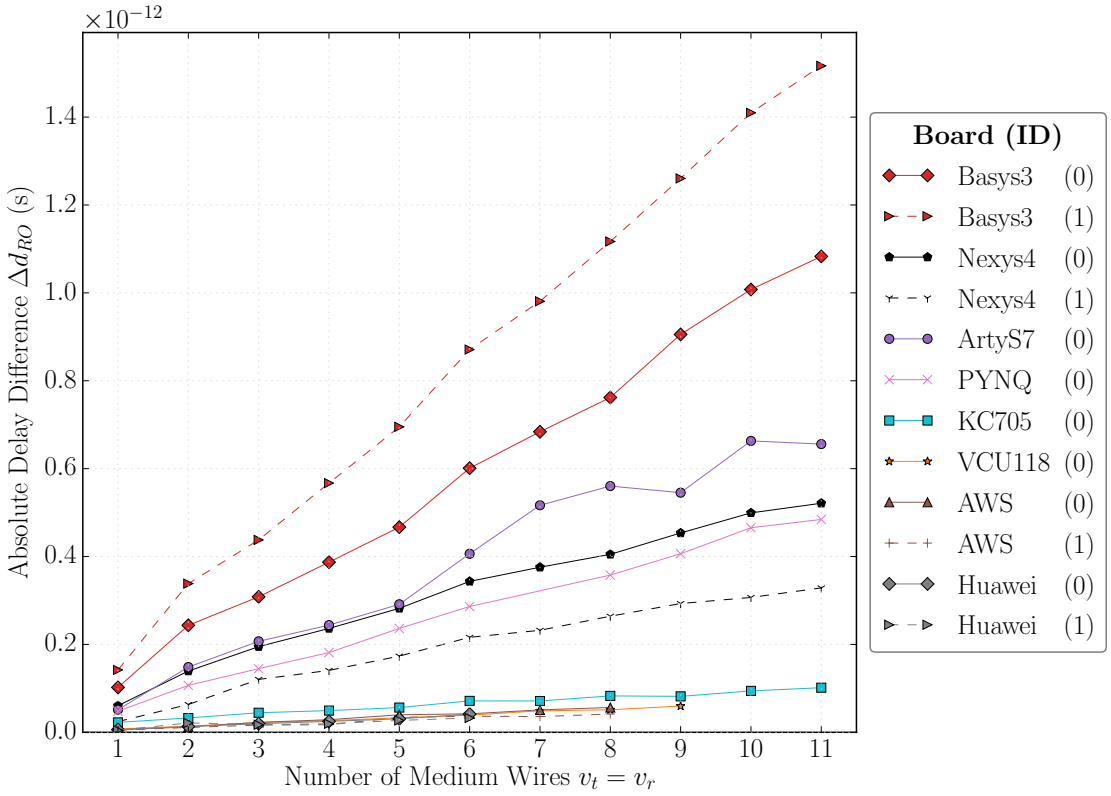


Figure B.8: Absolute Delay Difference (Δd_{RO}) for different numbers of transmitter and receiver medium wires $v_t = v_r$. Results from identical devices are shown in the same color.

Device	Part Number	Logic Cells	LUTs	Reference
ML509	XC5VLX110T-1FFG1136	110,592	69,120	[83, 367]
Basys 3	XC7A35T-1CPG236C	33,280	20,800	[81]
Arty S7	XC7S50-CSGA324	52,160	32,600	[80]
PYNQ-Z2	XC7Z020-1CLG400C	≈ 85,000	53,200	[339, 370]
Nexys 4 DDR	XC7A100T-1CSG324C	101,440	63,400	[82, 358]
ML605	XC6VLX240T-1FFG1156	241,152	150,720	[362, 368]
KC705	XC7K325T-2FFG900C	326,080	203,800	[358, 363]
VCU118	XCVU9P-L2FLGA2104E	2,586,150	1,182,240	[365, 366]

Table B.1: Size-related properties of the various boards tested for medium- and long-wire leakage.

trends. However, at least for devices that came after the Virtex 5, Figures B.7 and B.8 suggest that the size of the FPGA plays a role in the magnitude of the leakage: the board ordering broadly seems to follow the size of the device, shown in Table B.1.

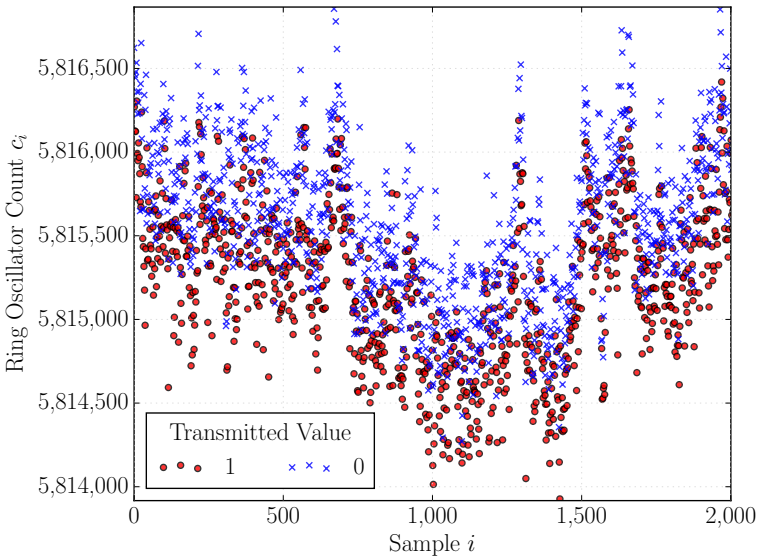


Figure B.9: Example of super-long-wire leakage on a Virtex UltraScale+ device.

B.6 Super-Long-Wire Leakage

As Section 2.2.4 explained in Chapter 2, Virtex UltraScale+ and other high-end Xilinx FPGAs are composed of multiple dies, called Super Logic Regions (SLRs). Super Logic Region (SLR) dies are connected through the silicon interposer, using routing resources called Super Long Lines (SLLs). These tap-less, bidirectional super-long wires span 60 CLBs, and are organized in channels of 24. It is therefore reasonable to wonder whether the values carried by SLLs can also be inferred by nearby wires crossing SLRs.

Some preliminary experiments (Figure B.9) suggest that they can, although the effect seems to be opposite that of regular long wires, with transmissions of a logic 1 resulting in lower RO counts than those of logic 0. However, the routing of a cross-SLR RO is much more complicated, so further experimentation with Time-to-Digital Converters (TDCs) or other structures is necessary to isolate any other unintended interactions. If these results persist across different setups, it might even be possible to exploit the phenomenon and inject timing violations in nearby logic with strict timing constraints.