

Understanding and Countermeasures against IoT Physical Side Channel Leakage

by

Michael Moukarzel

A Dissertation

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE


In partial fulfillment of the requirements for the

Degree of Doctor of Philosophy

in

Electrical and Computer Engineering

by

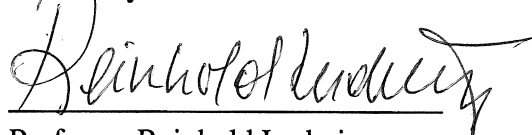


15 May 2019

APPROVED:



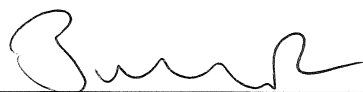
Professor Thomas Eisenbarth
Dissertation Committee
Institute for IT Security
University of Lübeck



Professor Reinhold Ludwig
Head of Department
ECE Department
Worcester Polytechnic Institute



Assistant Professor Matthew Hicks
Dissertation Committee
CS Department
Virginia Tech



Professor Berk Sunar
Dissertation Advisor
ECE Department
Worcester Polytechnic Institute

Abstract

With the proliferation of cheap bulk SSD storage and better batteries in the last few years we are experiencing an explosion in the number of Internet of Things (IoT) devices flooding the market, smartphone connected point-of-sale devices (e.g. Square), home monitoring devices (e.g. NEST), fitness monitoring devices (e.g. Fitbit), and smart-watches. With new IoT devices come new security threats that have yet to be adequately evaluated.

We propose μ Leech, a new embedded trusted platform module for next-generation power scavenging devices. Such power scavenging devices are already widely deployed. For instance, the Square point-of-sale reader uses the microphone/speaker interface of a smartphone for communications and as a power supply. Such devices are being used as trusted devices in security-critical applications, without having been adequately evaluated. μ Leech can securely store keys and provide cryptographic services to any connected smartphone. Our design also facilitates physical side-channel security analysis by providing interfaces to facilitate the acquisition of power traces and clock manipulation attacks. Thus μ Leech empowers security researchers to analyze leakage in next-generation embedded and IoT devices and to evaluate countermeasures before deployment.

Even the most secure systems reveal their secrets through secret-dependent computation. Secret-dependent computation is detectable by monitoring a system's time, power, or outputs. Common defenses to side-channel emanations include adding noise to the channel or making algorithmic changes to mitigate specific side-channels. Unfortunately, existing solutions are not automatic, not comprehensive, or not practical.

We propose an isolation-based approach for eliminating power and timing side-channels that is automatic, comprehensive, and practical. Our approach eliminates side-channels by leveraging integrated decoupling capacitors to electrically isolate trusted computation from the adversary. Software has the ability to request a fixed-power/time quantum of

isolated computation. By discretizing power and time, our approach controls the granularity of side-channel leakage; the only burden on programmers is to ensure that all secret-dependent execution differences converge within a power/time quantum.

We design and implement three approaches to power/time-based quantization and isolation: a wholly-digital version, a hybrid version that uses *capacitors for time tracking*, and a full-custom version. We evaluate the overheads of our proposed controllers with respect to software implementations of AES and RSA running on an ARM-based microcontroller and hardware implementations AES and RSA using a *22nm* process technology. We also validate the effectiveness and real-world efficiency of our approach by building a prototype consisting of an ARM microcontroller, an FPGA, and discrete circuit components.

Lastly, we examine the root cause of Electromagnetic (EM) side-channel attacks on Integrated Circuits (ICs) to augment the Quantized Computing design to mitigate EM leakage. By leveraging the isolation nature of our Quantized Computing design, we can effectively reduce the length and power of the unintended EM antennas created by the wire layers in an IC.

Acknowledgements

I would like to take this opportunity to offer my sincere gratitude to those who have made this work possible and who have provided support and guidance for my efforts. First, Profs. Berk Sunar and Thomas Eisenbarth for their time, effort, and advise that they have given me as my thesis advisers at WPI. The examples that they have set for me will remain influential for the remainders of my professional and personal life.

All my colleagues in the Vernam lab thank you for the friendships, memories, and heated discussions. With a special thanks to my lab mates and friends Yarkin Doroz, Wei Dai, and Gorka Irazoqui. I could not imagine completing this journey without there friendship and support.

Kevin Bush, Associate Group Leader at MIT Lincoln Labs, for his continued support and mentoring during both my internships and graduate studies. This thesis would not have been possible if not for the contacts, funding, and research opportunities afforded during my work with Kevin.

Prof. Matthew Hicks, Assistant Professor at Virginia Tech, for his friendship, advise, and mentoring during my graduate studies. He was instrumental in evolving my thinking process to a higher level. The lessons I have learned as part of his research group will remain influential for the remainder of my professional and personal life. He is a true friend and mentor.

My parents, Anthony and Nadia Moukarzel, and my sisters, Yasmina and Lea and Moukarzel, who have never stopped pushing, encouraging, and motivating me in my personal and academic development. Their continued love and support will aid me in all my future endeavors.

To all of you my sincere gratitude!

Michael Moukarzel

Contents

1	Introduction	1
1.1	Background	2
1.2	Problem Statement	4
1.3	Summary of Contributions	5
1.4	Dissertation Outline	6
2	μLeech: A Side-Channel Evaluation Platform for IoT	8
2.1	Side-Channel Evaluation Platforms	9
2.2	Overview of the Design	11
2.2.1	Power Siphoning	12
2.2.2	Capacitor Bank	14
2.2.3	Sleep Mode	15
2.2.4	Side-Channel Platform	16
2.3	Data Communication	17
2.3.1	Manchester Encoding	19
2.3.2	Transmit	20
2.3.3	Receive	23
2.4	Implementation Results	26
2.4.1	Sleep Mode	26

2.4.2	Active Mode	27
2.4.3	128-bit AES	28
2.5	Conclusion	28

3 Quantized Computing: On-demand Isolation as a Power and Timing Side-channel Defense 29

3.1	Background	32
3.1.1	Decoupling Capacitors	34
3.2	Threat Model	35
3.3	Quantization Controller Design	36
3.3.1	Isolation Controller	36
3.3.2	Timing Controller	37
3.3.3	Power Controller	38
3.3.4	Fault Attacks	39
3.3.5	End-to-end Flow	42
3.3.6	Illustrative Example	43
3.4	Quantization Controller Implementation	44
3.4.1	Wholly Digital Implementation	45
3.4.2	Analog Implementation	46
3.4.3	Hybrid Implementation	49
3.5	Selecting a Crypto Core	49
3.5.1	Capacitor Bank	50
3.5.2	Preliminaries	51
3.5.3	Computation Effects	52
3.5.4	Memory Effects	56
3.5.5	Analysis and Recommendations	59

3.6	Selecting a Microcontroller Configuration	60
3.6.1	Computation Time (Δt)	61
3.6.2	Average Current (I)	63
3.6.3	Capacitor Size	65
3.6.4	System Verification	65
3.6.5	Software Overhead	67
3.7	Discussion	68
3.8	Related Work	69
3.9	Conclusion	70
4	Quantized Computing:	
	Electromagnetic Augmentation	71
4.1	EM Leakage: Near-Field Radiation	72
4.2	EM Leakage: Wire Layers	75
4.2.1	Metal Layer: Modeling	78
4.3	Metal Layer: Simulating EM-Fields	81
4.4	Threat Model	81
4.5	Design	83
4.6	Discussion	84
4.7	Conclusion	85
5	Conclusion	87
5.0.1	Summary of Results	87
5.0.2	Recommendations for Future Work	90

List of Figures

2.1	Nexus 4 headset jack available power.	11
2.2	Power Circuit	13
2.3	Power Circuit Analysis	13
2.4	Sleep Circuit.	15
2.5	μ Leech Evaluation Board.	17
2.6	μ Leech Execution Cycle.	18
2.7	Manchester Encoding of header 0xDD	21
2.8	Manchester Encoding of length 0x0A	21
2.9	Manchester Encoding of data 0xFF	21
2.10	Manchester Encoding of data 0x00	21
2.11	μ Leech Transmit Flow Chart.	22
2.12	μ Leech Transmit Circuit.	22
2.13	μ Leech Transmission - Zoomed in.	23
2.14	μ Leech Transmission.	23
2.15	μ Leech Receive Flow Chart.	24
2.16	μ Leech Receive Circuit.	24
2.17	Smartphone Transmission - Zoomed in.	25
2.18	Smartphone Transmission.	25
3.1	Quantized Computing system overview.	30

3.2	Illustrative of the Quantization Controller’s effectiveness against power and timing side-channels for RSA. Note that the noise/variation is hard to see due to scaling.	43
3.3	RTL simulation of the wholly digital and hybrid versions.	45
3.4	Full-custom analog Quantization Controller circuit.	46
3.5	SPICE simulation of the full-custom analog version.	47
3.6	SPICE simulation of the analog parts of the hybrid version.	48
3.7	Time required to write to Flash and FRAM.	58
3.8	Quantization Controller MSP432 prototype.	60
3.9	Computation time for AES	62
3.10	Computation time for RSA	62
3.11	Average current for AES	64
3.12	Average current for RSA	64
3.13	Required capacitor size for AES	66
3.14	Required capacitor size for RSA	66
3.15	Software run time overhead across a range of secure execution invocation rates.	67
4.1	Wire Layer Geometry.	79
4.2	HFSS model of Intel 32nm Metal Layer - excluding Metal Layer 9.	80
4.3	HFSS model of Intel 32nm Metal Layer.	80
4.4	Metal Layer E-Field decay over distance.	82
4.5	Metal Layer E-Field contribution over distance.	82

List of Tables

2.1	Smartphone Transmission Burst.	23
2.2	Smartphone Transmission Burst.	25
2.3	μ Leech Power Consumption	27
3.1	Tradeoff space of the three Quantization Controller implementations using a <i>22nm</i> process technology.	45
3.2	Summary of key properties of the processors evaluated.	53
4.1	Pitch(nm)	76
4.2	Thickness(nm)	76
4.3	Aspect Ratio	76
4.4	Pitch(%) difference to Metal 1	77
4.5	Thickness(%) difference to Metal 1	77
4.6	Aspect Ratio(%) difference to Metal 1	77

Chapter 1

Introduction

With the proliferation of cheap bulk SSD storage and better batteries in the last few years we are experiencing an explosion in the number of Internet of Things (IoT) devices flooding the market, smartphone connected point-of-sale devices (e.g. Square), home monitoring devices (e.g. NEST), fitness monitoring devices (e.g. Fitbit), and smart-watches [1]. With new IoT devices come new security threats that have yet to be properly evaluated [2].

The Square point-of-sale unit, in particular, has gained traction in the market. The versatility and convenience of the design have propelled the product into widespread adoption. What makes the Square design unique is that it uses the microphone/speaker interface to power the unit as well as to facilitate the communication between the device and the smartphone. Since speaker/microphone is the only universal interface present in all smartphones the Square device is transparently interchangeable from one smartphone to the other.

The Square point-of-sale unit iterated through three editions, due to security threats [3]. The first version was susceptible to man in the middle attacks as all communication between the Square and smartphone was not encrypted [4]. To address this the next iteration implemented encrypted communication at the cost of power, requiring an internal

coin cell battery [4]. Their third edition addresses both encryption and power requirements with a customized copyrighted low power IC [4]. These version iterations were dictated by security vulnerabilities that were overlooked and compromised the intended objective of the device, i.e. secure transactions.

The focuses of this thesis are the security threat presented by physical side-channel leakage of IoTs such as the Square point-of-sale. To that end, we design and evaluate power and timing side-channel leakage of μ Leech. μ Leech is a custom IoT device modeled off of the Square point-of-sale that acts as a universal power scavenging Trusted Platform Module (TPM) for smartphones. We augment μ Leech to serve as a side-channel evaluation platform to evaluate side-channel leakage present in lower power scavenging IoT devices. In the course of our evaluation, we propose a new countermeasure centered around isolation. Our Quantization Controller design leverages power scavenging techniques to isolate power leakage during secure execution to create uniform power and timing footprints. We augment our countermeasure to include defenses against thermal, power glitching, memory fault, and EM attacks.

1.1 Background

All computing devices generate side-channel emanations as a byproduct of physical implementation and computation [5]. The key to side-channel analysis is the interpretation of these leaks to reveal secrets. As such, there are many different side-channel leaks used to capture secrets. These include: power [6], timing [7], electromagnetic [8], acoustic [9], memory remanence [10], and thermal [11]. The two most commonly exploited side-channels in literature are power and timing [12]. The popularity of these two side-channels is due to the high bit rate and fine-grain (e.g., per clock cycle) information that they expose to the attacker. To gain access to this wealth of secret-revealing information,

an attacker only needs to be able to measure the current consumed by the victim device.

Side-channel countermeasures focus on reducing the signal-to-noise ratio (SNR). Techniques such as noise introduction and incorporating randomness focus on raising the noise threshold [13]. Thereby making it challenging to differentiate between the leaked signal and random noise. On the other hand techniques such as leakage reduction and obfuscation aim to reduce the leaking signal strength within the noise threshold [13]. By reducing the SNR, any countermeasure can increase the difficulty of filtering out the leaking signal from the noise level. However, the real trick with any countermeasure is to not push leakage from one channel to another, as there is a symbiotic relationship between the different side-channels:

- **Power vs. Timing:** Implementing any limitations on a power rail would directly correlate to any variability in execution performance. Therefore a leakage hidden on the Power rail would be exposed through timing analysis.
- **Timing vs. Power:** Implementing any timing limitations to hide execution would directly translate into power consumption. Executions that obfuscate leakage by forcing constant execution times would leak variable power draw.
- **Timing vs. Fault:** Implementing counters to conceal timing differences creates opportunities for fault attacks that invert the expected ordering of system events. Temperature-based fault attacks affect the rate of leakage of a capacitor while not effecting digital counters.
- **Power vs. Electromagnetic:** Hiding the power capacitor recharge side-channel requires shorting the capacitor's voltage down to a fixed level. This creates a large current spike visible in the electromagnetic spectrum. The magnitude of this current spike indicates how much current was on the power capacitor after secure execution, thus revealing secret information.

Without care, eliminating the leak from one side-channel shifts the information to another channel. A complete solution would have to address these threats in unison.

1.2 Problem Statement

Accompanying the recent explosion of smartphones and tablets is a growing trend for new IoTs that can interface with those devices. These IoTs are designed with the express purpose of being small, easy to use, and convenient. Unfortunately, in many cases, IoTs are deployed as trusted devices in security critical applications before they have been properly evaluated. For the purpose of this work, we are interested in the physical side-channel leakage of such IoTs. While side-channel attacks have been extensively researched the effect of secure executions on such low power, embedded, and highly connected devices have not been properly evaluated. IoTs perform low power intermittent execution using energy harvesting inputs. As such IoTs have drastically different electrical characteristics that need to be properly evaluated for side-channel vulnerabilities.

Our threat model assumes an attacker with knowledge of the software running on as well as with full physical access to the IoT. The attacker's objective is to extract secret information through non-destructive, non-invasive, timing and power side-channel analysis. We assume that the attacker can also control the power supplied to the device as well as its environment to induce power and thermal faults.

We develop a custom low power IoT used for intermittent secure execution to observe any side-channel leakage, using a customized side-channel evaluation platform. In the course of our evaluation, we propose a new countermeasure centered around isolation. Our proposed design focuses on reducing the signal by leveraging power scavenging techniques to isolate power leakage during secure execution and creating uniform power and timing footprints. Following these observations, we develop and augment a compre-

hensive and mostly automatic power and timing side-channel defense that is immune to thermal and power fault attacks.

1.3 Summary of Contributions

- Designing a custom IoT device, μ Leech, that can mimic existing IoT communication protocols, power scavenging features, intermittent execution, and security-critical applications.
- Developed and optimized μ Leech as a low power TPM module for smartphones
- Designing a side-channel evaluation platform of μ Leech to facilitate the acquisition of physical side-channel leakage.
- Designing a countermeasure by leveraging energy harvesting techniques to isolate power and timing side-channel leakage.
- Augmenting our countermeasure to protect against temperature, fault, and power-glitching attacks.
- Prototype our countermeasure to test for side-channel leakage under real-world conditions
- Evaluating the source of electromagnetic side-channel leakage from ICs
- Modeling the primary source of electromagnetic side-channel leakage from ICs
- Extending the proposed countermeasure to reduce electromagnetic side-channel leakage

1.4 Dissertation Outline

- **Chapter 2** describes the design of μ Leech a new embedded TPM for IoTs and its development as a side-channel evaluation platform. μ Leech can mimic the communication protocols, power scavenging effects, and cryptographic executions of other IoTs. While such devices are used as trusted devices in security critical applications, they have not been properly evaluated yet. μ Leech can securely store keys and provide cryptographic services to any connected smartphone. Our design facilitates physical side-channel security analysis by providing interfaces to enable easier acquisition of power traces and clock manipulation attacks. Thus μ Leech empowers security researchers to analyze leakage in next generation embedded and IoT devices and to evaluate countermeasures before deployment.
- **Chapter 3** presents Quantization Controller, a side-channel isolation countermeasure that creates uniform power and timing side-channel footprints while protecting against temperature, fault, and power glitching attacks. Our design leverages integrated decoupling capacitors to electrically isolate trusted computation from the adversary. Software has the ability to request a fixed-power/time quantum of isolated computation. By discretizing power and time, our approach controls the granularity of side-channel leakage; the only burden on programmers is to ensure that all secret-dependent execution differences converge within a power/time quantum. We design and implement three approaches to power/time-based quantization and isolation: a wholly-digital version, a hybrid version that uses *capacitors for time tracking*, and a full-custom version. These designs are evaluated with respect to software implementations of AES and RSA running on an ARM-based microcontroller and hardware implementations of AES and RSA using a 22nm process technology. We also validate the effectiveness and real-world efficiency of our ap-

proach by building a prototype consisting of an ARM microcontroller, an FPGA, and discrete circuit components.

- **Chapter 4** examines the root cause of Electromagnetic (EM) side-channel attacks on Integrated Circuits (ICs) to augment the Quantization Controller design to mitigate EM leakage. EM attacks on ICs design have long followed a black box approach. We identify and examine the root cause of EM side-channel attacks, near field radiation from an ICs wire layers that act as unintended antennas. Combining these findings with IC power management protocols, we can identify that the major contributing EM leakage are the top two wire layers that disseminate power throughout the IC. These layers are intentionally thicker to carry more current thereby maintain power and timing benchmarks. Current IC technology are only exasperating the leakage from these top power layers, as shown by examining Intel's IC technology: 180nm [14], 130nm [15], 65nm [16], 45nm [17], and 32nm [18]. We propose an augmentation to our Quantization Controller design, by leveraging its isolation nature to reduce the length and power of the unintended EM antennas created by the wire layers.
- **Chapter 5** concludes this work with a summary of the results and recommendations for further work. We discuss the effectiveness of μ Leech as mimicking low power IoT devices, using μ Leech as a side-channel evaluation platform, leveraging isolation as a countermeasure in our Quantization Controller design, and our proposed Quantization Controller EM augmentation. Our proposed design would effectively reduce side-channel leakage by protecting against multiple side-channel attacks: power [6], timing [7], electromagnetic [8], thermal [11], power glitching [19], and Memory fault [20] attacks.

Chapter 2

μ Leech: A Side-Channel Evaluation

Platform for IoT

With Moore's law continuing to drive down the cost of computation, we are experiencing an explosion in the number of embedded devices flooding the market, including smart-phone connected point-of-sale devices, smart home devices or smart-watches. The Square point-of-sale unit, in particular, has gained traction in the market. The versatility and convenience of the design has propelled the product into widespread adoption. Such embedded, small footprint, highly connected, and low-power devices make up the backbone of the Internet of Things (IoT): they are attached to *things* in our environment we wish to track and control remotely. These devices are highly susceptible to physical attacks as there are many of them with loose isolation potentially in the reach of an attacker. Given physical access, an attacker can steal sensitive secrets, i.e. encryption keys and authentication credentials.

In this work for the first time we develop a side-channel evaluation platform, μ Leech, which mimicks the form factor and operational characteristics of IoT devices. Our design is modeled after the Square design, i.e. it uses the microphone/speaker interface to power

the unit and to communicate between the device and the smartphone. In the development of our design we started by examining the Hijack project [21]. Ultimately our communication protocol differs as we do not need to communicate continuous sensor data, but short bursts of data, draining less power in the process. Our μ Leech design acts as a universal power scavenging unit and may also be used as a Trusted Platform Module (TPM) for untrusted smartphones that can execute sensitive queries directly on the μ Leech processor, e.g. by using secret keys and authentication tokens that are programmed into μ Leech. The fact that these keys will never leave the device keeps them protected.

2.1 Side-Channel Evaluation Platforms

There are many platforms available that can be used to assess vulnerabilities of implementations to side-channel attacks. Some are commercially available and others developed for academic use. The following lists the most prominent:

- **Side-channel Attack Standard Evaluation Boards (SASEBO)** are standard evaluation boards developed by Tohoku University [22]. SASEBO Boards were developed to perform security tests for side-channel attacks. Their goal was to standardize the testing requirements of cryptographic modules. They are capable of evaluating side-channel attacks against cryptographic hardware with FPGAs and cryptographic software with microprocessor function.
- **Flexible Open-source Board for Side-channel analysis (FOBOS)** is an academic platform developed at George Mason University [23]. FOBOS is a platform for implementation attack resistance testing. FOBOS aims to provide an open-source platform. Their platform can be used to evaluate the effectiveness of side-channel analysis countermeasures on FPGA platforms. FOBOS supports multiple FPGA devices and the necessary software to run differential power analysis attacks.

- **Side-Channel Analysis Resistant Framework (SCARF)** is an open-source academic tool developed by the Electronics and Telecommunications Research Institute [24]. SCARF can be used for testing countermeasures for side-channel and fault attacks. They include a number of custom evaluation boards to test the attack resistance of smart-cards, microprocessors, and FPGAs. SCARF only supports testing of the devices included in its custom evaluation boards.

The following are a few tools that can be used with these side-channel evaluation platforms:

- **The DPA Workstation** is developed by Cryptography Research Inc. [25] and can be used to perform side-channel analysis including differential power or electromagnetic analysis on embedded systems. The DPA Workstation includes its own environment and proprietary software that can perform side-channel analysis on all major standard ciphers.
- **InspectorSCA** is a closed-source device developed by Riscure [26]. This platform can be used for side-channel analysis and fault analysis. It includes fault injection hardware and includes proprietary software that can perform side-channel and fault attacks on standard ciphers.
- **ChipWhisperer** is an open-source tool-chain for embedded hardware security research [27]. It can be used for side-channel power analysis and glitching and is known for its synchronous capture technology. Similar tools are commercially available but far more expensive and closed-source.

Numerous IoT devices have already been deployed as trusted devices in security critical applications, and yet they have not been properly evaluated for side-channel leakage. The existing platforms summarized above have characteristics drastically different from

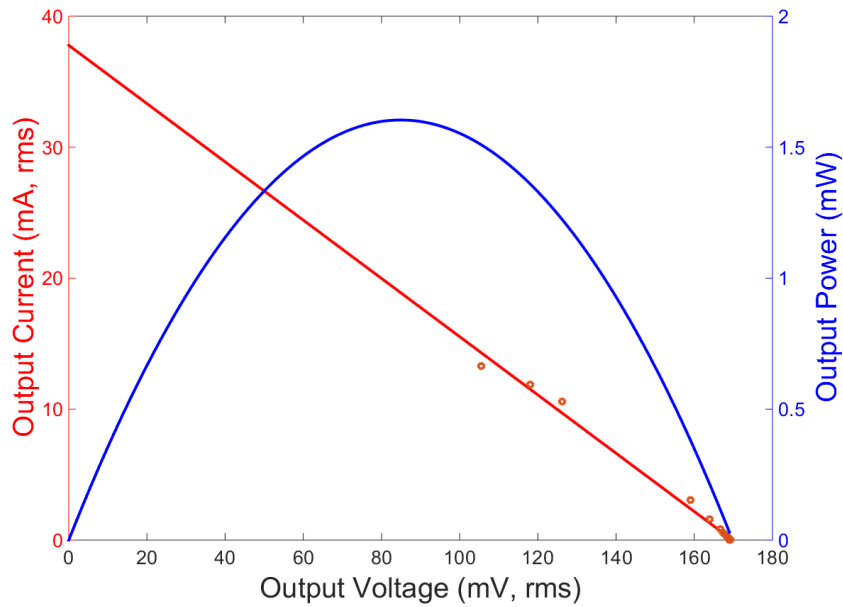


Figure 2.1: *Nexus 4 headset jack available power.*

IoT devices. μ Leech is a side-channel evaluation platform representative of a power scavenging IoT device: power is scavenged from the right audio channel, two-way communication through the left audio channel and the microphone. Our platform provides the ability for security researchers to analyze side-channel leakage in next-generation mobile attached embedded devices. Just like SASEBO, FOBOS, and SCARF the μ Leech evaluation platform will include high-quality onboard peripherals to our power and clock signals, allowing for easy access of signals necessary for side-channel analysis. This platform will allow for the development and enrollment of countermeasures.

2.2 Overview of the Design

Most cryptographic primitives require short and highly intensive computation. Taking advantage of this, we designed our power circuit with a capacitor bank that can be discharged and recharged. This allows our processor to perform any cryptographic operation

while draining the capacitor bank. We optimized our design to allow one round of AES to be computed before depleting the capacitor bank. Once the capacitor bank is depleted the processor will hibernate in sleep mode, allowing the capacitor bank to recharge again for another round of computations. To achieve this, we developed a low power sleep circuit that notifies the processor when to sleep and wake, thereby allowing the processor to operate off the limited power generated by our smartphone.

2.2.1 Power Siphoning

In keeping with the concept of a universal jack, our power has to be generated from the auxiliary jack. Therefore, we have to siphon power from an audio waveform. To determine the available power from our Nexus 4 headset port, we generated a 500 mV peak to peak AC audio waveform. A load resistance was connected between the right audio channel and the common line of the auxiliary jack. While measuring the output voltage across the resistor and the load current, the load resistance was varied from 0Ω to $15k\Omega$. We measured a total of 24 different resistor values. Using this data, we generated a linear fit, represented by the red linear IV curve in Figure 2.1 and the blue power transfer curve in Figure 2.1. The power transfer curves show that the maximum power transfer occurs at $85mV_{rms}$ and $18.87mA_{rms}$, with an ideal load of 4.51Ω .

The data in Figure 2.1 shows that it is possible to draw $1.61mW$ from an ideal matched load of 4.51Ω . For this power to be useful our power siphoning circuit will have to rectify the $500mV$ peak to peak waveform from AC to DC, boost it, and filter it. The power siphoning circuit we implemented was modeled off of the Hijack's power siphoning circuit [21], with a few modifications. The circuit we implemented is depicted in Figure

The input of the circuit Figure 2.2 is the Right audio channel of the smartphone. The app we implement will generate a continuous $500mV$ peak to peak audio waveform only onto the right audio line. Using this audio line and this circuit we are able to generate a

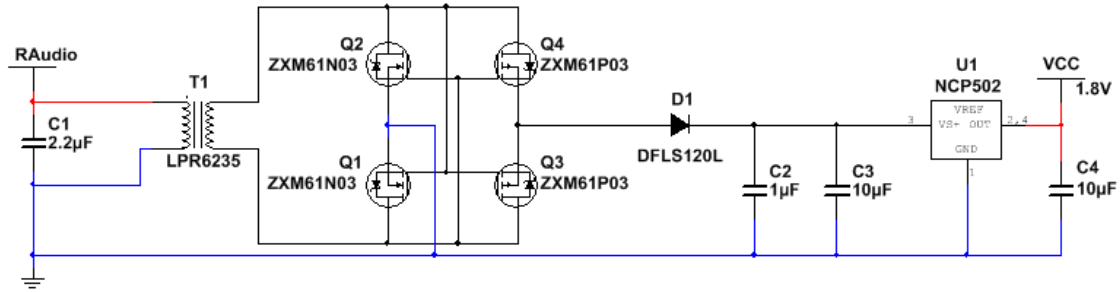


Figure 2.2: Power Circuit

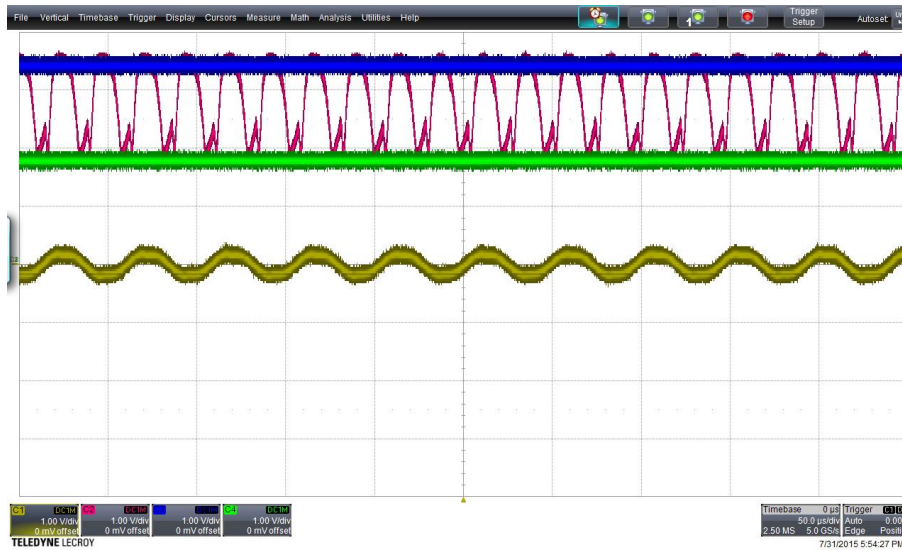


Figure 2.3: Power Circuit Analysis

1.8V source for our processor. Key points of the transformation of the 400mV peak to peak audio waveform through the circuit in Figure 2.2 were captured and illustrated in Figure 2.3.

The yellow signal is the raw waveform coming out of the smartphone. The first step in trying to use this audio frequency is to step up the low supply voltage level. Using a 1:20 micro-transformer we step up the incoming supply voltage level as shown by the red signal. Normally the next step would be to use a Schottky diode to perform low-loss blocking. However using a combination of N and P MOSFETs depicted in Figure 2.2, we can perform a FET-based rectification from AC current to DC. This allows us to be able

to use the negative part of our input waveform to generate power as well. The spikes on the low edge of the red signal are generated by the FET bridge that would have otherwise been clipped out, optimizing our power generating capability. Using the Schottky diode to provide low-loss blocking, will prevent the capacitor bank from being discharged through the FET bridge, illustrated by the Blue signal. The final step is to use a 1.8 voltage regulator to power our processor, the Green signal.

Our goal was to allow our processor to be able to perform more power intensive cryptographic computations, without using a battery and while staying within the limitations of our smartphone power generating capability. Therefore using this power siphoning circuit and a series of capacitors, that could be discharging and recharged merely by going into sleep mode, we are capable of allowing our device to remain charged for at least one round of AES.

2.2.2 Capacitor Bank

Every smartphone model will have its own power draw model and ideal matched load. To aid with power management, we implemented a capacitor bank that will aid us in granting enough power for one round of 128-bit AES. As our processor implementations continue to evolve, we will continue to optimize our power consumption. However, different cryptographic operations may consume more power than our current 128-bit AES implementation. This means that our capacitor bank needs to contain enough of a charge between the time our sleep circuit switches between its wake and sleep state. Knowing that our sleep circuit activates the processor's wake state at 1.95V and sleep state at 1.65V we can calculate the necessary capacitor bank we would need for one round of 128-bit AES, if we know the current consumption and amount of time it takes. As shown later AES consumes approximately $555 \mu\text{A}$ and takes $140\mu\text{s}$ for one round, see Table 2.3. Therefore we need a $0.28\mu\text{F}$ capacitor bank to power one AES round.

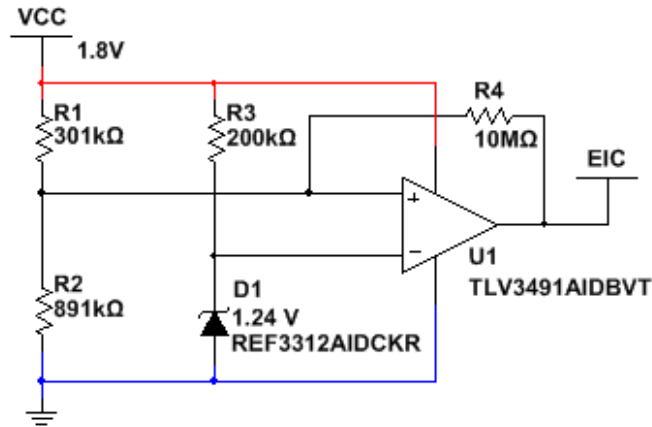


Figure 2.4: *Sleep Circuit.*

Currently, our capacitor bank was added after our 1.8 voltage regulator. This was done initially because our measured voltage change were measured after our voltage regulator. For optimal efficiency, the capacitor bank should be recalculated and inserted before the voltage regulator. Although the capacitor bank was initially designed for one round of 128-bit AES, the design still works with whatever implementation our processor is computing. As long as the power drops the sleep circuit will activate sleep mode allowing for the capacitor bank to recharge. Such functionally will be useful with any smartphones as the bank will aid in allowing for more continuous power.

2.2.3 Sleep Mode

To allow the capacitor bank to recharge as efficiently as possible, we had to minimize power consumption during sleep mode. In addition, we would need a way of telling the processor to go to sleep and not to wake up until the capacitor bank was adequately recharged. For this we built a sleep circuit, illustrated in Figure 2.4, that would tell the processor through an external interrupt when to go to sleep and when to wake up.

Our sleep circuit uses a total of four 1% tolerance resistors, one low power voltage reference, and one low power comparator. This circuit has minimal power consumption

and produces one output signal from the comparator. This signal is driven either high or low, and can be used as a digital input. When connected to the external interrupt controller of our processor this signal can be used to notifying our processor when to go to sleep, low for optimal power performance, and when to wake up.

Atmel's AVR UC3-L0 has multiple interrupts and external interrupts. The UC3-L0 has a total of seven sleep modes [28]. Only the two highest power consumption sleep modes can wake up from a synchronous source, ruling out using an internal interrupt routine. However, all sleep modes, except for shutdown mode can be achieved with an asynchronous source. Connecting the external signal generated from our sleep circuit to two external interrupt pin, we can use two external interrupt routines to go to sleep (low) and wake up (high). Both of these external sleep/wake interrupts have been given the highest priority to supersede any other interrupts. This is to guarantee that when our capacitor bank is almost drained the processor will go to sleep before losing power allowing the capacitor bank to charge up again.

Our processor operates at 1.8V, but using its power thresholds our comparator will notify the processor to go to sleep at 1.65V and to wake up at 1.95V. Measuring the time it takes for one round of AES we modified our capacitor bank for ample time to complete before recharging.

2.2.4 Side-Channel Platform

As a side-channel evaluation platform, our main goal was to provide easy acquisition of high-quality power signals for Differential Power Analysis (DPA). We added three 1Ω resistors in series to our design before the decoupling capacitors of our processor. The resistors were placed as close as possible to the processor's respective pins, and the copper pads were expanded, to minimize the amount of added noise. One resistor was added to the ground rail and the other two to the power rails. All three resistors can also

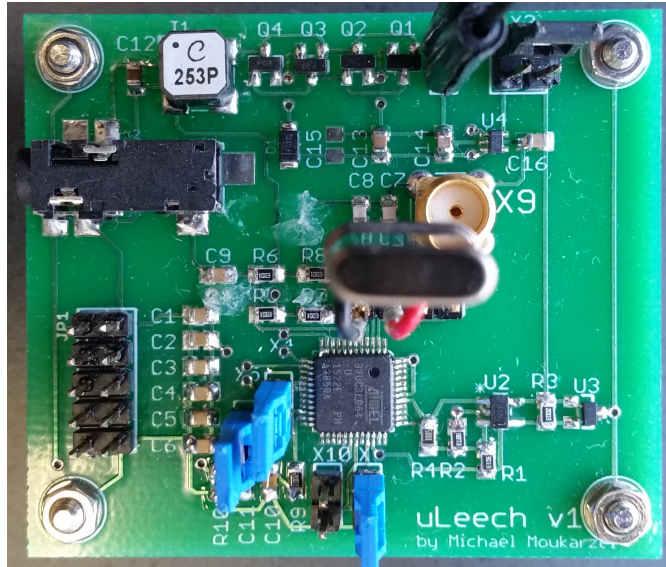


Figure 2.5: *μLeech Evaluation Board.*

be individually shorted to allow for custom measurements. The current version of our evaluation platform is illustrated in Figure 2.5. The three blue shunts are the jumpers to the three resistors. In addition, we added two external clock sources that can be selected for our Processor: a 1 MHz crystal and an SMA connector. The SMA connector can be used to feed a clock signal directly, thus also rendering the board ready for clock-glitching evaluation.

2.3 Data Communication

The communication protocol was modeled off of Hijack’s communication protocol [21]. This design does not require any analog to digital converters, which results in less power drain but slower communication speeds. The smartphone transmits analog signals and the processor transmits in digital, but each communication is handled in the time domain, using Manchester Encoding [29].

Manchester Encoding is a form of digital encoding where bits are represented by transitions from one logical state to the other, instead of being represented by a high or

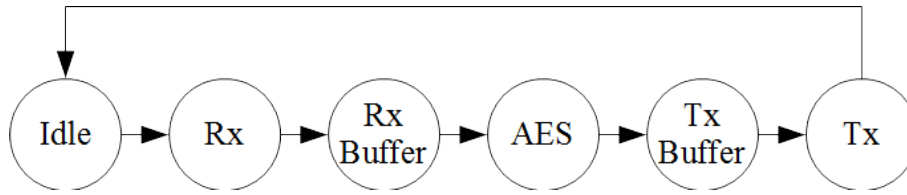


Figure 2.6: μ Leech Execution Cycle.

low signal. This means that instead of having to detect an analog high and low period, we now have to measure the periods between switches. Allowing a smartphone that can only output and read audio oscillating waveforms to transmit and receive a waveform with Manchester encoded data.

For the processor, transmitting in Manchester encoding to the smartphone is straight forward. The processor can generate a square wave at any frequency, doubling the period for a bit flipping, that can simply be interpreted by a smartphone as an oscillating waveform. On the other hand, decoding incoming data from a smartphone outputting an oscillating audio waveform instead of a digital signal to the processor is not as straightforward. The processor instead will have to determine the period lengths between the oscillations. Allowing the processor to determine if the incoming Manchester encoded data is flipping its bit, double the period, or not.

We have modified our communication protocol to work serially and in conjunction with the overall process of the processor. μ Leech is not a continuously operating device as such it does not need to transmit and receive data continuously. We only transmit and receive data in bursts when necessary after secure execution. This allows for more computation cycles and overall less power drain on our system. Figure 2.6 depicts a flow chart of μ Leech execution cycle for an AES implementation.

In Idle our processor is continuously waiting to receive a transmission from the smartphone. Once data is received it triggers our communication state-machine that grabs the incoming data. The state-machine then converts the raw incoming Manchester data into

binary data. This incoming data includes the instructions for our processor. Using the obtained data our processor then performs it AES encryption/decryption. It then triggers the transmit state of the state machine. The state machine will then converting the raw data into Manchester encoded data and transmit this to the smartphone. The transmission burst is repeated four times by our processor, after which it returns to Idle waiting to receive further instruction from the smartphone.

Currently, we are transmitting at approximately an average of 0.9kbps. Our current transmission speeds are unfortunately severely slower than regular digital communication. This is mainly due to the fact that our data is transmitted in Manchester Encoding that doubles the number of cycles. In addition, there are delays that are added through our communication state-machine. In the next iteration we will be optimizing our communication state-machine for intermittent communication.

2.3.1 Manchester Encoding

Manchester Encoding is a form of digital encoding in which data bits are represented by transitions from one logical state to the other. The encoding of digital data in Manchester format defines the binary states of a 1-bit and 0-bit to be transitions rather than static values. Manchester encoding was developed at the University of Manchester, where the coding was used to store data on the magnetic drum of a Manchester Mark 1 computer. There are two opposing conventions for Manchester encoding, both are used by numerous authors.

The first convention of Manchester Encoding was first published by G. E. Thomas in 1949. It stated that for a 0-bit the signal levels would be low-high and for a 1-bit the signal levels would be high-low. The second convention used by IEEE 802.4 and IEEE 802.3. It states that for a 0-bit the signal levels would be high-low and for a 1-bit the signal levels would be low-high. These two conventions are opposites of each other.

For our application, we followed the IEEE convention. Illustration examples of Manchester Encoded data can be found in Figures 6 through 9. These examples are illustrations of an implemented transmission using Hijack's communication protocol.

- Figure 2.7 is the header of the transmission.
- Figure 2.8 is the transmission byte length.
- Figure 2.9 and Figure 2.10 are two different data values that were transmitted.

We choose only to use two data values so that the illustration would be simpler to follow. The transmissions are handled one byte at a time. Each byte has a start bit of 0 and a parity bit at the end. The actual transmission streams of the processor and smartphone can be found in Figure 2.14 and Figure 2.18 respectively. The transmission captured match these illustrations.

Manchester Encoding allows for an overall reduced physical footprint and power consumption due to the lack of digital to analog and analog to digital converters. In addition, since Manchester Encoding ensures frequent line voltage transitions, which are directly proportional to their clock rate, an oscillating audio waveform can be manipulated into representing Manchester Encoded data. This makes it ideal for IoT communication systems through auxiliary jacks.

2.3.2 Transmit

Data sent from the processor to the smartphone will be sent along the microphone channel of the auxiliary port. The communication protocols used for creating this audio waveform were modeled off of Hijack's communication protocol. The major difference being that when in this state we only use the transmit portion of Hijack state-machine, as shown in Figure 2.6. The protocol we implemented to activate the transmit portion of the state machine is illustrated in Figure 2.11.

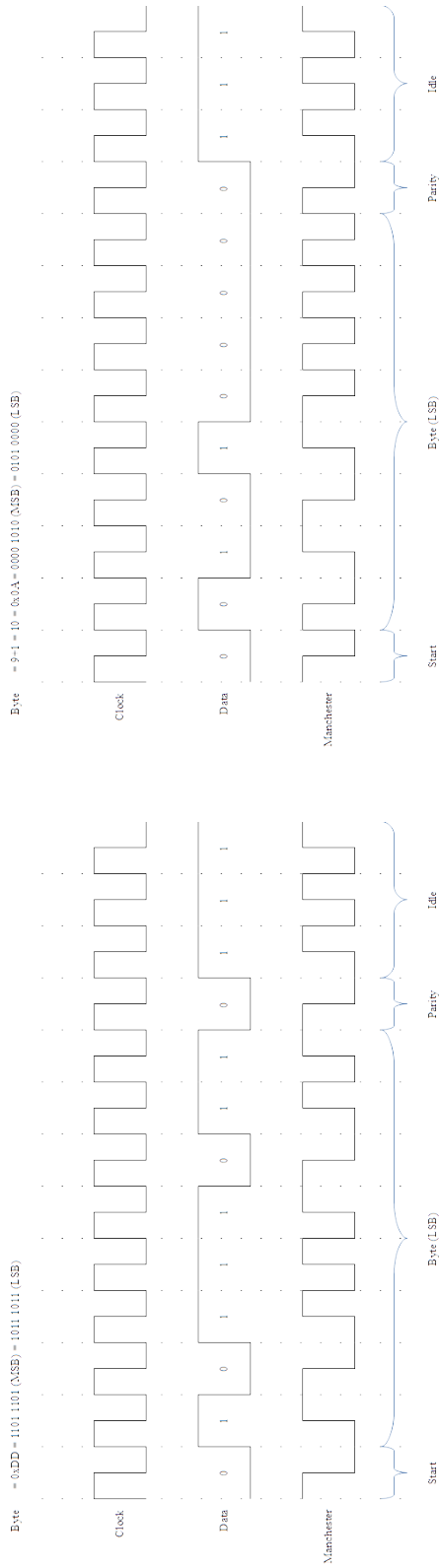


Figure 2.7: Manchester Encoding of header 0xDD

Figure 2.8: Manchester Encoding of length 0x0A

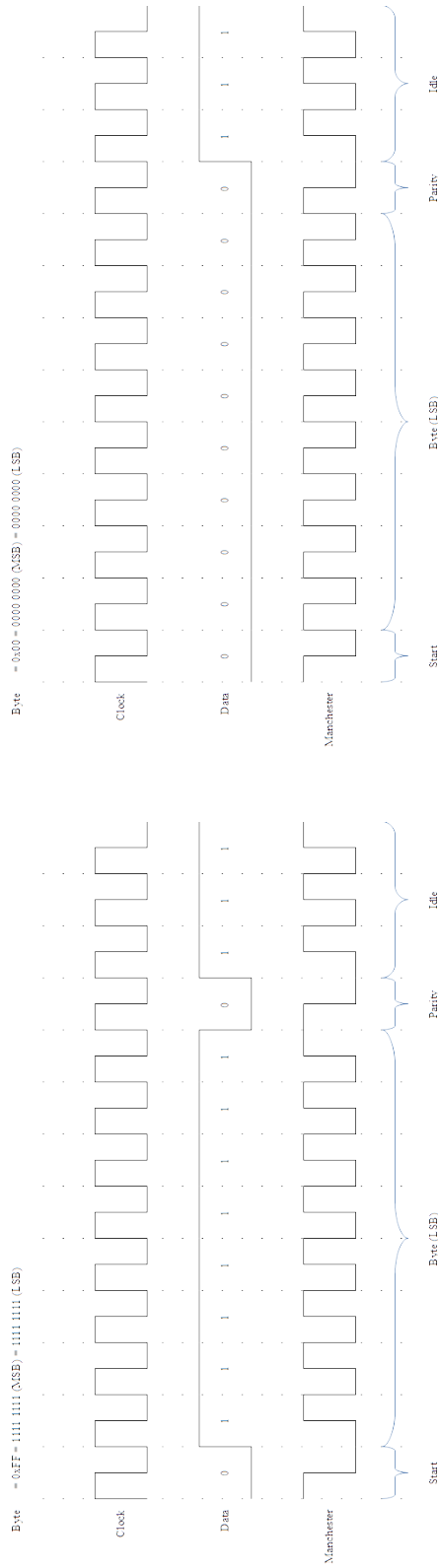


Figure 2.9: Manchester Encoding of data 0xFF

Figure 2.10: Manchester Encoding of data 0x00

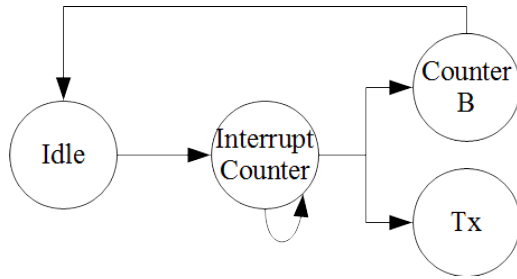


Figure 2.11: *μLeech Transmit Flow Chart.*

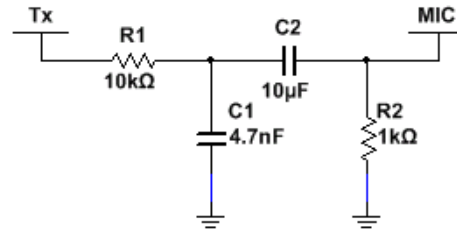


Figure 2.12: *μLeech Transmit Circuit.*

For our transmit protocol we used one of our processors built in synchronous counter and one asynchronous counter. The counters are only activated when the processor is in the transmit state (Tx), in Figure 2.6. When the counter is activated, it will trigger an interrupt every time it reaches a predetermined value. This value is the frequency of our generated signal, every time the interrupt is triggered is one clock cycle of our output stream. This event triggered interrupt will do two things: trigger the transmit state of the state-machine and increment a second asynchronous counter B. The second counter B is used to keep track of how many transmission have taken place, after four full transmissions the counter will exit the transmission stat and turn off the counter.

At the hardware level the only important thing with the transmission circuit is to have a 1k load resistor. The 1k Ohm resistor is necessary to activate the microphone channel on most smartphones; without it, the smartphone would not know there was data coming in on the microphone channel. The processor will then output an oscillating square waveform approximately 800mv pk to pk. The transmission circuit we implemented is shown in Figure 2.12.

An example of a transmission stream from our *μLeech* to our smartphone was captured and illustrated in Figure 2.14. Using the Manchester encoded byte examples in Figures 2.7 through 2.10 we can see the data captured in Figure 2.14. The communication protocol of the processor transmits in the format depicted in Table 2.1.

Each byte has a start bit of 0 and a parity bit at the end. This means each byte trans-

Byte	1	2	3	4	5	6	7	8	9	10	11	12
Content	Header	Length+1	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Check Sum
Value	0xDD	10	0xFF	0xFF	0x00	0xFF	0x00	0xFF	0x00	0xFF	0x00	-
Image	Figure 2.7	Figure 2.8	Figure 2.9	Figure 2.9	Figure 2.10	Figure 2.9	Figure 2.10	Figure 2.9	Figure 2.10	Figure 2.9	Figure 2.10	-

Table 2.1: Smartphone Transmission Burst.

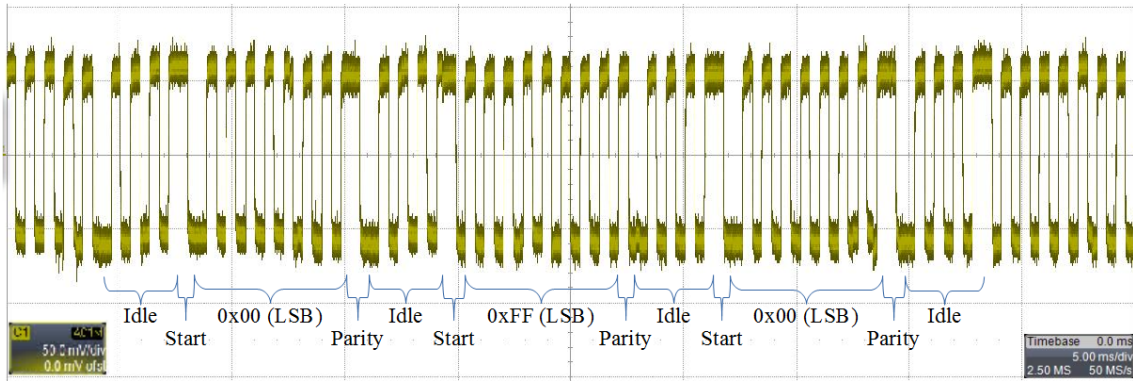


Figure 2.13: μ Leech Transmission - Zoomed in.

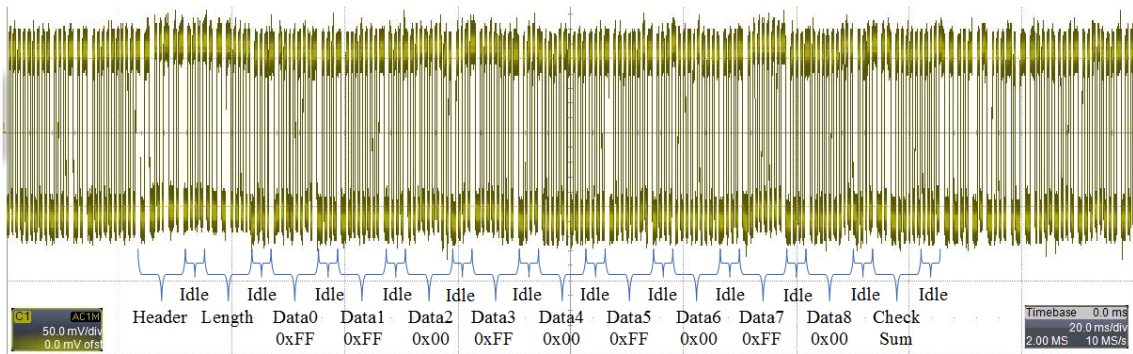


Figure 2.14: μ Leech Transmission.

mission is actually 10-bits which are then transmitted in LSB in Manchester encoding. However, in between each byte transmission the state-machine goes into an idle state, where it continues to transmit digital high for a fixed four delay count. Figure 2.13 depicts a zoomed in view of the Manchester encoded captured data burst shown in Figure 2.14.

2.3.3 Receive

Data sent from the smartphone to the processor will be sent along the left audio channel of the auxiliary port. The communication protocols used for interpreting this audio waveform were modeled off of Hijack's communication protocol. The major difference

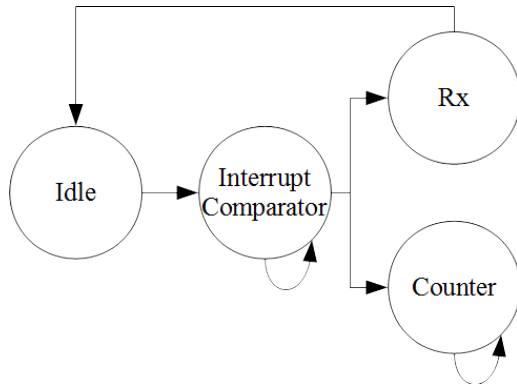


Figure 2.15: *μLeech Receive Flow Chart.*

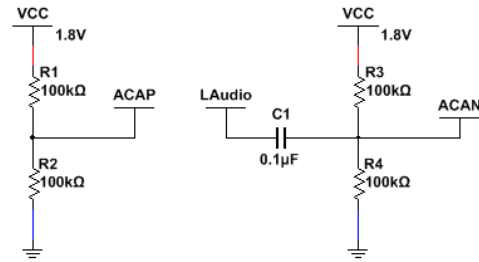


Figure 2.16: *μLeech Receive Circuit.*

being that when in this state we only use the receive portion of Hijack state-machine, as shown in Figure 2.6. The protocol we implemented to activate the receive portion of the state-machine is illustrated in Figure 2.15.

For our receive protocol, we used one of our processors built in synchronous comparator driven interrupt and counter. The comparator and counter are only activated when the processor is in the receiving state (Rx), in Figure 2.6. When the comparator is activated it will trigger an interrupt every time there is a bit flip on the incoming Manchester encoded data stream. This event triggered interrupt will do two things: trigger the receiving state of the state-machine passing it the counter value and reset the counter. This will allow the state-machine to know the length of time between bit-flips. Since the incoming data is represented in Manchester encoding, the state-machine will be able to determine if the previous bit was flipped or not. Allowing us to decode the income Manchester data into binary data.

What this means for our hardware, is that we have to feed the comparator in our processor two inputs: an oscillating audio waveform from the smartphone’s left audio channel and a constant signal that is the mean voltage level of the oscillating audio wave, as shown in Figure 2.16.

The smartphone will output an oscillating waveform of approximately 900mV pk to

Byte	1	2	3	4	5	6	7	8	9	10	11	12
Content	Header	Length+1	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Check Sum
Value	0xDD	10	0xFF	0xFF	0x00	0xFF	0x00	0xFF	0x00	0xFF	0x00	-
Image	Figure 2.7	Figure 2.8	Figure 2.9	Figure 2.9	Figure 2.10	Figure 2.9	Figure 2.10	Figure 2.9	Figure 2.10	Figure 2.9	Figure 2.10	-

Table 2.2: Smartphone Transmission Burst.

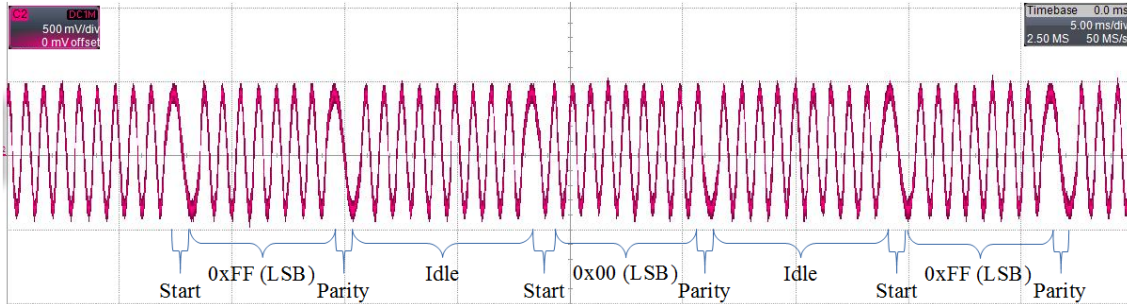


Figure 2.17: Smartphone Transmission - Zoomed in.

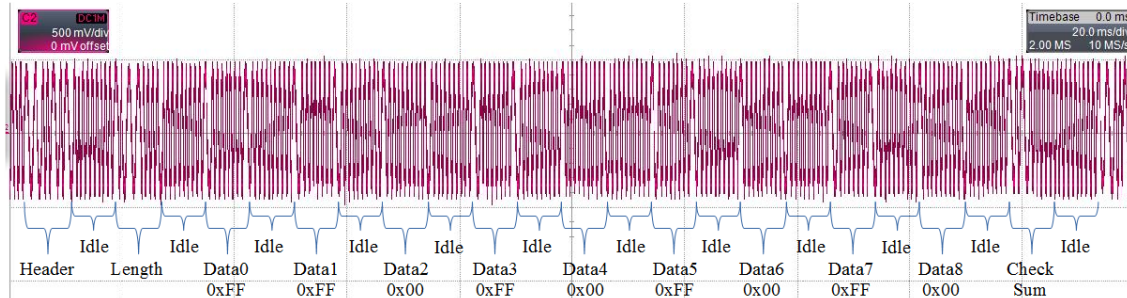


Figure 2.18: Smartphone Transmission.

pk with a lot of noise. The first step in our receive circuit is to drain out the noise with a capacitor, then using a pull-up resistor and a pull-down resistor we can generate an 800mV peak to peak oscillating at a mean of 800mV. However, when you take into account the additional pull-down resistor built into the processor, on the input pin, the signal is oscillating at a mean of approximately 900mV. This means, for our comparator to trigger we need to feed it another constant 900mV signal. Since our processor is operating at 1.8V using a simple voltage divider on our voltage source with equal resistors, we can divide the voltage level by half and generate a constant input signal of 900mV, with minimal power loss. This will allow our comparator to trigger exactly when our oscillating audio signal is switching from low to high, allowing us to retrieve the period length determining if the Manchester encoded audio waveform is bit flipping or not.

An example of a transmission stream from our smartphone to our μ Leech was captured and illustrated in Figure 2.18. Using the Manchester encoded byte examples in Figures 2.7 through 2.10 we can see the data captured in Figure 2.18. The communication protocol of the smartphone transmits in the format depicted in Table 2.2.

Each byte has a start bit of 0 and a parity bit at the end. This means each byte transmission is actually 10-bits which are then transmitted in LSB in Manchester encoding. However, in between each byte transmission the state-machine goes into an idle state, where it continues to transmit digital high for a fixed twenty delay count. Figure 2.17 depicts a zoomed in view of the Manchester encoded captured data burst shown in Figure 2.18.

2.4 Implementation Results

The performance of our device is evaluated with a focus on power consumption. Our device is designed to operate solely off of the power generated from an audio waveform. As a result, we have made numerous modifications to optimize performance within our power generating capabilities. We optimized for power by modifying our capacitor bank to at least allow for one round of AES encryption before depletion. For this, we measured the power consumption of our different modes: idle, sleep, communication, and AES. Then we measured the duration of each of these modes so that we could accurately estimate our capacitor bank.

2.4.1 Sleep Mode

There are many different sleep modes available to the UC3-L0. Table 2.3 shows the power consumption of the lowest three power consumption sleep modes available for a single supply mode design. The main differences between these three sleep modes are

Table 2.3: *μLeech Power Consumption*

Processor State	Power Consumption
Idle	539 μA
Receiving Data	588 μA
Transmitting Data Burst	575 μA
128-bit AES Encryption	555 μA
128-bit AES Decryption	555 μA
Sleep Mode	Power Consumption
Stop	92.6 μA
DeepStop	68.3 μA
Static	56.7 μA

which clock sources are left enabled. For this reason, our design uses an external clock crystal that allows us to use DeepStop and Static sleep modes. Static mode requires us to reinitialize some of our clock signals, and therefore the modules using them, on every wake-up. For this reason currently, our design uses DeepStop which is less of a power drain on the overall system for every wake-up cycle.

2.4.2 Active Mode

In active mode, our UC3-L0 is: idling, communicating, or performing AES. Our processor only checks for incoming data when it is idling and waiting for instructions. Computations are only executed when the processor receives new incoming data or new instruction sets. The processor transmissions are executed only when there is a change in the outgoing data and computations have been completed. The transmissions are in short bursts repeated four times for redundancy error checking. The measured power consumption of these processor states is listed in Table 2.3. Currently, our design's highest power consumption is in our communication protocol.

2.4.3 128-bit AES

Atmel's AVR UC3-L0 combines low power consumption and computational capacity. 128-bit AES was implemented in software and tested for power consumption. The power consumption of 128-bit AES encryption and decryption are shown in Table 2.3. The measured performances of our processor are operating at a clock speed of 1 MHz. This clock is generated using an external 1 MHz clock crystal. Encryption and decryption were timed at 1.378 ms and 1.400 ms respectively. Modifying this external crystal we can achieve higher performance implementations of AES, but in the process increase our power consumption.

2.5 Conclusion

μ Leech is a low power IoT side-channel evaluation platform. As such, it was optimized for minimal power consumption. The power consumption of μ Leech was optimized by utilizing the various sleep and active states of the processor. μ Leech provides a secure low power scavenging trusted cryptographic platform for any smartphone. Our design also includes interfaces to facilitate easy acquisition of high-quality power signals for Differential Power Analysis, as well as an SMA connector to manipulate the clock source. μ Leech enables security researchers to analyze leakage in next-generation mobile attached embedded devices and to develop and enroll countermeasures to protect future IoT devices.

Chapter 3

Quantized Computing: On-demand Isolation as a Power and Timing Side-channel Defense

Side-channel emanations compromise secure execution by revealing secret data [5]. There are a wide range of side-channels attackers use to extract secrets, with power [6] and timing [7] side-channels being the two most readily exploited [12]. All electronic devices generate power and timing information during execution. This information is observable by monitoring the power rail—using relatively pedestrian equipment—to capture execution emanations [6]. Attackers analyze these captures either directly or using differential analysis [6] to deduce otherwise secret information. The fundamental issue is that the power rail provides *fine-grain* information about security-critical execution.

Existing side-channel defenses focus on the attacker’s signal-to-noise ratio: increasing the noise or decreasing/hiding the signal. Countermeasures include: reducing the magnitude of side-channel emanations [30], the addition of noise to mask side-channel emanations [31], obfuscation to hide the relative timing of the secret-revealing emana-

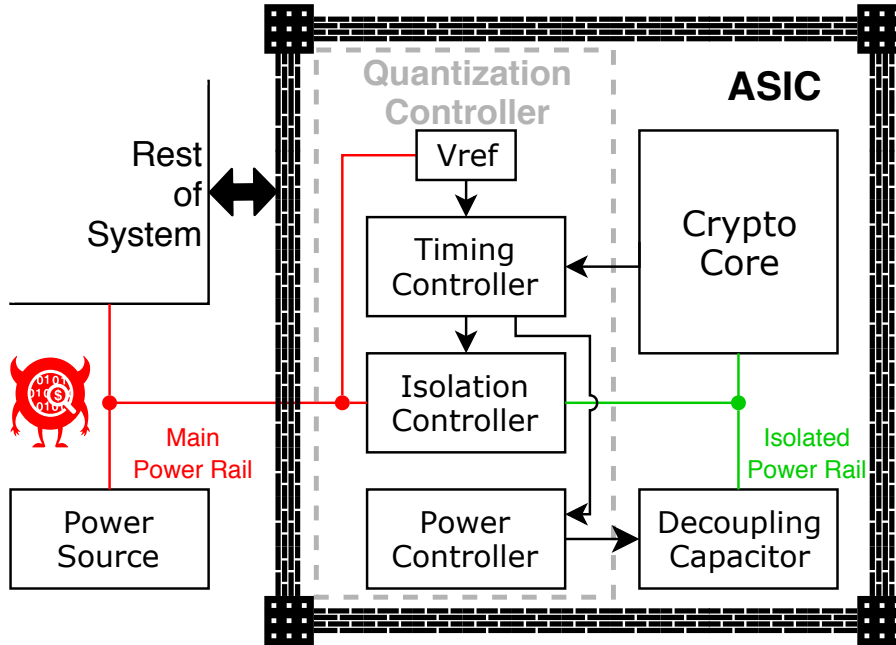


Figure 3.1: *Quantized Computing system overview.*

tion [32], and incorporating randomness at the software level [13].

Alternatively, we advocate *isolation*. Our countermeasure works by performing computation with secret-revealing side-channels in fixed-sized power/time chunks isolated from the rest of the system—and the attacker. By forcing secure execution into uniform power/time chunks, we effectively quantize the information revealed by power and timing side-channels. Thus, our approach provides control over the granularity of attacker-visible side-channel information. Assuming no secret-dependent execution-differences span quantized executions, then, from the attacker’s perspective, the power and timing requirements are uniform and secret-independent. The challenge is to design an isolation system that eliminates both power and timing side-channels.¹ We achieve via a Quantization Controller: on-chip control logic that sits between a security-critical core and the rest of the system. As shown in Figure 3.1, the Quantization Controller dictates a secure

¹Many techniques for eliminating power side-channels result in increasing the information in timing side-channels and vice versa [33, 34, 35]. Given that there is *no* added burden for an attacker capable of measuring power to measure time, a defense must address both.

processor’s power source; either from the system’s main power rail—attacker visible—or from an internal—attacker invisible—energy storage capacitor (by repurposing existing integrated decoupling capacitors). To ensure uniform (hence secret independent) secure execution times, the Quantization Controller employs execution-agnostic counters (either digital or analog). To avoid leaking information during decoupling capacitor recharge, the Quantization Controller incorporates a discharge circuit to remove any charge remaining on the decoupling capacitor after secure execution. To protect against thermal fault attacks, we use state versioning to ensure secure computation atomicity [36, 37]. Lastly, to prevent voltage fault attacks, we add an internal voltage reference.

To demonstrate the effectiveness of our approach and explore its trade-off space, we implement three Quantization Controllers: a wholly-digital, a hybrid, and a full-custom variant. For the hybrid and full-custom controllers, we use capacitors and their natural leakage as execution-independent counters—as opposed to digital logic. All three designs quantize power and timing to eliminate their respective side-channel leakage, but our analysis reveals that the hybrid design is preferred due to its compactness and simplicity. We show that our approach defends against power and timing side-channels in software implementations of AES and RSA running on an ARM Cortex-m4-based microcontroller and hardware implementations of AES and RSA on an FPGA. Our evaluation also shows that existing decoupling capacitors provide ample energy for our defense. Lastly, we analyze the software run time overheads of real software running on a complete (e.g., capacitor powered) prototype of our system.

Quantized Computing makes the following contributions:

- We use execution-independent counters to quantize secure execution time. This eliminates external fine-grain timing side-channels.
- We replace digital counters with capacitors to protect against thermal faults and

decrease area and power.

- We leverage on-demand isolation to make side-channel protection a dynamic and software-level decision.
- We design and implement three variants of our proposed Quantization Controller; we evaluate the effectiveness and efficiency of these implementations against both software and hardware implementations of AES and RSA.
- We verify our approach with a capacitor-powered prototype that we use to understand the software run time overheads experienced by software that relies on AES functionality.

3.1 Background

All computing devices generate side-channel emanations as a byproduct of physical implementation and computation [5]. The key to side-channel analysis is the interpretation of these leaks to reveal secrets. As such, there are many different side-channel leaks used to capture secrets. These include: power [6], timing [7], electromagnetic [8], acoustic [9], memory remanence [10], and thermal [11]. The two most commonly exploited side-channels in literature are power and timing [12]. The popularity of these two side-channels is due to the high bit rate and fine-grain (e.g., per clock cycle) information that they expose to the attacker. To gain access to this wealth of secret-revealing information, an attacker only needs to be able to measure the current consumed by the victim device. Given our goal is to counter these side-channels comprehensively and automatically, we address the following:

- **Power Analysis:** non-invasive analysis of the power rail during execution to reveal unintended leakage through current consumption. Depending on the signal-

to-noise ratio, attackers employ one of three power analysis techniques (in order of increasing complexity): Simple Power Analysis (SPA) [6], Differential Power Analysis (DPA) [6], and High-order Differential Power Analysis (HO-DPA) [38]. SPA is useful when the difference in current due to secure information is large relative to noise. When the signal is buried in noise, DPA uses statistical analysis of multiple executions to detect differences. To address cases of extremely small signal-to-noise ratios, HO-DPA extends DPA by correlating the statistical changes in multiple variables to uncover secret-dependent current changes. No matter the power analysis technique used, the measurement system consists of an oscilloscope and a probe; common equipment, even for university labs.

- **Timing Analysis:** non-invasive analysis of execution to reveal unintended leakage through response latency or performance. Attackers use timing analysis when an algorithm or processor has secret-dependent execution time. Possible sources of timing variation include performance optimization [39], branching and conditional statements [7], processor instructions [40], RAM [41], and cache hits [42]. The infrastructure required for timing analysis varies depending on the granularity of the timing difference in secret-dependent execution. In the most extreme case, timing analysis requires the same complexity setup as power analysis.
- **Fault Attacks:** induce faults via unexpected environmental conditions to reveal unintended leakage through inconsistent system state. Depending on the target system there are many ways of inducing faults, including: temperature [43], power [19], overclocking [44], electromagnetic fields [45], and ionizing radiation [46]. Our design defends against temperature and power fault attacks since these can have a targeted effect and are available to a side-channel attacker.

Addressing all three threat vectors at once is a challenging task, but essential for

true improvements in security. Creating a comprehensive countermeasure is challenging due to the inherent reciprocal relationship between side-channels. Mitigating one side-channel in isolation often results in shifting the information to another side-channel. For example, consider a *broken* version of our approach that uses only isolation from the power rail to protect against fine-grain power analysis. This means that secure execution time—hence the time the crypto core is disconnected from the power rail—is dictated by secret information. Thus, the attacker sees a square wave on the power rail that divulges the exact execution time of the secure execution. Therefore, we must eliminate both power and timing side-channels and guard against thermal and power fault attacks.

3.1.1 Decoupling Capacitors

System designers add decoupling capacitors to an Integrated Circuit (IC) to filter out noise from the power supply. Conceptually, the power supply provides a constant voltage to the chip. Practically, the highly-variable load caused by different amounts of switching transistors results in voltage drops as the power supply struggles to supply enough current to maintain the desired voltage. Making matters worse, power rails are shared by many ICs on the same Printed Circuit Board (PCB). Large voltage drops cause the IC to have a power fault; resulting in an inconsistent state or a reset.

To protect against voltage drops, system designers add capacitors between the power rail and the IC. These capacitors decouple the IC from the power rail, acting as a filter for power supply noise; when voltage would drop, the decoupling capacitor sources additional charge to maintain the desired voltage. We leverage the observation that even when the power rail goes to 0V, *the decoupling capacitor supplies enough current to keep the IC running for a short amount of time.*

Traditionally, system designers add decoupling capacitors external to the IC as part of the PCB layout. Recent advances in IC integration enable adding decoupling capacitors

inside the IC [47, 48, 49]. Internally-integrated decoupling capacitors have the advantage of lowering the cost of adding the IC to a PCB, reducing the chance of layout errors, and providing IC designers quality guarantees about incoming power. We leverage internal decoupling capacitors to hide current consumption information from an attacker who is unable to non-destructively interpose between them and the IC.

3.2 Threat Model

Our threat model assumes an attacker with knowledge of the software running on as well as with full physical access to an IC. The attacker’s objective is to extract secret information through non-destructive, non-invasive, timing and power side-channel analysis. We assume that the attacker can also control the power supplied to the device as well as its environment to induce power and thermal faults.

On the other hand, we assume that the attacker has no capability to non-destructively interpose between the crypto core and the Quantization Controller. Traditionally, the external nature of decoupling capacitors made it easy for attackers to interpose by removing them from the circuit board. With the recent trend of integrated decoupling capacitors, they are now an integral part of the circuit itself, making interposing destructive. In cases of nation-state adversaries, it is common to employ anti-tamper techniques to prevent attackers from non-destructively analyzing the contents of the chip [50].

Following our observation on isolating integrated decoupling capacitors, we develop a comprehensive and mostly automatic power and timing side-channel defense that is immune to thermal and power fault attacks.

3.3 Quantization Controller Design

At the core of our Quantization Controller design is the observation that isolating integrated decoupling capacitors from the external power rail prevents the attacker from seeing fine-grain power side-channel information. Unfortunately, isolation alone is insufficient as coarse-grain information still reveals secrets. To form a comprehensive solution that addresses power and timing side-channels completely, we layer on top of isolation components that mask variations in timing and power by ensuring that the power and time required by secure execution is execution-invariant—from the attacker’s perspective. Lastly, we address several thermal and power fault attacks created by our approach. This section details each component of our Quantization Controller.

3.3.1 Isolation Controller

The trend of integrating decoupling capacitors into the Integrated Circuit (IC) enables our approach. We observe that decoupling capacitors provide enough energy for modern low-power devices to continue execution for a short period after all power is removed. This presents an opportunity, because, as our evaluation shows (Section 3.6), decoupling capacitors provide enough energy to execute popular cryptographic algorithms. As Figure 3.1 shows, our Isolation Controller uses this opportunity by selectively isolating (in an electrical sense) a crypto core from the main power rail when it performs secure execution (i.e., execution that requires protection from power and timing side-channels). During regular (i.e., insecure) execution, the external power rail continuously provides power to both the crypto core and the decoupling capacitor; allowing the decoupling capacitor to charge. Isolation is triggered by the crypto core when it is ready to start secure execution, allowing the software to balance security guarantees and performance dynamically. *Thus, the Isolation Controller ensures that all fine-grain power and timing side-channel*

information generated by the crypto core is hidden from an attacker with access to the main power rail and only exposed as coarse-grain information. System designers have the ability to control the granularity of side-channel information exposed to an attacker by changing the size of the decoupling capacitor.²

3.3.2 Timing Controller

Although isolating the crypto core from the main power rail eliminates fine-grain power and timing side-channels, it makes coarse-grain power and timing side-channel information more evident to the attacker. One source of coarse-grain side-channel information is the amount of time required for secure execution; an attacker measures this as the amount of time the crypto core is isolated from the main power rail. The mitigation for coarse-grain timing side-channels is computation-invariant secure execution length.

To eliminate coarse-grain timing side-channels, the Quantization Controller includes a Timing Controller that uses a timer to equalize time spent in secure execution—making it computation invariant. When the crypto core requests secure execution, the Timing Controller resets the timer to a seed value. As secure execution occurs, the timer decrements. Once the timer reaches zero, the Timing Controller ends secure execution by reconnecting to the main power rail. The only constraint is that the secure computation completes before the timer reaches zero. In Section 3.3.4 we employ data versioning to address cases where, through fault attacks or poor software design, secure computation does not complete before reconnecting to the main power rail.

From a high level, the Timing Controller quantizes secure computation into a series of fixed-time isolated executions. *Thus, adding the Timing Controller makes each secure execution constant time with respect to an attacker.*

²Larger decoupling capacitors have benefits beyond enabling longer periods of secure execution in our approach: larger decoupling capacitors are more effective at filtering power supply noise and are also a power source for a range of anti-tamper techniques.

3.3.3 Power Controller

While adding the Timing Controller eliminates coarse-grain timing side-channels, coarse-grain power side-channels remain. One example of a coarse-grain power side-channel that isolation creates is the amount of charge required to recharge the decoupling capacitor once the Timing Controller reconnects to the main power rail. The amount of charge required for recharge indicates how much energy the crypto core consumed during secure execution—revealing secret-dependent differences in computation. Uniform-time secure executions do not mitigate this threat, because, from an energy perspective, they act as a constant energy expenditure (the idle current times the time of the secure execution) on top of which a security-dependent delta energy is added.

To eliminate coarse-grain power side-channels, the Quantization Controller includes a Power Controller that ensures that secure executions are uniform energy from an attacker’s perspective. The Power Controller achieves this by removing any remaining charge from the decoupling capacitor before reconnecting to the main power rail. We call this Discharge Mode. To support Discharge Mode, we add a timer to the Timing Controller. Going into secure execution mode is the same, except both timers are loaded with their seed value. When the first timer reaches zero, the Timing Controller goes into Discharge Mode, shorting the decoupling capacitor. When the second timer reaches zero, the Timing Controller goes into Recharge Mode by connecting the decoupling capacitor to the main power rail. The difference in seed values between the two timers must guarantee that there is enough time to completely discharge the decoupling capacitor.

Adding the Power Controller makes our secure execution both uniform in time and power with respect to an attacker.

3.3.4 Fault Attacks

Although the goal of our design is to eliminate power and timing side-channels, we realize that our defense also defeats power-based fault attacks, as well as creating a range of new attacks based on injecting faults around secure execution. Thus, our design also addresses a range of fault attacks: Power [19], Memory [20], and Thermal [11].

By isolating secure execution from the main power rail, the Quantization Controller prevents fault attacks induced through power glitching during secure execution. Power glitching attacks result in a range of malicious effects: instruction failing [51], instruction skipping [52], propagation delays [53], and clock skewing [19]. Power faults require the attacker to have direct control over the crypto core’s power source during secure execution. Typically, attackers reduce the voltage dramatically and for a short duration; this potentially skips individual or a range of instructions. This technique is useful for skipping loop iterations or security-critical checks. In addition to effecting instruction execution directly, power fluctuations can also effect execution indirectly through clock frequency manipulation. Decreasing the supply voltage reduces the drive strength of transistors; resulting in increased signal transition times. By decreasing supply voltage enough, attackers can cause the signal propagation delay to be longer than a clock cycle. This causes subtle execution errors. *By powering the crypto core using an internal power supply (i.e., integrated decoupling capacitor), our approach eliminates any direct influence of the attacker over the power rail.*

Though isolation addresses power-based fault attacks during secure execution, our approach (up to this point) enables power-based fault attacks outside of secure execution. Attackers can manipulate the power source before secure execution starts, allowing them to control the amount of energy stored in the decoupling capacitors. Through this control, attackers can cause power to run out before secure execution completes. The timing of power loss and the intermediate software state exposes secrets. Another threat is that

the crypto core assumes that secure execution completed and uses incomplete results for future computation. For example, with the Chinese Remainder Theorem should an attacker compromises either prime number p or q , the other prime is extractable [54].

Algorithm 1 Secure Mode

Input: Mode, Vref

Effect: Isolated Execution

```
1: if Mode == SECURE then
2:   if Vref == OKAY then
3:     Isolate = 1 // Isolate internal power rail
4:     Recharge_Timer = 0 // Initialize timer
5:     Discharge_Timer = 0 // Initialize timer
6:     ...Isolated Secure Execution...
7:     ...Commit results to non-volatile memory...
8:   else
9:     // attempted undervolt attack
10:    Isolate = 1 // Isolate internal power rail
11:    Discharge = 1 // Activate discharge circuit
12:   end if
13: end if
```

To address fault attacks that leverage premature, unexpected, secure execution termination, we implement state versioning. With state versioning, the crypto core prepares a temporary state that it operates on during secure execution. The last action of secure execution is a commit of the temporary state to a permanent state by a writing to a flag variable that software reads when resuming execution after secure execution. In the event of incomplete secure execution, software sees that the flag was not updated and retries secure execution. By doing this, *state versioning provides to software the notion of secure execution-level atomicity*.

Unfortunately, this creates another potential side-channel as the attacker can use the lack of forward progress and their control over the voltage of the main power rail to systematically identify how much charge (as voltage across the decoupling capacitor) is required for the crypto core to complete and commit secure computation. To eliminate this threat, we add an internal reference voltage circuit (V_{ref} in Figure 3.1). When the crypto core signals to start a secure execution, the Timing Controller checks the internal

reference voltage to ensure that the attacker has not been undervolting the main power rail. In the event of attack, Discharge Mode is started to reset the core. From a high level, *V_{ref}* guarantees that there is enough charge stored in the decoupling capacitor to start secure execution by bounding the attacker’s influence over the initial energy stored in the decoupling capacitor.

Another mechanism—albeit coarse-grain—for inducing premature termination of secure execution is a thermal attack. Observe that digital circuits (i.e., the Quantization Controller’s timers³) are mostly immune to thermal variation, while analog circuits (i.e., the decoupling capacitors) are not. By increasing temperature in a targeted way, an attacker increases the power requirement of the crypto core and the natural leakage of the decoupling capacitor [43]; the digital counters used for controlling the time of secure execution are unaffected. This results in power loss before software completes its secure execution. *State versioning protects against premature power loss.*

Freezing attacks on digital counters also possible. Attackers can desynchronize the timing between the secure execution and our timers. *Replacing digital timers with leakage-based capacitor timers allows our timers to be affected in the same way as the power capacitor.* Freezing either hybrid or analog designs causes all three capacitors to change in the same way—maintaining the expected ordering of events.

Algorithm 2 Discharge Mode

Input: Discharge_Timer

Effect: Uniform Power

```
1: if Discharge_Timer == CONSTANTD then  
2:   Discharge = 1 // Activate discharge circuit  
3:   ...Short-circuit decoupling capacitor..  
4: end if
```

³In Section 3.4.3, we present the idea of using capacitors for time tracking due to their compactness and consistency in behavior to decoupling capacitors given thermal fault attacks.

Algorithm 3 Recharge Mode

Input: Recharge_Timer

Effect: Uniform Time

```
1: if Recharge_Timer == CONSTANTR then  
2:   Isolate = 0 // Connect to main power rail  
3:   Discharge = 0 // Disable discharge circuit  
4:   ...Insecure execution...  
5: end if
```

3.3.5 End-to-end Flow

Previous sections discuss the individual components that comprise our approach, now we describe how they work together to form our Quantization Controller. From a high level, the Quantization Controller operates in one of three modes: Secure Mode (Algorithm 1), Discharge Mode (Algorithm 2), and Recharge Mode (Algorithm 3). The system powers-on in Recharge Mode where non-security-critical computation occurs and the decoupling capacitor charges in preparation for the next secure execution. The Isolation Controller connects the decoupling capacitor to the main power rail and the Power Controller disconnects the shorting circuit. This is the only time where the attacker has access to fine-grain power and timing side-channel information. When the crypto core signals, through Mode, that it needs to execute securely, the Timing Controller resets both timers and commands the Isolation Controller to disconnect from the main power rail. After a secure-computation-invariant amount of time passes, Discharge_Timer fires. This causes the Power Controller to connect the decoupling capacitor to the shorting circuit to remove all charge. Finally, after additional secure computation-invariant time, Recharge_Timer fires, causing a return to Recharge Mode.

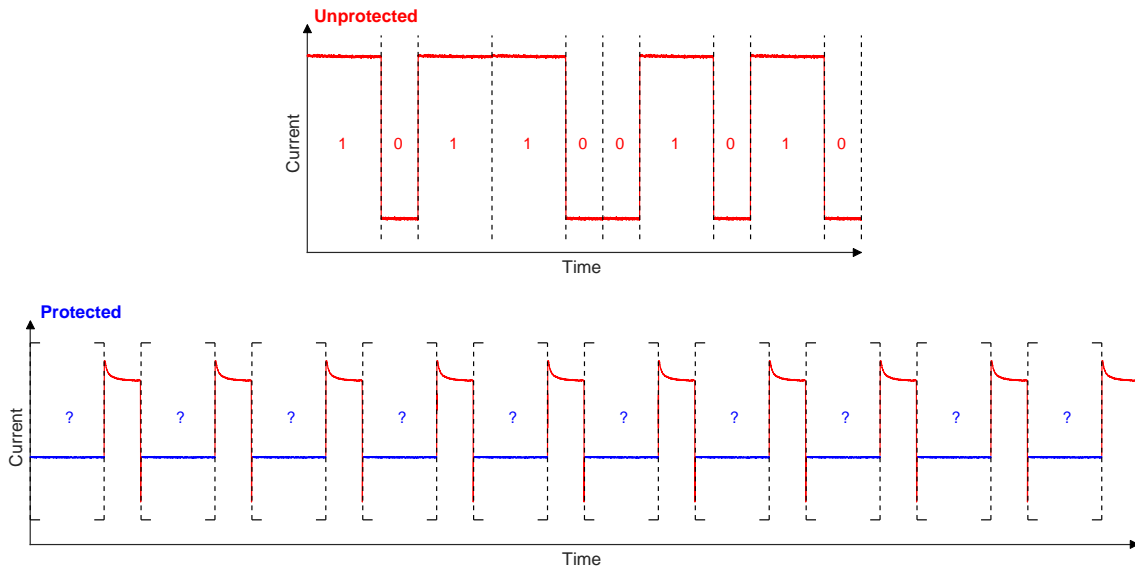


Figure 3.2: Illustrative of the Quantization Controller’s effectiveness against power and timing side-channels for RSA. Note that the noise/variation is hard to see due to scaling.

3.3.6 Illustrative Example

To illustrate the impact our design has on an side-channel attacker, we compare two runs of RSA: one unprotected and one protected by the Quantization Controller. As the shown in Figure 3.2, for the unprotected case, when a key bit is 1, RSA executes modular exponentiation, which consumes more time and current than a 0 bit. This enables an attacker to extrapolate the key based on both power and timing side-channels. Portions of the trace shown in red represent when the attacker gains meaningful information and portions in blue represent when the attacker sees only noise from the main power rail. The span between vertical lines represents the time required by that round of RSA. Notice how each round in the protected case is uniform in length, while it is secret-dependent in the unprotected case. Also notice, that even though there is current information available to the attacker during recharge, recharge always starts with the same current (i.e., the decoupling capacitor is at the same voltage at the start of Recharge Mode). Thus the Quantization Controller eliminates power and timing side-channels.

3.4 Quantization Controller Implementation

As Figure 3.1 shows, we implement the Quantization Controller in hardware, as a part of an existing crypto core’s Integrated Circuit (IC). The Quantization Controller interposes between the main power rail (attacker visible) and the internal power rail (attacker invisible). This allows the Quantization Controller to control when power side-channel information is exposed. In addition to the input from the main power rail, the Quantization Controller takes a one wire input from the crypto core. This wire carries the `Mode` information used by the Quantization Controller in Algorithm 1 to determine when to start secure execution. Lastly, the Quantization Controller includes a power output used to connect the integrated decoupling capacitor to the discharging circuit. These connections enable the Quantization Controller to eliminate power and timing side-channels, for arbitrary crypto cores, on-demand.

The Quantization Controller consists of three sub-controllers: timing, isolation, and power. The `Timing Controller` ensures that secure execution takes a secret-independent amount of time, the `Isolation Controller` manages isolation from the main power rail, and the `Power Controller` ensures that secure execution takes a secret-independent amount of energy. From a high level, these controllers are a combination of counters and switches. From a low level, we describe three implementation strategies: wholly digital, full-custom analog, and a hybrid design. A driving observation of our implementations is that *capacitors act as area-efficient counters*. We validate each implementation using a combination of Register Transfer Level (RTL) simulation (for digital logic) and Simulation Program for Integrated Circuits Emphasis (SPICE) simulations (for analog circuits). While each implementation strategy eliminates power and timing side-channels, they present tradeoffs in terms of complexity and hardware overheads. Table 3.1 details the tradeoff space between the three variants.

	Wholly Digital	Analog	Hybrid
Cells	105	3	5
Area	221	8	14
Lines HDL	65	0	29
Design Effort	low	high	low
Integration Effort	low	high	medium

Table 3.1: Tradeoff space of the three Quantization Controller implementations using a 22nm process technology.

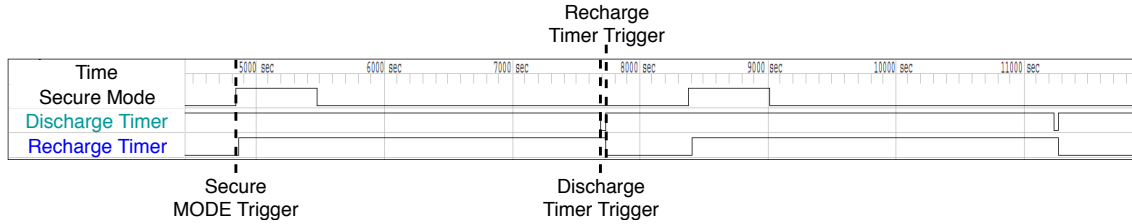


Figure 3.3: RTL simulation of the wholly digital and hybrid versions.

3.4.1 Wholly Digital Implementation

The wholly digital implementation of the Quantization Controller uses only digital logic to create the required counters and switches. The advantages of using only digital logic are that it is much easier to design and tune compared to using analog components and it is easier for hardware designers to integrate into their designs. The disadvantages are that digital logic requires logic to save and update potentially large counters and using digital counters to track time presents the opportunity for an attacker to use thermal fault attacks to invert the expected relationship between the timing of power running out and secure execution completing.

Figure 3.3 shows our RTL-level validation of the wholly digital Quantization Controller. The Quantization Controller powers-on in Recharge Mode. Eventually the crypto core triggers Secure Mode by driving Mode to 1. This causes the Isolation Controller to disconnect the crypto core from the main power rail, isolating it. Eventually software completes its security-critical computation. At a later—secret-invariant—time, the Timing Controller signals to the Power Controller to go to Discharge

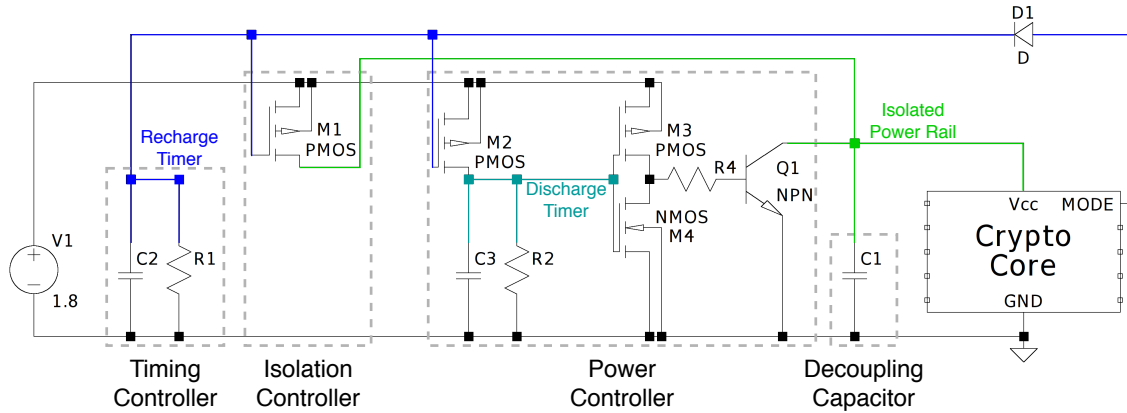


Figure 3.4: Full-custom analog Quantization Controller circuit.

Mode. The Power Controller connects the decoupling capacitor to the short-circuit, draining all energy. After additional secret-invariant time, the Timing Controller signals to the Isolation Controller to return to Recharge Mode.⁴

3.4.2 Analog Implementation

Figure 3.4 shows our full-custom analog Quantization Controller implementation. An analog implementation enables two optimizations not possible in the digital implementation. First, we replace expensive digital counters with capacitors (C2 and C3) to control our PMOS recharge (M1) and discharge (M2) switches directly. The observation that makes this possible is that the natural leakage of capacitors creates a countdown effect, albeit in the analog domain. When the capacitor’s voltage is feed to PMOS logic, there is a voltage where the digital value changes from 1 to 0. We treat this event as the counter firing. This enables us to replace hundreds of logic gates required by digital counters with two capacitors.⁵ Beyond their compactness, capacitor-based counters have an advantage

⁴There is no need for a counter to determine how long to stay in recharge mode because software controls that transition. This avoids adding unnecessary overhead when security is not required and reduces Quantization Controller complexity. A pull-down resistor ensures that Mode wire defaults to 0V (aka insecure execution).

⁵We do *not* add capacitors; because our capacitor-based counters only need to track small amounts of time and only lose energy due to leakage, their capacitance values are very small. To implement our

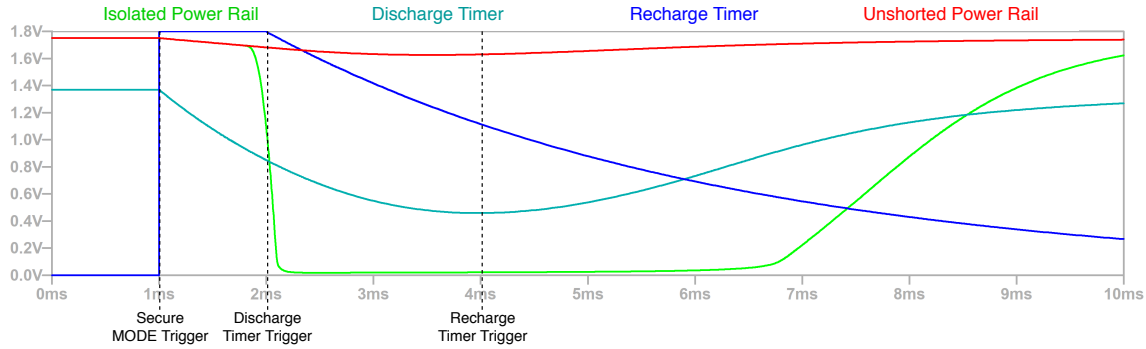


Figure 3.5: SPICE simulation of the full-custom analog version.

in that they encode a relative notion of time with respect to each other and the decoupling capacitor—a critical property given temperature-based fault attacks.

The second optimization made possible by our analog implementation is we use `Mode` from the crypto core to control directly our timing counter. When the crypto core initiates secure mode by driving `Mode` to 1, this powers the recharge counter (C2). When the discharge counter (C3) fires, it short-circuits the decoupling capacitor (C1), that in-turn browns out the crypto core. This causes `Mode` to return to its default value of 0. This causes the recharge counter (C2) to start (via leakage). This optimization reduces the size of both the recharge and discharge counters by having them run sequentially as opposed to concurrently.

The tradeoff with these optimizations is increased design and integration complexity. Since the counter capacitors are directly controlling the switching PMOS logic, they do not switch instantly but gradually as the capacitor continues to discharge. Thus, hardware designers must simulate the analog domain switching behavior to ensure correct operation.

Figure 3.5 shows our SPICE simulation-based validation of the full-custom analog Quantization Controller. The crypto core starts executing in a continuously-powered fashion, with side-channels exposed. This mode, while insecure, allows the **decoupling capacitor-based counters** we vary the implementation of transistors and the wires used to connect them [55].

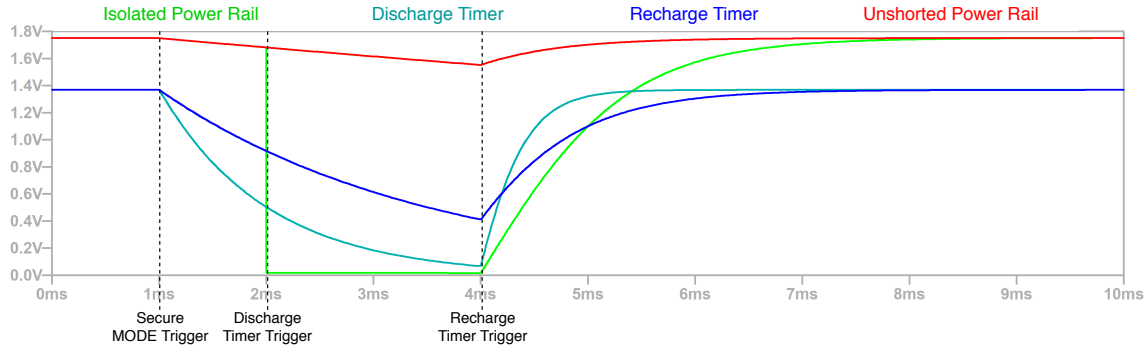


Figure 3.6: SPICE simulation of the analog parts of the hybrid version.

capacitor (C1), recharge counter (C2), and the discharge counter (C3) to charge. The crypto core then signals to the Isolation Controller that it wants to perform security-critical computation by driving Mode to 1. This disconnects the decoupling capacitor from the main power rail—isolating the crypto core from the attacker. Now disconnected, the decoupling capacitor begins to discharge due to crypto core computation and leakage. Eventually enough charge dissipates from the decoupling capacitor such that the processor browns out due to insufficient voltage. After enough secret-invariant leakage of the discharge counter capacitor, the Power Controller is triggered shorting the decoupling capacitor, ensuring that the amount of energy required for recharge is secret-invariant.⁶ At a later time, dictated by the recharge counter capacitor’s leakage, the Power Controller disconnects the short circuit and the Isolation Controller reconnects to the main power rail. No matter the load used to simulate secure execution, all events except the time the discharge trigger (which is not attacker observable) and the recharge trigger occur at the same time and all capacitors have consistent charge times when reconnected.

3.4.3 Hybrid Implementation

Incorporating most of the benefits of the reduced complexity and ease of integration of the digital implementation and the compact and thermal fault resistant capacitor-based counters of the analog implementation is our hybrid implementation. As Table 3.1 shows, by replacing the digital counters with capacitors, the hybrid implementation approaches the minimal area of the heavily-optimized analog implementation, without having to deal with analog circuit behavior. What is not shown is the complexity of the implementation. By replacing the counters with capacitors, the hybrid implementation reduces the amount of HDL code as well: from 65 lines for the wholly-digital implementation to 29. The disadvantage of the hybrid implementation compared to the digital implementation is the added burden on the hardware designer to create appropriately sized capacitors as part of their IC design.

Figure 3.6 shows the SPICE simulation of the hybrid implementation. Unlike the analog implementation's SPICE simulation (Figure 3.5), the transitions between states are instantaneous. This is evident when the `discharge` and `recharge` timer capacitors trigger the `Power Controller` and the `Timing Controller`, respectively. The `decoupling` capacitor is discharged instantly and starts recharging instantly for the hybrid simulation, while gradually discharging and recharging in the analog simulation. The instantaneous transitions obviate the need for analog domain simulation (e.g., SPICE).

3.5 Selecting a Crypto Core

With the design and implementation of the Quantization Controller set, the next step is to incorporate it into a real system. There are two benefits to this: (1) understanding how

⁶Had the decoupling capacitor not been shorted, it would have continued to discharge as indicated by the `unshorted power rail`.

crypto core properties affect Quantization Controller performance and (2) to show that our Quantization Controller is effective at eliminating power and timing side-channels.

The first step of a full-system implementation is deciding cryptographic exemplars. AES [56] and RSA [57] are ideal examples: (1) they are two of the most popular cryptographic algorithms in use today; (2) both have known power [58, 59] and timing [60, 7] side-channel vulnerabilities; and (3) from a benchmarking perspective, AES and RSA explore complementary aspects of system performance. AES is memory bound as it relies on many table look-ups and simple computation. Alternatively, RSA is compute bound as it relies on complex modular exponentiation on register values. For these reasons, we base our full-system implementation and evaluation on 128-bit AES and 64-bit RSA executions. We avoid longer keys because it is unlikely that integrated decoupling capacitors for the foreseeable future will be able to support them in a single secure execution and the trends shown for shorter keys are true for longer keys.

Both AES and RSA are implementable as software running on a processor or as hardware accelerators. Given that our approach is agnostic to whether the crypto core is a general-purpose processor that runs cryptographic algorithms or a hardware accelerator, we focus on software implementations of AES and RSA.

3.5.1 Capacitor Bank

The centerpiece of our approach is the decoupling capacitor. Our approach works only when the decoupling capacitor is able to provide enough charge to complete meaningful security-critical computation (i.e., all secret-dependent execution differences converge). Thus, when evaluating system design options it is essential to base the evaluation on maximizing the amount of computation possible given the size of existing integrated decoupling capacitors. To this end, we derive an equation for calculating the required capacitance based on features of the crypto core and secure execution.

Capacitance (C) depends on charge (Q) and voltage (V).

$$C = \frac{Q}{V} \quad (3.1)$$

The total amount of charge (Q) is given by multiplying average current (I) by the length of computation (Δt).

$$C = \frac{I\Delta t}{V} \quad (3.2)$$

Since most devices do not use all the charge held by the capacitor (because they fail to operate below some non-zero voltage), we replace the voltage (V) with a value that accounts for the operational voltage range of the crypto core (ΔV).

$$C = \frac{I\Delta t}{\Delta V} \quad (3.3)$$

Equation 3.3 defines the relationship between average current, voltage range, and computation time for a capacitor. By plugging the crypto core and secure computation measurements into Equation 3.3, we can solve for the capacitance required for secure execution. Since the goal is to minimize the capacitor required for a unit of computation, system design decisions should focus on minimizing the average current and total time of that computation, while maximizing the operating voltage range of the device performing the computation.

3.5.2 Preliminaries

One way to implement cryptographic algorithms is as software running on a general-purpose processor. Software implementations, while being slower than hardware implementations, are updateable, have the ability to support a variety of algorithms, as well as allowing the processor to run a larger application with non-security-critical functionality.

Given software implementations of AES and RSA, this sections explores the impact on required capacitor size given key processor properties. We divide the discussion into two parts, one focused on computation and another focused on memory. For each part, we relate processor properties to the effect they have on each term on the right hand side of Equation 3.3.

To explore the trade-space of processor properties, we analyze AES and RSA software implementations on three microcontrollers. We select the three microcontrollers based on their popularity and their coverage of the microcontroller design space.⁷ Table 3.2 provides a summary of our findings for each microcontroller. Remember that the goal is to maximize the amount of computation that can fit within the range of existing embedded processor decoupling capacitors.

3.5.3 Computation Effects

3.5.3.1 Computation Time (Δt)

Reducing the time required to perform a given security-critical computation (Δt) reduces the numerator in Equation 3.3, resulting in a smaller capacitor (C). The two most significant properties of a microcontroller that dictate how long a given computation takes are the maximum clock frequency (Max clk) and microcontroller performance (ISA Efficiency). Multiplying these two values provides a measure of peak performance. Further, any secure computation can be expressed as a worst-case number of instructions, and dividing this value by the peak performance reveals an estimate for Δt .

Maximum clock frequency is a product of several hardware design decisions, including, process technology, targeted deployments, and micro-architectural complexity. Sub-32-bit microcontrollers focus on cost where 32-bit microcontrollers focus on perfor-

⁷We focus on microcontrollers as opposed to desktop or mobile processors because current and future decoupling capacitors are not able to provide enough energy for such high-power processors to execute.

Processor	AT32UC3L064	MSP432P401R	MSP430FR6989
Max CPU clk MHz	50	48	16
ISA bit width	32	32	16
ISA Efficiency DMIPS/MHz	1.28	1.20	0.288
Voltage Range V	1.62 - 3.6	1.62 - 3.7	1.8 - 3.6
Power Efficiency μ Amps/MHz	165	80	100
NV-Mem Type	Flash	Flash	FRAM
NV-Mem write time μ s/64B (AES)		88.4	66.8
NV-Mem write current μ A @ 64B (AES)		4.07	2.14
NV-Mem write time μ s/32B (RSA)		66.9	31.2
NV-Mem write current μ A @ 32B (RSA)		3.97	2.11

Table 3.2: Summary of key properties of the processors evaluated.

mance, power, and size. By focusing on cost, sub-32-bit microcontrollers tend to target toys and bare-bones appliances that have very low performance requirements. The primary way to reduce cost is to fabricate the microcontroller using larger and slower transistors (i.e., an older process node). This reduces the maximum clock frequency. Another reason for reduced clock frequency is the number of pipeline stages; the MSP430 is unpipelined [61] while the UC3L0 [62] and the MSP432 [63] have three stage pipeline stages. Pipelining increases the maximum frequency while also increasing development costs and current consumption.

Microcontroller performance, on the other hand, is dictated by instruction set design and word-width. Instruction set design dictates how many instructions it takes to execute a given cryptographic operation. For example, RSA relies on modular arithmetic. Unfortunately, implementing a modulus operation on the MSP430 is expensive because it does not have a modulus instruction. The UC3L0 and the MSP432 do, making them have to execute fewer instructions to perform one round of RSA. A second consideration—especially for cryptographic algorithms that deal with large numbers—is the word width of the microcontroller. The UC3L0 and MSP432 have twice the word size of the MSP430, thus they have roughly twice the throughput when working with large data words.

Plugging in the numbers from Table 3.2 into Equation 3.3 shows that both the UC3L0 and MSP432 are the best options for reducing computation time (Δt). In the case of our three processors, there is a correlation between maximum clock frequency and ISA efficiency that makes peak performance differences more exaggerated than individual property differences. Thus, keeping all variables besides Δt in Equation 3.3 equal, the MSP432 and the MSP430 require an 11% and 1289% larger capacitor compared to the UC3L0, respectively.

3.5.3.2 Average Current (I)

Reducing the average current (I) reduces the numerator in Equation 3.3, resulting in a smaller capacitor (C). The current consumption of the crypto core dictates the rate at which charge is used from the capacitor. To approximate the average current of each microcontroller, we multiply the Power Efficiency and the Max CPU clk in Table 3.2. This number tells us the expected current for the clock frequency that minimizes computation time (Δt) for a given secure computation.

Notice that both the computation time (Δt) and average current calculations use Max CPU clk, but one as a divider and one as a multiplier. Given this, the way to reduce average current—without increasing computation time—is to increase the power efficiency of the microcontroller. There are two ways hardware designers accomplish this: move to a smaller process node and/or power/clock gate the design. A smaller process node reduces the current required to switch transistors (although it increases the static current). Clock gating prevents transistors from switching when their results are not needed. Finally, power gating removes both static and dynamic current consumption of transistors that are not needed.

The MSP432, being newer, uses a more recent process node than the other two. The designers of the MSP432 also focus more power gating to reduce current consumption. Even though that makes it the most power efficient, the MSP430 actually has the lowest peak current. This is mainly due to its low Max CPU clk. Plugging the average currents into Equation 3.3, keeping all other variables equal, the UC3L0 and the MSP432 require a 416% and 140% larger capacitor compared to the MSP430, respectively.

3.5.3.3 Voltage Range (ΔV)

Increasing the operating voltage range of the crypto core (ΔV) increases the denominator in Equation 3.3, resulting in a smaller capacitance (C). To better understand why this is

the case, recall that the amount capacitance is fixed, therefore as a capacitor loses charge (through leakage or by powering computation), the voltage across the capacitor decreases (Equation 3.1). Unfortunately, as voltage decreases, digital components begin to act as analog components, causing them to fail. Given that this failure occurs when there is a non-zero voltage across the capacitor, the capacitor still holds charge. This charge is wasted. Thus, we can only consider the amount of charge that covers the operating voltage range of the core when sizing the capacitor; wider voltage ranges are better.

For microcontrollers, the primary factor influencing voltage range is the internal voltage regulator. Voltage regulator options are common across microcontrollers; meaning all three microcontrollers have similar voltage ranges. Thus, the UC3L0 and the MSP430 require a 5% and 16% larger capacitor than the MSP432, respectively.

3.5.3.4 Summary

When considering only the effects of computation, the MSP432 is the preferred microcontroller; the UC3L0 requires a capacitor twice as large, while the MSP430 requires a capacitor six times as large.

3.5.4 Memory Effects

After secure computation completes, the crypto core must commit results to non-volatile memory before it loses power and all volatile state disappears. Depending on non-volatile memory technology, this can require as much charge as the computation itself. Thus, in this section we explore non-volatile memory technology's impact on capacitor size.

The microcontrollers we evaluate offer either Flash or FRAM non-volatile memory. Flash is the industry standard, while FRAM is a newcomer targeted at ultra-lower-power and energy harvesting devices. FRAM advantages include lower power, single cycle writes. Alternatively, Flash comes in larger sizes and services reads at three times the

frequency. Given that both the MSP432 and the UC3L0 use Flash, but the MSP432 has a lower compute cost, this section compares the Flash on the MSP432 against the FRAM on the MSP430.

3.5.4.1 Write Time (Δt)

Write time is the most significant factor that differentiates FRAM and Flash. Writing to FRAM requires changing the polarity of a magnetic field. This is possible at low current and low voltage [64]. On the other hand, writes to Flash require much more time because they must collect enough current to create a high enough voltage to force charge across a dielectric. The Flash controller must then hold this high voltage for enough time for sufficient charge to flow to change the cell's state.

Normally, every write to Flash memory incurs this waiting time. Fortunately, the MSP432 includes an optimization that allows software to write to Flash in 16 byte bursts, up to four at a time. Doing this amortizes a single waiting time across many contiguous writes. Figure 3.7 shows the impact of this optimization.⁸ From a high level, the ability to amortize the writes to Flash significantly impacts software and how the MSP432 compares to the MSP430. From a microcontroller comparison perspective, Flash is prohibitively expensive when committing state in anything other than multiples of 16-bytes. Thus, software may need to incorporate padding to leverage 16-byte bursts. For example, AES requires 32 bytes for intermediate results and 1 bit for version tracking. To minimize write time we add padding to fill up 48-bytes.

Another takeaway from Figure 3.7 is that increases in clock frequency do little to hasten Flash writes. This is because the time spent waiting for the write to complete far exceeds the time spent moving data between buffers. A similar trend exists for FRAM,

⁸Notice that even though the burst size is 16-bytes, Figure 3.7 shows that there is no speed-up for 16 bytes. We identified a bug in TI's Flash driver when it checks the size of the write to see if it can use a burst write: a < is used instead of a <=. This causes bursts to work for multiples of 16 bytes, but not for a single 16-byte burst.

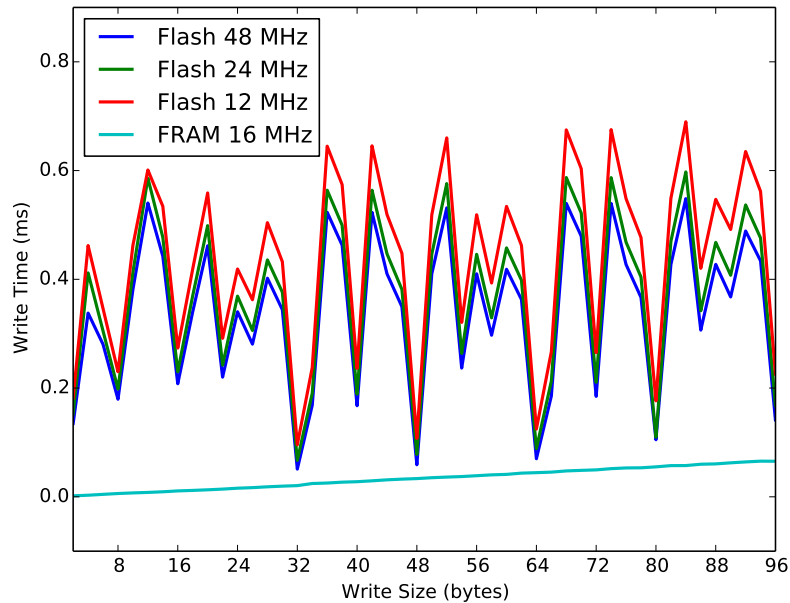


Figure 3.7: Time required to write to Flash and FRAM.

but the critical frequency is $8MHz$ as that is the maximum frequency of FRAM.

Plugging the write times from Table 3.2 into Equation 3.3, keeping all other variables equal, writing to the Flash memory on the MSP432 requires a 32% and 114% larger capacitor for AES and RSA compared to the MSP430, respectively.

3.5.4.2 Average Current (I)

While the current required by Flash and FRAM is similar for read operations, write currents are much higher for Flash. Writing values to Flash memory involves flipping 1 bits to 0 bits, as appropriate. Flipping a 1 to a 0 requires a high enough voltage to force charge to flow across a dielectric; this voltage is much higher than the chip's supply voltage. To create a sufficiently high voltage, the Flash controller uses a charge pump to essentially convert high current into high voltage.

Table 3.2 shows the average Flash and FRAM write currents for a range of write

sizes. The results show that writing to FRAM requires similar current to performing computation, while writing to Flash almost doubles the current required compared to computation. Plugging the average currents into Equation 3.3, keeping all other variables equal, writing to the Flash memory on the MSP432 requires up to a 278% larger capacitor compared to the MSP430.

3.5.4.3 Voltage Range (ΔV)

The crypto core must commit the results of secure computation to non-volatile memory at the end of that secure computation—otherwise, the results will be lost when power cycles. In the case of microcontrollers, they include brown out circuitry that ensures that if it is on, voltage is sufficient to write to non-volatile memory. Thus, the operating voltage range for memory is the same as it is for computation.

3.5.4.4 Summary

When considering only the effects of non-volatile memory accesses, the MSP430 is the preferred microcontroller; depending on the amount of data that software needs to commit at the end of secure computation, the MSP432 requires between 118% and 249% as large of a capacitor as the MSP430.

3.5.5 Analysis and Recommendations

The MSP432 is better for compute-limited secure computation, but the MSP430 is better for memory-limited secure computation. This is a trade space that system designers can exploit when selecting a microcontroller for Quantized Computing deployments. Combining our analyses into a single equation reveals that when the secure computation spends 70% of its time writing to Flash memory, it is best to move to an FRAM-based microcontroller. For future Quantized Computing deployments, we recommend augmenting

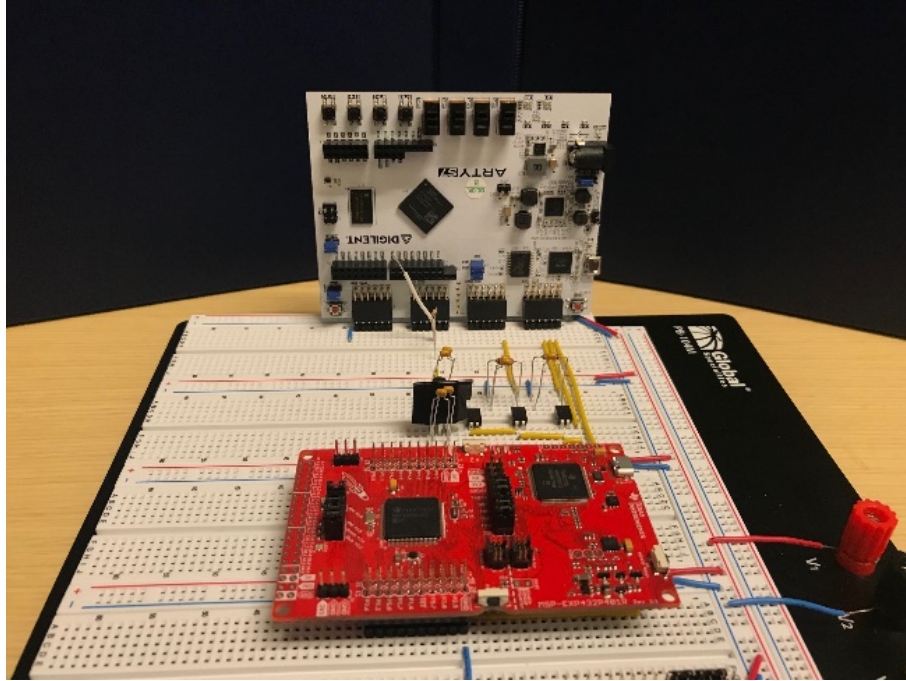


Figure 3.8: *Quantization Controller MSP432 prototype.*

the MSP432 with a small (e.g., 128 byte) FRAM scratch pad that secure computation can use to commit results before power loss. Given that neither AES or RSA approach 70% of their time spent writing to Flash, we select the MSP432.

3.6 Selecting a Microcontroller Configuration

The MSP432 supports a range of configuration options that impact duration of secure computation (Δt) and current consumption (I). To explore the impact of these configuration options, we create a prototype of the Quantization Controller with the MSP432 as the crypto core. Using this platform, we run AES and RSA. The goal is to discover the configuration settings that minimize the required capacitor.

Figure 3.8 shows our prototype. The prototype is split into three components: a Xilinx Artix-35T FPGA, a collection of discrete circuit components, and a MSP432 development board. The FPGA implements the digital logic of our Quantization Controller, the discrete

circuit components implement the analog aspects of the Quantization Controller (i.e., the switches, decoupling capacitors, and timing capacitors), and the MSP432 acts as the crypto core.

Using this platform, we evaluate all centered clock frequencies of the MSP432: $1.5MHz$, $3MHz$, $6MHz$, $12MHz$, $24MHz$, and $48MHz$.⁹ We only present the three highest frequencies as lower frequencies disproportionately sacrifice computation time for reduce current. Also, we evaluate two different $24MHz$ configurations [63]: with and without a memory wait state. Adding a wait state decreases memory throughput, but allows for lower current operation. Because the MSP432 uses a read buffer that leverages access locality, the impact of wait states is program dependent.

To capture an upper bound on computation time and current, we use worst case inputs for AES and RSA. Evaluating under the worst case inputs ensures that we are evaluating the secure algorithms under there highest energy usage, yielding a worst case capacitor size. For example, RSA modular exponentiation only executes when a key bit is 1, which consumes more power and time. Thus, to guarantee that we capture worst case behavior, we use a key of all 1's.

To minimize the influence of noise on our results we average 10 trails. Averaging 10 trails results in a relative standard deviation of 5.61% for current and 0.04% for computation time. Another tool that we use to reduce measurement noise is a differential probe that eliminates common mode noise.

3.6.1 Computation Time (Δt)

To determine the effect that clock frequency has on computation time, we record the computation time of all possible sizes of secure computation (i.e., 1 through 12 rounds

⁹Centered frequencies are the coarse-grain clock frequencies supported by the MSP432; i.e., operational set points.

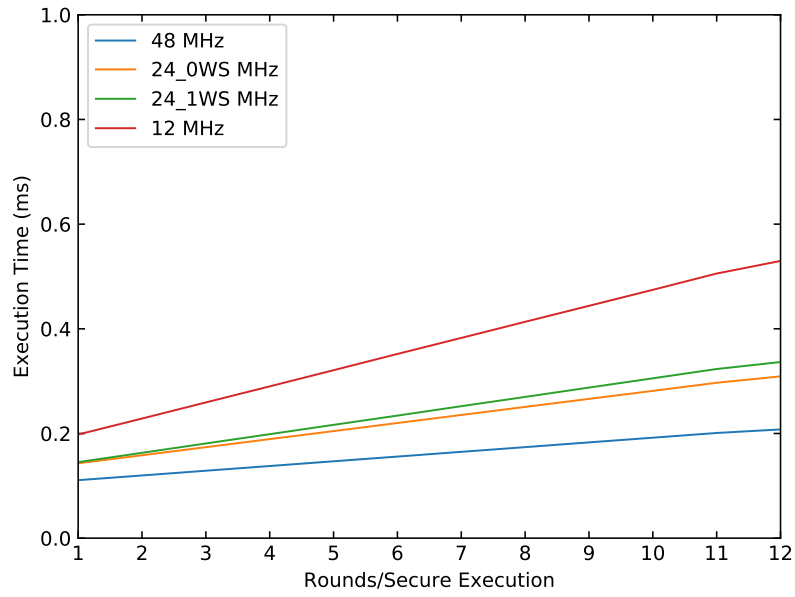


Figure 3.9: *Computation time for AES*

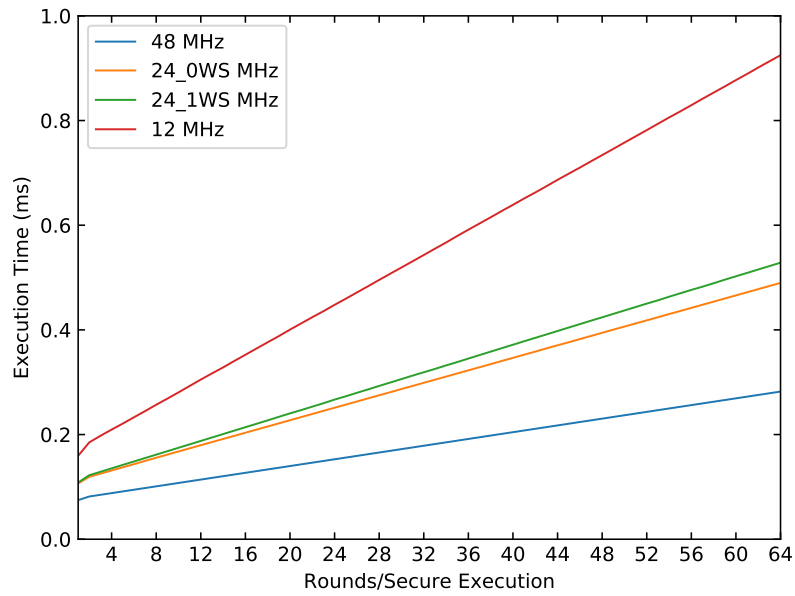


Figure 3.10: *Computation time for RSA*

per secure execution for AES and 1 through 64 rounds per secure execution for RSA) at every frequency and wait state configuration. Figures 3.9 and 3.10 show the results of this experiment for AES and RSA, respectively. When comparing computation times, RSA takes approximately double the computation time of AES due to its modular arithmetic. For example, 12 rounds of AES take as long as 32 rounds of RSA.

As expected, there is a linear relationship between the number of rounds per secure execution and the computation time. The same is not true for clock frequency: doubling the clock frequency from $12MHz$ to $24MHz$ results in a 50% reduction in time. However, doubling from $24MHz$ to $48MHz$ results in only a 33% reduction. This non-linearity is due to memory pressure stemming from the $24MHz$ limit on Flash reads. This is why the memory-bound AES shows less improvement when moving to $48MHz$ than the compute-bound RSA.

3.6.2 Average Current (I)

Similarly, Figures 3.11 and 3.12 show the current results for AES and RSA, respectively. The results suggest that current largely depends on clock frequency, as opposed to software. Software does have a small impact, as RSA generally has a low current—even though its computation takes longer. AES's slightly higher current is due to increased memory accesses that require more current than computation from registers. Current appears proportional to clock frequency. However, the current offset does not scale with the clock frequency, there is a 30% increase when going from $12MHz$ to $24MHz$, but a 65% increase when going from $24MHz$ to $48MHz$. There is also 7% variation with and without a wait state at $24MHz$ for AES compared to a minimal difference for RSA. AES, being memory-bound, benefits more from the read-buffer than RSA, lowering its current.

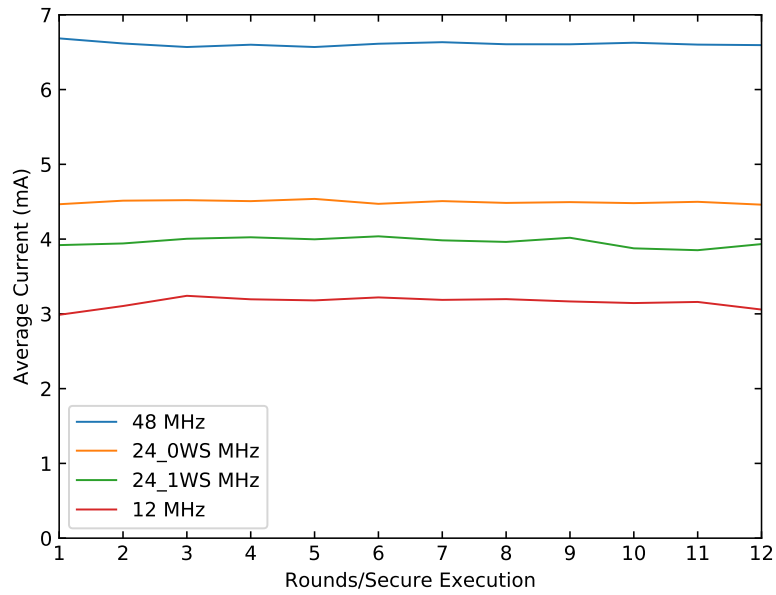


Figure 3.11: Average current for AES

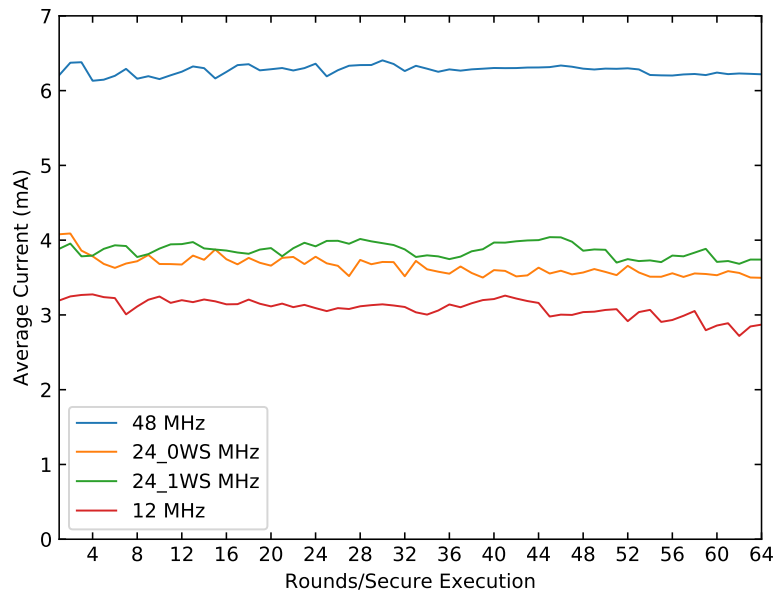


Figure 3.12: Average current for RSA

3.6.3 Capacitor Size

With computation time and current tradeoffs measured, we use Equation 3.3 to determine their combined effect on required decoupling capacitor size. Figures 3.13 and 3.14 show the required capacitor size of AES and RSA, respectively. The best clock frequency is $24MHz$ with zero memory wait states, since that is the best option for RSA and among the best for AES. To put these results into context, the MSP432 that we use has $400nF$ of decoupling capacitors. This allows for 5 rounds of AES and 25 rounds of RSA per secure execution.

3.6.4 System Verification

To validate our system we show the effectiveness of our Quantization Controller in eliminating both power and timing side-channel leakage by leveraging the MSP432's existing decoupling capacitors to produce uniform power and timing footprints for secure execution. We conduct the system verification using the prototype in Figure 3.8 with a hybrid Quantization Controller, an MSP432 running at $24MHz$ and 0 wait states, the existing $400nF$ decoupling capacitor, using AES (5 rounds per secure execution) and RSA (25 rounds per secure execution) implementations vulnerable to power and timing side-channel attacks (which we verify using our measurement setup). Our validation shows that both power and timing side-channels are eliminated by the Quantization Controller and that both AES and RSA complete across several secure executions.

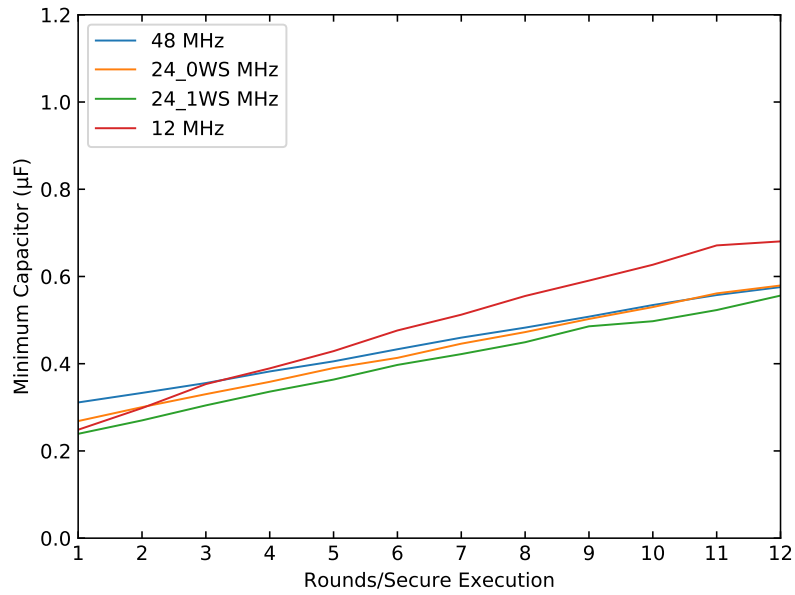


Figure 3.13: Required capacitor size for AES

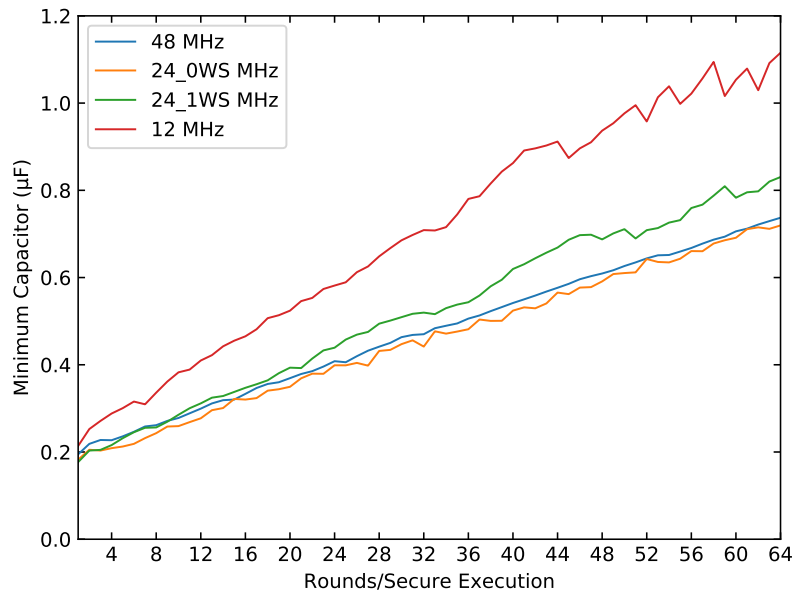


Figure 3.14: Required capacitor size for RSA

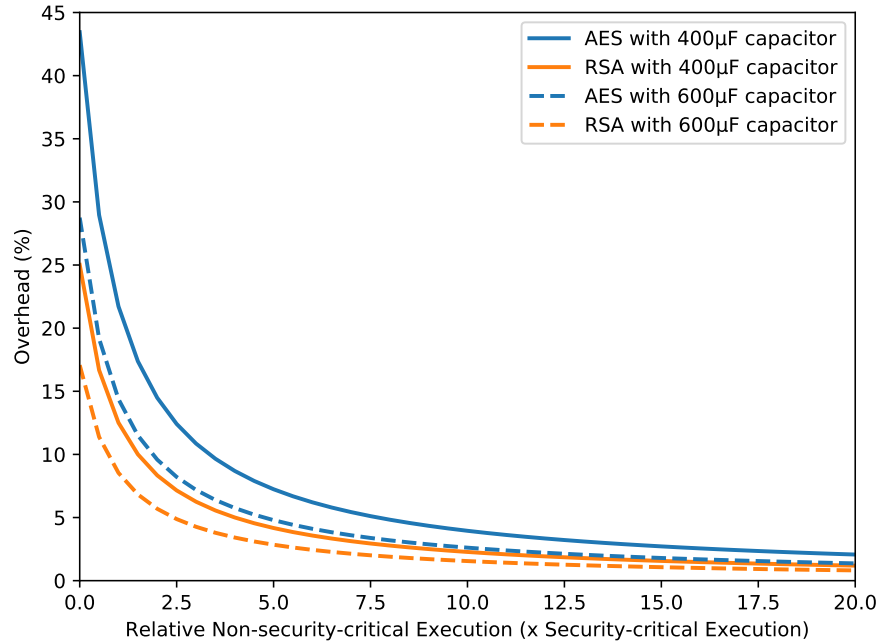


Figure 3.15: *Software run time overhead across a range of secure execution invocaton rates.*

3.6.5 Software Overhead

An important feature the Quantization Controller is the ability to protect software on-demand. When operating in recharge mode, there is no protection, but also no cost. This allows software designers to reason about security guarantees and run time overhead due to protection. To explore this trade space afforded by the Quantization Controller, we explore how software’s run time overhead changes as the frequency of secure executions decreases (i.e., more time spent on non-security-critical computation). Figure 3.15 shows the results.

The overhead of our design using both AES and RSA follow similar patterns with an offset. The offset comes from two differences between AES and RSA: (1) RSA rounds are more fine grain and easier to tightly pack in a secure execution and (2) RSA spends less time writing to Flash since it has to write half the date. One option system designers might explore is increasing decoupling capacitor size (as shown in 600nF lines). While

this does reduce overhead, finding ways to reduce the rate of secure executions is much more effective and easier since it is a software change.

3.7 Discussion

Given that our design eliminates power and timing side-channel leakage while also addressing a range of fault attacks, advanced attackers are forced to use more complex and expensive techniques to capture side-channel information. One such technique is exploiting Electromagnetic (EM) side-channel leakage. EM side-channel attacks are similar to power side-channel attacks in that they both extract power information [12]. The tradeoff is between fidelity and access: where power side-channel attacks use low cost probes [65] that attach directly to the power rail, providing direct access to changes in current, EM side-channel attacks use expensive probes [66] placed above sensitive locations on the IC, indirectly measuring changes in current via the radiating electromagnetic field. The challenge with EM side-channel attacks is extracting the desired signal from the noise as all wires in a system that carry current contribute to the EM field [67, 68].

While our approach is not intended to deal with EM attacks, it does make them more challenging. When the Quantization Controller isolates the crypto core for secure execution, it dramatically shortens the wires in the IC that carry the bulk of the current. This makes them a less effective antenna, reducing the signal the attacker is able to capture. While this reduces EM radiation, it does not remove it with respect to a nation-state level attacker. Fortunately, effective IC-level countermeasures against EM emanations exist. These include shielding [69] and adding power and ground planes [70].

3.8 Related Work

Quantized Computing is an on-demand, automatic, practical, and comprehensive defense against power, timing, and fault attacks. Our approach centers on isolation, which diverges from traditional defenses to power and timing side-channel attacks: traditional defenses play the signal-to-noise ratio game focusing on eliminating, reducing, hiding, or adding noise to mask the signal.

Resdesigning cryptographic circuits to consume uniform (i.e., independent of data) energy eliminates power side-channels [6, 71, 72]. Balanced based signal reduction circuits have been achieved by focusing on dual-rail precharge logic [73, 74, 75], current modes [76, 77], and asynchronous logic styles [78, 79]. Other techniques include compensating circuitry and physical shielding [80, 81]. While such approaches are effective against power, they are not comprehensive (timing and fault attacks remain). Additionally, they require significant hardware redesign while dramatically increasing hardware's power and area.

Other approaches focus on increasing noise in hopes of masking the signal [82]. Noise is added by circuits that consume a variable amount of current, or using circuit elements to perform calculations that are uncorrelated to the secure computation. Temporal noise is introduced by inserting variations in timing and execution order. Methods include using deliberately decorrelated and varying clocks, random wait states, random execution re-ordering, use of dummy operations, and random branching [82]. Countermeasures based on introducing noise do not remove secret-dependent information as well as not addressing timing or fault attacks.

The recent trend of integrated decoupling capacitors is a key enabler of our security guarantees. Before this trend, others have attempted to decouple system power from security-critical circuit power, but have failed for a variety of reasons: not eliminating

both power and timing side-channels [83, 34, 84], not addressing fault attacks [83, 34, 84], and/or were trivially attacker-bypassable [83, 34]. Alternatively, Quantized Computing eliminates both power and timing side-channels, is immune to a range of fault attacks, and presents a strong security perimeter that forces the adversary to undertake expensive and slow destructive analysis.

3.9 Conclusion

This work shows how system designers can leverage existing integrated decoupling capacitors for security, specifically, to protect security-critical computation from attackers seeking to exploit power and timing side-channel emanations. We propose the idea of a Quantization Controller that utilizes on-demand isolation and task-level atomicity guarantees to control the granularity of power and timing side-channels emanations, while masking any secret-dependent variations. Our experimental results with real hardware show that it is possible and practical to accomplish meaningful security-critical computation while being powered only by a decoupling capacitor. We see our isolation-based approach enabling the construction of more secure systems; whether they be ultra-low-power embedded devices or commodity servers.

Chapter 4

Quantized Computing:

Electromagnetic Augmentation

With growing demand for secure low power IoT devices, embedded processors are expected to perform secure cryptographic algorithms. These algorithms are implemented on physical platform that leak secure side-channel information [5]. These channels include power [6], timing [7], electromagnetic [8], acoustic [9], memory remanence [10], and thermal [11]. The two most commonly exploited side-channels in literature are power and timing [12].

Existing side-channel defense focus on the attacker's signal-to-noise ratio: increasing the noise or decreasing/hiding the signal. Countermeasures include: reducing the magnitude of side-channel emanations [30], the addition of noise to mask side-channel emanations [31], obfuscation to hide the relative timing of the secret-revealing emanation [32], and incorporating randomness at the software level [13]. The Quantization Controller in Chapter 3 is a defensive design that reduces the visible signal through *isolation*. As such it is able to defend against power, timing, fault, and power-glitching attacks. Forcing an attacker to consider more complex and expensive techniques to capture side-channel

information, such as Electromagnetic (EM) side-channel leakage.

EM side-channel attacks are similar to power side-channel attacks in that they both extract power information [12]. The challenge with EM side-channel attacks is extracting the desired signal from the noise, as all wires in a system that carry current contribute to the EM field [67, 68]. There are existing EM side-channel countermeasures that reduce the signal to noise ratio through shielding [69], adding power and ground planes [70], and noise insertion [12]. However, EM shielding incurs high cost of packaging and noise injection comes with significant power overheads making these unpractical for most applications.

While the Quantization Controller is not intended to deal with EM attacks, it does make them more challenging. When the Quantization Controller isolates the crypto core for secure execution, it dramatically shortens the wires in the IC that carry the bulk of the current. This makes them a less effective antenna, reducing the signal the attacker is able to capture [84]. This work will make the following contributions:

- Understanding of the root cause of EM leakage
- Analysis of EM leakage within an IC
- Proposed EM augmentation defense to the Quantization Controller

4.1 EM Leakage: Near-Field Radiation

EM emissions are the cause of current carrying traces in an IC. These traces act as antennas that produce a signal proportional to the current in the trace. In CMOS based IC designs sequential logic is executed on clock edges, causing a short burst of current through the power traces. The sudden change in current following a clock edge causes a change in the EM field surrounding the trace and a finite electromotive force given by

Lenz' Law, $E = d\Phi_B/dt$, where Φ_B is the magnetic flux, i.e., $\Phi_B = \int_S \mathbf{B} \cdot d\mathbf{S}$.

Compared to power side-channel attacks, which measure a collective signal from all processes occurring within the chip, EM side-channel attacks can measure emanations from localized components on the device that are positioned in close proximity to the EM probe.

The EM field that is produced from a current carrying trace can generally be classified into two regimes – a *near-field* and *far-field* signal. The far-field is what is traditionally referred to as "electromagnetic radiation," and radiates energy from the source to infinity as a result of accelerating charges in the source. Far-field radiation is typically found at distances greater than $2D^2/\lambda$ from the trace, where D is the size of the trace and λ is the wavelength of the electromagnetic signal. Correspondingly, the *near-field* is found within approximately $2D^2/\lambda$ of the source and stores energy that is generated by the inductive nature of the trace. Near-field EM probes, which can be bought off-the-shelf or fabricated from a small metal plate or metal coil, effectively siphon energy from the near-field of an antenna, thereby producing a current in a shunt resistor in the probe. In other words, the current from the trace *induces* a current in the EM probe. The boundary between the near and far field is not definitive, but generally a near-field probe must be within a few wavelengths of the trace to measure a signal. For a 1 GHz clock with a fundamental wavelength of 30 cm, mostly all measurable signals are from the near-field.

The EM field, whether near-field or far-field, is generally represented by the scalar and vector potentials given by

$$\Phi(\mathbf{r},t) = \int_V \frac{\rho(t-r/c)}{4\pi\epsilon_0 r} dV \quad (4.1)$$

$$\mathbf{A}(\mathbf{r},t) = \int_V \frac{\mu_0 \mathbf{J}(t-r/c)}{4\pi\epsilon_0 r} dV \quad (4.2)$$

where ρ is the resistivity, r is the distance from the source, dV is the differential volume

of the source, t is time, c is the speed of light, ϵ_0 and μ_0 are the vacuum permittivity and permeability, and \mathbf{J} is the current density.

In the case of a short wire with radius, d , length ℓ , and current $I(t)$ that is aligned along the \hat{z} direction with uniform current density, $\mathbf{J} = I(t)/\pi d^2 \hat{z}$, the vector potential is given as

$$\mathbf{A} = \frac{\mu_0 \ell}{4\pi r} I_0 e^{-jkr} (\hat{r} \cos \theta - \hat{\theta} \sin \theta) \quad (4.3)$$

from which the magnetic and electric fields are derived as

$$\mathbf{B} = \frac{jkI_0 \ell}{4\pi r} e^{-jkr} \left(1 + \frac{1}{jkr}\right) \hat{\phi} \quad (4.4)$$

$$\mathbf{E} = \frac{jkI_0 \mu_0}{4\pi \epsilon_0} e^{-jkr} \left\{ \left(\frac{1}{jkr} + \frac{1}{(jkr)^2}\right) 2 \cos \theta \hat{r} + \left(1 + \frac{1}{jkr} + \frac{1}{(jkr)^2}\right) \sin \theta \hat{\theta} \right\} \quad (4.5)$$

where k is the wave number, $2\pi/\lambda$.

In the far-field region, $kr \gg 1$,

$$\mathbf{E} \approx \frac{jkI_0 \ell \mu_0}{4\pi \epsilon_0 r} e^{-jkr} \sin \theta \hat{\theta} \quad (4.6)$$

$$\mathbf{B} \approx \frac{jkI_0 \ell}{4\pi r} e^{-jkr} \sin \theta \hat{\phi} \quad (4.7)$$

indicating the expected $1/r$ dependence.

In the near-field, $kr \ll 1$,

$$\mathbf{E} \approx \frac{I_0 \ell}{j\omega 4\pi \epsilon_0 r^3} (2 \cos \theta \hat{r} + \sin \theta \hat{\theta}) \quad (4.8)$$

$$\mathbf{B} \approx \frac{I_0 \ell}{4\pi r^2} \sin \theta \hat{\phi} \quad (4.9)$$

the latter of which is commonly referred to as the Biot-Savart law. Evidently, the magnetic field from a power rail on an IC, which induces a current in the EM probe, is proportional to the current in the line and the length of the line, and drops off as $1/r^2$.

It is evident from equation 4.9 that reducing the length and current of the secret carrying power rails could provide a decrease in the radiated signal.

4.2 EM Leakage: Wire Layers

Traditional multilayered ICs are composed of a semiconductor silicon layer (device layer) at the bottom with multiple wire layers (metal layers) above interconnecting the device layer [85]. Following equation 4.9 it is evident that the wire layers are the root cause of the near-field radiation, acting as antennas, carrying current to throughout the device layer. With the advancement of semiconductor technology, transistor sizes are scaling down, the number of transistors in an IC are increasing. Therefore more interconnects and power traces are required. As such IC process technologies have reduced in node sizes and increased in wire layers. [18]

For the purposes of this work we focus on Intel's process technologies. Since 1998 Intel's semiconductor technology has evolved from 180nm to 10nm currently. However, Intel has only released wire geometry information for there 180nm [14], 130nm [15], 65nm [16], 45nm [17], and 32nm [18]. Tables 4.1, 4.2, and 4.3 are a collection of the dimensions provided by Intel.

As Intel's semiconductor technology evolved so did the number of metal layers and wire thicknesses. This is a result of having smaller transistor nodes, increasing the total number of transistors per IC requires more routing and power. This is reflecting in Tables 4.1, 4.2, and 4.3. The thickness of metal layer 1 grows smaller to connected to the smaller semiconductors while the higher metal layers are thicker to distribute more current throughout the IC. Traditionally ICs use a power grid to distribute power through the ICs semiconductors. These power grids consume the top two thicker layers (one for vertical and the other for horizontal routing) to grantee power and timing standards. To

Technology	180	130	65	45	32
Metal 1	500	293	210	160	112.5
Metal 2	640	425	210	160	112.5
Metal 3	640	425	220	160	112.5
Metal 4	1080	718	280	240	168.8
Metal 5	1600	1064	330	280	225
Metal 6	1720	114	480	360	337.6
Metal 7	-	-	720	560	450.1
Metal 8	-	-	1080	810	566.5
Metal 9	-	-	-	30500	19400

Table 4.1: *Pitch(nm)*

Technology	180	130	65	45	32
Metal 1	480	280	170	144	95
Metal 2	700	360	190	144	95
Metal 3	700	360	200	144	95
Metal 4	1080	570	250	216	151
Metal 5	1600	900	300	252	204
Metal 6	1720	1200	430	324	303
Metal 7	-	-	650	504	388
Metal 8	-	-	975	720	504
Metal 9	-	-	-	7000	8000

Table 4.2: *Thickness(nm)*

Technology	180	130	65	45	32
Metal 1	1.9	1.7	1.6	1.8	1.7
Metal 2	2.2	1.7	1.8	1.8	1.7
Metal 3	2.2	1.7	1.8	1.8	1.7
Metal 4	2	1.6	1.8	1.8	1.8
Metal 5	2	1.7	1.8	1.8	1.8
Metal 6	2	2.1	1.8	1.8	1.8
Metal 7	-	-	1.8	1.8	1.7
Metal 8	-	-	1.8	1.8	1.8
Metal 9	-	-	-	0.4	1.5

Table 4.3: *Aspect Ratio*

Technology	180	130	65	45	32
Metal 1	-	-	-	-	-
Metal 2	28	45	0	0	0
Metal 3	28	45	5	0	0
Metal 4	116	145	33	50	50
Metal 5	220	263	57	75	100
Metal 6	244	290	129	125	200
Metal 7	-	-	243	250	300
Metal 8	-	-	414	406	404
Metal 9	-	-	-	18963	17144

Table 4.4: *Pitch(%) difference to Metal 1*

Technology	180	130	65	45	32
Metal 1	-	-	-	-	-
Metal 2	46	29	12	0	0
Metal 3	46	29	18	0	0
Metal 4	125	104	47	50	59
Metal 5	233	221	76	75	115
Metal 6	258	329	153	125	219
Metal 7	-	-	282	250	308
Metal 8	-	-	474	400	431
Metal 9	-	-	-	4761	8321

Table 4.5: *Thickness(%) difference to Metal 1*

Technology	180	130	65	45	32
Metal 1	-	-	-	-	-
Metal 2	16	0	13	0	0
Metal 3	16	0	13	0	0
Metal 4	5	-6	13	0	6
Metal 5	5	0	13	0	6
Metal 6	5	24	13	0	6
Metal 7	-	-	13	0	0
Metal 8	-	-	13	0	6
Metal 9	-	-	-	-78	-12

Table 4.6: *Aspect Ratio(%) difference to Metal 1*

evaluate the evolution in Intel’s process technologies we calculate the percent difference with respect to Metal Layer 1 in Tables 4.4, 4.5, and 4.6. We can observe a consistent trend with Intel’s process technology, as semiconductors become smaller the difference between its smallest metal layer (Metal 1) and the highest metal layers increases significantly. This is a direct result of needing more wires and current to connect to more smaller semiconductors throughout the IC. However with respect to EM side-channel attacks Intel’s current trend would increase its visible near-field radiation footprint, as the length and current values in equation 4.9 would increase.

4.2.1 Metal Layer: Modeling

To better understand and evaluate Intel’s current trend on the EM side-channel leakage we need to measure the near field radiation of these wire layers. The net radiation can be split into contributions originating from each interconnect in the routing. A simple structure that can be used to analyze the radiation properties of different metal layers is a vertical stack of interconnects of the same length, joined by vias. While this would not accurately depict the interconnects of an actual IC layout, it will allow us to quantify the effects of thickness on the near-field. In essence evaluating the near field, equation 4.9, using a constant antenna length, l , and variable current, I_0 , that is based on the metal layer dimensions. The larger the available trace, the more current that can pass.

For our model we used the dimensions provided by Intel’s 32nm [18] technology, values of which are provided in Tables 4.1, 4.2, and 4.3. Following standard IC interconnects guidelines [86] we can calculate the exact geometry of each wire layer. Figure 4.1 is an illustration of the wire layers geometry and variables. Using Equations 4.10 and 4.11 we are able to calculate the additional parameters need to create an exact model of Intel’s 32nm wire layer.

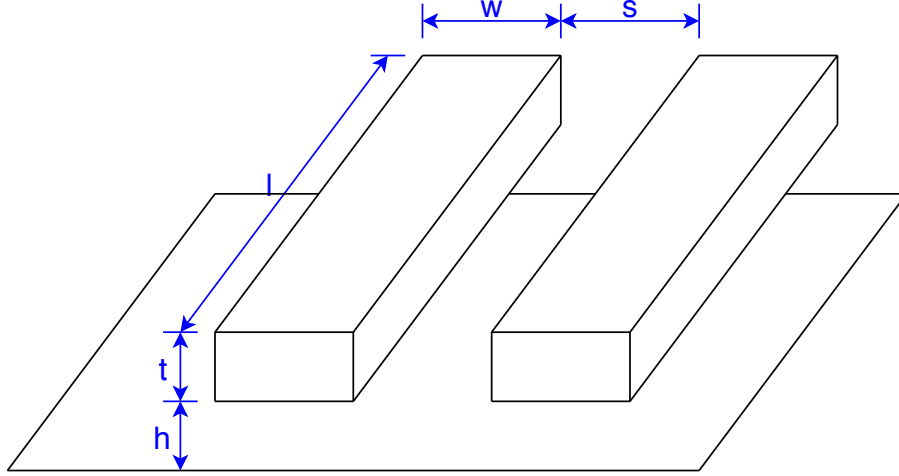


Figure 4.1: *Wire Layer Geometry.*

$$AspectRatio = \frac{t}{w} \implies w = \frac{t}{AspectRatio} \quad (4.10)$$

$$Pitch = w + s \implies s = Pitch - w \quad (4.11)$$

The simple structure layout we modeled for this experiment is shown in Figures 4.2 and 4.3. The length of each interconnected layer in this model is set to be $3\mu m$. While there is a slight increase in thickness between metal layers, Figure 4.3 illustrates the significant difference between metal layer 9 and the rest.

These models were built using Ansoft HFSS, a finite element method (FEM) based EM simulator to solve Maxwell's equations. The excitation for our interconnect is provided via a lumped port in HFSS between the bottom-most metal layer and a perfect electric conductor (PEC) plate functioning as a ground. This style of excitation is similar to a dipole antenna and is justified due to the similarity of the system to an infinitesimal dipole. Representing the close proximity of an attacker a radiation boundary of $1mm$ radius enclosing the interconnect stack is used to simulate the region. Limiting the analysis and eliminating reflection of incident radiation from the outer surface.

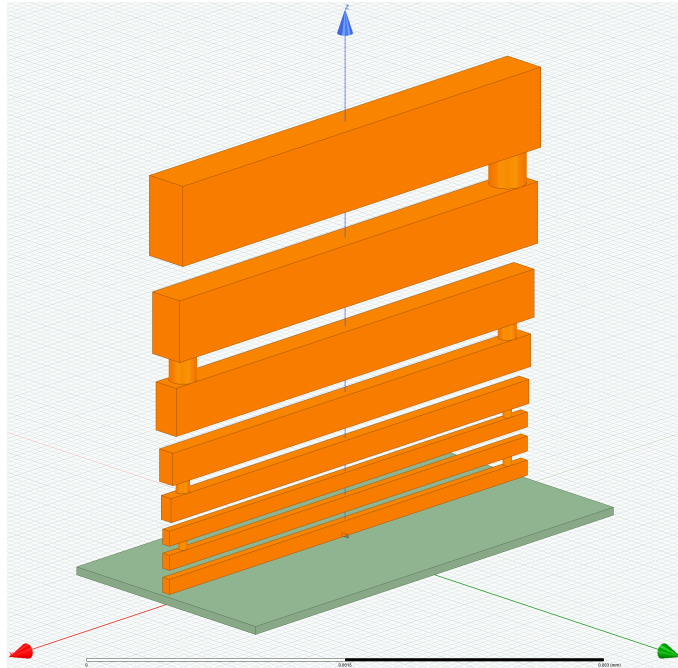


Figure 4.2: *HFSS model of Intel 32nm Metal Layer - excluding Metal Layer 9.*

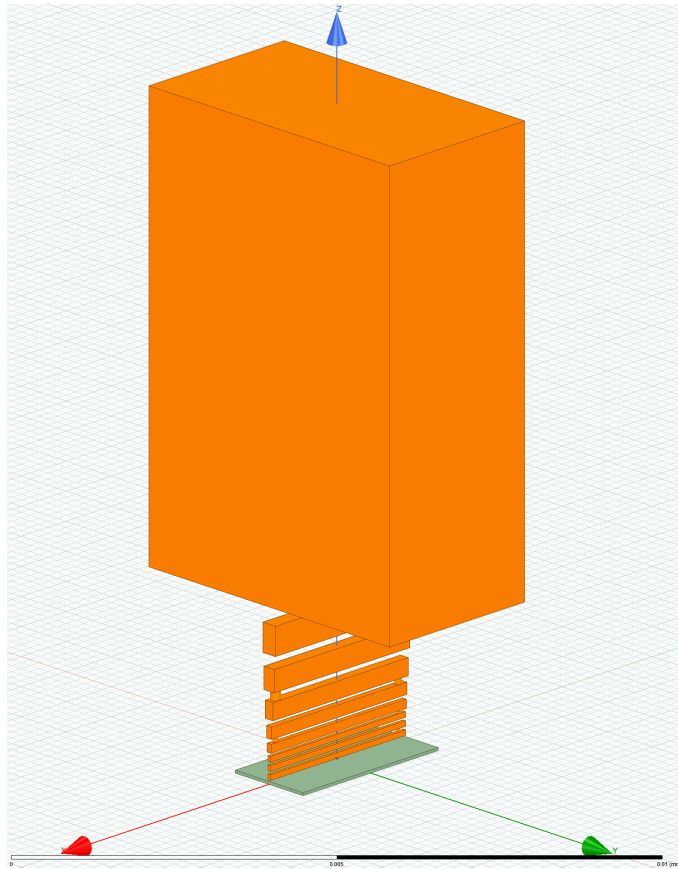


Figure 4.3: *HFSS model of Intel 32nm Metal Layer.*

4.3 Metal Layer: Simulating EM-Fields

The EM radiation of an IC primarily originates from the metal layer interconnects acting as mini-antennas. To develop a better understanding of these mini-antennas the net radiation can be split into contributions originating from each metal layer in the routing. The model in Figure 4.3 is excited at 1 GHz, and the electric field amplitude is measured. The magnetic field is negligible in this model as electrically activated plates dominate the electric field in the near-field region, whereas with current loops magnetic fields would dominate the near-field. Therefore with closed loop differential elements in a chip, we would want to measure the magnetic field using a similar analysis.

The near-field electrical radiation patterns are shown in Figure 4.4. To analysis the contributions of each metal layer in the routing we repeat the simulation multiple times, each time removing the topmost metal layer. This allows us to calculate there electrical radiation contribution of each metal layer depicted in Figure 4.5. The electric field contribution of metal layers 1 through 8 appear to increase linear, but metal layer 9 contributes significantly more than the rest of the metal layers. As metal layer 9 is part of the power grid that disseminates current throughout the IC, it is ideal for leaking power information. Any significant EM countermeasure would have to mitigate the leakage cause my metal layer 9 routing.

4.4 Threat Model

Our threat model assumes an attacker with knowledge of the software running on as well as with full physical access to an IC. The attacker's objective is to extract secret information through non-destructive, non-invasive, EM side-channel analysis. We assume that the attacker can also control the device's environment to induce power and thermal faults.

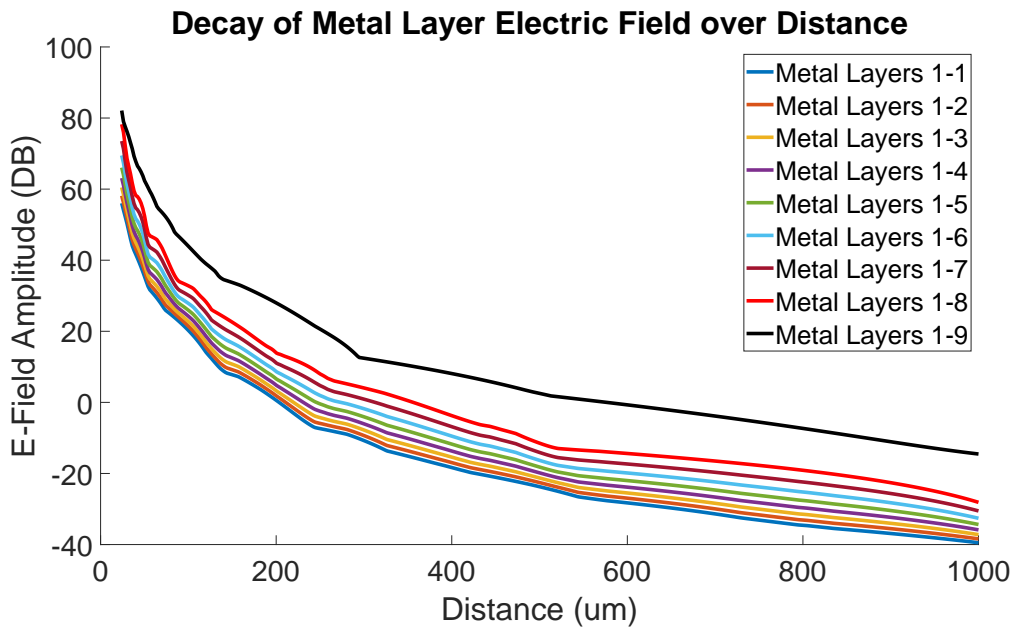


Figure 4.4: Metal Layer E-Field decay over distance.

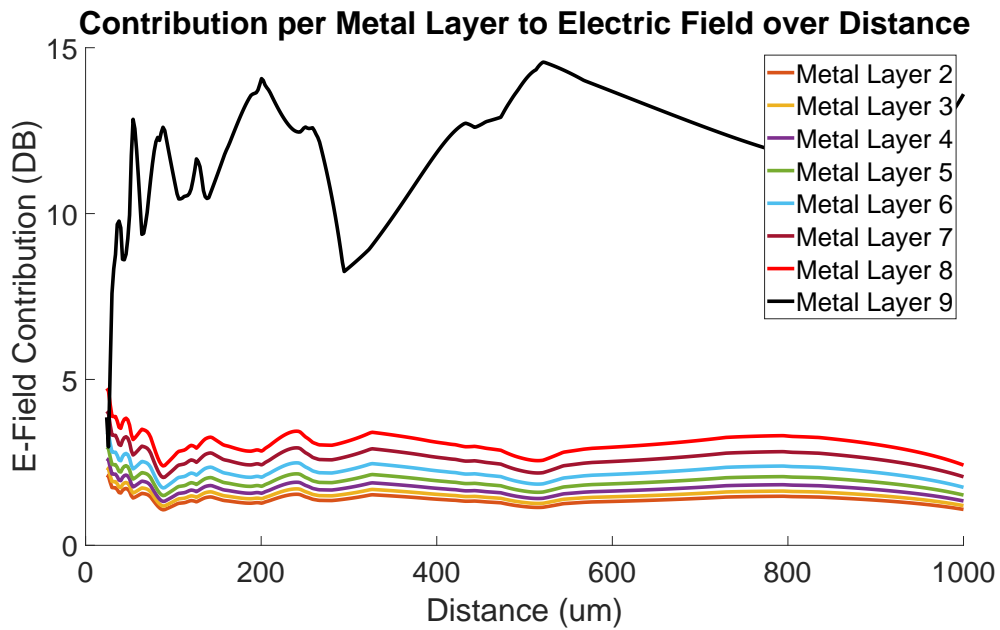


Figure 4.5: Metal Layer E-Field contribution over distance.

Following our observations of the metal layers contribution to EM radiation, we develop an augmentation to the Quantization Controller to mitigate EM radiation from the top two metal layers. Our design inherits the Quantization Controller defenses against power, timing, thermal, power glitching, and memory fault attacks while significantly reducing EM leakage.

4.5 Design

Equation 4.9 quantifies the near field radiation strength of these mini-antennas based on two variable properties the length and current of the antennas. Any proposed design trying to reduce EM radiation effectively will need to reduce the visible footprints from the metal layers mini-antennas. In this work, we have identified that the top tiers generate the majority of the near-field radiation footprint. These top tiers are used as a power grid for disseminating power throughout the IC. To augment the Quantization Controller with an EM countermeasure we need to effectively reduce both the current draw and length of the power grid on these top tier metal layers.

During secure execution the Quantization Controller uses internal decoupling capacitors to power an internally isolated power grid. Following Kirchhoffs Current Law for parallel capacitors we can replace the internal decoupling capacitor with a bank of n smaller capacitors that total the same value:

$$C_{total} = C_1 + C_2 + \dots + C_n \quad (4.12)$$

$$I_{total} = C_1 \frac{dv}{dt} + C_2 \frac{dv}{dt} + \dots + C_n \frac{dv}{dt} \quad (4.13)$$

By spreading out these smaller capacitors along the internally isolated power grid, we would effectively be reducing the length and current of the mini antennas. Potentially

removing the current draw across the power grid during secure execution by using localized capacitors to power semiconductor cells. Such a design would significantly reduce the EM footprint during secure execution for the Quantization Controller.

At the time of writing this thesis parts of Quantization Controller - EM Augmentation are under restricted access. The design aspect of this work was completed at MIT Lincoln Labs and is currently being reviewed for release. For this reason, we are unable to provide details, designs, simulations, and evaluations of the proposed design.

4.6 Discussion

Given that our design augments the Quantization Controller with EM mitigation we effectively reduce the visible footprint of power, timing, and EM side-channel leakage while also addressing a range of fault attacks. Our design forces attackers to use more complicated and expensive techniques to capture side-channel information.

Currently, the weakest design requirement of the EM augmentation is its uniform execution. To optimize our design and minimize its overhead we require that all cell execution uniformly drain power from their capacitor banks. Thereby preventing capacitor banks of different cells from sharing charge and inducing a current through the top two tiers. Although uniform execution is a common EM side channel countermeasure due to the nature of our proposed design attackers could potentially force specific cells to execute non-uniformly thereby causing undesired charge flow across the top two tiers.

Should our proposed design effectively mitigate leakage against non-invasive passive attacks, attackers with full physical access maybe be forced to induce leakage actively. Active attacks would require significantly more complicated, expensive, and invasive techniques such as etching and targeted thermal fluctuations. By etching away the cover of the IC and exposing the substrate and metal layers, attackers can manipulate the charge

of the power capacitors. Attackers can then add or remove charge forcing a non-uniform discharge of the power capacitors. Causing the capacitors to share charge, thereby inducing a current through the top two tiers. Such attacks would raise the threat level of an attacker to a nation-state, due to the complexity and cost of such an attack.

To counter uniform execution, we examine isolating each cell during secure execution. However, we determine that such a design would incur significant overhead. By adding PMOS switches between each cell, we would add a large area footprint and a constant power overhead on the internal power rail in both secure and insecure mode. We would also add power limitations due to current flow through serial PMOS switches. While isolating each cell during secure execution with PMOS switches would allow for non-uniform executions the current design overhead is not ideal.

4.7 Conclusion

Our evaluation into the source of EM leakage has shown that the metal layers of an IC, used to interconnect the device layer, act as mini-antennas that produce near field radiation that leakage information. Equation 4.9 quantifies the near field radiation strength of these mini-antennas based on two variable properties the length and current of the antennas. We simulate a simple wire layer struct to measure the contributions of each wire layers. Figures 4.4 and Figures 4.5 show EM that the EM footprint grows with each wire layer depending on its thickness. We observed a significantly larger increase with the top tier as it is substantially larger than the previous metal layers.

Traditionally the top tiers are used as a power grid to disseminate power throughout the IC, the thicker the metal layer the more current that can pass. As semiconductors continue to emerge smaller consequently: they will be more densely packed, requiring small wires to connect to, more interconnecting metal layers, and thicker layers for more current

flow. With respect to EM side-channel attacks, the consequences of smaller semiconductors will only increase the near-field radiation footprint. Analyzing the metal layer dimensions of Intel's IC technology nodes for 180nm [14], 130nm [15], 65nm [16], 45nm [17], and 32nm [18] we can observe in Tables 4.1 through 4.6 an increase in the number of wire layers, smaller initial metal layers, and significantly large top metal layers.

To augment the Quantization Controller with an EM countermeasure we need to effectively reduce both the current draw and length of these metal layer mini-antennas created in any design layout. During secure execution the Quantization Controller uses internal decoupling capacitors as a power source. By leveraging this isolation feature, we can distribute the decoupling capacitors along the internal isolated power rail reducing the effective length and current draw of the metal layer mini antennas. Such a design would significantly reduce the EM footprint during secure execution for the Quantization Controller.

Chapter 5

Conclusion

With the increasing demand of new IoT devices comes new security threats that have yet to be properly evaluated [2]. In this work we developed a customized IoT, μ Leech, that acts as a TPM for smartphones. μ Leech mimics the properties of low power embedded IoTs allowing use to develop an evaluation platform to study the side-channel leakage of IoTs. Over the course of this evaluation we observed that low power scavenging techniques could be leveraged to create a side-channel isolation countermeasure. We design our Quantization Controller to create uniform power and timing footprints during secure execution. We augment the Quantization Controller design to defend against thermal, power glitching, memory fault, and EM attacks.

This chapter concludes the thesis by presenting a summary of the results and recommendations for future work.

5.0.1 Summary of Results

5.0.1.1 μ Leech

μ Leech is a low power platform on which to run cryptographic operations. As such, its performance is mainly determined in power consumption. The less power it consumes

the less power it ultimately needs to drain from a smartphone. The power consumption of μ Leech was evaluated by looking at different sleep and active implementation states.

μ Leech power consumption of sleep states are available in Table 2.3. These results show that the lowest power consuming sleep state is Static Sleep at $56.7\mu\text{A}$. Currently we are using Deep Stop Sleep which consumes $68.3\mu\text{A}$. Using Static Sleep would require us to reset our interrupts, counter, and comparator clocks. As a result there would be a clock initialization phase at the start of every wake cycle, ultimately consuming more power. In addition, depending on when the sleep state was activated an event reset may be required at wake, due to the clock reset, resulting in wasted cycles/power. For these reasons we are currently using Deep Stop Sleep, the second lowest power consuming sleep state available. With the prototype completed now, further testing of the different wake protocols may show which sleep state is ultimately more efficient.

Table 2.3 also shows μ Leech power consumption of active states. The highest power consuming phase of our μ Leech is its communication phase. Transmitting and receiving data is consuming $575\mu\text{A}$ and $588\mu\text{A}$ respectively. While 128-bit AES is consuming $555\mu\text{A}$. These results show that there are two things that need to be researched further, to optimize our design. First, can we achieve lower power consumption from our communication protocol implementation. Second, how far can we push our AES implementation while keeping an eye on our power consumption and capacitor bank. Possibly allowing us to increase our CPU clock speed, currently 1 MHz.

5.0.1.2 Quantized Computing

Our Quantization Controller design shows how system designers can leverage existing integrated decoupling capacitors for security, specifically, to protect security-critical computation from attackers seeking to exploit power and timing side-channel emanations. Simulations (Figures 3.5, 3.6) of proposed design illustrate how we leverage integrated

decoupling capacitors to create uniform timing and power footprints. This effect is also illustrated in Figure 3.2, that depicts the measured footprint of a real hardware implementation of our Quantization Controller. Our design is centered around the relationship between the size of the decoupling capacitors and their discharge rate during execution, quantified by equation 3.3. To optimize this relationship we analyzed three different embedded processor, shown in Table 3.2. Evaluating the different parameters of these processors and their effects on equation 3.3 i.e. computation time, average current, and voltage range. We measure memory performances (Figure 3.7) and execution performances (Figures 3.9, 3.10, 3.11, 3.12, 3.13 and 3.14) using AES and RSA implementations, to optimize our Quantization Controller implementation.

Our Quantization Controller utilizes on-demand isolation and task-level atomicity to control the granularity of power and timing side-channels emanations, while masking any secret-dependent variations. Our experimental results with real hardware show that it is possible and practical to accomplish meaningful security-critical computation while being powered only by a decoupling capacitor. We see our isolation-based approach enabling the construction of more secure systems; whether they be ultra-low-power embedded devices or commodity servers.

5.0.1.3 Quantized Computing: EM Augmentation

The two most commonly exploited side-channels in literature are power and timing [12]. Our Quantization Controller in Chapter 3 is a defensive design that reduces the visible signal through *isolation*. As such it is able to defend against power, timing, fault, and power-glitching attacks. Forcing attackers to consider more complex and expensive techniques to capture side-channel information, such as EM side-channel leakage.

Our evaluation into the source of EM leakage has shown that the metal layers of an IC, used to interconnect the device layer, act as mini-antennas that produce near field

radiation that leakage information. The strength of the near field radiation is dictated by the length and current of these antennas, quantified in equation 4.9. We analyze Intel's IC technology nodes for: 180nm [14], 130nm [15], 65nm [16], 45nm [17], and 32nm [18]. As tables 4.4, 4.5, and 4.6 show there is an increase in the size of the wire layer as you go up the IC. With a significant increase between the top metal layer and the bottom metal layer.

This is due to the fact that as smaller technology nodes are developed semiconductors become smaller and are therefore more densely packed. Smaller technology nodes will require smaller wire to interconnect, more wire layers, and thicker wires to carry current throughout the IC. For this reason the top two layers are traditionally used as a power grid, the thicker layer are able to provide more current throughout the IC. However with respect to EM side-channel attacks these layers would increase the near-field radiation footprint, as the length and current values in equation 4.9 would increase. Figures 4.4 and Figures 4.5 support this claim as the EM footprint grows with each wire layer depending on its thickness.

5.0.2 Recommendations for Future Work

5.0.2.1 μ Leech

Currently the biggest bottle neck in our design is our communication protocols. We initially developed a simple communication protocol that is constantly communicating. This was done to mimic other IoT communication protocols, i.e. Square-point-of-sale, and Hijack. Such IoTs require continuous execution and communication, requiring more power and timing overhead. μ Leech executes and communicates in burst cycles. Therefore, a more efficient protocol for μ Leech can be implemented that takes advantage of its burst nature. This would optimize the libraries we use on the smartphone app and the proces-

processor's state-machine. In doing so, our communication protocol could:

- **Only need 1 Idle cycle.** At most we only need 1 Idle cycle between byte transmissions. Currently there are four idle cycles for our processor and twenty idle cycles for our smartphone between every byte transmission. This would ultimately improve transmission speeds.
- **Possibly attain higher baud rates.** Our data is predetermined at the start of the transmission, therefore the transmission buffer can be full pre-loaded allowing us to optimize timing. Allowing us to achieve higher baud rates, the limitation would turn to the smartphone's frequency modulation.
- **Not need a state-machine.** The communication protocols could be moved to the interrupt level entirely. We would not need a state machine to determine when to go to idle, transmit, receive, and when to build the transmit or receive buffers. Our communication would be in burst serial sequences. Therefore receiving and transmitting would be a serial process and could never happen simultaneously. This would allow us to pre-encode buffers using Manchester encoding and pre-load the data buffers as the data would be predetermined before transmitting.
- **Not need a counter.** The counter used to determine the space between comparisons when receiving data can be removed. Our AVR comparator is a synchronous comparator and can be used to decode the Manchester input directly, without using a counter. This would improve incoming data speeds and reduce power consumption.
- **Allow our smartphone app to transmit in bursts.** Having modified our processor to only transmit and receive when necessary this smartphone app could be modified to do the same. Having the smartphone continuously transmit on the power channel

but only transmit and receive data when necessary and in bursts, would significantly reduce the power overhead of the smartphone.

Such a communication protocol would allow for a much more efficient implementation protocol for μ Leech. However we would still be using Manchester encoding. There maybe other methods to achieving higher baud rates for our communication protocol that do not involve Manchester encoding. Currently Manchester encoding doubles the clock rate since ever bit is represent by two bits. The reason for using Manchester encoding is due to the fact that smartphone tones are a continuously oscillating waveform, and cannot represent high or low. Therefore Manchester encoding allows us to transmit by affecting the frequency without affecting the amplitude of the waveform. Other options may include modifying the amplitude of the waveform without affecting its frequency, such as volume control. If we could create a communication protocol that does not use Manchester encoding we could at the very least double our baud rate.

5.0.2.2 Qunatized Computing

The Quantization Controller was tested using real hardware, Figure 3.8, to verify functionality, Figure 3.2. The Quantization Controller was tested to effectively protect against power, timing, thermal, power glitching, and memory fault attacks. However we used a breadboarded prototype, therefore the next steps would include:

- **Fabrication.** The proposed design takes advantage of being integrated within the IC, this restricts attacker access to our design and the internal rail. We prototype our design using real hardware on a breadboard, Figure 3.8, to verify functionality, Figure 3.2. The final iteration should be fabricated within an IC design to maintain our threat model.
- **Testing and benchmarks.** With a customized fabricated IC we will be able to

perform tests using real hardware. This would allow use to quantify the effects of our countermeasures using an IC. We would expect these effects to closely resemble our pro-typed hardware version illustrated in Figure 3.2. An IC iteration would allow use to evaluate breakpoints and benchmarks of our design allowing use to further optimize the final design.

5.0.2.3 Quantized Computing: EM Augmentation

At the time of writing this thesis parts of Quantization Controller - EM Augmentation are under restricted access. The design aspect of this work was completed at MIT Lincoln Labs and are currently being reviewed for release. Should the proposed design be accepted for release the next steps would include:

- **Integration with the Quantization Controller.** The proposed design takes advantage of the isolation nature of the Quantization Controller. The Quantization Controller effectively uses the integrated decoupling capacitors as power sources. We would need to create a customized IC design that integrate the Quantization Controller with the proposed design.
- **Fabrication.** The nature of the problem we are trying to mitigate forces use to fabricate to perform a real hardware test. The source of EM radiation we are trying to mitigate is caused by the IC wire layers acting as antennas. As these wire layer antennas are shorter then there wave length, evaluations with larger antennas would be meaningless. Therefore the only real verification would need to be preformed using a Fabricated IC with an integrated Quantization Controller and the proposed EM augmentation design.
- **Testing and benchmarks.** With a customized fabricated IC we will be able to perform tests using real hardware. This would allow use to quantify the effects

of our proposed design on EM leakage of real hardware. We would also be able to evaluate breakpoints and benchmarks of our design allowing use to further optimize the final design.

Bibliography

- [1] Rolf H Weber. Internet of things–new security and privacy challenges. *Computer law & security review*, 26(1):23–30, 2010.
- [2] Ari Juels et al. Rfid security and privacy: A research survey. *IEEE journal on selected areas in communications*, 24(2):381–394, 2006.
- [3] Alexandra Mellen, John Moore, and Artem Losev. Mobile point of scam: Attacking the square reader.
- [4] Thiago Olson. Inside the square card reader, 2012. [Accessed 28 August 2015].
- [5] YongBin Zhou and DengGuo Feng. Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. *IACR Cryptology ePrint Archive*, 2005:388, 2005.
- [6] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO, pages 789–789, August 1999.
- [7] Paul C Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO, pages 104–113, August 1996.
- [8] Wim Van Eck. Electromagnetic radiation from video display units: An eavesdropping risk? *Computers & Security*, 4(4):269–286, 1985.
- [9] Adi Shamir and Eran Tromer. Acoustic cryptanalysis. *presentation available from <http://www.wisdom.weizmann.ac.il/tromer>*, 2004.
- [10] Yossef Oren, Ahmad-Reza Sadeghi, and Christian Wachsmann. On the effectiveness of the remanence decay side-channel to clone memory-based pufs. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 107–125. Springer, 2013.
- [11] Josep Altet, Antonio Rubio, Emmanuel Schaub, Stefan Dilhaire, and Wilfrid Claeys. Thermal coupling in integrated circuits: application to thermal testing. *IEEE Journal of Solid-State Circuits*, 36(1):81–91, 2001.

- [12] Dakshi Agrawal, Bruce Archambeault, Josyula R Rao, and Pankaj Rohatgi. The em sidechannel (s). In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 29–45. Springer, 2002.
- [13] Rambus. Licensed countermeasures - rambus, 2017. [Accessed 28 July 2017].
- [14] S Yang, S Ahmed, B Arcot, R Arghavani, P Bai, S Chambers, P Charvat, R Cotner, R Gasser, T Ghani, et al. A high performance 180 nm generation logic technology. In *International Electron Devices Meeting 1998. Technical Digest (Cat. No. 98CH36217)*, pages 197–200. IEEE, 1998.
- [15] Scott Thompson, Mohsen Alavi, Makarem Hussein, Pauline Jacob, Chris Kenyon, Peter Moon, Matthew Prince, Sam Sivakumar, Sunit Tyagi, and Mark Bohr. 130nm logic technology featuring 60nm transistors, low-k dielectrics, and cu interconnects. *Intel Technology Journal*, 6(2), 2002.
- [16] Peng Bai, C Auth, S Balakrishnan, M Bost, R Brain, V Chikarmane, R Heussner, M Hussein, J Hwang, D Ingerly, et al. A 65nm logic technology featuring 35nm gate lengths, enhanced channel strain, 8 cu interconnect layers, low-k ild and 0.57/spl mu/m/sup 2/sram cell. In *IEDM Technical Digest. IEEE International Electron Devices Meeting, 2004.*, pages 657–660. IEEE, 2004.
- [17] Peter Moon, Vinay Chikarmane, Kevin Fischer, Rohit Grover, Tarek A Ibrahim, Doug Ingerly, Kevin J Lee, Chris Litteken, Tony Mule, and Sarah Williams. Process and electrical results for the on-die interconnect stack for intel’s 45nm process generation. *Intel Technology Journal*, 12(2), 2008.
- [18] S Natarajan, M Armstrong, M Bost, R Brain, M Brazier, C-H Chang, V Chikarmane, M Childs, H Deshpande, K Dev, et al. A 32nm logic technology featuring 2 nd-generation high-k+ metal-gate transistors, enhanced channel strain and 0.171 μm^2 sram cell size in a 291mb array. In *2008 IEEE International Electron Devices Meeting*, pages 1–3. IEEE, 2008.
- [19] Ross Anderson and Markus Kuhn. Tamper resistance-a cautionary note. In *Proceedings of the second Usenix workshop on electronic commerce*, volume 2, pages 1–11, 1996.
- [20] Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall, and Claire Whelan. The sorcerer’s apprentice guide to fault attacks. *Proceedings of the IEEE*, 94(2):370–382, 2006.
- [21] Ye-Sheng Kuo, Sonal Verma, Thomas Schmid, and Prabal Dutta. Hijacking power and bandwidth from the mobile phone’s audio interface. In *Proceedings of the First ACM Symposium on Computing for Development*, ACM DEV ’10, pages 24:1–24:10, New York, NY, USA, 2010. ACM.

- [22] Toshihiro Katashita, Yohei Hori, Hirofumi Sakane, and Akashi Satoh. Side-channel attack standard evaluation board sasebo-w for smartcard testing. *Power*, 3(2012):400, 2012.
- [23] Rajesh Velegalati and Jens-Peter Kaps. Introducing fobos: Flexible open-source board for side-channel analysis. In *Constructive Side-Channel Analysis and Secure Design (COSADE), Third International Workshop on: Work in Progress Session*, 2012.
- [24] Juhan Kim, Kyunghee Oh, Dohoo Choi, and Howon Kim. Scarf: profile-based side channel analysis resistant framework. In *Proceedings of the International Conference on Security and Management (SAM)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012.
- [25] Rambus: Dpa workstation analysis platform.
- [26] Riscure: Inspector sca.
- [27] Colin OFlynn and Zhizhang David Chen. Chipwhisperer: An open-source platform for hardware embedded security research. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*, pages 243–260. Springer, 2014.
- [28] ATMEL. *32-bit Atmel AVR Microcontroller*, 1 2012.
- [29] Roger Forster. Manchester encoding: opposing definitions resolved. *Engineering Science and Education Journal*, 9(6):278–280, 2000.
- [30] Zhe Liu, Johann Großschädl, and Ilya Kizhvatov. Efficient and side-channel resistant rsa implementation for 8-bit avr microcontrollers. In *Workshop on the Security of the Internet of Things-SOCIOT*, volume 10, 2010.
- [31] Jean-Sébastien Coron and Ilya Kizhvatov. Analysis and improvement of the random delay countermeasure of ches 2009. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 95–109. Springer, 2010.
- [32] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer Science & Business Media, 2008.
- [33] Stefan Tillich and Christoph Herbst. Attacking state-of-the-art software countermeasures case study for aes. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 228–243. Springer, 2008.
- [34] Thomas Plos. Evaluation of the detached power supply as side-channel analysis countermeasure for passive uhf rfid tags. In *Cryptographers Track at the RSA Conference*, pages 444–458. Springer, 2009.

- [35] Stefan Tillich, Christoph Herbst, and Stefan Mangard. Protecting aes software implementations on 32-bit processors against power analysis. In *Applied Cryptography and Network Security*, pages 141–157. Springer, 2007.
- [36] Brandon Lucia and Benjamin Ransford. A simpler, safer programming and execution model for intermittent systems. In *Conference on Programming Language Design and Implementation*, PLDI, pages 575–585, June 2015.
- [37] Alexei Colin and Brandon Lucia. Chain: tasks and channels for reliable intermittent programs. In *International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, OOPSLA, pages 514–530, November 2016.
- [38] Paul Kocher, Joshua Jaffe, Benjamin Jun, and Pankaj Rohatgi. Introduction to differential power analysis. *Journal of Cryptographic Engineering*, 1(1):5–27, 2011.
- [39] Paul Kocher, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. Spectre attacks: Exploiting speculative execution. *arXiv preprint arXiv:1801.01203*, 2018.
- [40] David Brumley and Dan Boneh. Remote timing attacks are practical. *Computer Networks*, 48(5):701–716, 2005.
- [41] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. Flipping bits in memory without accessing them: An experimental study of dram disturbance errors. In *ACM SIGARCH Computer Architecture News*, volume 42, pages 361–372. IEEE Press, 2014.
- [42] Joseph Bonneau and Ilya Mironov. Cache-collision timing attacks against aes. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 201–215. Springer, 2006.
- [43] Michael Hutter and Jörn-Marc Schmidt. The temperature side channel and heating fault attacks. In *International Conference on Smart Card Research and Advanced Applications*, pages 219–235. Springer, 2013.
- [44] Michel Agoyan, Jean-Max Dutertre, David Naccache, Bruno Robisson, and Assia Tria. When clocks fail: On critical paths and clock faults. In *International Conference on Smart Card Research and Advanced Applications*, pages 182–193. Springer, 2010.
- [45] Amine Dehbaoui, Jean-Max Dutertre, Bruno Robisson, and Assia Tria. Electromagnetic transient faults injection on a hardware and a software implementations of aes. In *2012 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 7–15. IEEE, 2012.

- [46] Sergei P Skorobogatov and Ross J Anderson. Optical fault induction attacks. In *International workshop on cryptographic hardware and embedded systems*, pages 2–12. Springer, 2002.
- [47] Yuan-Liang Li, Teong-Guang Yew, Chee Yee Chung, and DG Fugueroa. Design and performance evaluation of microprocessor packaging capacitors using integrated capacitor-via-plane model. *IEEE transactions on advanced packaging*, 23(3):361–367, 2000.
- [48] E. Song, J. S. Pak, and J. Kim. Tsv-based decoupling capacitor schemes in 3d-ic. In *Electronic Components and Technology Conference*, pages 1340–1344, May 2012.
- [49] Yongki Min, Reynaldo Olmedo, Michael Hill, Kaladhar Radhakrishnan, Kemal Aygun, Mostafa Kabiri-Badr, Rahul Panat, Sriram Dattaguru, and Haluk Balkan. Embedded capacitors in the next generation processor. In *Electronic Components and Technology Conference*, ECTC, pages 1225–1229, May 2013.
- [50] Srivaths Ravi, Anand Raghunathan, and Srimat Chakradhar. Tamper resistance mechanisms for secure embedded systems. In *VLSI Design, 2004. Proceedings. 17th International Conference on*, pages 605–611. IEEE, 2004.
- [51] Alessandro Barenghi, Guido M Bertoni, Luca Breveglieri, Mauro Pelliccioli, and Gerardo Pelosi. Low voltage fault attacks to aes. In *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*, pages 7–12. IEEE, 2010.
- [52] Jörn-Marc Schmidt and Christoph Herbst. A practical fault attack on square and multiply. In *Fault Diagnosis and Tolerance in Cryptography, 2008. FDTC'08. 5th Workshop on*, pages 53–58. IEEE, 2008.
- [53] Konrad J Kulikowski, Mark G Karpovsky, and Alexander Taubin. Power attacks on secure hardware based on early propagation of data. In *On-Line Testing Symposium, 2006. IOLTS 2006. 12th IEEE International*, pages 6–pp. IEEE, 2006.
- [54] Dan Boneh, Richard A DeMillo, and Richard J Lipton. On the importance of eliminating errors in cryptographic computations. *Journal of cryptology*, 14(2):101–119, 2001.
- [55] Kaiyuan Yang, Matthew Hicks, Qing Dong, Todd Austin, and Dennis Sylvester. A2: Analog malicious hardware. In *Symposium on Security and Privacy*, Oakland, pages 18–37, May 2016.
- [56] Joan Daemen and Vincent Rijmen. Aes proposal: Rijndael. 1999.
- [57] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

- [58] Stefan Mangard. A simple power-analysis (spa) attack on implementations of the aes key expansion. In *International Conference on Information Security and Cryptology*, pages 343–358. Springer, 2002.
- [59] Bert den Boer, Kerstin Lemke, and Guntram Wicke. A dpa attack against the modular reduction within a crt implementation of rsa. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 228–243. Springer, 2002.
- [60] Daniel J Bernstein. Cache-timing attacks on aes. 2005.
- [61] Texas Instruments. *Migrating to the SimpleLink™MSP432™Family™*, 3 2015.
- [62] Atmel. *32-bit Atmel AVR Microcontroller AT32UC3L064, AT32UC3L032, AT32UC3L016*, 1 2012.
- [63] Texas Instruments. *MSP432P401R, MSP432P401M SimpleLink Mixed-Signal Microcontrollers datasheet*, 9 2017. Rev. G.
- [64] Texas Instruments. *FRAM FAQs*, 2014.
- [65] Pico Technology. *TA058 50 MHz +/-700 V Differential Probe Users Manual*, 2008.
- [66] Langer EMV-Technik. *MEASURING SET-UP NEAR FIELD MEASURING*, 2014.
- [67] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In *International workshop on cryptographic hardware and embedded systems*, pages 251–261. Springer, 2001.
- [68] Cheuk Wong. Analysis of dpa and dema attacks. 2012.
- [69] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *Smart Card Programming and Security*, pages 200–210. Springer, 2001.
- [70] Kris Tiri, David Hwang, Alireza Hodjat, B Lai, Shenglin Yang, Patrick Schaumont, and Ingrid Verbauwhede. A side-channel leakage free coprocessor ic in 0.18 μm cmos for embedded aes-based cryptographic and biometric processing. In *Proceedings of the 42nd annual Design Automation Conference*, pages 222–227. ACM, 2005.
- [71] Joshua M Jaffe, Paul C Kocher, and Benjamin C Jun. Balanced cryptographic computational method and apparatus for leak minimizational in smartcards and other cryptosystems, January 21 2003. US Patent 6,510,518.
- [72] Joshua M Jaffe, Paul C Kocher, and Benjamin C Jun. Hardware-level mitigation and dpa countermeasures for cryptographic devices, November 25 2003. US Patent 6,654,884.

- [73] Alex Bystrov, Danil Sokolov, Alex Yakovlev, and A Koelmans. Balancing power signature in secure systems. In *Proc. 14th UK Asynchronous Forum*, volume 2003, 2003.
- [74] Kris Tiri and Ingrid Verbauwhede. A logic level design methodology for a secure dpa resistant asic or fpga implementation. In *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, volume 1, pages 246–251. IEEE, 2004.
- [75] Danil Sokolov, Julian Murphy, Alex Bystrov, and Alex Yakovlev. Improving the security of dual-rail circuits. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 282–297. Springer, 2004.
- [76] François Macé, François-Xavier Standaert, Ilham Hassoune, Jean-Didier Legat, Jean-Jacques Quisquater, et al. A dynamic current mode logic to counteract power analysis attacks. In *Proc. 19th International Conference on Design of Circuits and Integrated Systems (DCIS)*, pages 186–191, 2004.
- [77] Zeynep Toprak and Yusuf Leblebici. Low-power current mode logic for improved dpa-resistance in embedded systems. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 1059–1062. IEEE, 2005.
- [78] Simon Moore, Ross Anderson, Paul Cunningham, Robert Mullins, and George Taylor. Improving smart card security using self-timed circuits. In *Asynchronous Circuits and Systems, 2002. Proceedings. Eighth International Symposium on*, pages 211–218. IEEE, 2002.
- [79] Zhong Chuan Yu, Stephen B Furber, and Luis A Plana. An investigation into the security of self-timed circuits. In *Asynchronous Circuits and Systems, 2003. Proceedings. Ninth International Symposium on*, pages 206–215. IEEE, 2003.
- [80] Patrick Rakers, Larry Connell, Tim Collins, and Dan Russell. Secure contactless smartcard asic with dpa protection. *IEEE Journal of Solid-State Circuits*, 36(3):559–565, 2001.
- [81] Girish B Ratanpal, Ronald D Williams, and Travis N Blalock. An on-chip signal suppression countermeasure to power analysis attacks. *IEEE Transactions on Dependable and Secure Computing*, 1(3):179–189, 2004.
- [82] Paul C Kocher, Joshua M Jaffe, and Benjamin C Jun. Using unpredictable information to minimize leakage from smartcards and other cryptosystems, December 4 2001. US Patent 6,327,661.
- [83] Adi Shamir. Protecting smart cards from passive power analysis with detached power supplies. In *Conference on Cryptographic Hardware and Embedded Systems*, CHES, pages 71–77, August 2000.

- [84] Andreas Gornik, Amir Moradi, Jürgen Oehm, and Christof Paar. A hardware-based countermeasure to reduce side-channel leakage: Design, implementation, and evaluation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(8):1308–1319, 2015.
- [85] Andrey V Mezhiba and Eby G Friedman. Impedance characteristics of power distribution grids in nanoscale integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(11):1148–1155, 2004.
- [86] Neil HE Weste and David Harris. *CMOS VLSI design: a circuits and systems perspective*. Pearson Education India, 2015.