Draft Project Proposal.

# Introduction

This is a proposal for a research project on applying side-channel attacks to non-cryptographic applications. In section 2, we describe what side channel attacks are. In section 3, we present the research questions that this project will attempt to address. Section 4 shows the path forward.

# Side Channel Attacks

Computers are physical realizations of abstract machines. Indeed, the origins of computation are rooted in purely mathematical designs like the Turing Machine. Even von Neumann’s architecture, upon which our modern systems are based, is an abstract system (albeit one that lends itself easily to a real implementation).

When we program computers, it is easiest to think abstractly. We hide complexity behind interfaces. Systems are organized into layers of abstraction with the highest-level ideas and operations at the top, the bits and bytes further down, and the physical implementation – wires and voltages on them -- still further down. The lower layers are available for us to use, but we do not need to understand their internal function to get our work done. Likewise, designers of the lower layers do need to know in advance all of the things we will use their layer for.

Thinking about computers this way makes them manageable. It is impossible to remember all of the details, but if the details can be summarized succinctly, divided into layers and regions with boundaries, we can remember the summaries and get on with our work without worrying about the details.

Seeing computers this way also causes problems. In particular, security vulnerabilities arise when a user at one level misuses the functionality provided by another level.

Seeing computers abstractly, we spend most of our effort thinking about explicit communication paths, for they are the ones that matter. The user does not care exactly how much power their computer is using, how much noise it makes, or how long it takes to perform an operation, as long as it stays within reasonable bounds. These variations are called side channels, and they tell us a little bit about what is going on inside the computer.

Indeed, side channels sometimes leak sensitive information. Research [CITE] has shown that by taking a close look at the power usage signature of a smart card, we can tell which bits of its secret key are 1 and which are 0 (because processing the 1 bits uses more power than the 0 bits), thus extracting the secret key. Attacks like these, which use side channels to gather information about a system are called side channel attacks. Many more examples are known.

TODO: Cite more.

Side channels are interesting because a computer program can be logically correct yet still vulnerable to a side channel attack. The program can respond to all inputs in all the right ways, always getting the right answer, but still be leaking its secrets out through the time it takes or the amount of power it uses.

Side channel attacks are still not well understood. The next section presents the open questions about side channels that this project will attempt to answer.

# Research Problem

While many examples of side channels have been produced, they are still poorly understood. This is mainly because we do not yet know everything that is possible. To get a better understanding, we need to see more attacks, and in particular, more general attacks, before we can effectively defend our systems.

Most side channels are applied in the context of cryptography. Cryptography is especially easy to attack with side channels, because small information leaks, like leaking a secret key, can be devastating to the entire system. Compounding the issue, crypto usually makes heavy use of the few secret bits that it has, making them more exposed to side channel attacks.

TODO: Cite counterexamples and respond to that.

We ask whether this focus on cryptography is because non-cryptographic things are hard to attack, or if cryptography is just the low hanging fruit that gets all of the attention. This is an important question, since if we have been putting undue focus on cryptography, we may be overlooking significant and exploitable flaws in the programs we rely on every day [EXAMPLE/ILLUSTRATE] (WORDS: Dormant, undiscovered).

TODO: something about defense

TODO: motivation that side channels are hard to fix, because they are in the hardware (so even if they are sev:lo they can be so expensive it makes it worthwhile)

In particular, we ask the following questions:

* What is the impact of side channel attacks on non-cryptographic applications?
* Do all developers, not just the ones working on cryptoraphy, need to be educated about side channels and design their code to resist them, or is it ok to only consider side channels in security-critical code?
* If the above two questions are answered in the affirmative, what can we do to reduce the risk? Are there automated technological defenses? Are there easy-to-follow coding guidelines that, when followed, make side channels less of a risk?
* Given the broadened scope, can side channels be applied for good instead of bad? Can we use side channels as a means of identifying attacks?

If this research finds that such attacks exist and are widespread, it will motivate the community to broaden their horizons beyond crypto.

The next section discusses the methods we can use to answer these questions, and what the result of the project will be if it is successful.

# Steps Forward

HERE, NOT above, note FLUSH+RELOAD as being particularly applicable to non-crypto attacks.

* Attack Non-crypto apps with FLUSH+RELOAD.
* Strong experimental / reproducible (scientific) basis.
* Learn from the attacks to see if there are effective defenses.

The essence of this research problem is to determine whether side channels are realistic threats against non-cryptographic applications. The best way to answer this is to try to attack them.

TODO: Address project failure here too (i.e. what happens if no attacks are found)?

# Conclusion

Just summarize the previous sections.