Side-channel power attacks

With demos veryoooo

What is a side-channel?

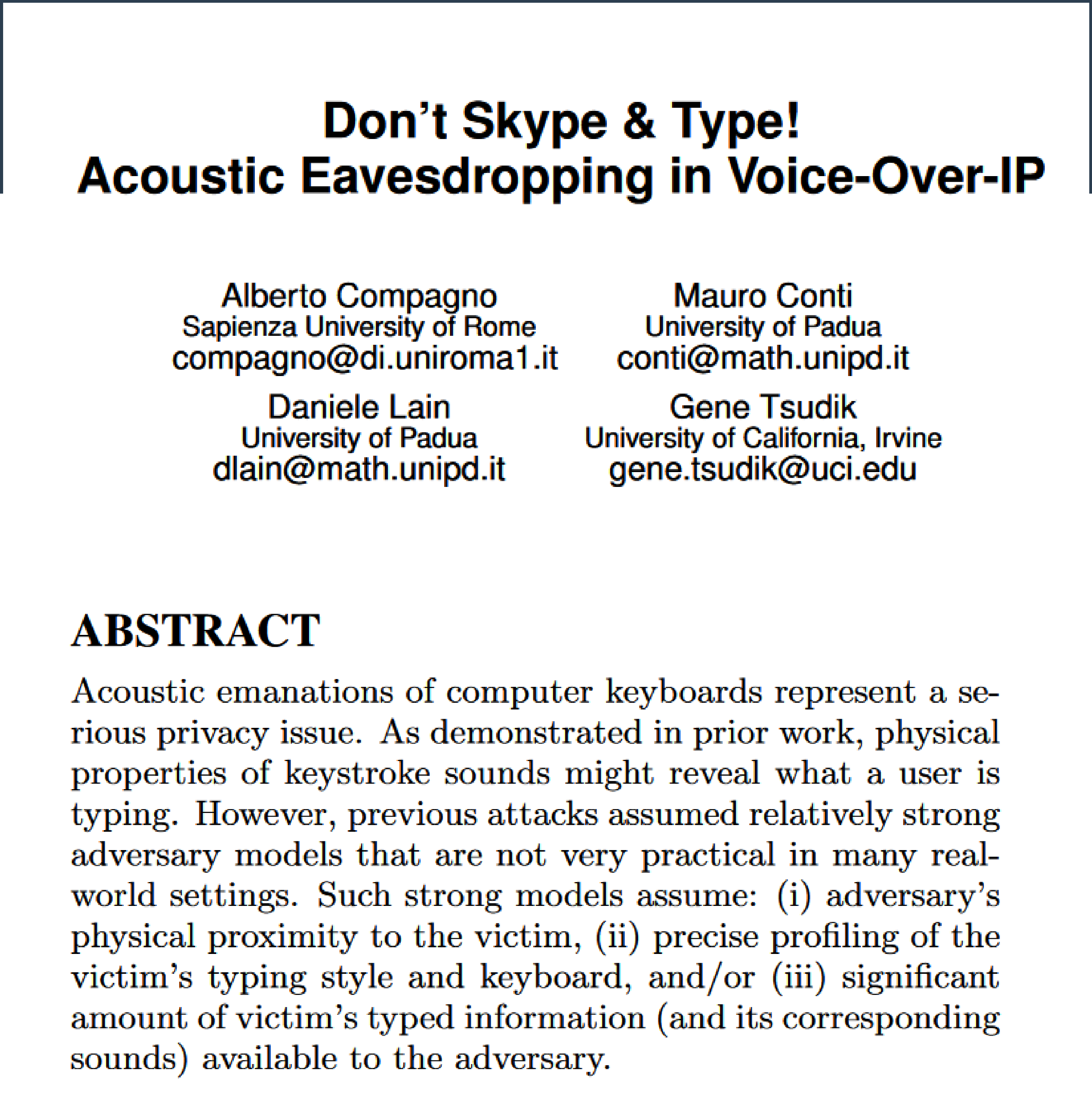
- Don’t be silly, you learned about this with Meltdown
- A side-channel is unintended leaked information through a secondary channel
- Side channel attacks exploit this leaked data to perform some attack

What leaks information?

- On systems
- On small systems
- On large systems
- On services (sql, etc)

What is our attack vector? What is leaked?

- The time that it takes to do operations
- The power consumption of the device
- The EM radiation of the device
- The sound the device makes
- Anything you can think of



Don’t Skype & Type! Acoustic Eavesdropping in Voice-Over-IP

Alberto Compagno
Sapienza University of Rome
compagno@di.uniroma1.it

Mauro Conti
University of Padua
conti@math.unipd.it

Daniele Lain
University of Padua
dlain@math.unipd.it

Gene Tsudik
University of California, Irvine
gene.tsudik@uci.edu

ABSTRACT

Acoustic emanations of computer keyboards represent a serious privacy issue. As demonstrated in prior work, physical properties of keystroke sounds might reveal what a user is typing. However, previous attacks assumed relatively strong adversary models that are not very practical in many real-world settings. Such strong models assume: (i) adversary’s physical proximity to the victim, (ii) precise profiling of the victim’s typing style and keyboard, and/or (iii) significant amount of victim’s typed information (and its corresponding sounds) available to the adversary.

What is our attack vector? What is leaked?

- The time that it takes to do operations
- The power consumption of the device
- The EM radiation of the device
- The sound the device makes
- Anything you can think of

Lets focus on this!

What does it mean to leak information over power?

- If you do expensive operations, you’re loud, hot, and take a lot of resources
- If you do less expensive things…
- The more bits you flip the more power you use

What level of detail can we expect?

- General code structure
- (potentially) The instructions that you run
- (potentially) The time that they take
- And even, (possibly) the data that instructions were run with!

Data-Operation level snooping

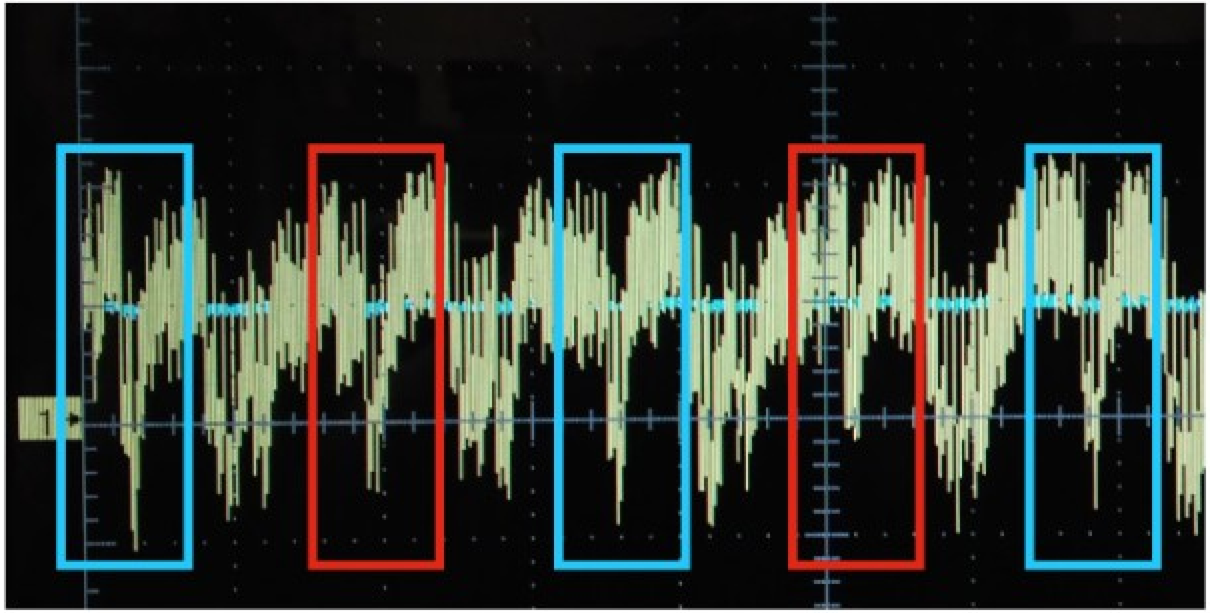
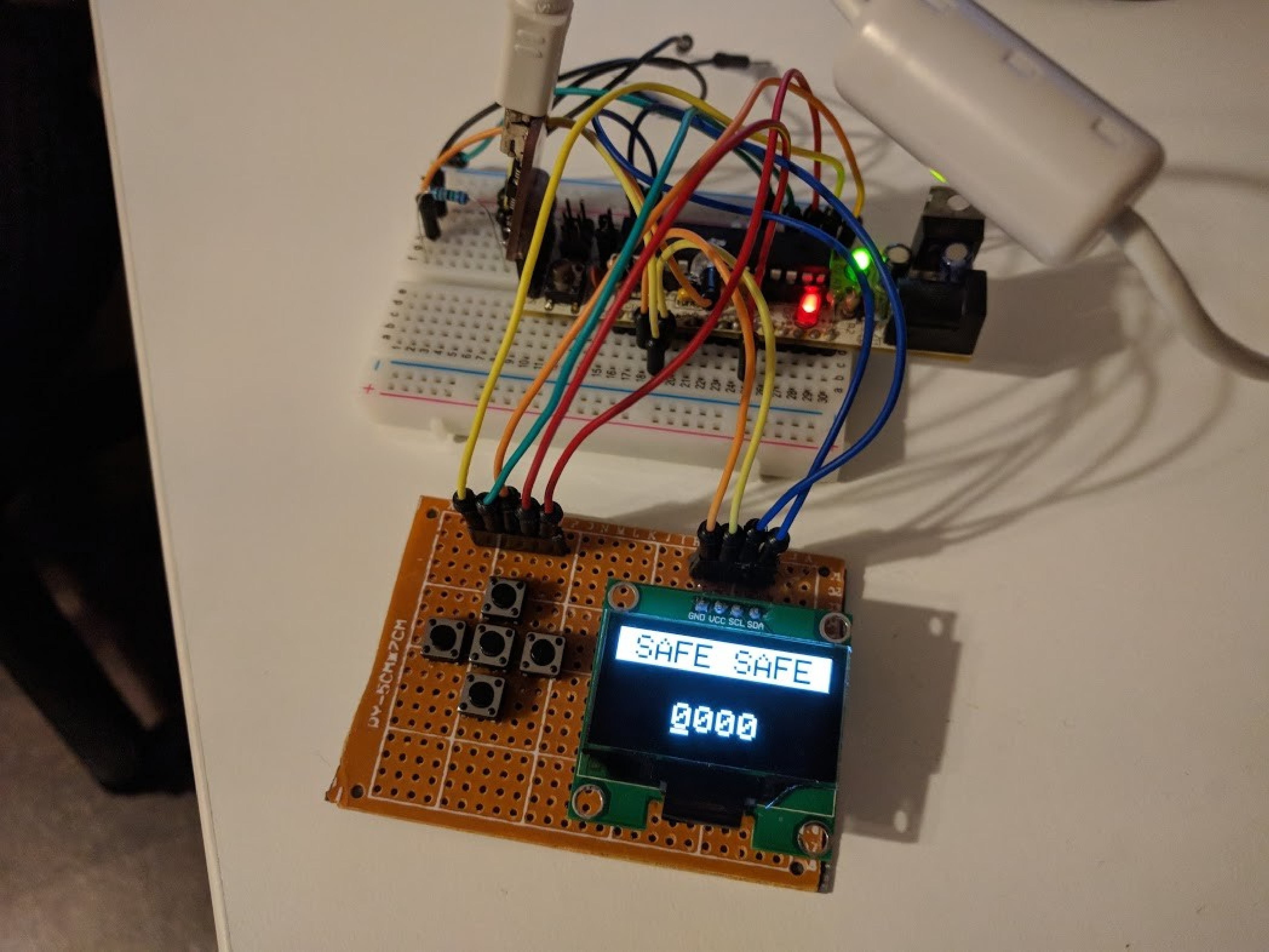
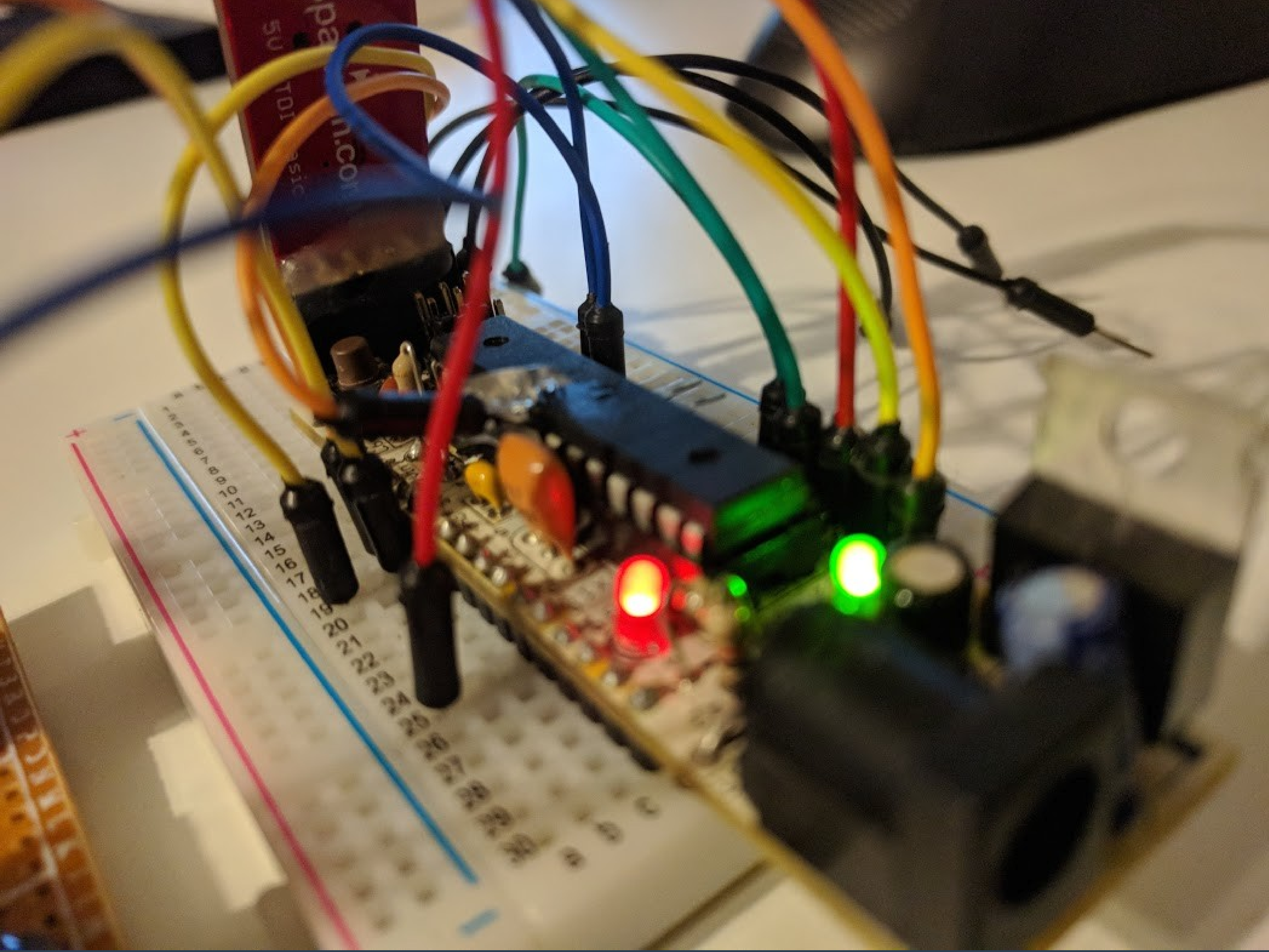
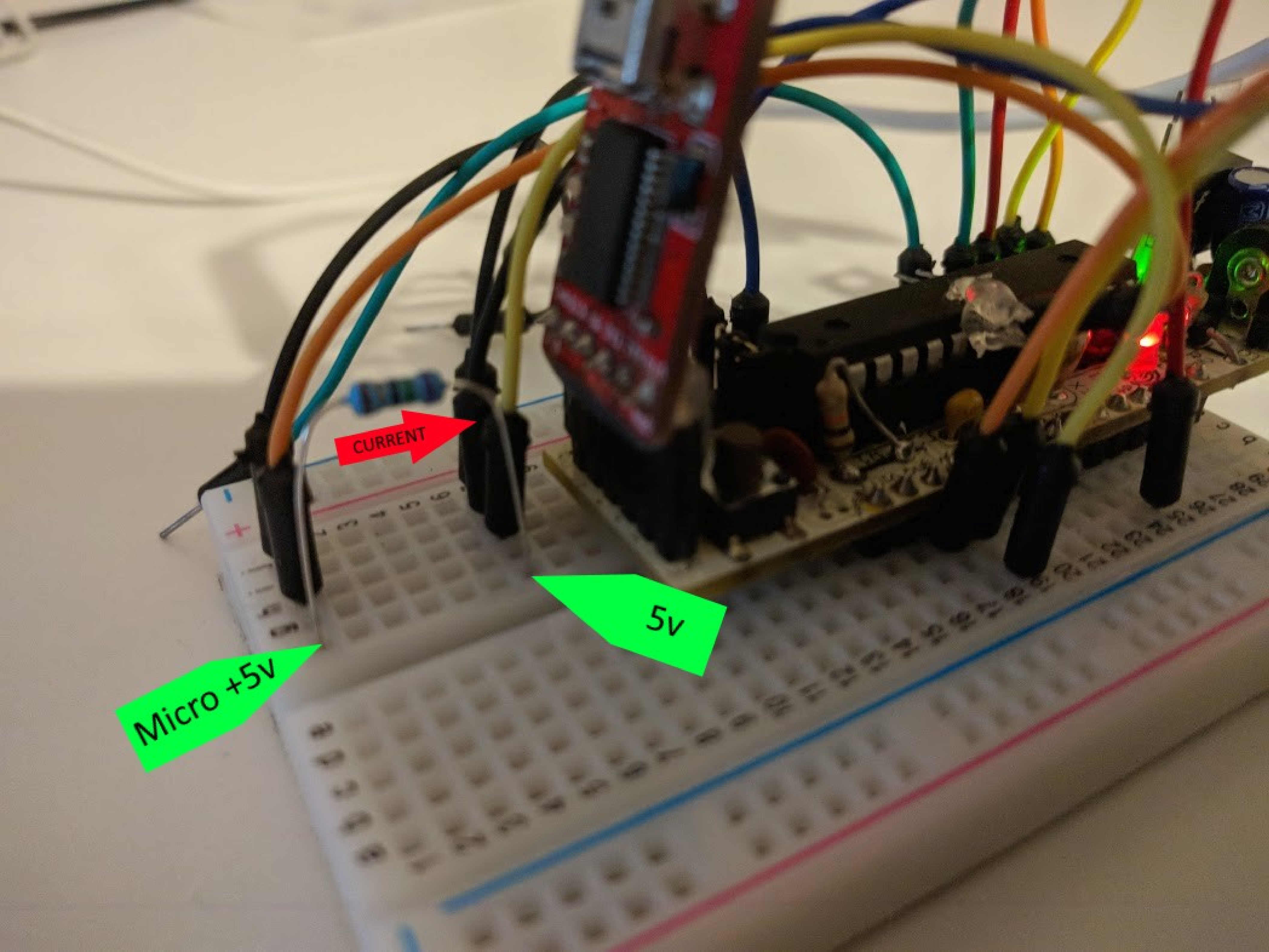


Fig. 3: XOR operations alternated between results of all ”0”s and all ”1”s. The power consumption was noticeably different depending on the Hamming Weight of the result.

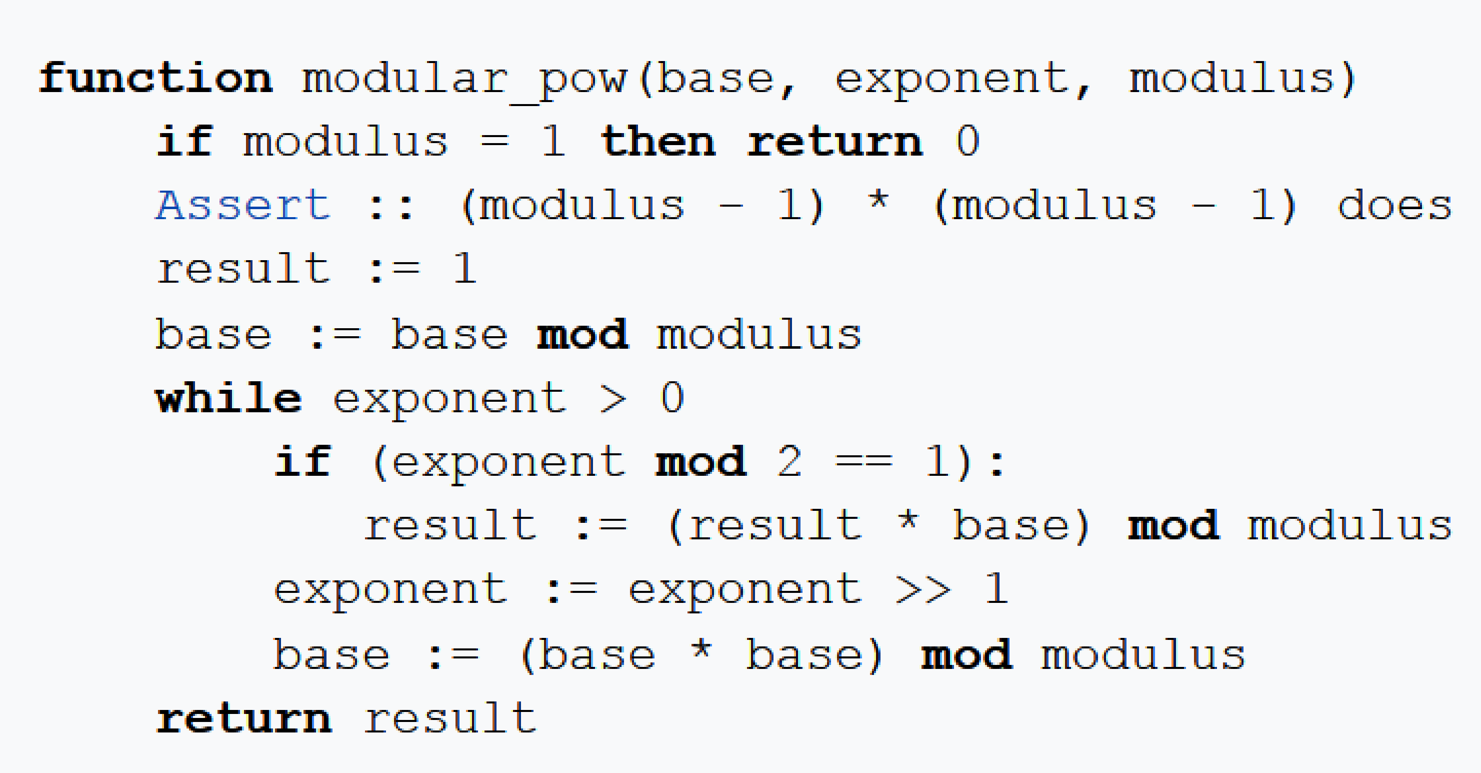

GND VCC SCL SDA
SAFE SAFE
0000
DY-5CM*7CM


5V
TDI
m.co
asic


CURRENT
5v
Micro +5v

But RSA

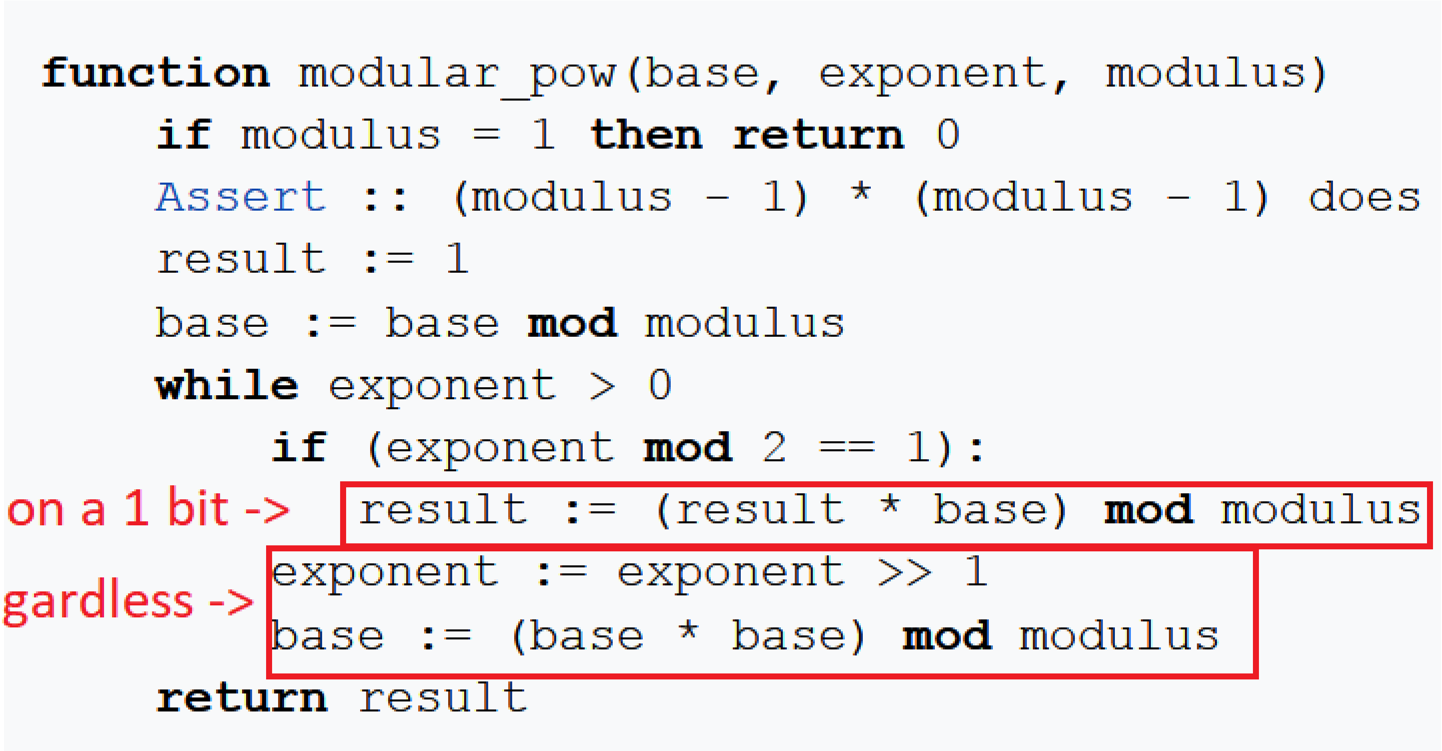
- Lets look at a key part of the RSA algorithm: the modular exponentiation algorithm



```
function modular_pow(base, exponent, modulus)
    if modulus = 1 then return 0
    Assert :: (modulus - 1) * (modulus - 1) does
    result := 1
    base := base mod modulus
    while exponent > 0
        if (exponent mod 2 == 1):
           result := (result * base) mod modulus
        exponent := exponent >> 1
        base := (base * base) mod modulus
    return result
```

But RSA

- Lets look at a key part of the RSA algorithm: the modular exponentiation algorithm



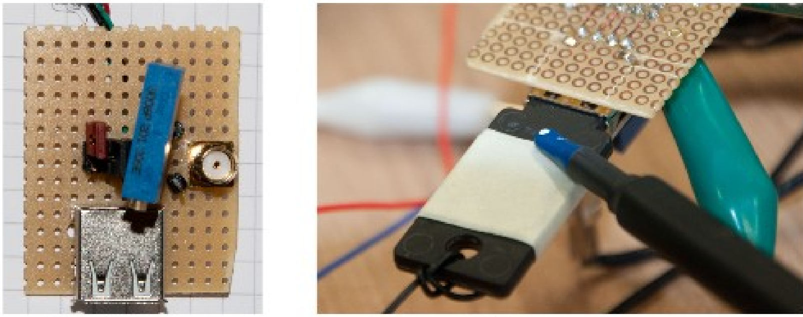
```
function modular_pow(base, exponent, modulus)
    if modulus = 1 then return 0
    Assert :: (modulus - 1) * (modulus - 1) does
    result := 1
    base := base mod modulus
    while exponent > 0
        if (exponent mod 2 == 1):
            result := (result * base) mod modulus
        exponent := exponent >> 1
        base := (base * base) mod modulus
    return result
```

Do a thing on a 1 bit ->

Do these regardless ->

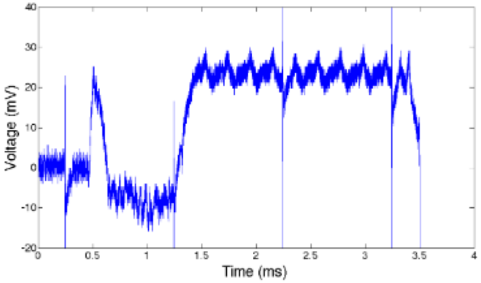
Other examples?

- The Yubikey 2 is a two-factor authentication key-fob
- It's vulnerable to EM and power based side-channel attacks
- Completely non-invasive
- "The EM trace allows us to separately observe every clock cycle, while the power consumption trace only shows the overall round structure

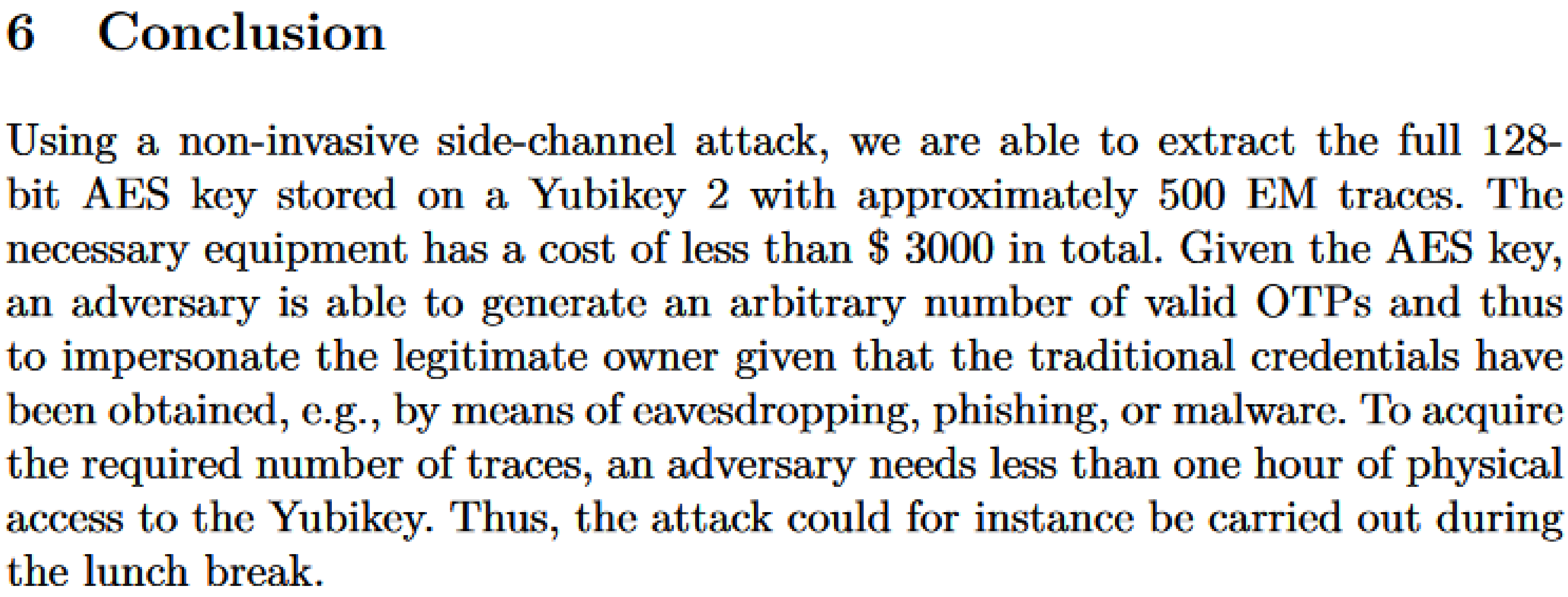


(a) USB adaptor (b) EM probe

the voltage drop, a pattern can be observed that resembles a structure with ten rounds, each approximately 200 µs long, cf. Fig. 8.



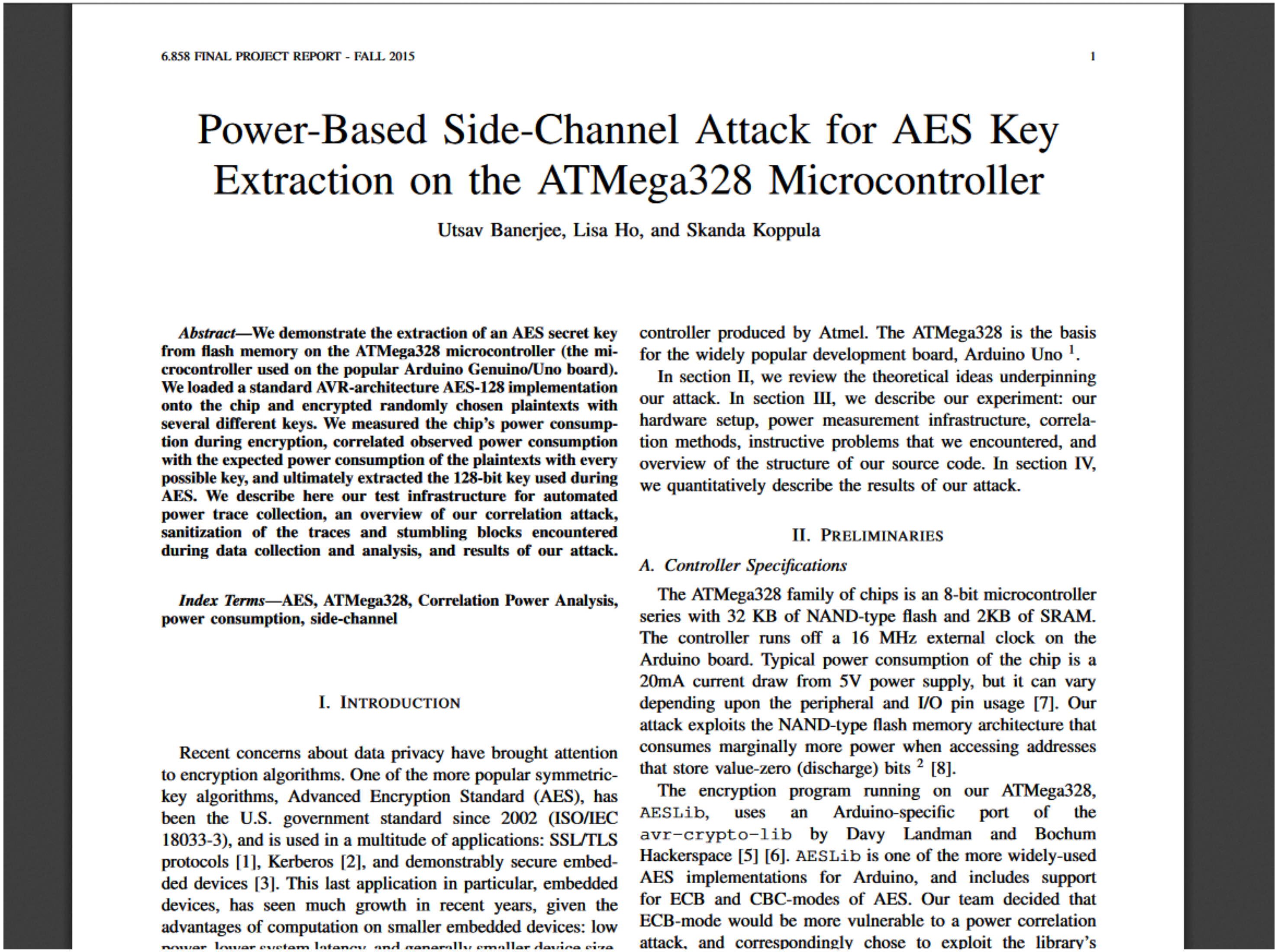
Yubikey 2



6 Conclusion

Using a non-invasive side-channel attack, we are able to extract the full 128-bit AES key stored on a Yubikey 2 with approximately 500 EM traces. The necessary equipment has a cost of less than $ 3000 in total. Given the AES key, an adversary is able to generate an arbitrary number of valid OTPs and thus to impersonate the legitimate owner given that the traditional credentials have been obtained, e.g., by means of eavesdropping, phishing, or malware. To acquire the required number of traces, an adversary needs less than one hour of physical access to the Yubikey. Thus, the attack could for instance be carried out during the lunch break.

AES Key extraction on the Arduino!



Power-Based Side-Channel Attack for AES Key Extraction on the ATMega328 Microcontroller

Utsav Banerjee, Lisa Ho, and Skanda Koppula

***Abstract*—We demonstrate the extraction of an AES secret key from flash memory on the ATMega328 microcontroller (the microcontroller used on the popular Arduino Genuino/Uno board). We loaded a standard AVR-architecture AES-128 implementation onto the chip and encrypted randomly chosen plaintexts with several different keys. We measured the chip's power consumption during encryption, correlated observed power consumption with the expected power consumption of the plaintexts with every possible key, and ultimately extracted the 128-bit key used during AES. We describe here our test infrastructure for automated power trace collection, an overview of our correlation attack, sanitization of the traces and stumbling blocks encountered during data collection and analysis, and results of our attack.**

***Index Terms*—AES, ATMega328, Correlation Power Analysis, power consumption, side-channel**

I. Introduction

Recent concerns about data privacy have brought attention to encryption algorithms. One of the more popular symmetric-key algorithms, Advanced Encryption Standard (AES), has been the U.S. government standard since 2002 (ISO/IEC 18033-3), and is used in a multitude of applications: SSL/TLS protocols [1], Kerberos [2], and demonstrably secure embedded devices [3]. This last application in particular, embedded devices, has seen much growth in recent years, given the advantages of computation on smaller embedded devices: low power, lower system latency, and generally smaller device size controller produced by Atmel. The ATMega328 is the basis for the widely popular development board, Arduino Uno [1].

In section II, we review the theoretical ideas underpinning our attack. In section III, we describe our experiment: our hardware setup, power measurement infrastructure, correlation methods, instructive problems that we encountered, and overview of the structure of our source code. In section IV, we quantitatively describe the results of our attack.

II. Preliminaries

A. Controller Specifications

The ATMega328 family of chips is an 8-bit microcontroller series with 32 KB of NAND-type flash and 2KB of SRAM. The controller runs off a 16 MHz external clock on the Arduino board. Typical power consumption of the chip is a 20mA current draw from 5V power supply, but it can vary depending upon the peripheral and I/O pin usage [7]. Our attack exploits the NAND-type flash memory architecture that consumes marginally more power when accessing addresses that store value-zero (discharge) bits [2] [8].

The encryption program running on our ATMega328, `AESLib`, uses an Arduino-specific port of the `avr-crypto-lib` by Davy Landman and Bochum Hackerspace [5] [6]. `AESLib` is one of the more widely-used AES implementations for Arduino, and includes support for ECB and CBC-modes of AES. Our team decided that ECB-mode would be more vulnerable to a power correlation attack, and correspondingly chose to exploit the library's

References

- “Side-Channel Attacks on the Yubikey 2 One-Time Password Generator” https://www.emsec.rub.de/media/crypto/veroeffentlichungen/2014/02/04/paper_yubikey_sca.pdf
- Power-Based Side-Channel Attack for AES Key Extraction on the ATMega338 Microcontroller https://css.csail.mit.edu/6.858/2015/projects/utsav-lisayz-skoppula.pdf
- Don’t Skype & Type! Acoustic Eavesdropping in Voice-Over-IP https://arxiv.org/pdf/1609.09359.pdf