Side-Channel Analysis and Fault Injection: CPA & DFA Seminar

PV204 Security Technologies

Łukasz Chmielewski

CRoCS,
Masaryk University,
chmiel@fi.muni.cz

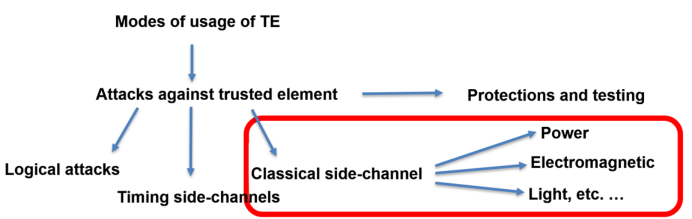
Part of the content is reused with permission of Lejla Batina, Radboud University Nijmegen, The Netherlands.

April 20, 2023

Outline

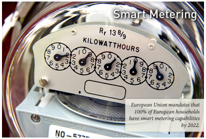
1. Introduction
2. ChipWhisperer
3. Practical Side-channel Analysis
4. Typical Attack Procedure
5. Capturing AES traces with ChipWhisperer
6. Conclusions

Plan for Today



1. ChipWhisperer Installation
2. How to build setup? When is it good enough? Etc...
3. Analysis of some captured by you traces
4. Even if we have enough devices: please work in pairs! Discuss you solutions.

Known challenge: embedded crypto devices



European Union mandates that 100% of European households have smart metering capabilities by 2022.

Disclaimers

- Goal: show you how building a setup for SCA looks and what the hurdles in very short time...
- This seminar is for you and there is no homework. We can look at what you want, so please let me know when you have questions...
- Since there are many technical components, things might get shaky...

ChipWhisperer

ChipWhisperer

What is chipwhisperer?

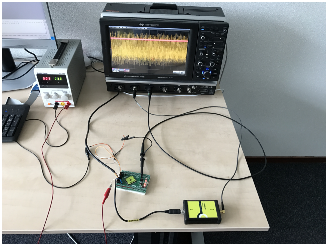
To perform a side channel attack, two things is needed,

1. A capture hardware:

   oscilloscope: captures small signals with a precisely synchronized clock.

2. A target board:

   processor: is programmed to perform secure operation.



Setting up the hardware for side channel attacks is not easy!

CW1101 ChipWhisperer-Nano resolves difficulties, but hard to be customized!

CW1101 ChipWhisperer-Nano:

- comes with the capture hardware and the target together on a single board.
- has ARM Cortex-M0 processor.
- we have 20 of them in the lab (and 2 CW1173 ChipWhisperer-Lite based on ARM Cortex-M4F)

CW1101 ChipWhisperer Nano

The ChipWhisperer Nano comes with two main parts:

1. a multi-purpose power analysis capture hardware, and
2. a microcontroller (target board) which you can implement algorithms onto.

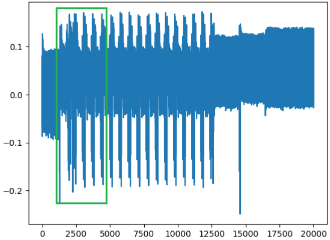


Figure: Figure from: https://myrelabs.com/breaking-rsa-with-chipwhisperer/

Documentation can be found at:

- https://www.newae.com/products/NAE-CW1101
- https://chipwhisperer.readthedocs.io/en/latest/
- https://wiki.newae.com/Main_Page.

CW1101 ChipWhisperer Nano (Cont'd)

Open-source toolchain for hardware security research and education

Hardware: The ChipWhisperer uses a capture hardware and a target board.

- Schematics and PCB layouts for capture hardware & target board

Firmware: Three separate pieces of firmware are used on the hardware.

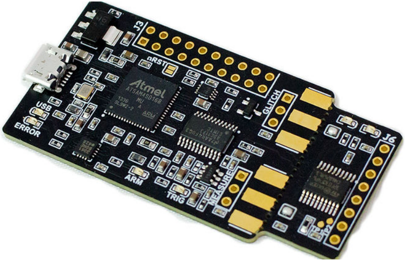
- The capture hardware:
  a USB controller (in C) & an FPGA for high-speed captures (in Verilog)
  In “hardware/capture” of the ChipWhisperer Github Repo.
- The target device has its own firmware (mostly in C)
  In “hardware/victims/firmware” of the ChipWhisperer Github Repo.

Software: The ChipWhisperer software includes

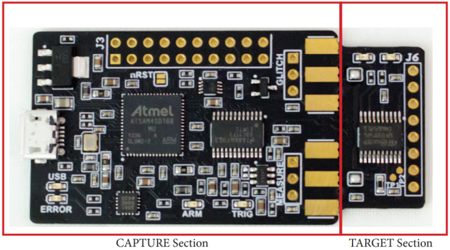
- a Python API for talking to hardware (Capture), and
- a Python API for processing power traces from hardware (Analyzer).

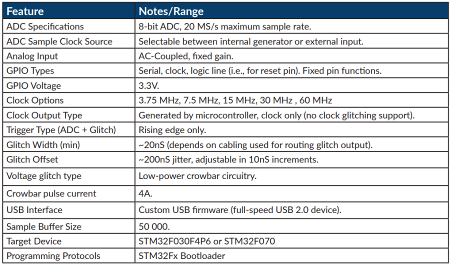
Hardware Specification

- Hardware documentation can be found at:
  https://rtfm.newae.com
- More specifically look to:
  https://media.newae.com/datasheets/NAE-CW1101_datasheet.pdf



Hardware Specification





Feature	Notes/Range
ADC Specifications	8-bit ADC, 20 MS/s maximum sample rate.
ADC Sample Clock Source	Selectable between internal generator or external input.
Analog Input	AC-Coupled, fixed gain.
GPIO Types	Serial, clock, logic line (i.e., for reset pin). Fixed pin functions.
GPIO Voltage	3.3V.
Clock Options	3.75 MHz, 7.5 MHz, 15 MHz, 30 MHz , 60 MHz
Clock Output Type	Generated by microcontroller, clock only (no clock glitching support).
Trigger Type (ADC + Glitch)	Rising edge only.
Glitch Width (min)	~20nS (depends on cabling used for routing glitch output).
Glitch Offset	~200nS jitter, adjustable in 10nS increments.
Voltage glitch type	Low-power crowbar circuitry.
Crowbar pulse current	4A.
USB Interface	Custom USB firmware (full-speed USB 2.0 device).
Sample Buffer Size	50 000.
Target Device	STM32F030F4P6 or STM32F070
Programming Protocols	STM32Fx Bootloader

Figure: from https://media.newae.com/datasheets/NAE-CW1101_datasheet.pdf

- **REMARK:** there are different versions of ChipWhisperer, see https://rtfm.newae.com/Capture/

Hardware set up

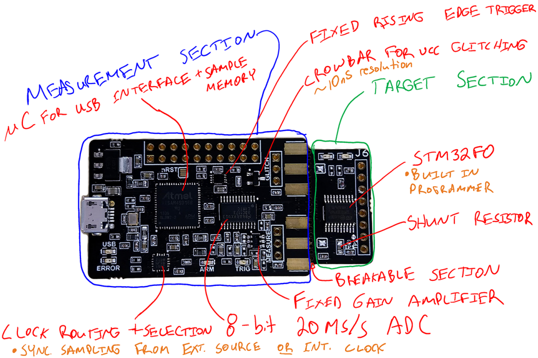


Figure: from https://rtfm.newae.com/Capture/ChipWhisperer-Nano/

Use a micro USB cable to connect the ChipWhisperer to a computer

Done? Then,

follow software guide at https://chipwhisperer.readthedocs.io/en/latest/

Software installation

Chipwhisperer has an open-source Python library for

controlling the capture hardware and
communicating with the target.

There are two modes (basic & advanced) for chipwhisperer installation.

There are two ways for basic installation

- Windows Installer
- Virtual Machine (VirtualBox)

There are different choices for advanced installation and prerequisites

- GNU/Linux (preferred)
- Windows
- Mac OS X
- Virtual machine (VMWare)

Detailed documentation can be found at:
https://chipwhisperer.readthedocs.io/en/latest/index.html.

Prerequisites

Python

- Any version $\geq$ 3.6 should work, e.g., 3.9.x.
- Python 2.x does **NOT** work with chipwhisperer codebase

Install Jupyter notebook

Select the operating system

- GNU/Linux
- Windows
- Mac

You can also use VirtualBox and the image from the internet or the one provided by me.

With Jupyter Notebooks

- If you’ve installed via the Windows Installer: run the ChipWhisperer shortcut.
- If you’ve installed natively: make sure you are using a bash like terminal.
- If you installed on Windows/Mac you may have to install a bash like terminal.
- If you installed Git to install chipwhisperer you already have git-bash available to you.
- Here are a few bash-like terminals available on other windows:
  (Recommended) Git-Bash (select to install git bash during the installation of Git.)
  MinGW, Cygwin, etc.
- Start the bash terminal. Make sure you have access to chipwhisperer in the terminal using:
  - python3 -c “import chipwhisperer as cw”
- Navigate to the chipwhisperer directory and Start the Jupyter Server there by: - jupyter notebook
- The Jupyter Notebook Server interface should be automatically opened in your browser.
- You should see the chipwhisperer folder in your browser, the tutorials are in the jupyter folder
- Look to: https://chipwhisperer.readthedocs.io/en/latest/starting.html#starting
- For jupyter notebook tutorial, look to
  https://jupyter-notebook.readthedocs.io/en/stable/

Without Jupyter Notebooks

- The chipwhisperer software can also be used without Jupyter Notebooks.
- Use chipwhisperer as you normally would any python package:

  - import chipwhisperer as cw

  - help(cw)
- For information about what functions, and classes are available to you, look to :
  https://chipwhisperer.readthedocs.io/en/latest/api.html#api
- For the tutorial look to:
  https://chipwhisperer.readthedocs.io/en/latest/tutorials.html/#tutorials

Connecting to ChipWhisperer

We can connect to the ChipWhisperer with:

- import chipwhisperer as cw & - scope = cw.scope()

By default, ChipWhisperer will try to autodetect the type of device your're running

Scope type can be specified manually (see API documentation)

Some sane default settings can be set using: - scope.default_setup()

You can print the scope default by: - print(scope)

The default values are:

- Sets the scope gain to ∼25dB

- Sets the scope to capture 5000 samples

- Sets the scope offset to 0 (aka it will begin capturing as soon as it is triggered)

- Sets the scope trigger to rising edge

- Outputs a 7.38MHz clock to the target on HS2

- Clocks the scope ADC at 4×7.38MHz. Note that this is synchronous to the target clock on HS2

- Assigns GPIO1 as serial RX

- Assigns GPIO2 as serial TX

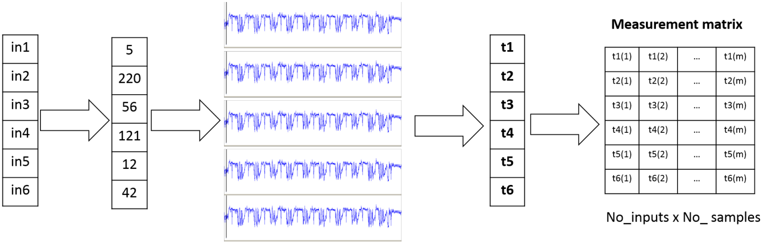
ChipWhisperer is now setup and ready to attack a target.

Disconnect the scope/target before connecting again: **scope.dis()** and **target.dis()**

Practical Side-channel Analysis

Practical Side-channel Analysis:
Acquisition of traces (and signal processing)

Side-Channel Example: CPA on AES (recall)



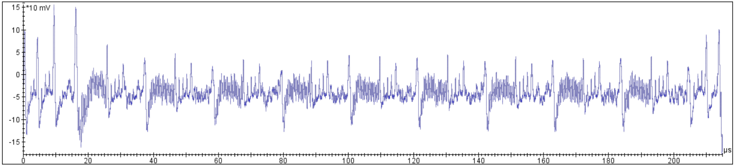
Measurement matrix

t1(1)	t1(2)	...	t1(m)
t2(1)	t2(2)	...	t2(m)
t3(1)	t3(2)	...	t3(m)
t4(1)	t4(2)	...	t4(m)
t5(1)	t5(2)	...	t5(m)
t6(1)	t6(2)	...	t6(m)

No_inputs x No_ samples

- Make predictions based on the key guesses, compute correlations, and determine the maximum one: $\rho_k(L, HW(V_g)) = \frac{cov[L,HW(V_g)]}{\sqrt{Var[L] \cdot Var[HW(V_g)]}}$
- We see a tace but where does it come from?
- “Magically”? From Łukasz :-) ?
- What do we need to do before we do CPA?

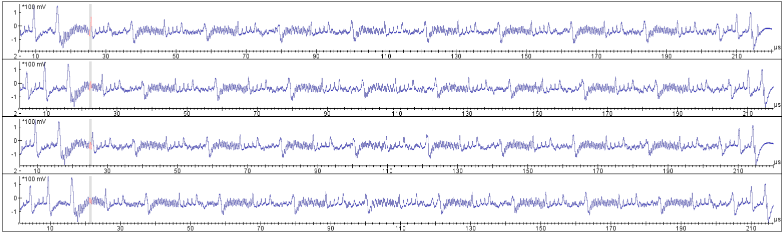
Side-channel Traces



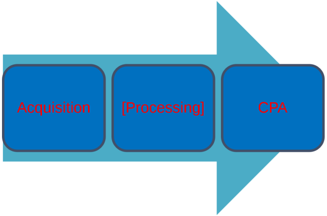
- What it is? AES
- What are typical side-channels? power, EM, time, sound, temeperature, light...
- smartcards vs. embedded devices
- What to do first? Build the setup :-)

Problems with Side-channel Traces

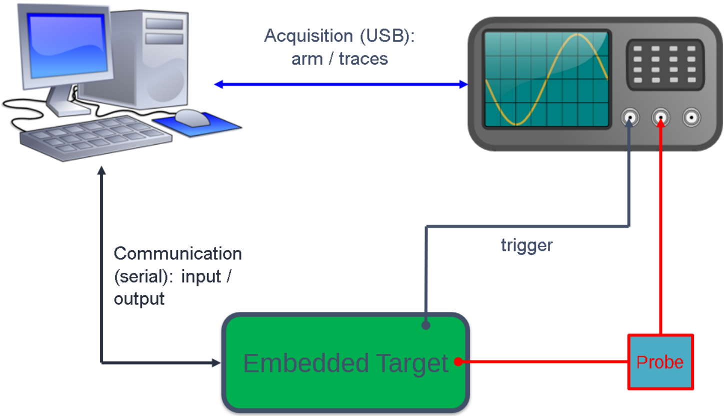
- Misalignment:



- Noise
- How to minimize these problems? (1) Build a good “enough” setup. (2) Do processing of the traces (e.g., alignment, compression, etc.).



Typical Setup Components

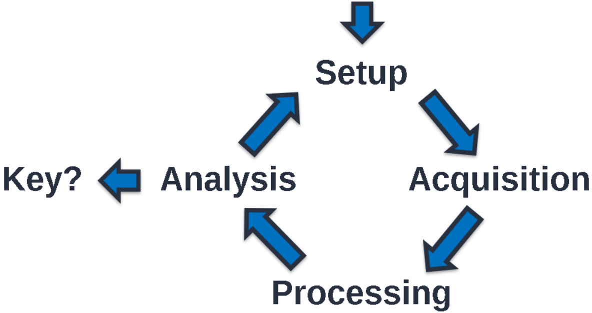


Typical Attack

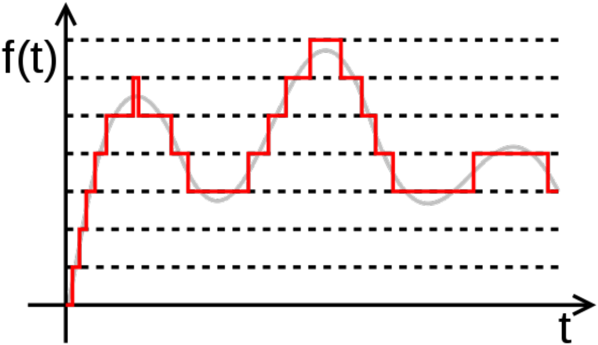
1. Build Setup & CharacterizationCharacterization
2. Collect data for the attack:
   - Using passives probes, for example: current or EM.
3. Optionally: perform signal processing, for example, alignment or compression.
4. Perform the attack on the collected data, for example:
   - Correlation Power Analysis (CPA) or Simple Power Analysis (SPA).

Typical challenges: noise, jitter, misalignment...

What if it fails?



Typical easy mistake (quantization)[1]



[1] wikimedia

Building Setup - discussion about equipment

Let's discuss which different setups we could build here...

Different Types Software

- ChipWhisperer (NewAE Technology):
  - `https://www.newae.com/chipwhisperer`
- AISyLab (TU Delft):
  - `https://github.com/AISyLab`
- Jlsca (Riscure):
  - `https://github.com/Riscure/Jlsca`
- Side-Channel Marvels:
  - `https://github.com/SideChannelMarvels`
- Pysca:
  - `https://github.com/ikizhvatov/pysca`
- ...
- Commercial:
  - Riscure's Inspector, Secure IC, Rambus ...

Messages to Remember

- Building a suitable (“good enough”) setup is crucial for a successful SCA attack
- Setups differ significantly for different targets
- Different side-channels require different setups
- Characterization is a crucial step
- Signal Processing is usually important too
- Most of that step are not necessary for ChipWhispere :-)
- All that complexity, makes SCA attacks are challenging and fun :-)

Capturing AES traces with ChipWhisperer

- Even if your setup works please work in pairs.
- If you installed everything then let's run `Lab 3_3 - DPA on Firmware Implementation of AES (MAIN)` in `http://localhost:8888/notebooks/jupyter/courses/sca101/`
- Let's experiment with acquisition and different setting values.
- Compute correlation between input and output values and the traces.
- If you have time try to implement CPA!

Conclusions

- The main goal of this seminar is to go more into detail of Side-Channel Analysis by experimenting with ChipWhisperer.
- Investigate trace acquisition and correlation.
- In general, provide more background on Side-Channel Attacks.
- Thank you for the attendance :-)

Questions

?