Side-channels and Fault Attacks

Roman Korkikian, Nicolas Oberli

February 23, 2023

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

About us

About us: Roman Korkikian

- Independent contractor in hardware attacks (working with Sony and other companies)
- >10 years of industrial and academic experience (Kudelski Security, STMicroelectronics, Altis Semiconductor)
- PhD (ENS, Paris)
- roman.korkikian@gmail.com

About us: Nicolas Oberli

- Security engineer by day, hardware hacker by night
- Open source tools (Hydrabus), Speaker at conferences, Blackalps
- HEIG alumni
- nicolas.oberli@gmail.com

Teaching language

- Either we do everything in French (except slides)
- Alternatively, Roman will give classes in English

Teaching language

- Either we do everything in French (except slides)
- Alternatively, Roman will give classes in English
- Shall we vote (anonymously)?

Examples of side-channel and fault attacks

Example: Side-channel attack (power) on Jade

- Jade crypto-wallet is based on ESP32 (where various security features are activated: Secure Boot, Flash encryption, etc.)
- Applying statistics on current consumption measurements during Flash decryption Ledger guys could get flash encryption key
- Side-channel analysis applies statistics (math) on physically measured events to get secret information

- Karim Abdellatif, Olivier Hériveaux, Adrian Thillard. Unlimited Results: Breaking Firmware Encryption of ESP32-V3. Cryptology ePrint Archive. 2023

Figure 31: Jade wallet before opening the package

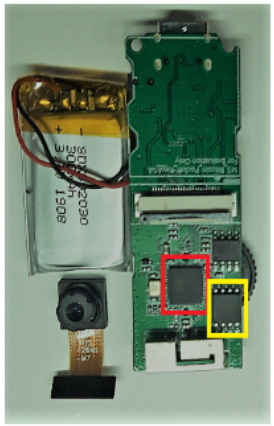


Figure 32: Jade wallet after opening the package

```
7010h: 27 E6 F3 D4 43 F1 95 38 79 CF E3 4C D9 4B 8B 6F  'æóOCñ•8yÏãLUK‹o
7020h: 38 7C 28 AC 1D 9E CE D5 EB F2 9A 3F 95 0F CE E8  8|(¬.žÎÕëò š?•.Îè
7030h: 6C 7E 30 FF B9 CA 09 1C BA EE D6 8E EA BD 46 8B  l~0ÿ'Ê..°îÖŽê½F‹
7040h: 20 0C 87 BC C1 E2 F7 66 8F F3 82 E9 BF BF 89 61   .‡¼Áâ÷f.ó.é¿¿‰a
7050h: DE 56 0E C4 D1 16 E2 71 9A AA BE D7 29 A5 E5 2B  ÞV.ÄÑ.âqš ª¾×)¥å+
7060h: C8 C0 D3 AF 50 4E E1 21 6E 7E 21 F2 EB F5 DB 91  ÈÀÓ¯PNá!n~!òëõÛ'
7070h: 86 AE A0 D0 86 D4 D3 C4 4C 3A B7 D1 70 5E 69 82  †® ĐTÔÓÄL:·Ñp^i‚
7080h: 3A DB 1B A7 64 7D 81 EA 15 12 E3 C0 B2 2F 38 B0  :Û.§d}.ê..ãÀ²/8°
7090h: 17 7E 61 FB 99 70 99 4E 6D B2 61 32 8E EF CF 93  .~aû™p™Nm²a2ŽïÏ"
70A0h: 50 43 44 FE 2A B4 27 65 F5 EA F1 C9 A2 8F 05 3D  PCDþ*´'eõêñÉ¢..=
70B0h: E4 C6 8E A1 B4 81 EB 7F DB EA C8 45 B3 1D C8 A8  äÆŽ¡´.ë.ÛêÈE³.È¨
70C0h: 4E 3D E7 BE 61 9E 30 1E 13 CD 62 2A F2 94 85 2F  N=ç¾až0..Íb*ò"…/
```

(a) Encrypted firmware

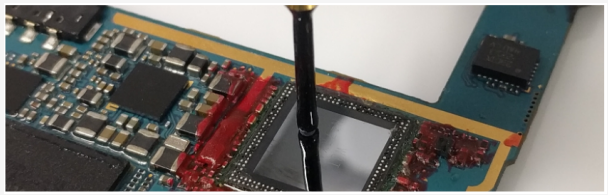
```
7010h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
7020h: AA 50 01 00 00 D0 00 00 00 20 00 00 6F 74 61 64  ªP...Đ... ..otad
7030h: 61 74 61 00 00 00 00 00 00 00 00 00 01 00 00 00  ata.............
7040h: AA 50 01 01 00 F0 00 00 00 10 00 00 70 68 79 5F  ªP...ð......phy_
7050h: 69 6E 69 74 00 00 00 00 00 00 00 00 01 00 00 00  init............
7060h: AA 50 00 10 00 00 01 00 00 70 17 00 6F 74 61 5F  ªP.......p..ota_
7070h: 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  0...............
7080h: AA 50 00 11 00 00 19 00 00 70 17 00 6F 74 61 5F  ªP.......p..ota_
7090h: 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  1...............
70A0h: AA 50 01 04 00 70 30 00 00 10 00 00 6E 76 73 5F  ªP...p0.....nvs_
70B0h: 6B 65 79 00 00 00 00 00 00 00 00 00 01 00 00 00  key.............
70C0h: EB EB FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ëëÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

(b) Decrypted firmware

Figure 33: Jade's firmware before and after decryption

Example: Side-channel attack (electromagnetic) on a mobile chipset

- A non-disclosed mobile processor encrypts Flash (part of the Secure Boot process)
- Electromagnetic measurements during AES decryption were used to get the Flash encryption key
- Philippe Maurine (LIRMM) is one of the academic key figures in side-channel and fault attacks



- Aurélien Vasselle, Philippe Maurine, Maxime Cozzi. Breaking Mobile Firmware Encryption through Near-Field Side-Channel Analysis. ASHES 2019 - 3rd Attacks and Solutions in Hardware Security Workshop, Nov 2019, London, United Kingdom. pp.23-32

Example: Optical Side-Channel Attacks

- Optical Beam Induced Current (OBIC), Laser Voltage Probing (LVP), Electro-Optical Frequency Mapping (EOFM) and Photon Emission Analysis (PEM)
- Those methods (if work) provide spacial and temporal side-channel information and they shall be more powerful than power or electromagnetic side-channel analysis

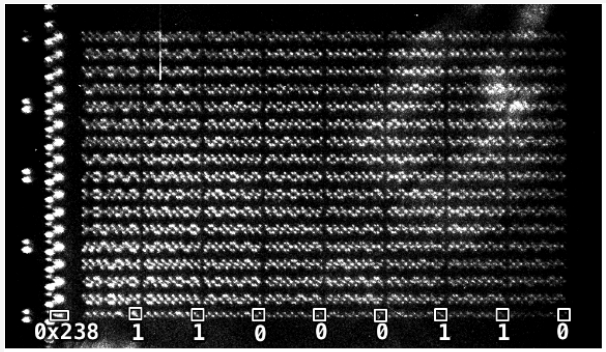
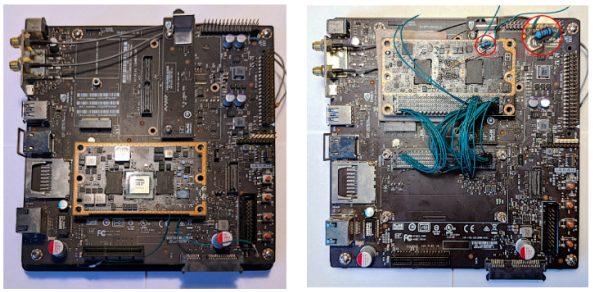


Fig. 3. Optical emission image of the S-Box in memory. The 256 bytes of the S-Box are located from `0x23F` to `0x33E`, see Table 3 in the appendix. The address `0x23F` is the seventh byte of the `0x238` SRAM line, i.e. the S-Box has an offset of 7 bytes. The emissions of the row drivers are clearly visible to the left of the memory bank. The image allows direct readout of the bit-values of the stored data. The first byte for example, as shown in the overlay, corresponds to $01100011_2 = 63_{16}$, the first value of the AES S-Box.

- Alexander Schlösser, Dmitry Nedospasov, Juliane Krämer, Susanna Orlic, Jean-Pierre Seifert. Simple photonic emission analysis of AES: photonic side-channel analysis for the rest of us. Cryptographic Hardware and Embedded Systems–CHES 2012: 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings 14. pp. 41-57

Example: Fault attack (glitch) on Nvidia Tegra

- The initial booting stage is embedded into the ROM code (a memory which is hardcoded, pretty often companies want to keep ROM code secret)
- ROM code usually has the highest privileges in the system
- By injecting a power glitch, researchers could run their code on Nvidia Tegra X2 with the highest privileges and dump the entire ROM (and use this code for further attacks)



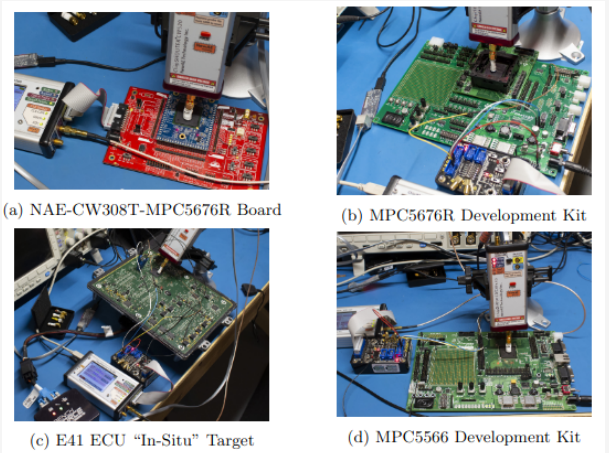
(a) Module without adapter (b) Module with adapter

Figure 4: TX2 module mounted on the developer kit carrier board.

- Otto Bittner, Thilo Krachenfels, Andreas Galauner, Jean-Pierre Seifert. The forgotten threat of voltage glitching: a case study on Nvidia Tegra X2 SoCs. 2021 Workshop on Fault Detection and Tolerance in Cryptography (FDTC). pp. 86-97

Example: Fault attack (electromagnetic) on an automotive ECU

- Modern cars have plenty of chips inside (including secure chips)
- The security of chips is also linked to safety (so you don't want a hacker to rewrite your engine control unit)
- This work shows that access to an ECU can be gained with Electromagnetic Fault Injection.



(a) NAE-CW308T-MPC5676R Board (b) MPC5676R Development Kit

(c) E41 ECU "In-Situ" Target (d) MPC5566 Development Kit

Fig. 3: Hardware targets used to validate the EMFI effectiveness.

- Colin O'Flynn. BAM BAM!! On Reliability of EMFI for in-situ Automotive ECU Attacks. Cryptology ePrint Archive 2020

Example: Fault attack (laser) on ATECC608A

- Microchip has a secure chip ATECC608A to manage secure keys and run various crypto operations (encryption, authentication)
- Laser fault injection could be used to get a special secret Seed value
- Olivier Hériveaux. Defeating a Secure Element with Multiple Laser Fault Injections. Symposium sur la sécurité des technologies de l'information et des communications-SSTIC 2021

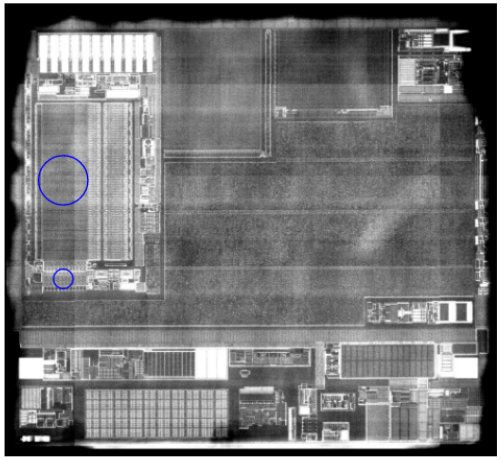


Fig. 5. Top circle area in EEPROM: sets bits to zero. Bottom circle area in EEPROM: sets bits to one.

What hardware attacks can be used for?

- Extract cryptographic keys

What hardware attacks can be used for?

- Extract cryptographic keys
- Bypass security measures (PIN-code, signature, etc.)

What hardware attacks can be used for?

- Extract cryptographic keys
- Bypass security measures (PIN-code, signature, etc.)
- Create one-time software vulnerability (bypass buffer length check to be used for buffer overflow)

What hardware attacks can be used for?

- Extract cryptographic keys
- Bypass security measures (PIN-code, signature, etc.)
- Create one-time software vulnerability (bypass buffer length check to be used for buffer overflow)
- Reverse-engineer software or algorithm

The cost of hardware attacks

- Hardware attacks require special equipment (digital oscilloscopes, lasers, etc.)

The cost of hardware attacks

- Hardware attacks require special equipment (digital oscilloscopes, lasers, etc.)
- Hardware attacks shall be applied if the attack cost is smaller than the cost of the asset.

The cost of hardware attacks

- Hardware attacks require special equipment (digital oscilloscopes, lasers, etc.)
- Hardware attacks shall be applied if the attack cost is smaller than the cost of the asset.
- My personal estimation:
  - Side-channel attacks are applied only when fault attacks can not work (Flash encryption is the main example)
  - If you can apply fault attacks - use them
  - On the field side-channel attacks - 20%; fault attacks - 80%

About the course

About the course

- We will introduce you to power side-channel attacks on RSA and AES and glitch fault attacks on AES and common CPU flow

About the course

- We will introduce you to power side-channel attacks on RSA and AES and glitch fault attacks on AES and common CPU flow
- A series of lectures, exercises and practical lab hands-on experience

About the course

- We will introduce you to power side-channel attacks on RSA and AES and glitch fault attacks on AES and common CPU flow
- A series of lectures, exercises and practical lab hands-on experience
- Four periods during 14 weeks: February 23rd, 2023 - June 29th, 2023
  - 7 weeks for side-channel attacks
  - 7 weeks for fault attacks

About the course

- We will introduce you to power side-channel attacks on RSA and AES and glitch fault attacks on AES and common CPU flow
- A series of lectures, exercises and practical lab hands-on experience
- Four periods during 14 weeks: February 23rd, 2023 - June 29th, 2023
  - 7 weeks for side-channel attacks
  - 7 weeks for fault attacks
- Final grade depends on labs and final evaluation performance

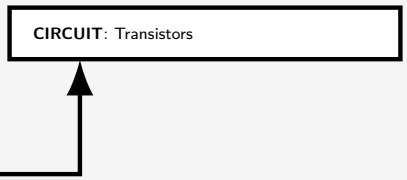
About the course

- We will introduce you to power side-channel attacks on RSA and AES and glitch fault attacks on AES and common CPU flow
- A series of lectures, exercises and practical lab hands-on experience
- Four periods during 14 weeks: February 23rd, 2023 - June 29th, 2023
  - 7 weeks for side-channel attacks
  - 7 weeks for fault attacks
- Final grade depends on labs and final evaluation performance
- Since hardware attacks are mainly applied against cryptography, you will see some math and crypto

Instead of history

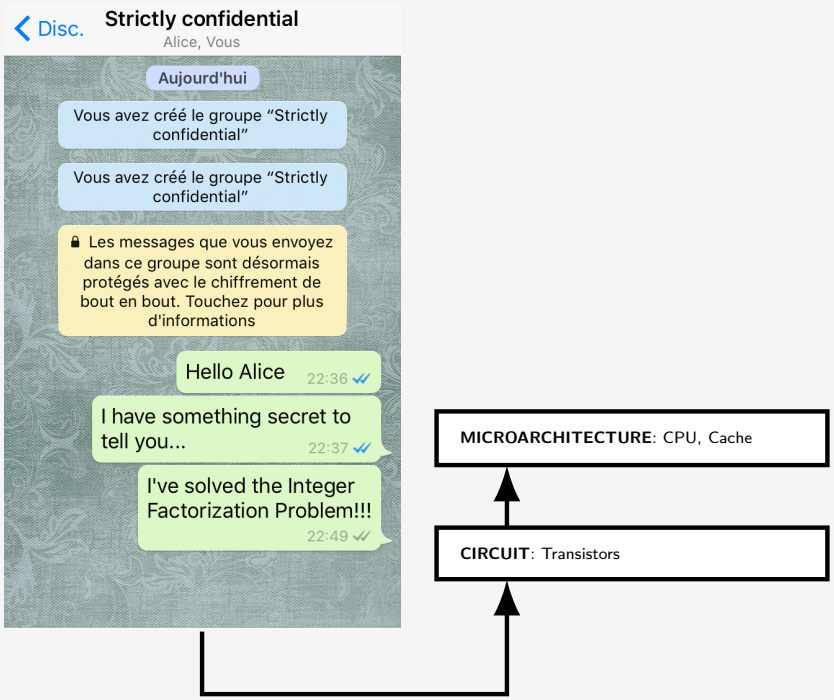
Introduction

What happens when you press 



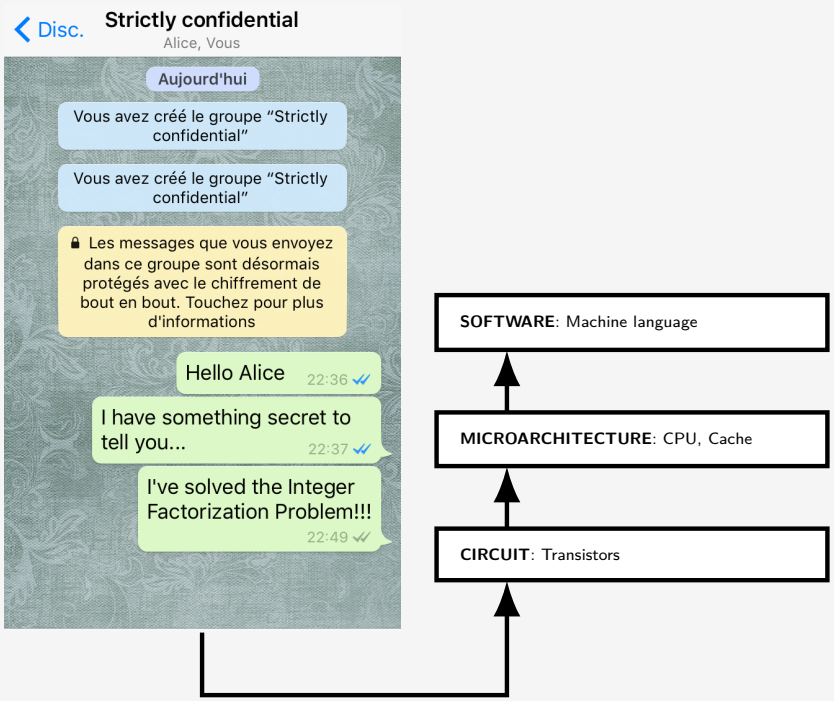
Introduction

What happens when you press 



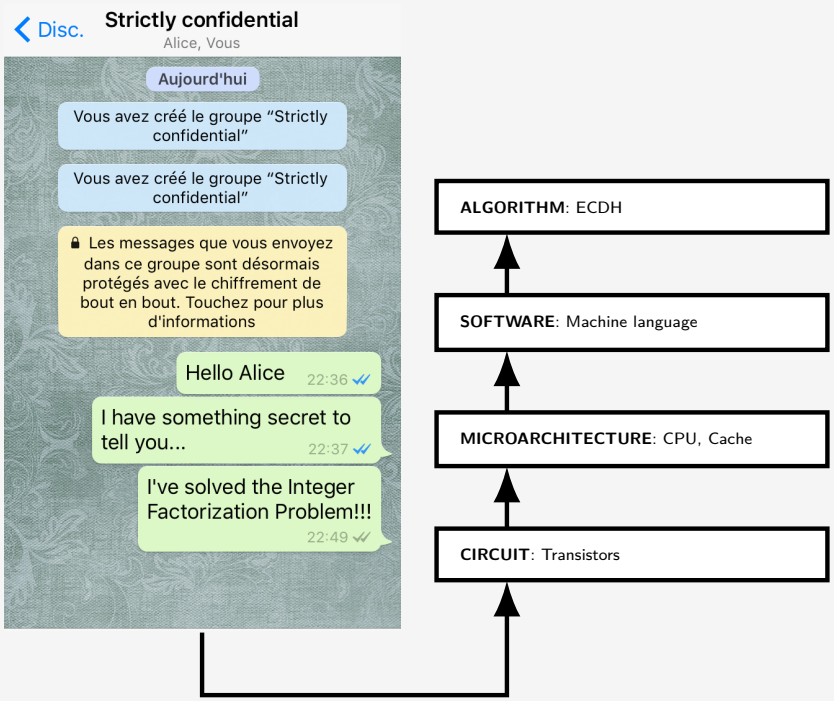
Introduction

What happens when you press 



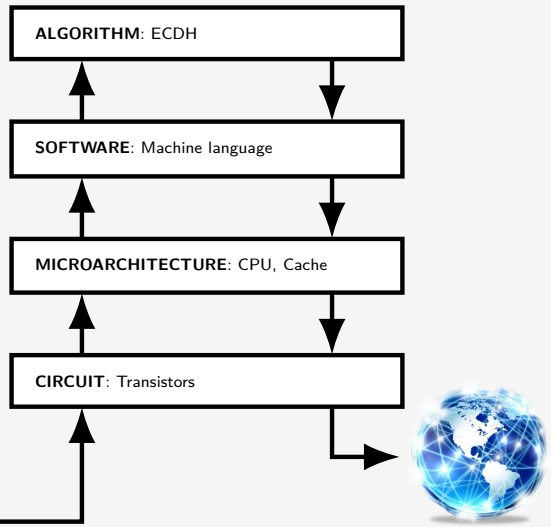
Introduction

What happens when you press 



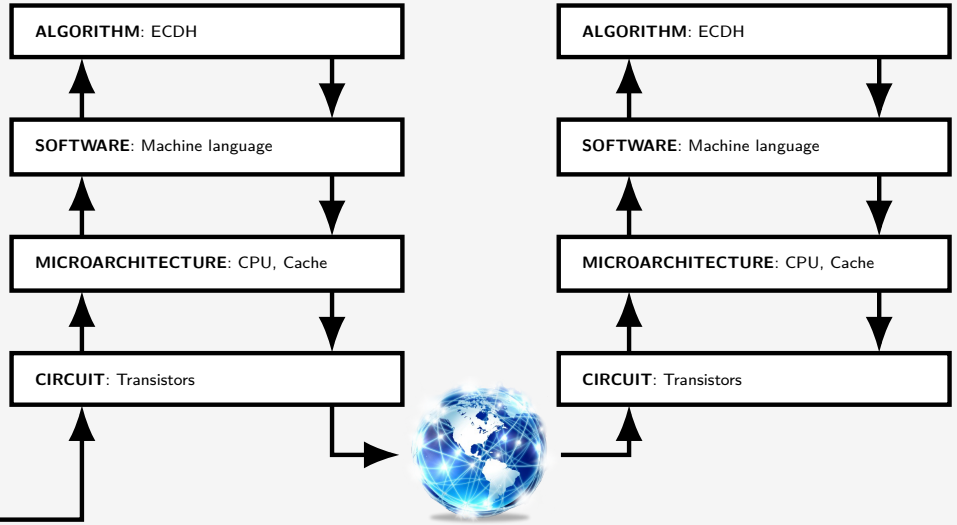
Introduction

What happens when you press 



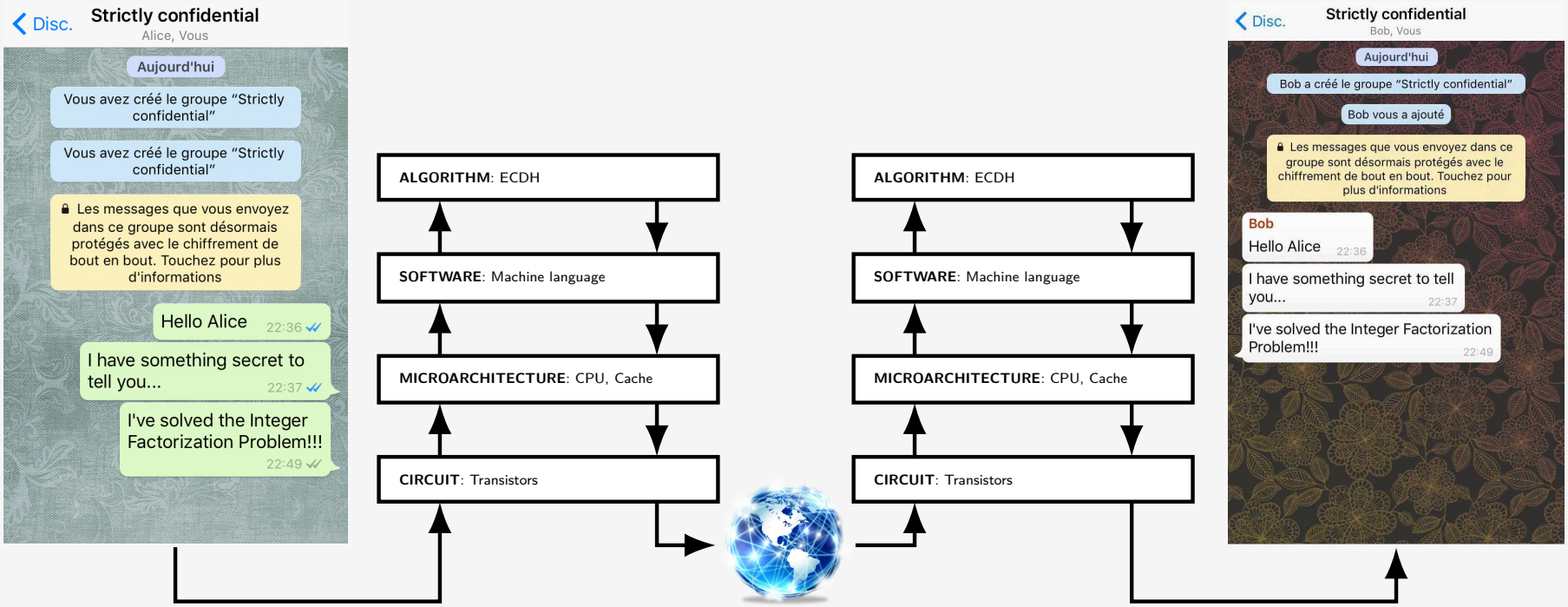
Introduction

What happens when you press 



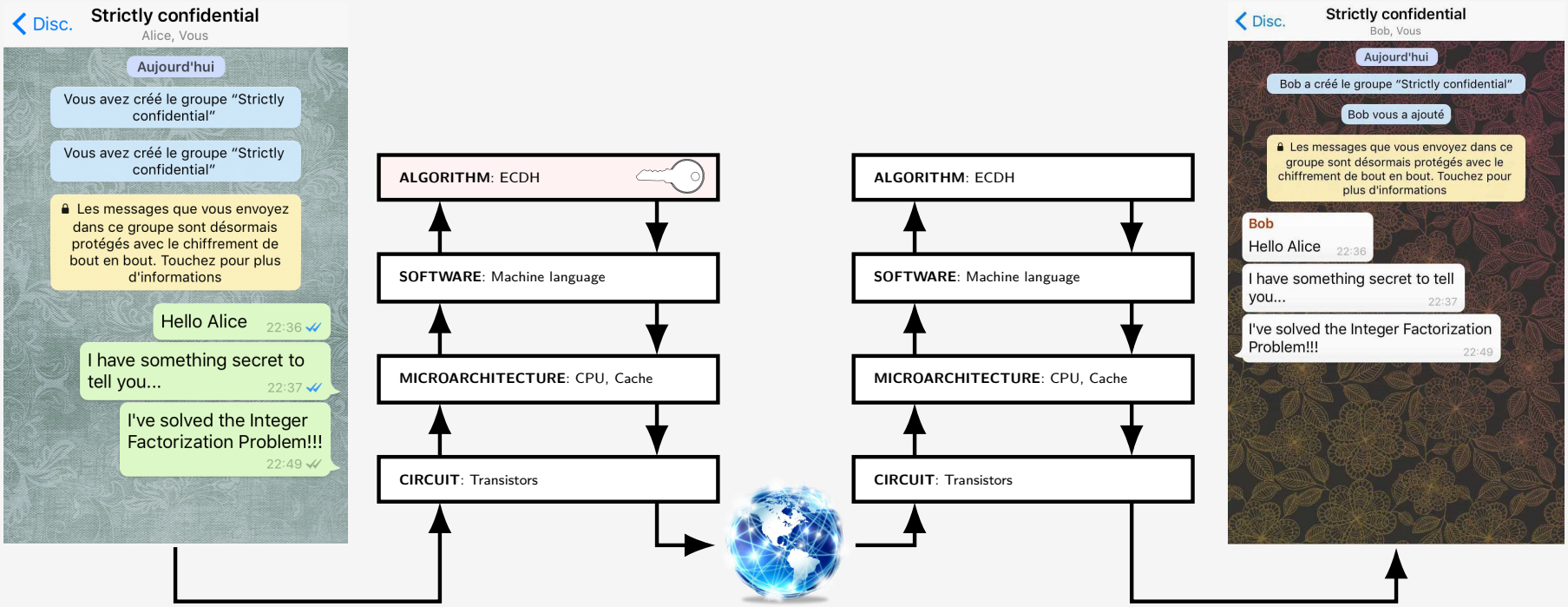
Introduction

What happens when you press 



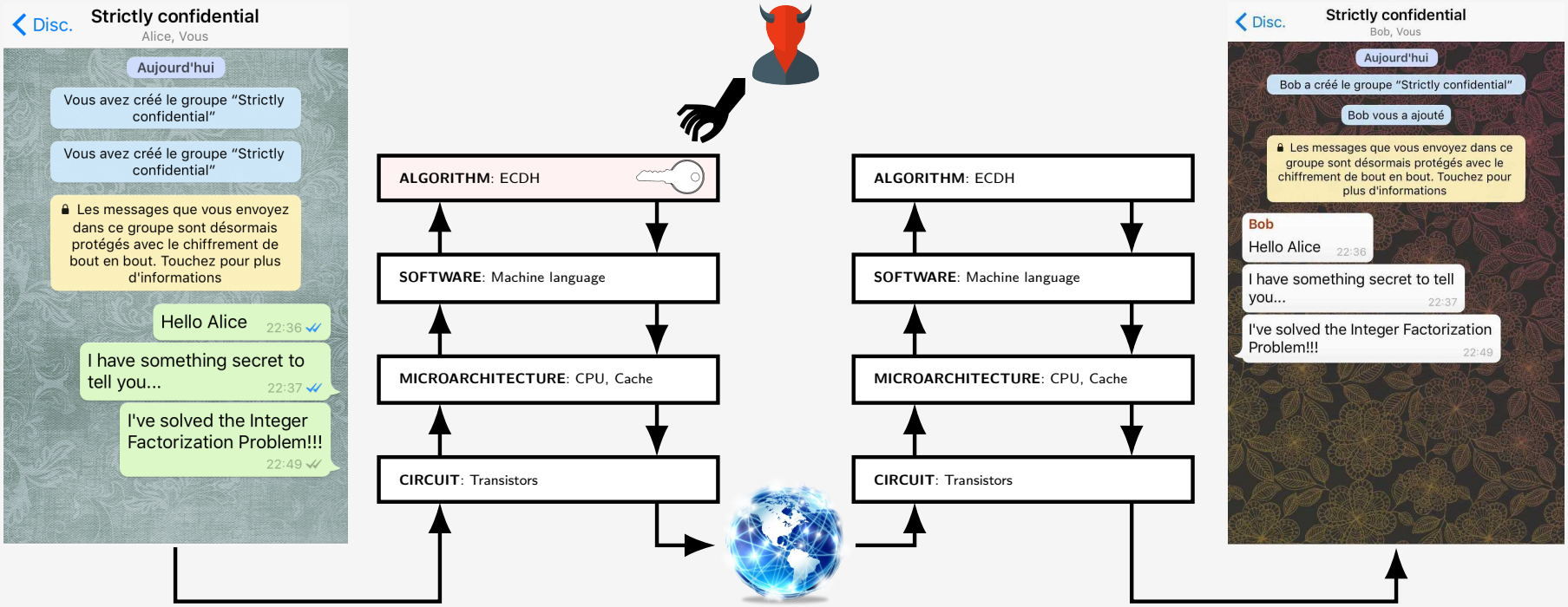
Introduction

To secure communication, the algorithm applies a secret key:



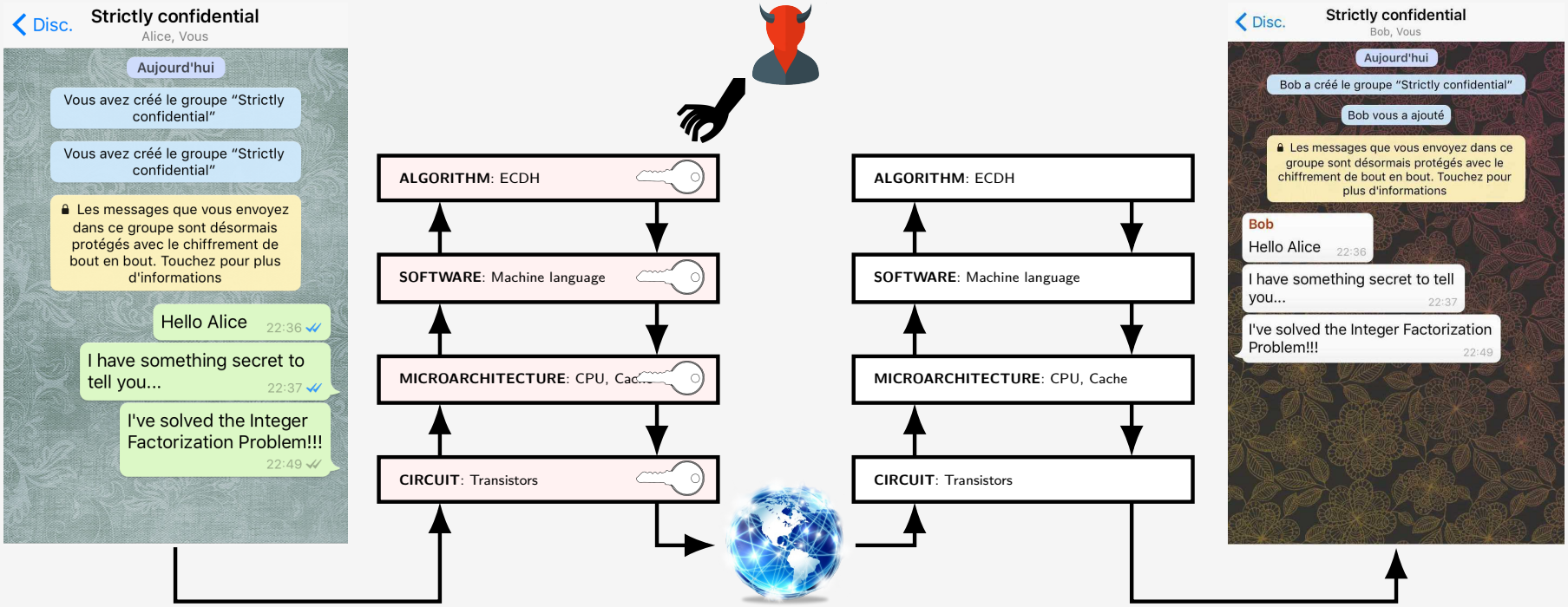
Introduction

To secure communication, the algorithm applies a secret key:



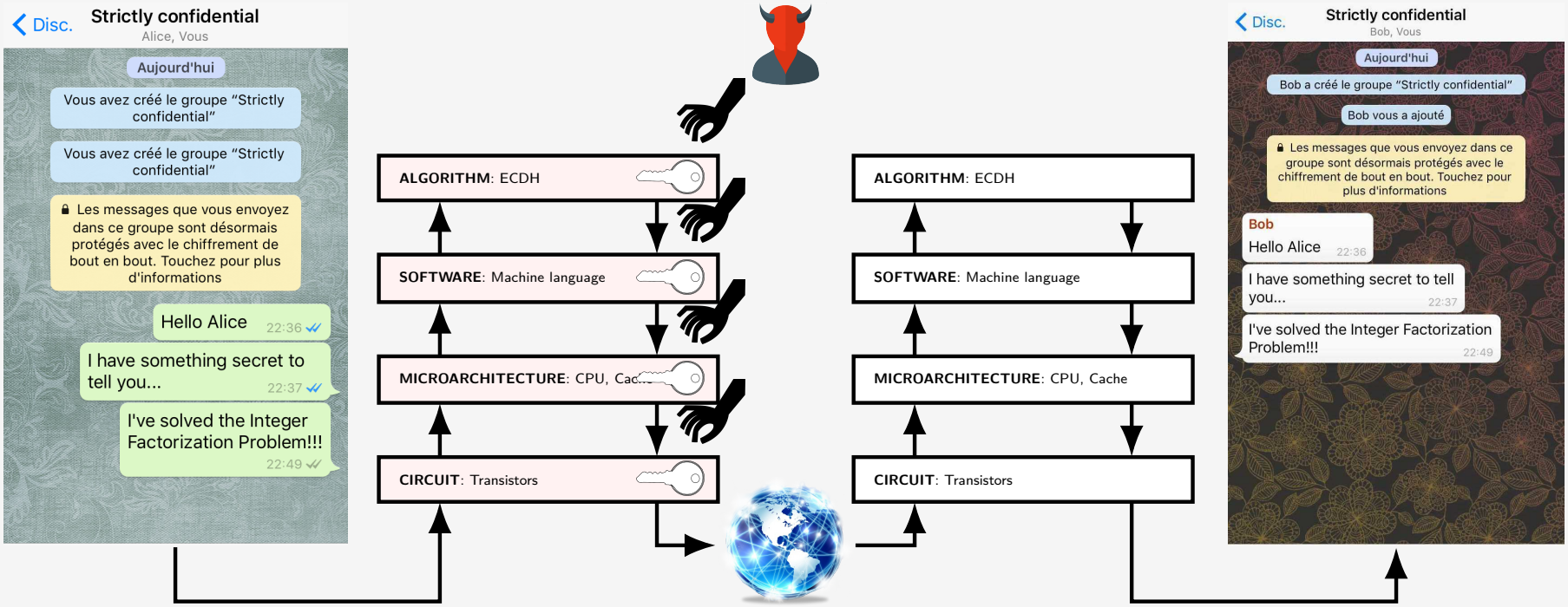
Introduction

To secure communication, the algorithm applies a secret key:



Introduction

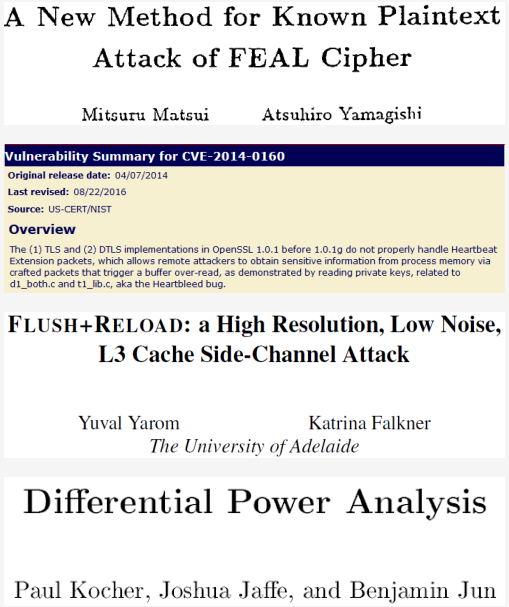
To secure communication, the algorithm applies a secret key:



Introduction

- Each abstraction layer faces threats capable of leaking the secret key:

Algorithm	*e.g.* Linear cryptanalysis
Software	*e.g.* Heartbleed
Microarchitecture	*e.g.* Flush+Reload
Circuit	*e.g.* Side-Channel Attacks Fault Attacks



A New Method for Known Plaintext Attack of FEAL Cipher

Mitsuru Matsui Atsuhiro Yamagishi

Vulnerability Summary for CVE-2014-0160

Original release date: 04/07/2014

Last revised: 08/22/2016

Source: US-CERT/NIST

Overview

The (1) TLS and (2) DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a buffer over-read, as demonstrated by reading private keys, related to d1_both.c and t1_lib.c, aka the Heartbleed bug.

FLUSH+RELOAD: a High Resolution, Low Noise, L3 Cache Side-Channel Attack

Yuval Yarom Katrina Falkner

The University of Adelaide

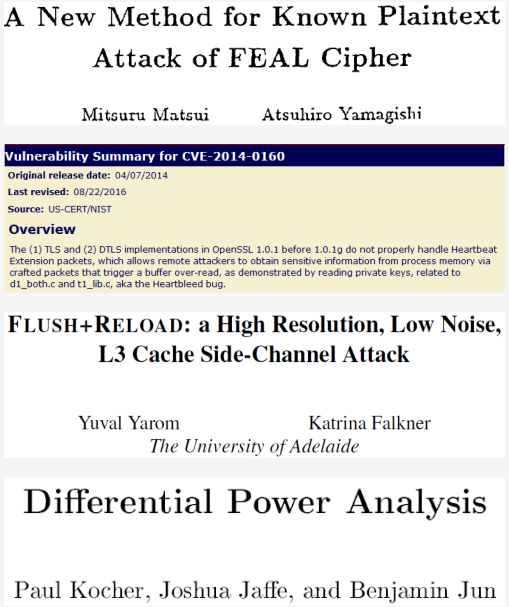
Differential Power Analysis

Paul Kocher, Joshua Jaffe, and Benjamin Jun

Introduction

- Each abstraction layer faces threats capable of leaking the secret key:

Algorithm	*e.g.* Linear cryptanalysis
Software	*e.g.* Heartbleed
Microarchitecture	*e.g.* Flush+Reload
Circuit	*e.g.* Side-Channel Attacks Fault Attacks


A New Method for Known Plaintext Attack of FEAL Cipher

Mitsuru Matsui Atsuhiro Yamagishi

Vulnerability Summary for CVE-2014-0160

Original release date: 04/07/2014

Last revised: 08/22/2016

Source: US-CERT/NIST

Overview

The (1) TLS and (2) DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a buffer over-read, as demonstrated by reading private keys, related to d1_both.c and t1_lib.c, aka the Heartbleed bug.

FLUSH+RELOAD: a High Resolution, Low Noise, L3 Cache Side-Channel Attack

Yuval Yarom Katrina Falkner

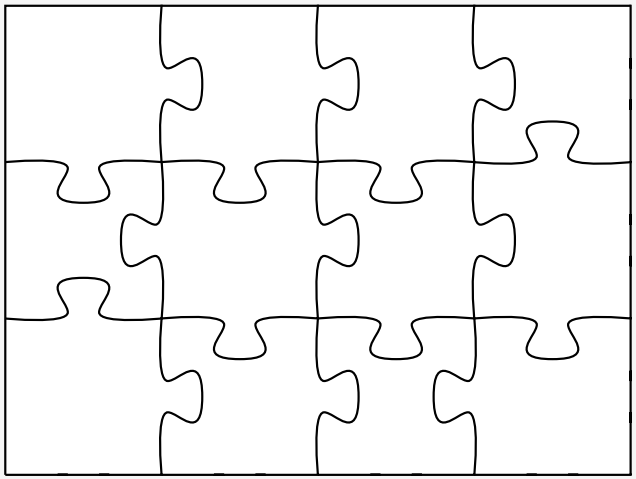
The University of Adelaide

Differential Power Analysis

Paul Kocher, Joshua Jaffe, and Benjamin Jun

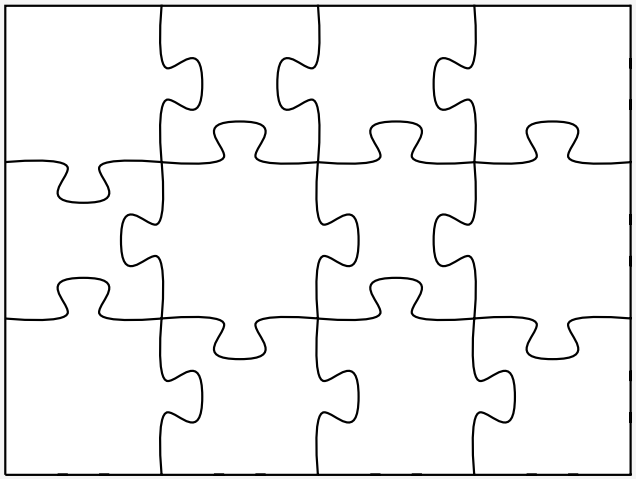
- The course focuses on circuit-level attacks, a.k.a., hardware attacks

Big puzzle



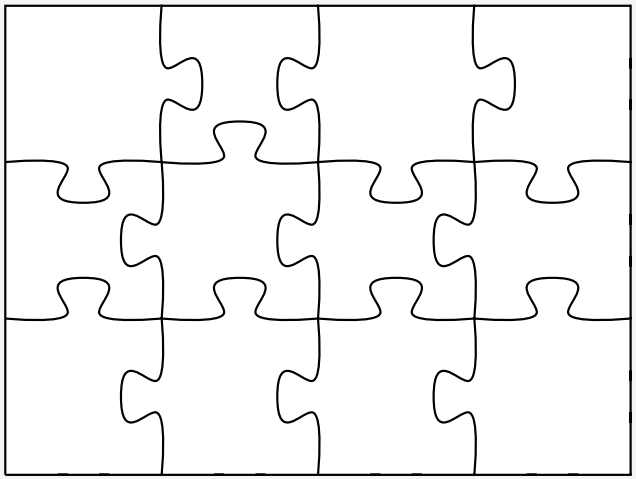
- Knowledge and skills required to master side-channel attacks can be considered a big puzzle

Big puzzle



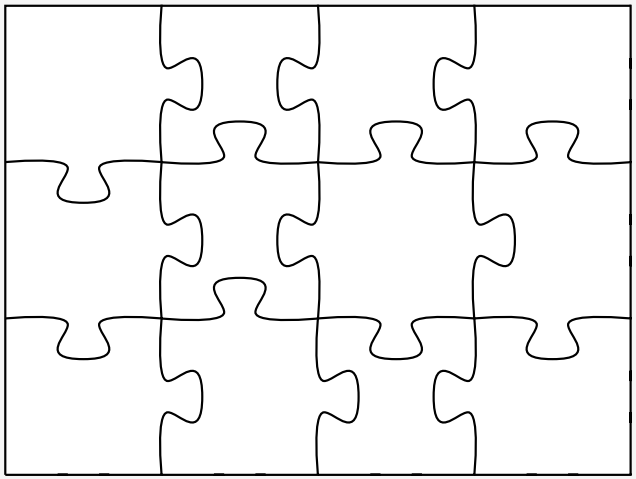
- Knowledge and skills required to master side-channel attacks can be considered a big puzzle
- Some pieces of information will be difficult to digest but DON'T WORRY. You will get there

Big puzzle



- Knowledge and skills required to master side-channel attacks can be considered a big puzzle
- Some pieces of information will be difficult to digest but DON'T WORRY. You will get there
- If you get the keys (by science of chance) you don't care about the theory (don't say that to other teachers)

Big puzzle



- Knowledge and skills required to master side-channel attacks can be considered a big puzzle
- Some pieces of information will be difficult to digest but DON'T WORRY. You will get there
- If you get the keys (by science of chance) you don't care about the theory (don't say that to other teachers)
- In my practice I had one attack which I still cann't understand how and why it worked

I beg you know side-channels

Side-channel information examples

We use side-channel information in various ways.

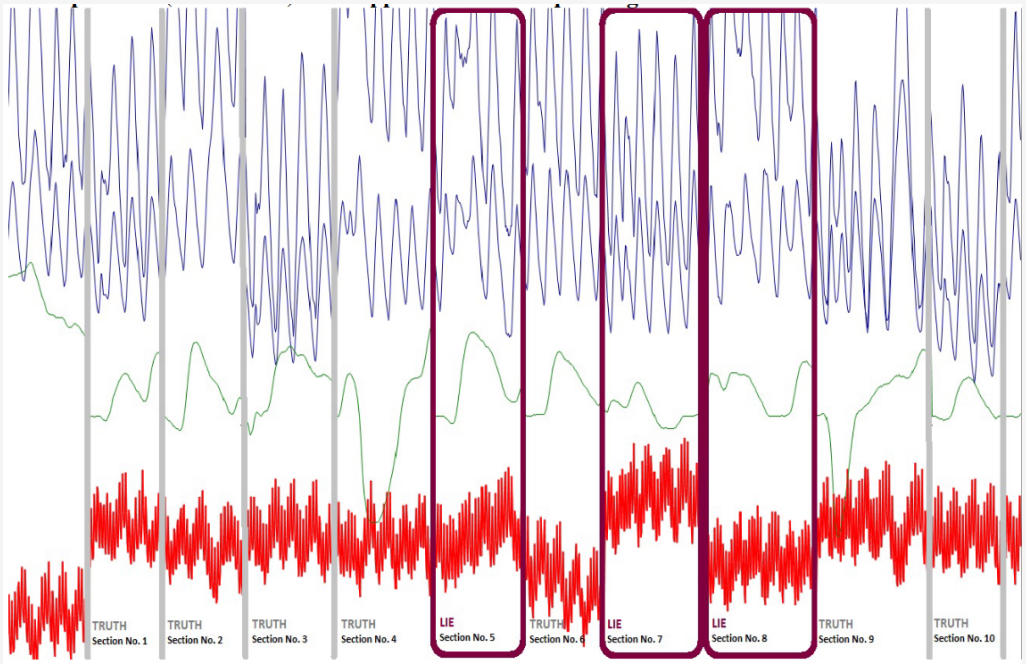


Figure: Do you know what those graphs are?

Side-channel information examples

We use side-channel information in various ways.

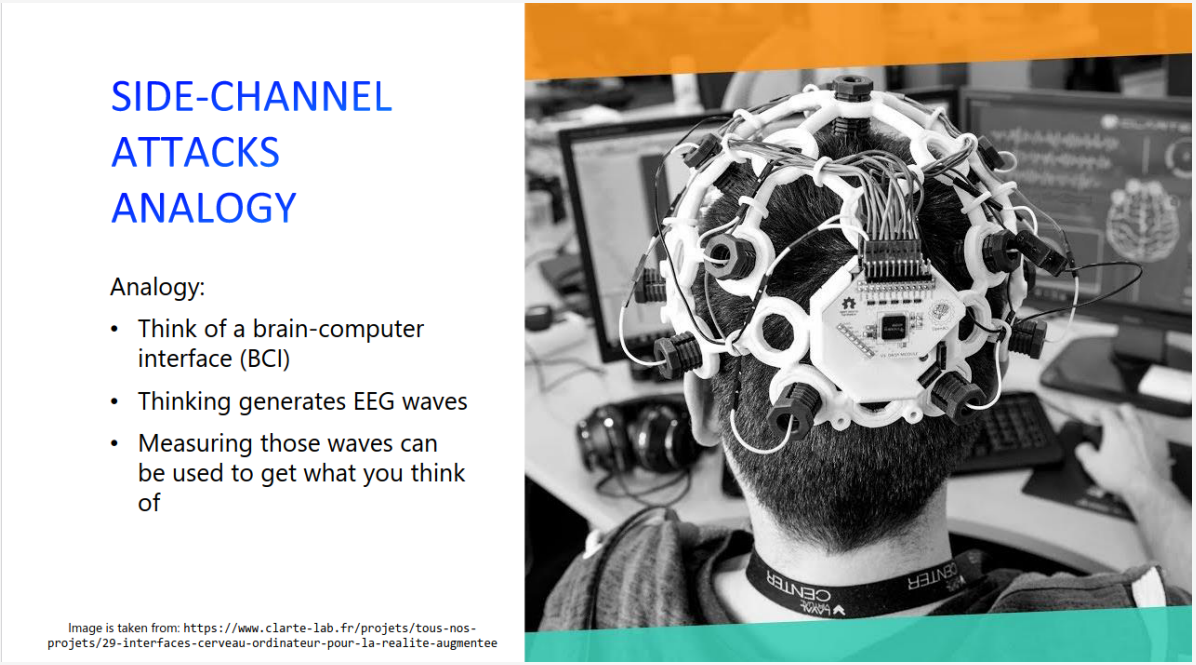


Figure: Can we get a password by looking at our brain activity? Yes... in a way.

Side-Channel Attacks Against the Human Brain: the PIN Code Case Study

Side-channel information examples

We use side-channel information in various ways.

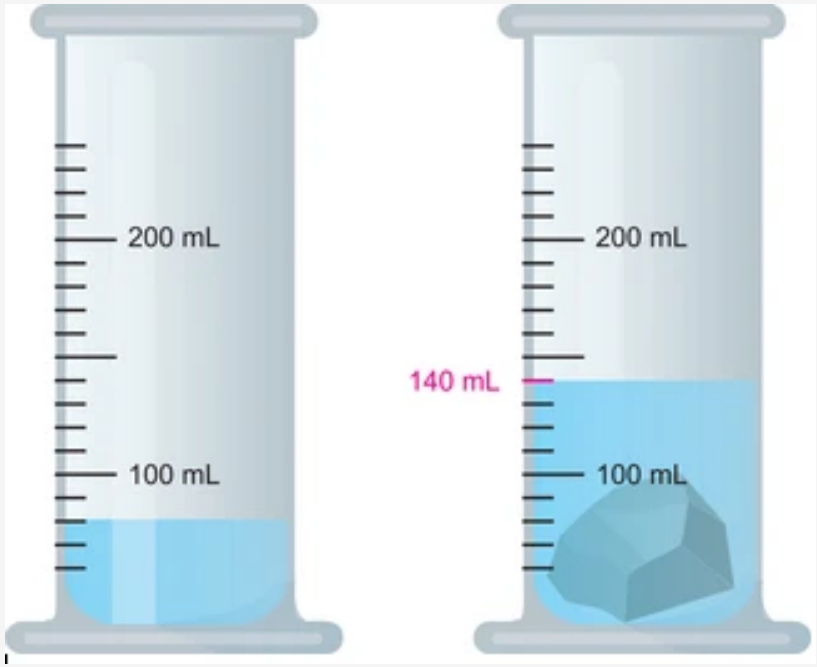


Figure: We don't need to cut objects into small pieces to measure the volume.

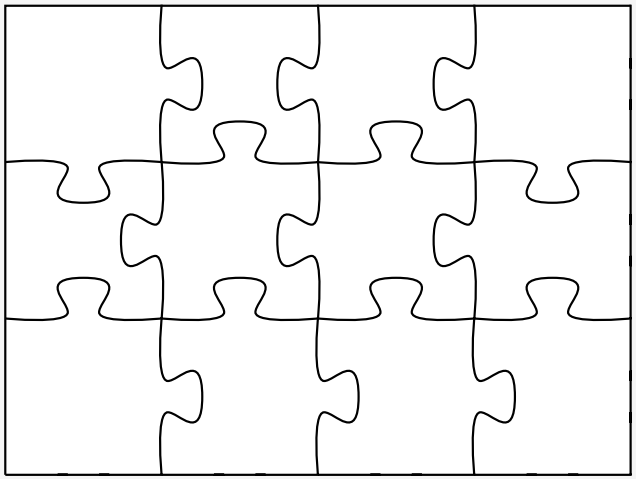
Side-channel information examples

Practical exercise...

Figure: Look carefully at the properties of a pin code entry to get the value.

Pin has 10 digits

Puzzle 1: Power consumption in modern semiconductor devices



A bit of phylosophy

Computations take time and consume energy.

A bit of phylosophy

Computations take time and consume energy.

Side-channel attacks can be applied when execution time, drained current (or other physical parameters) depend on the processed data.

A bit of technology

- A mobile phone is a good example (not for an attack but for visualisation)

A bit of technology

- A mobile phone is a good example (not for an attack but for visualisation)
- More intensive usage will drain your battery faster

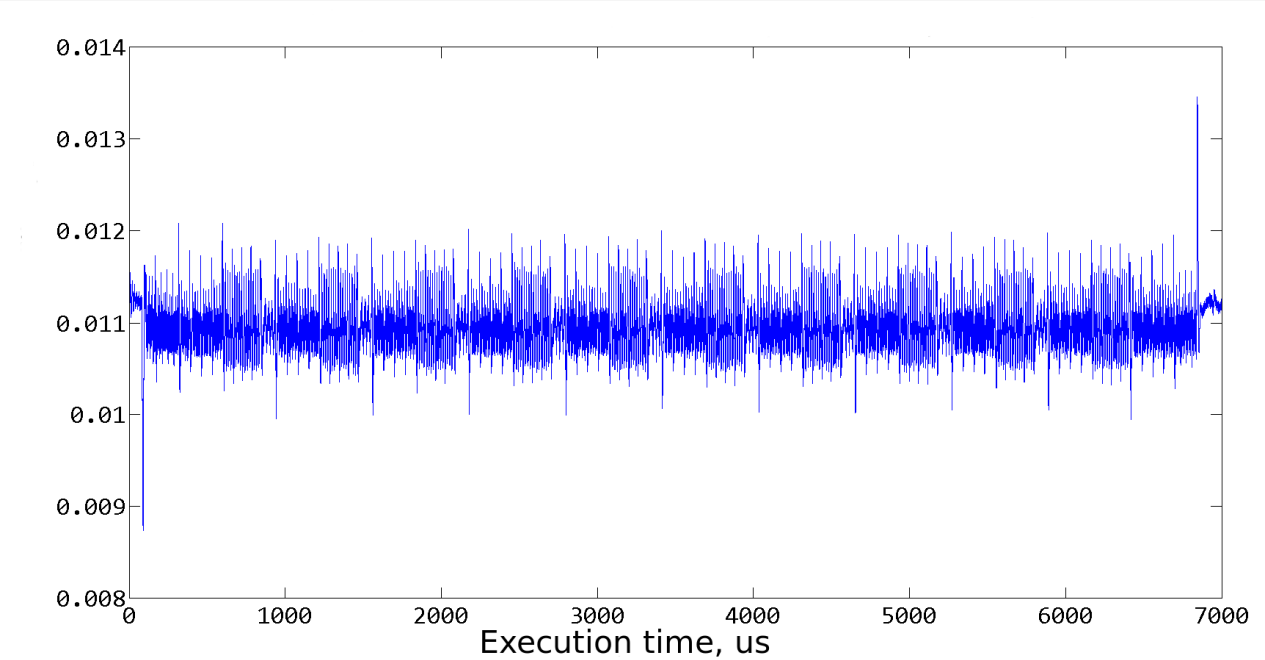
A bit of technology

- A mobile phone is a good example (not for an attack but for visualisation)
- More intensive usage will drain your battery faster
- The battery charge is dropped because electrons (current) travel from anode to cathode (or ions travelling from cathode to anode)

A bit of technology

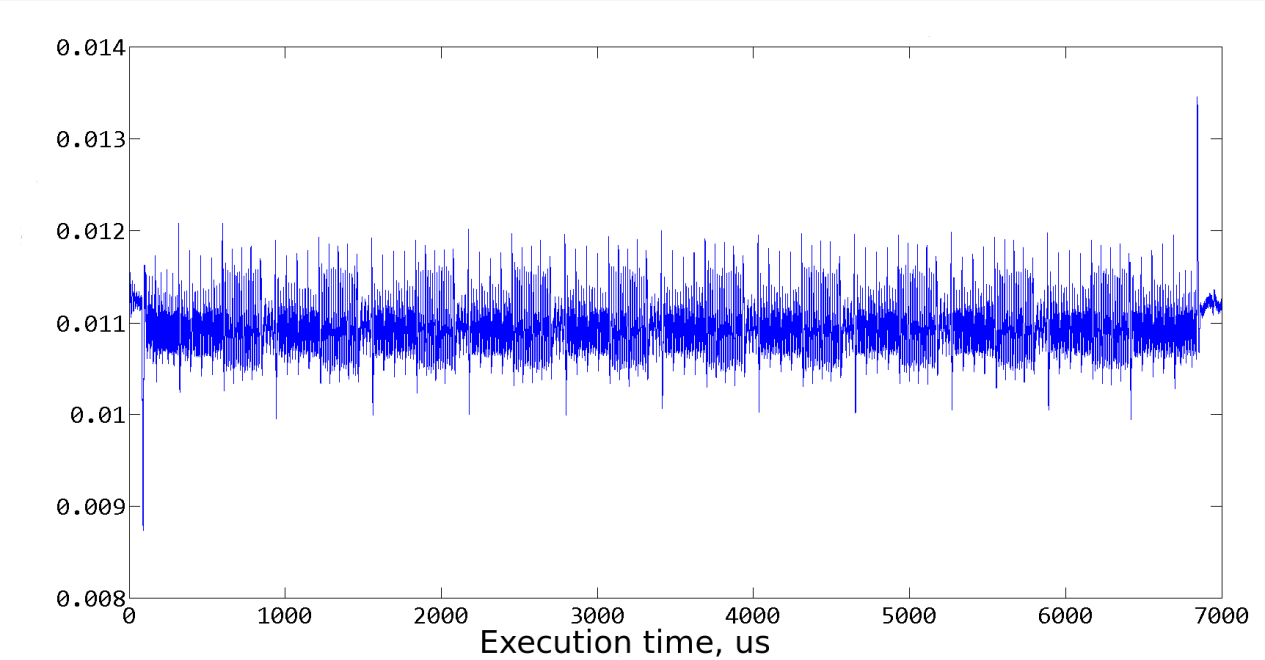
- A mobile phone is a good example (not for an attack but for visualisation)
- More intensive usage will drain your battery faster
- The battery charge is dropped because electrons (current) travel from anode to cathode (or ions travelling from cathode to anode)
- The level of "discharge" is defined by an operation itself (XOR, OR, MOV) and the processed data.

AES-128 in software running on STM8



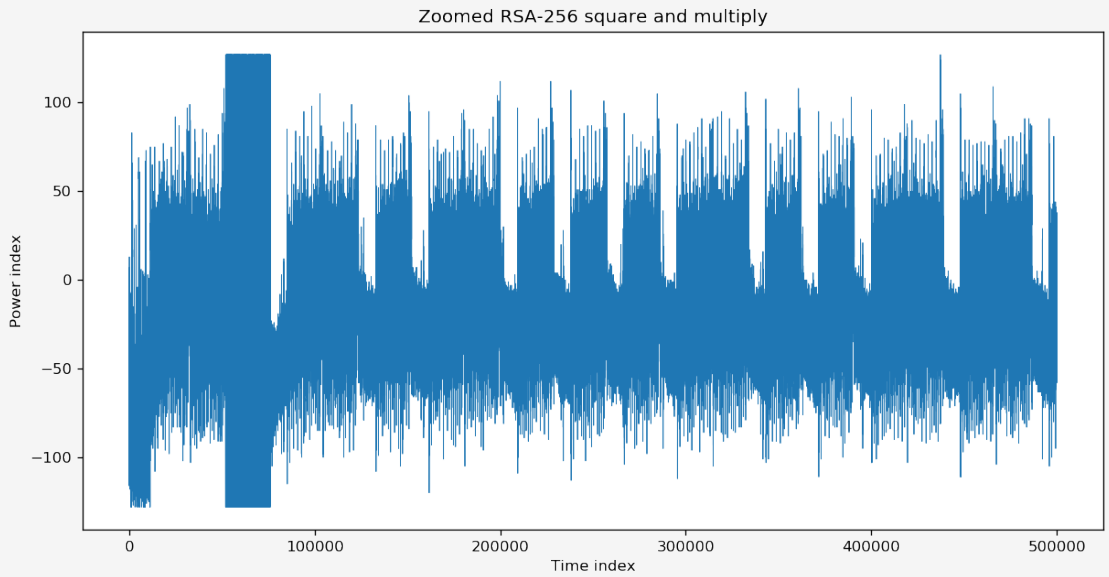
- Measured current drain (energy consumption) during AES-128
- Can AES-128 operations be detected?
  - One round with KeyAddition only
  - Nine rounds with Sbox, ShiftRows, MixColumn and KeyAddition operations
  - One round with Sbox, ShiftRows and KeyAddition operations

AES-128 in software running on STM8



- Measured current drain (energy consumption) during AES-128
- Can AES-128 operations be detected?
  - One round with KeyAddition only
  - Nine rounds with Sbox, ShiftRows, MixColumn and KeyAddition operations
  - One round with Sbox, ShiftRows and KeyAddition operations
- One trace is not sufficient to get the AES-128 key (most probably)

RSA algorithm running on ESP32



- RSA algorithm consumes energy as well
- Straightforward RSA execution contains a loop over the size of a private key:
  - When a key bit is equal to 0: only one multiply operation is performed
  - When a key bit is equal to 1: two multiply operations are performed
- Computations take time and consume energy (and we will use this)

Side-channel information examples

Practical exercise for RSA

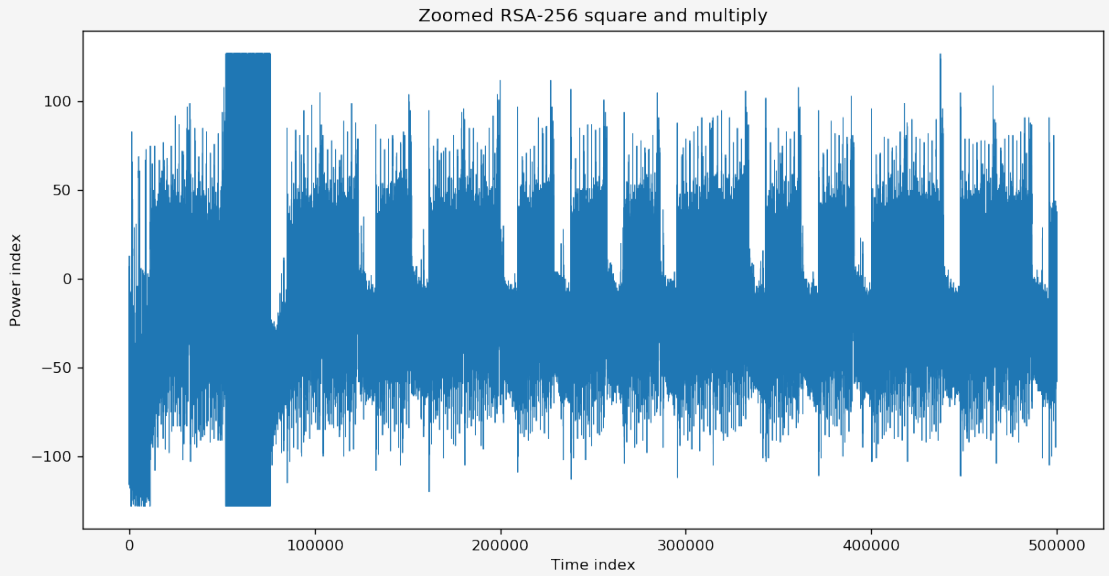


Figure: You will need to get the full RSA key of 256 bits

Side-channel information examples

Practical exercise for RSA

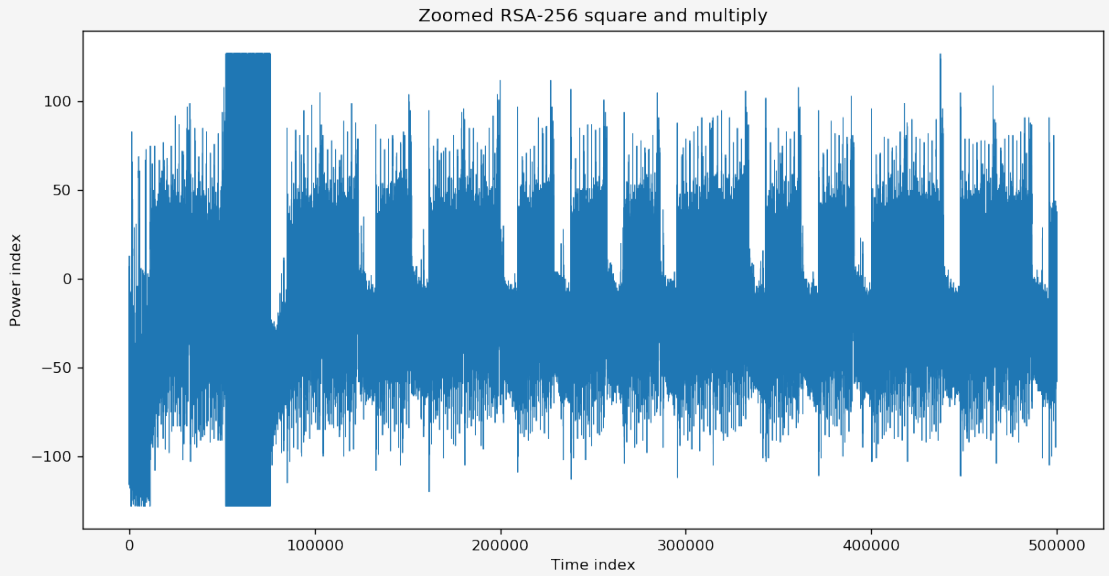


Figure: You will need to get the full RSA key of 256 bits

We will do the task but first let us recap the RSA algorithm

RSA task

RSA is based on two properties:

- Big number factorisation problem.

  For a given $N = p \cdot q$ we cann't find integers p and q

RSA task

RSA is based on two properties:

- Big number factorisation problem.

  For a given $N = p \cdot q$ we cann't find integers p and q

- Fermat–Euler theorem or Euler's totient theorem:

  $a^{\varphi(N)} \equiv 1 \mod N$, where $\varphi(N)$ is Euler's totient function

RSA task

RSA is based on two properties:

- Big number factorisation problem.

  For a given $N = p \cdot q$ we cann't find integers p and q

- Fermat–Euler theorem or Euler's totient theorem:

  $a^{\varphi(N)} \equiv 1 \mod N$, where $\varphi(N)$ is Euler's totient function

1. Select two primes: p, q

RSA task

RSA is based on two properties:

- Big number factorisation problem.

  For a given $N = p \cdot q$ we cann't find integers p and q

- Fermat–Euler theorem or Euler's totient theorem:

  $a^{\varphi(N)} \equiv 1 \mod N$, where $\varphi(N)$ is Euler's totient function

1. Select two primes: p, q
2. Compute the modulus: $N = p \cdot q$

RSA task

RSA is based on two properties:

- Big number factorisation problem.

  For a given $N = p \cdot q$ we cann't find integers p and q

- Fermat–Euler theorem or Euler's totient theorem:

  $a^{\varphi(N)} \equiv 1 \mod N$, where $\varphi(N)$ is Euler's totient function

1. Select two primes: p, q
2. Compute the modulus: $N = p \cdot q$
3. Select a public exponent: e (often e is equal to 0x10001)

RSA task

RSA is based on two properties:

- Big number factorisation problem.

  For a given $N = p \cdot q$ we cann't find integers p and q

- Fermat–Euler theorem or Euler's totient theorem:

  $a^{\varphi(N)} \equiv 1 \mod N$, where $\varphi(N)$ is Euler's totient function

1. Select two primes: p, q
2. Compute the modulus: $N = p \cdot q$
3. Select a public exponent: e (often e is equal to 0x10001)
4. Compute the Euler's totien function: $\varphi(N) = (p-1)(q-1)$

RSA task

RSA is based on two properties:

- Big number factorisation problem.

$$\text{For a given } N = p \cdot q \text{ we cann't find integers } p \text{ and } q$$

- Fermat–Euler theorem or Euler's totient theorem:

$$a^{\varphi(N)} \equiv 1 \mod N, \text{ where } \varphi(N) \text{ is Euler's totient function}$$

1. Select two primes: p, q
2. Compute the modulus: $N = p \cdot q$
3. Select a public exponent: e (often e is equal to 0x10001)
4. Compute the Euler's totien function: $\varphi(N) = (p-1)(q-1)$
5. Compute private key $d = e^{-1} \mod \varphi(N)$

RSA task

RSA is based on two properties:

- Big number factorisation problem.

  For a given $N = p \cdot q$ we cann't find integers p and q

- Fermat–Euler theorem or Euler's totient theorem:

  $a^{\varphi(N)} \equiv 1 \mod N$, where $\varphi(N)$ is Euler's totient function

1. Select two primes: p, q
2. Compute the modulus: $N = p \cdot q$
3. Select a public exponent: e (often e is equal to 0x10001)
4. Compute the Euler's totien function: $\varphi(N) = (p-1)(q-1)$
5. Compute private key $d = e^{-1} \mod \varphi(N)$
6. Public key: (e, N), private key: (d, N)

RSA task

RSA signature:

- In asymmetric cryptography signature is generated by an owner of a private key (d, N):

$$s = m^d \mod N$$

RSA task

RSA signature:

- In asymmetric cryptography signature is generated by an owner of a private key (d, N):

$$s = m^d \mod N$$

- Exponentiation m^d is the longest operation in RSA algorithm

RSA task

RSA signature:

- In asymmetric cryptography signature is generated by an owner of a private key (d, N):

$$s = m^d \mod N$$

- Exponentiation m^d is the longest operation in RSA algorithm
- Signature verification $s^e \equiv m \mod N$ is done with the public key (e, N), we also need to remember that $a^{\varphi(N)} \equiv 1 \mod N$:

$$\begin{aligned} s^e \mod N &= \left(m^d\right)^e \mod N = \\ &= m^{e \cdot d} \mod N = \\ &= m^{1 \mod \varphi(N)} \mod N = \\ &= m \cdot \left(m^{k \cdot \varphi(N)}\right) \mod N = \\ &= m \cdot \left(m^k\right)^{\varphi(N)} \mod N = m \mod N \end{aligned}$$

RSA task

- Square and Multiply algorithm computes $m^d \mod N$

RSA task

- Square and Multiply algorithm computes $m^d \mod N$
- The algorithm iterates over a private exponent bits $d = [d_{k-1}, d_{k-2}, ..., d_0]$

RSA task

- Square and Multiply algorithm computes $m^d \mod N$
- The algorithm iterates over a private exponent bits $d = [d_{k-1}, d_{k-2}, ..., d_0]$
- Two ways to compute the exponent (left-to-right and right-to-left approaches)

RSA task

Left-to-Right

$$m^d = m^{d_0} \cdot \left(m^{d_1} \cdot \left(\ldots \left(m^{d_{k-1}} \right)^2 \right)^2 \right)^2$$

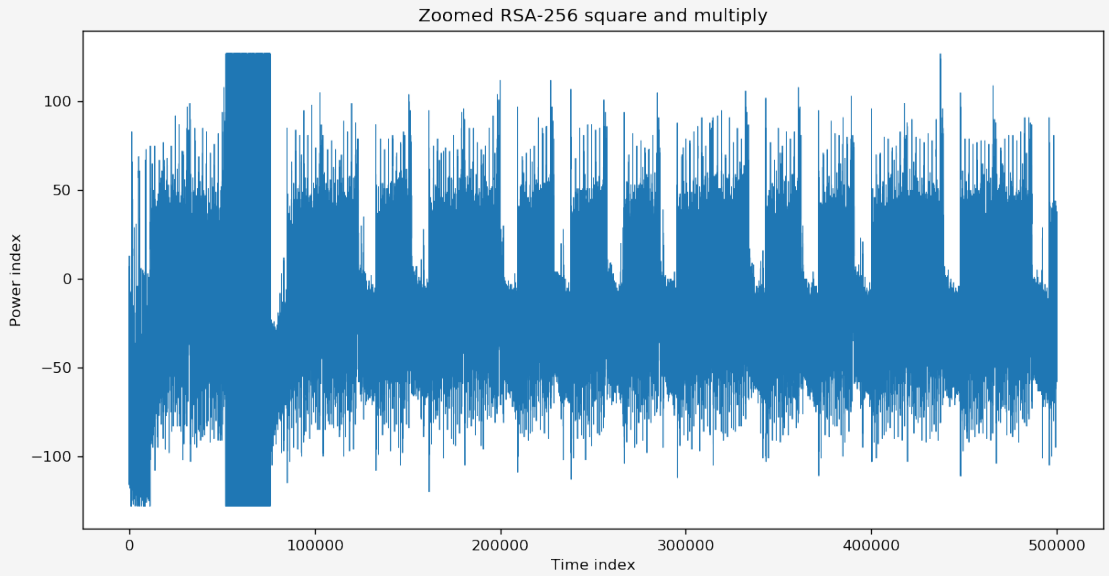
Input: m, d, N
Output: $m^d \mod N$
$a \leftarrow 1$;
for $i = k - 1$ **to** 0 **do**
  $a \leftarrow a^2 \mod N$;
  if $d_i = 1$ **then**
    $a \leftarrow a \times m \mod N$;
  end
end
return a;

Right-to-Left

$$m^d = m^{d_{k-1} 2^{k-1}} \cdot m^{d_{k-2} 2^{k-2}} \cdot \ldots \cdot m^{d_0}$$

Input: m, d, N
Output: $m^d \mod N$
$a \leftarrow 1$; $b \leftarrow m$;
for $i = 0$ **to** $k - 1$ **do**
  if $d_i = 1$ **then**
    $a \leftarrow a \times b \mod N$;
  end
  $b \leftarrow b^2 \mod N$;
end
return a;

RSA task



for $i = k - 1$ **to** 0 **do**
  $a \leftarrow a^2 \mod N$;
  if $d_i = 1$ **then**
    $a \leftarrow a \times m \mod N$;
  end
end

for $i = 0$ **to** $k - 1$ **do**
  if $d_i = 1$ **then**
    $a \leftarrow a \times b \mod N$;
  end
  $b \leftarrow b^2 \mod N$;
end

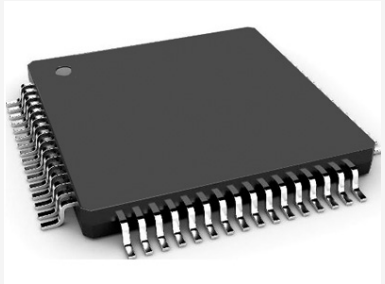
RSA task

Jupyter lab window...

Main terms and concepts

- We need to agree on terminology and understand the main concepts
  - Current drain (power consumption is a more general term)
  - Leakage
  - Side-channel attacks

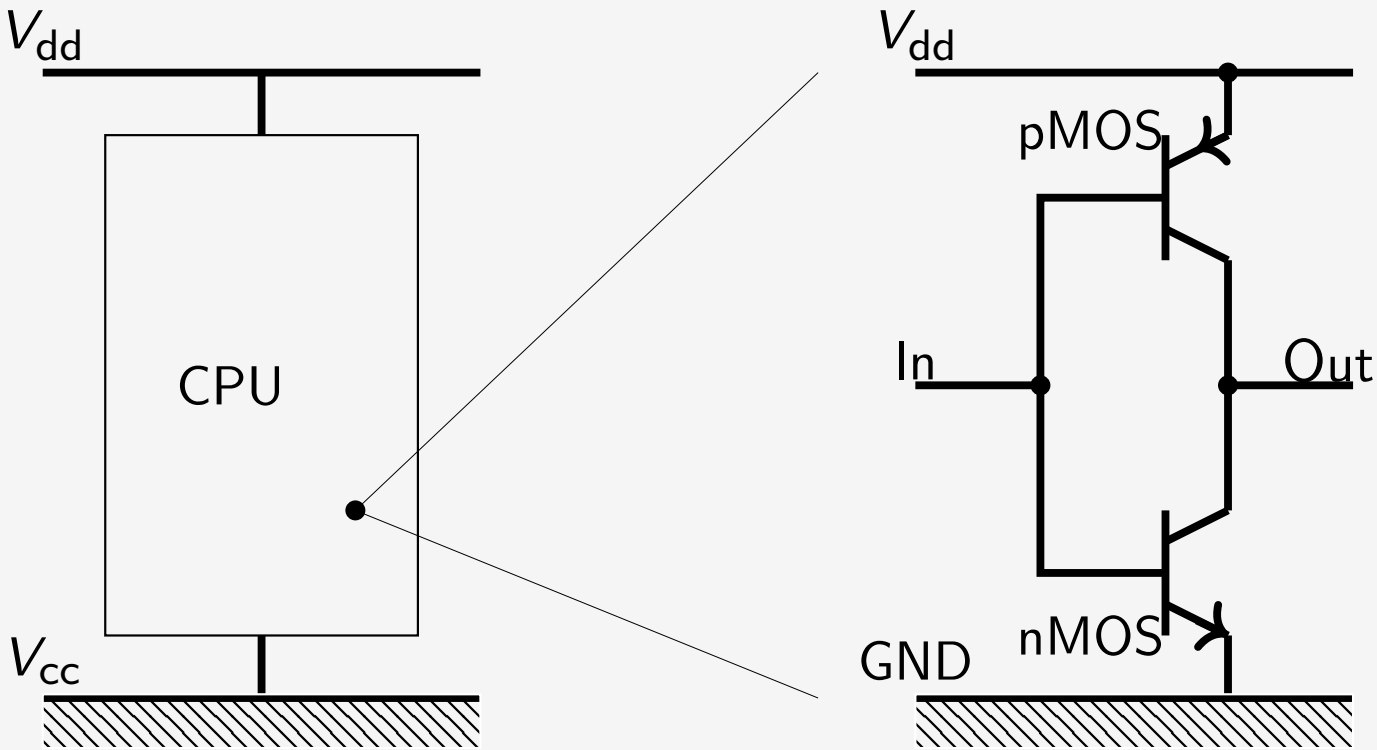
A bit of phylosophy



- All algorithms (crypto, AI, blockchains, communication, etc.) are executed by hardware, such as a smart card, a microcontroller, and a "big" CPU...
- Complementary metal–oxide–semiconductor (CMOS) is an absolutely dominant technology for consumer semiconductor devices
- CMOS "converts" electricity into computations (remember your phone)

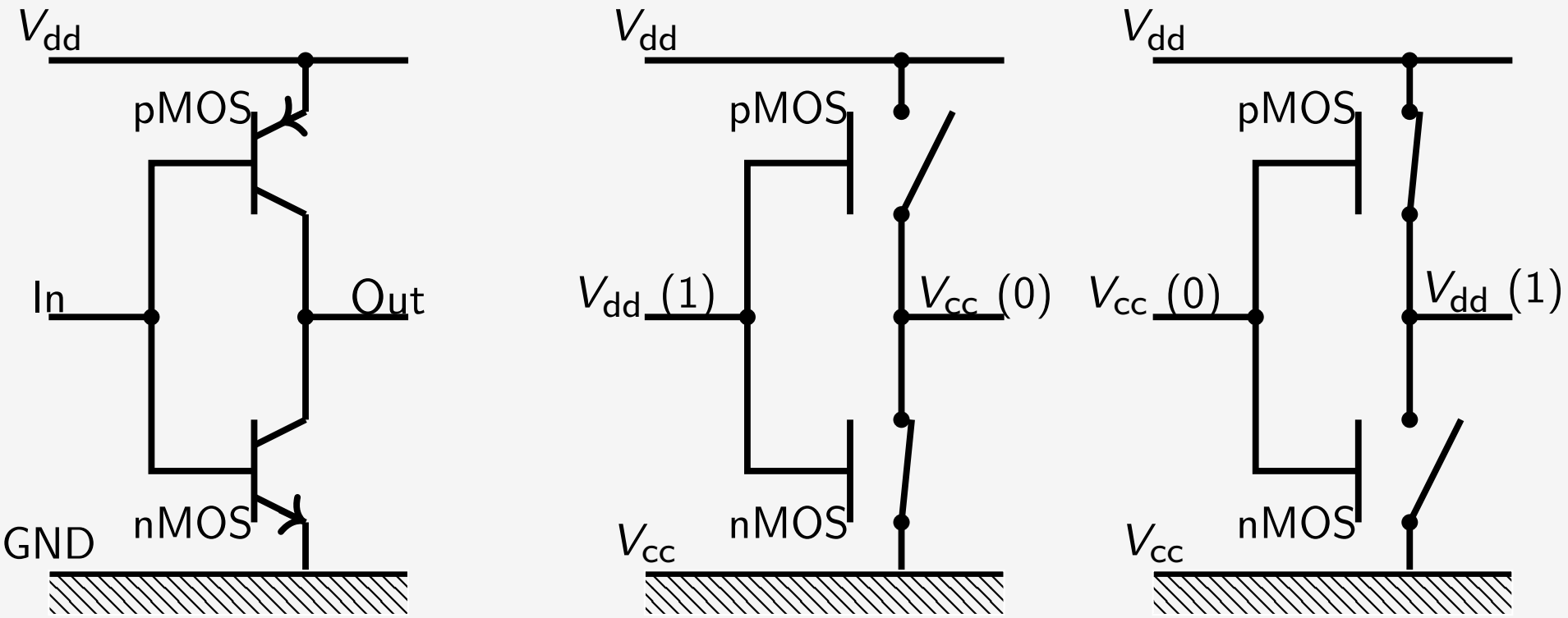
Currently, companies don't invest in other mass-market semiconductor technologies. The CMOS ecosystem is huge. We will be using CMOS technology by inertion. Good point: CMOS technology has a clear evolution roadmap (we saw shrinkage from 350nm to 3nm).

Inverter (NOT gate)



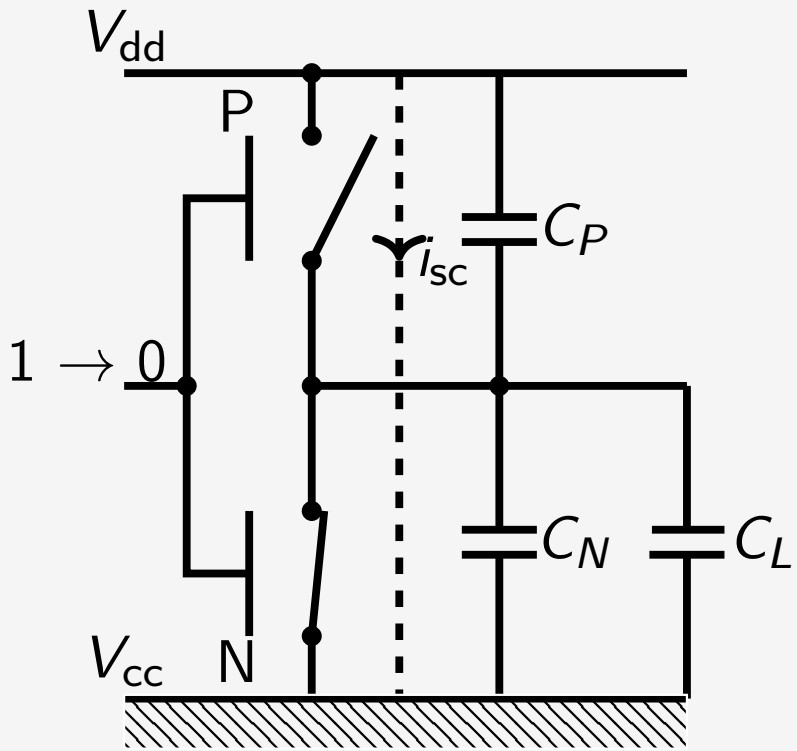
- Inverter (NOT gate) is the "smallest" logical element
- This element inverts an input signal (1 to 0, 0 to 1)

Inverter (NOT gate) in static mode



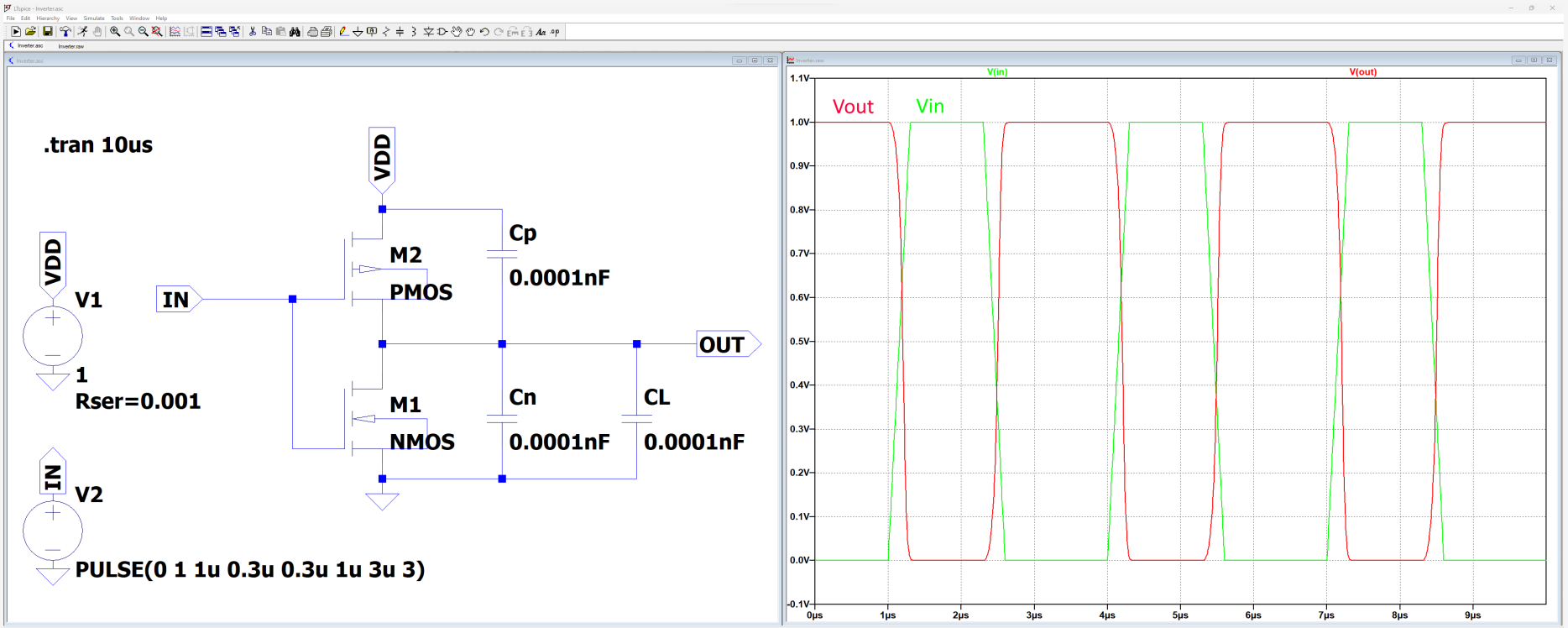
- This element inverts an input signal (1 to 0, 0 to 1)
- When nMOS transistor receives a voltage around 0, the connection from the source to the drain will be broken (open circuit)
- The pMOS transistor works counter to the nMOS transistor

Inverter (NOT gate) in switching mode



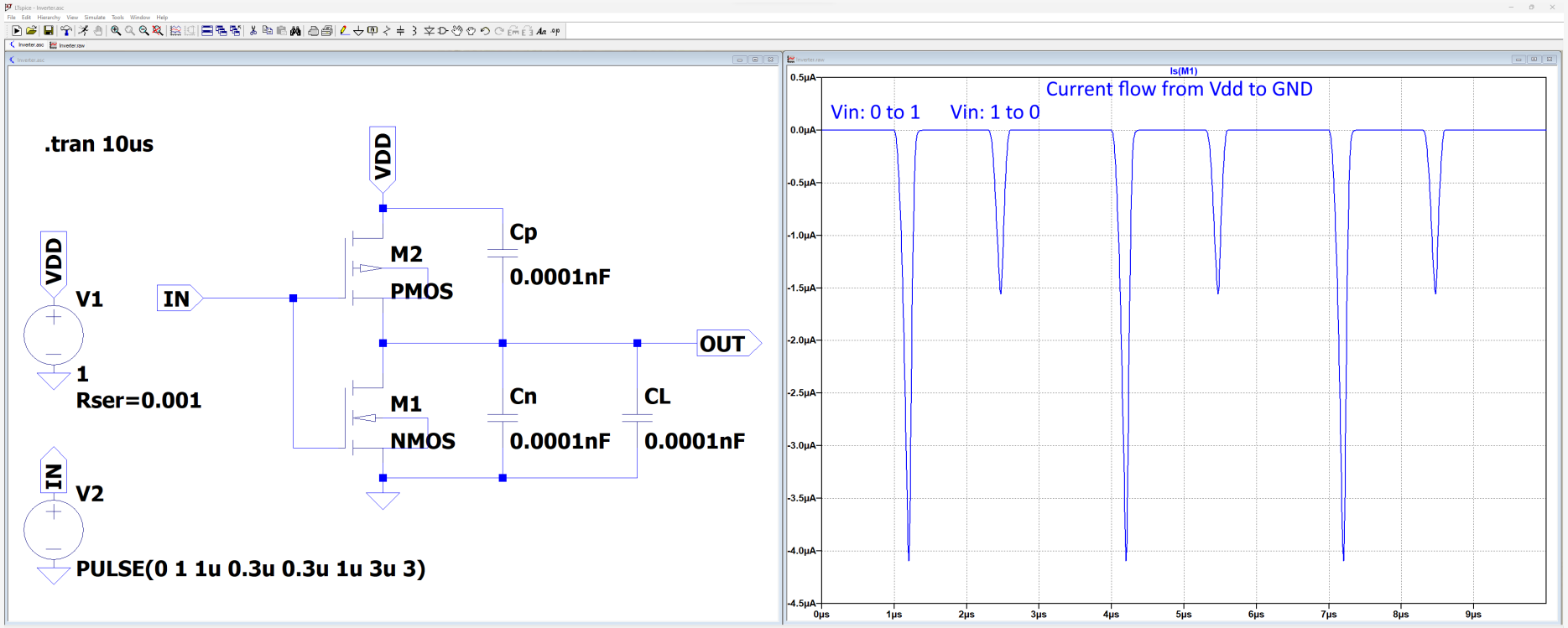
- When switching an inverter drains current from V_{dd} to V_{cc} (GND)
- pMOS capacitance C_p
- nMOS capacitance C_n
- Load capacitance C_L (following circuit and components)

Simulating current drain of an inverter



- LTSpice simulation illustrates the work of an inverter
- V_{out} and V_{in} are in counter positions (V_{in} is inverted on V_{out})
- Current drain occurs during a transition

Simulating current drain of an inverter



- Current is irreversibly drained from V_{dd} to GND (consumed) during transition
- Current drain causes a battery discharge due to electrons travel
- V_{in} transition from 0 to 1 consumes more than the transition from 1 to 0

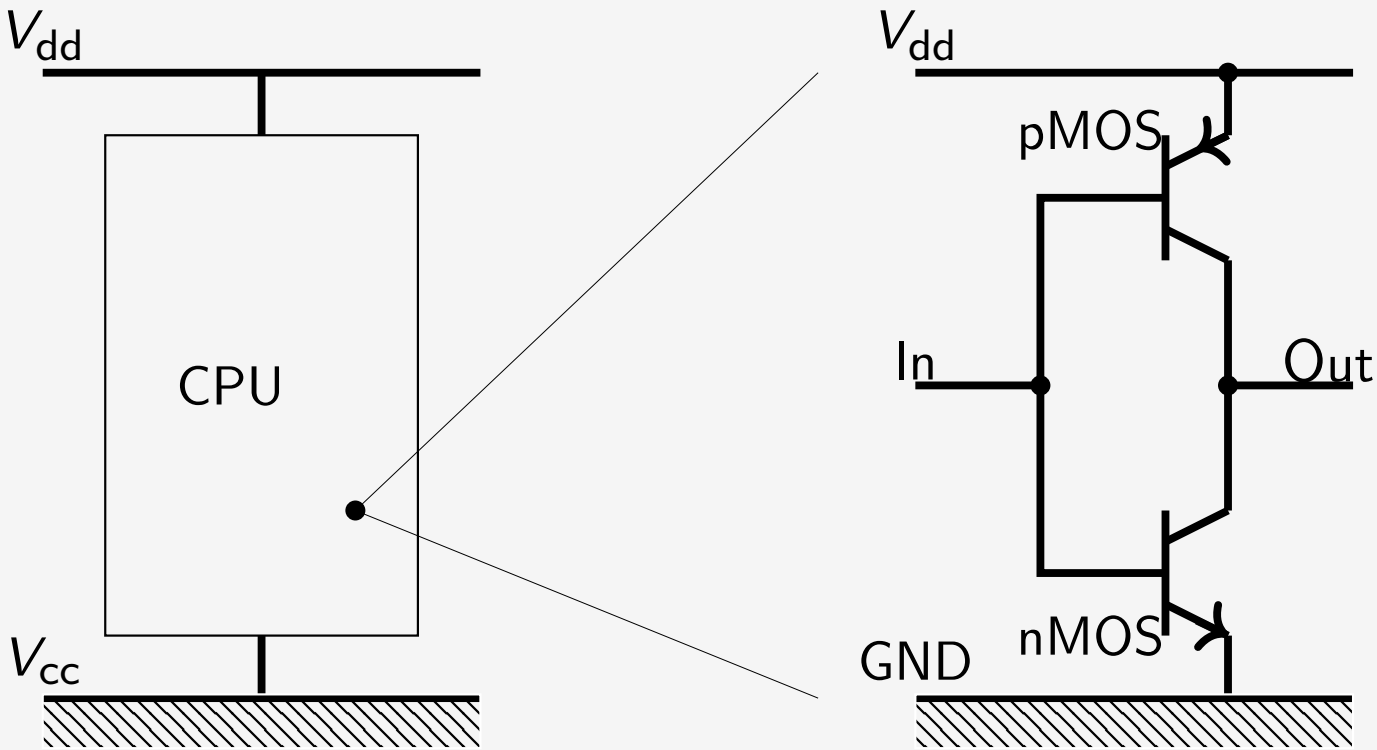
Simulating current drain of an inverter

Current values during transition of V_{in}

Transition of V_{in}	Value of current i
$0 \rightarrow 0$	$i \approx 0$
$1 \rightarrow 1$	$i \approx 0$
$1 \rightarrow 0$	$i \approx i_{\text{pn}}$
$0 \rightarrow 1$	$i \approx i_{\text{pn}} + i_L$

- i_{pn} current through pMOS and nMOS
- i_L current through the load (following elements)

Inverter (NOT gate)



- Inverter's current drain is tiny
- Reducing production technology (90ns to 45ns and below) reduces current drain (that is why your phone battery shall last longer)

Logic gates

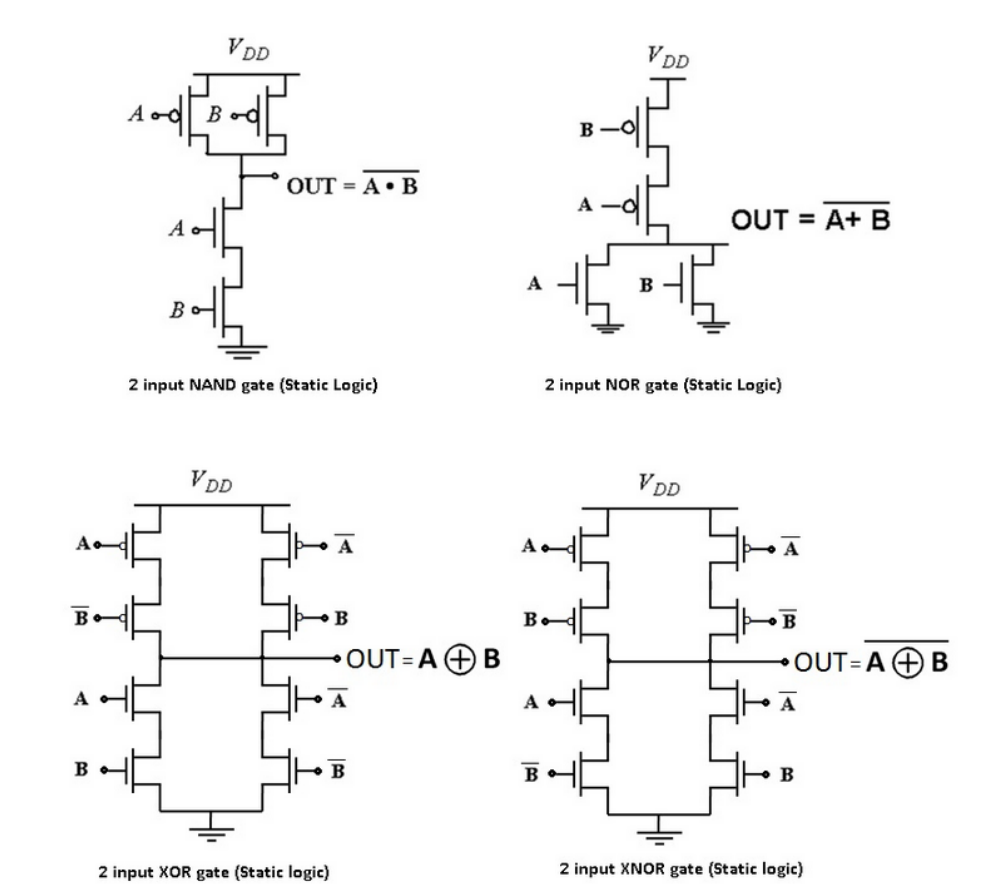
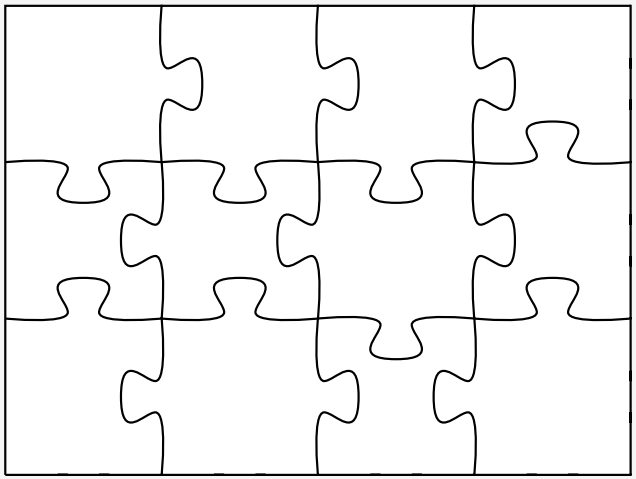


Fig.3: The circuit diagram for 2-input NAND, NOR, XOR and XNOR gates in CMOS static logic.

- Logic gates (NOT, OR, NAND...) are made of nMOS and pMOS
- Assembled logic gates implement registers, CPU instructions and other digital elements

Puzzle 2: Law of large numbers A bit of mathematics... who said side-channels are easy



Law of large numbers

- This is the cornerstone law for side-channel attacks

Law of large numbers

- This is the cornerstone law for side-channel attacks

The average of the results obtained from a large number of trials should be close to the expected value and tend to become closer to the expected value as more trials are performed

$$\lim_{x \to \inf} \sum_{i=0}^{n} \frac{X_i}{n} = \bar{X}$$

Law of large numbers

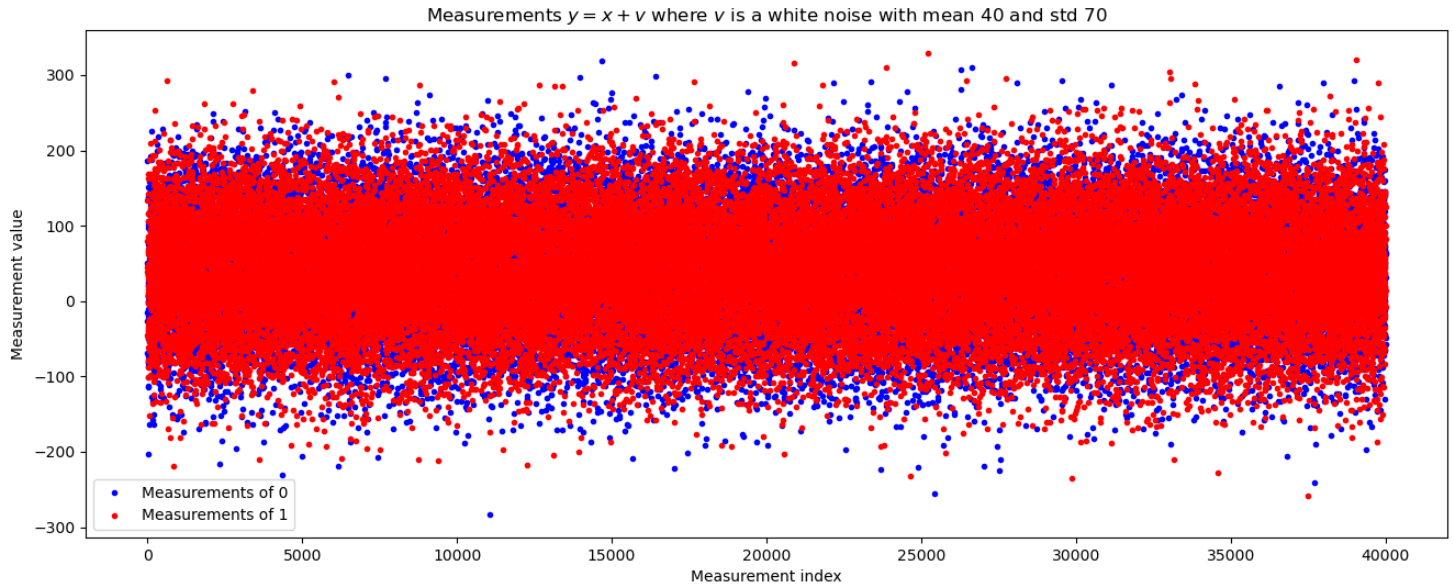
- This is the cornerstone law for side-channel attacks

> The average of the results obtained from a large number of trials should be close to the expected value and tend to become closer to the expected value as more trials are performed

$$\lim_{x \to \inf} \sum_{i=0}^{n} \frac{X_i}{n} = \bar{X}$$

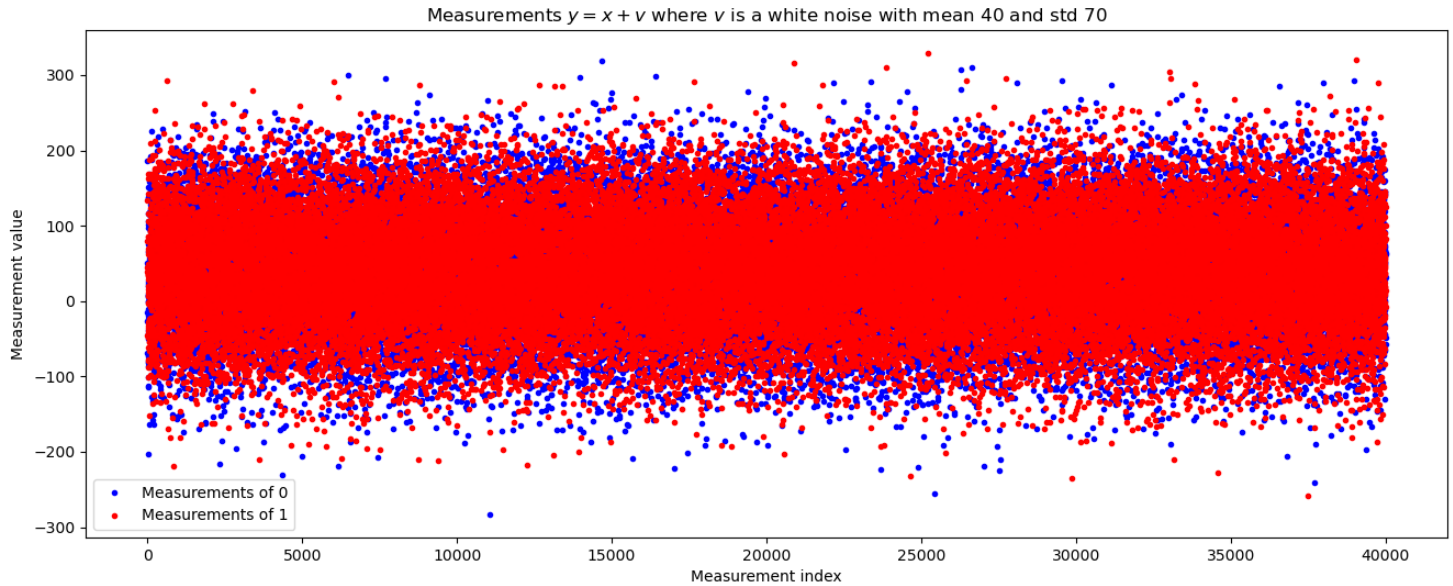
- What does this mean in practice?

Law of large numbers: white noise



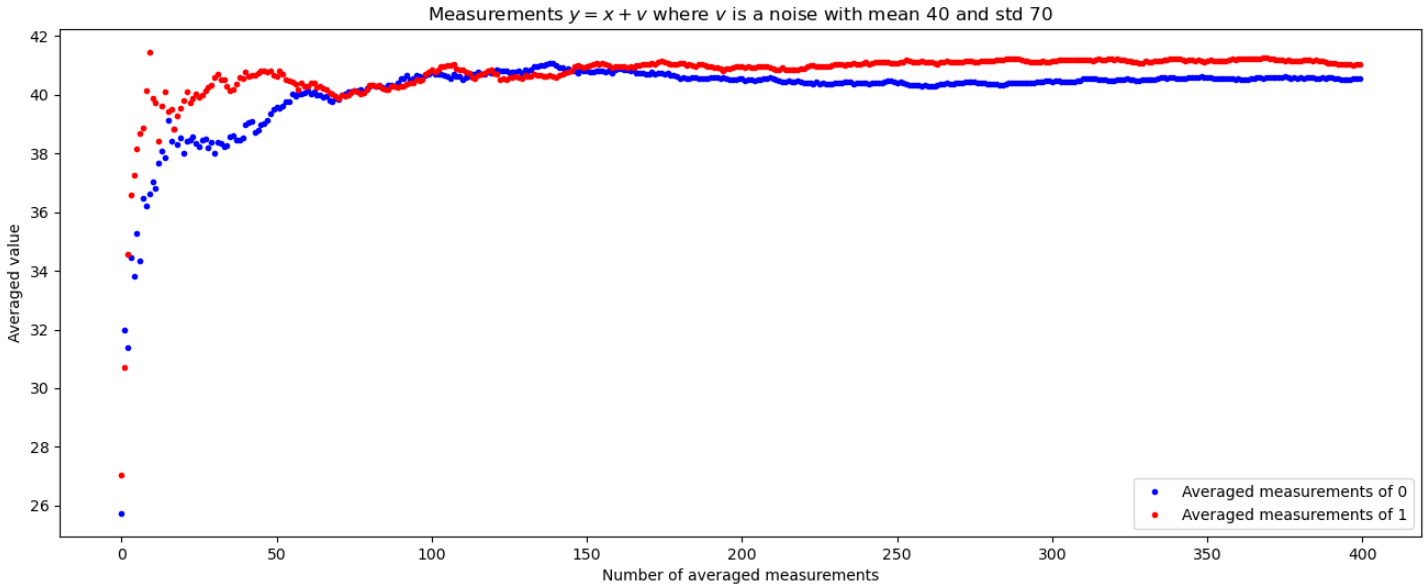
- Consider a situation when we measure a process: $y = x + \nu$, where x is 0 or 1 and ν is a white noise with mean $\mu = 40$ and std $\sigma = 70$
- Use python to emulate measurements

Law of large numbers: white noise



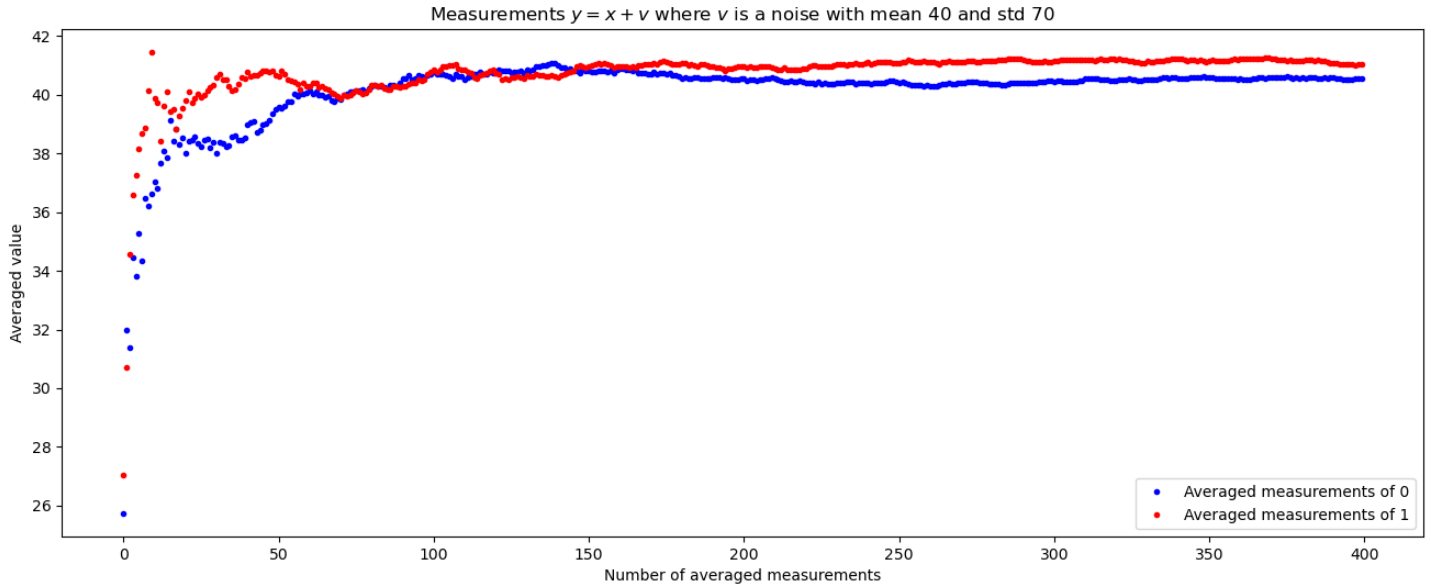
- Consider a situation when we measure a process: $y = x + \nu$, where x is 0 or 1 and ν is a white noise with mean $\mu = 40$ and std $\sigma = 70$
- Use python to emulate measurements
- By looking at raw measurements, 0 and 1 can not be distinguished

Law of large numbers: white noise



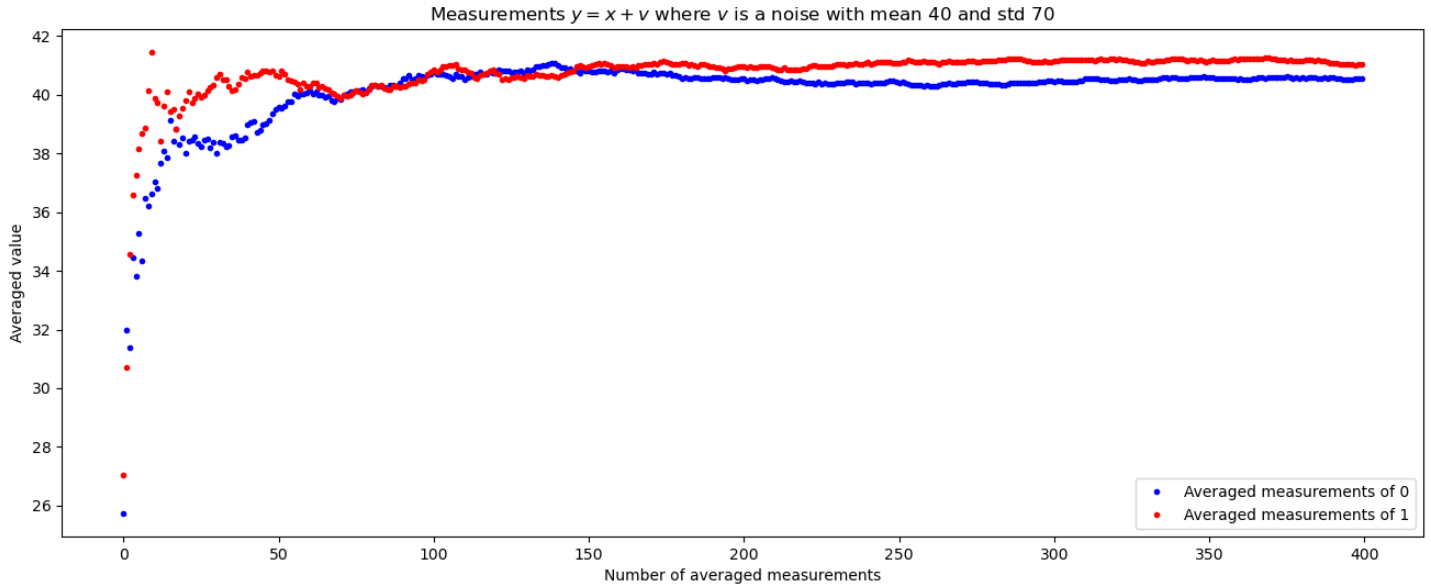
- Here comes the Law of Large Numbers

Law of large numbers: white noise



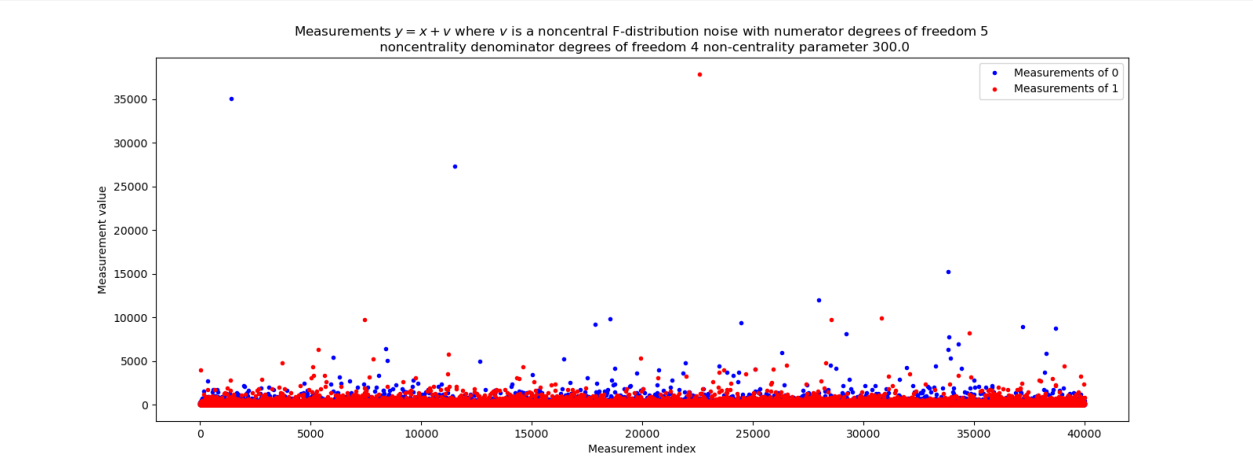
- Here comes the Law of Large Numbers
- For $x = 0$ the curve converges to 40
- For $x = 1$ the curve converges to 41

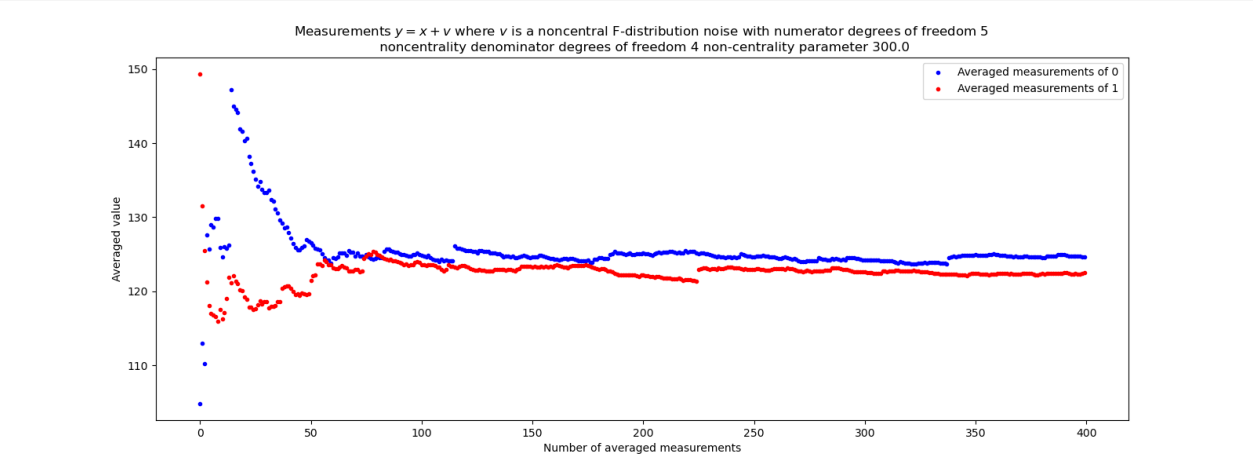
Law of large numbers: white noise



- Here comes the Law of Large Numbers
- For $x = 0$ the curve converges to 40
- For $x = 1$ the curve converges to 41
- Now we see a clear difference

Law of large numbers: more complex noise



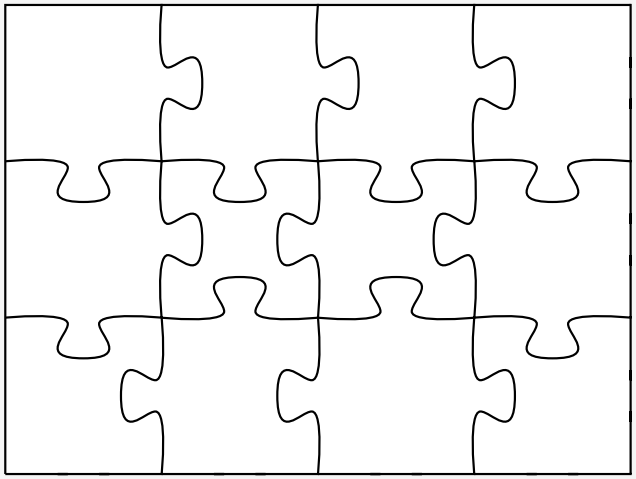


- Measurements with the noncentral F-distribution noise

Law of large numbers: conclusion

- LLN shows that averaging a "sufficiently" big number of samples will converge noise to constant
- In certain cases, we can directly average measurements for side-channel attacks
- Nevertheless, by averaging, we can lose important information about random numbers

Puzzle 3: Power models



Power models

- We cannot model the exact current drain (power consumption):
  - A digital circuit is too complex, and the number of switching transistors is huge
  - Current drain will depend on technology, DC voltage, temperature and other parameters

Power models

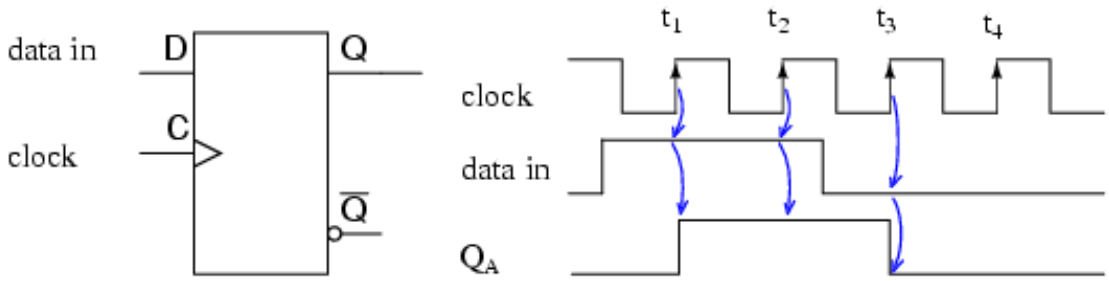
- We cannot model the exact current drain (power consumption):
  - A digital circuit is too complex, and the number of switching transistors is huge
  - Current drain will depend on technology, DC voltage, temperature and other parameters
- Side-channel attacks don't try to have the exact current drain value (I would like to see if someone does it)

Power models

- We cannot model the exact current drain (power consumption):
  - A digital circuit is too complex, and the number of switching transistors is huge
  - Current drain will depend on technology, DC voltage, temperature and other parameters
- Side-channel attacks don't try to have the exact current drain value (I would like to see if someone does it)
- Instead, side-channel attacks work with power consumption models

Registers

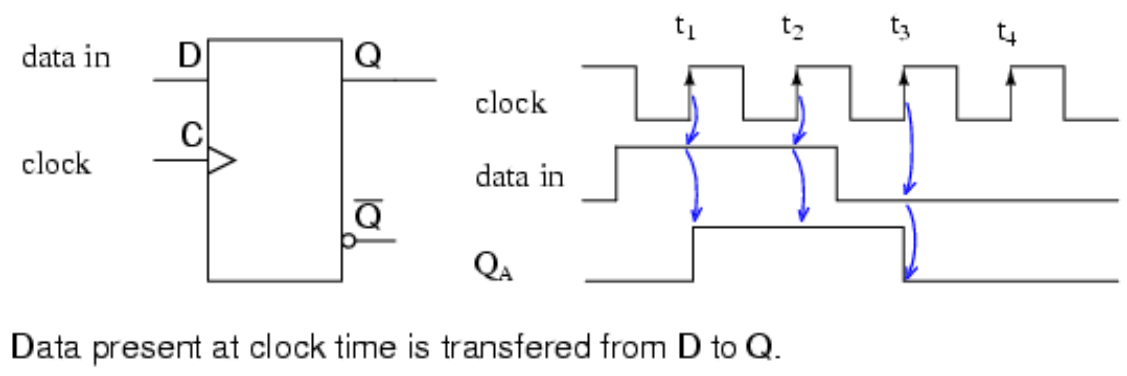
- Digital logic contains multiple registers (e.g., CPU registers)
- Registers are made of flip-flops

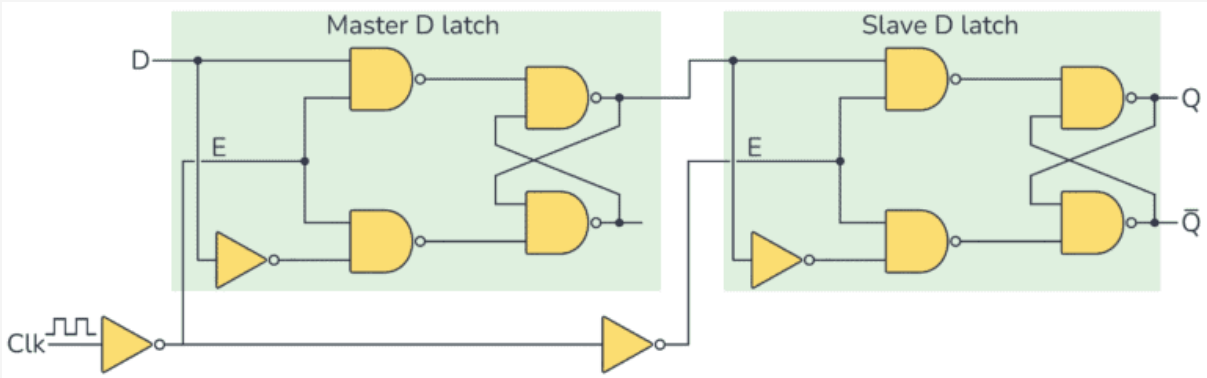


Data present at clock time is transfered from D to Q.

Registers

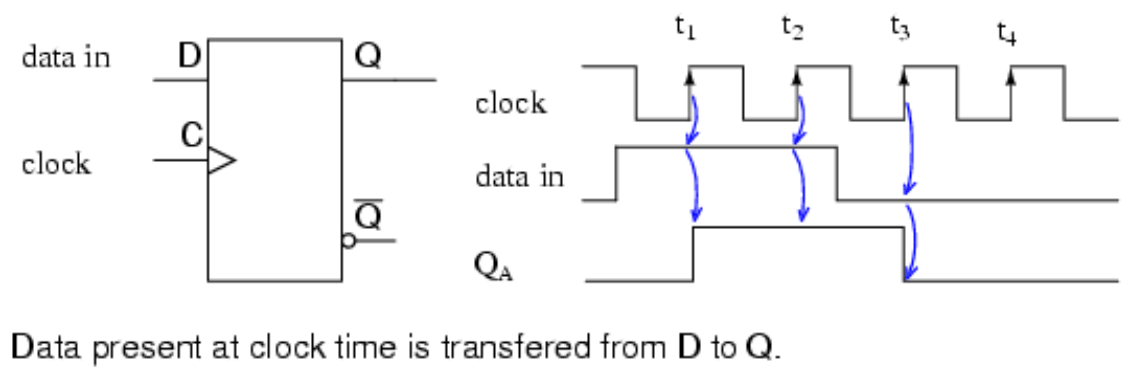
- Digital logic contains multiple registers (e.g., CPU registers)
- Registers are made of flip-flops

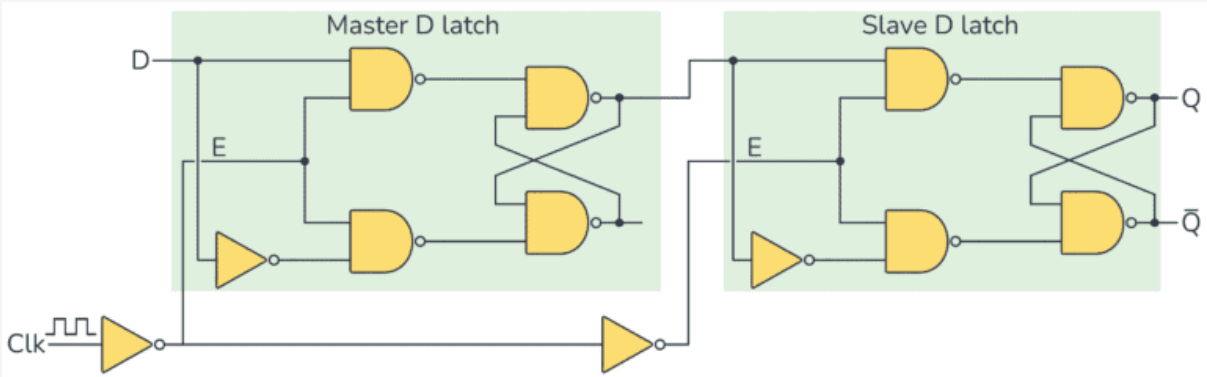




Registers

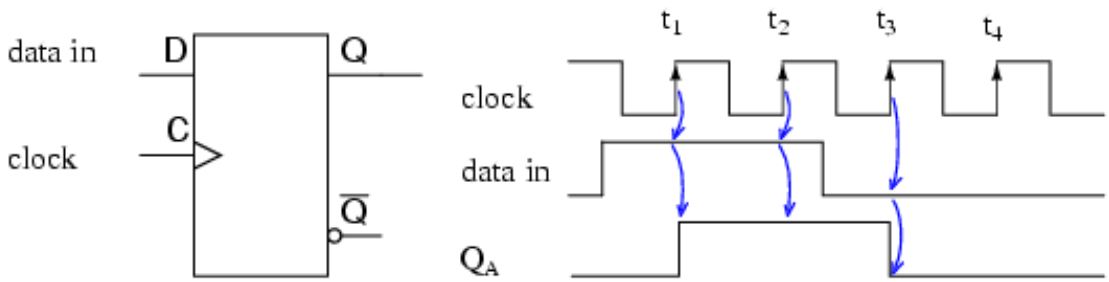
- Digital logic contains multiple registers (e.g., CPU registers)
- Registers are made of flip-flops





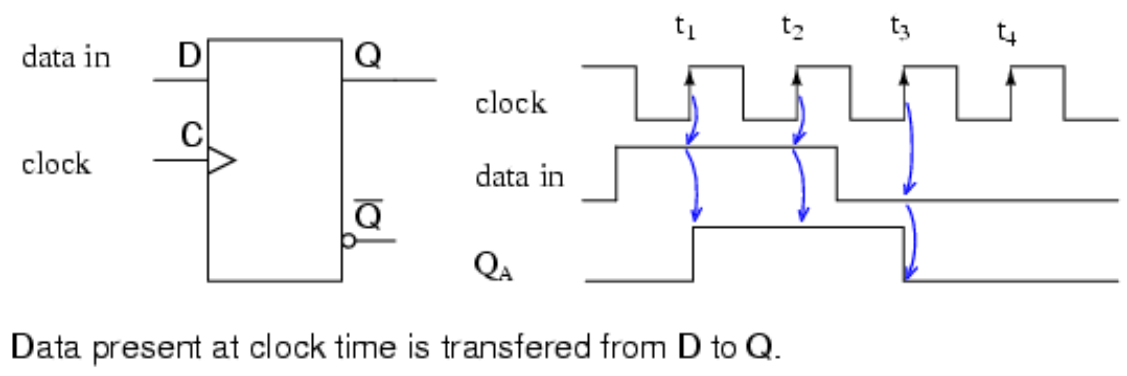
- D flip-flop is made of combinational logic, but the value is updated only on clock rising edge
- D flip-flop elements drain current

Registers



Data present at clock time is transfered from D to Q.

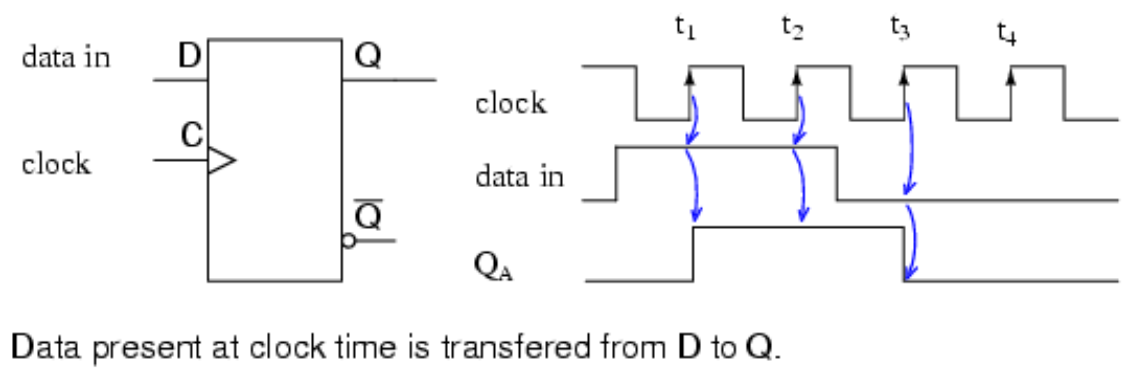
Registers



Data present at clock time is transfered from D to Q.

- Flip-flops consume current during transitions $0 \rightarrow 1$ and $1 \rightarrow 0$

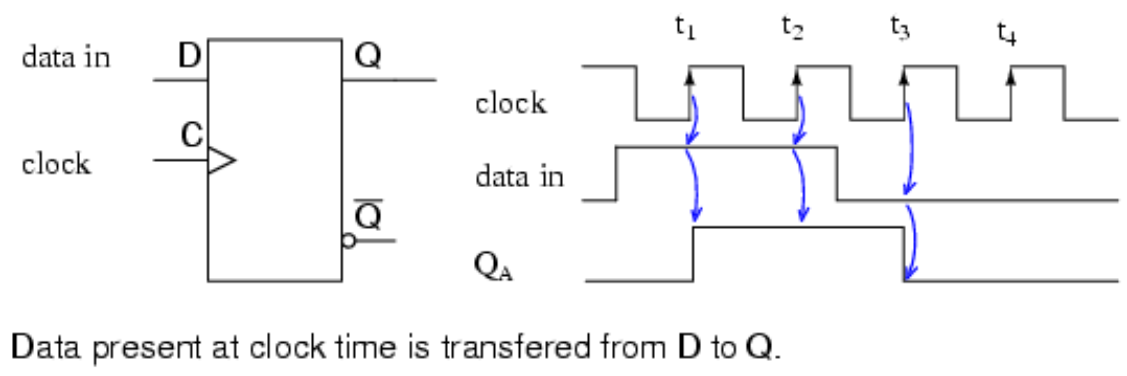
Registers



Data present at clock time is transfered from D to Q.

- Flip-flops consume current during transitions $0 \rightarrow 1$ and $1 \rightarrow 0$
- As seen with the inverter, consumption during the transitions $0 \rightarrow 1$ and $1 \rightarrow 0$ differ

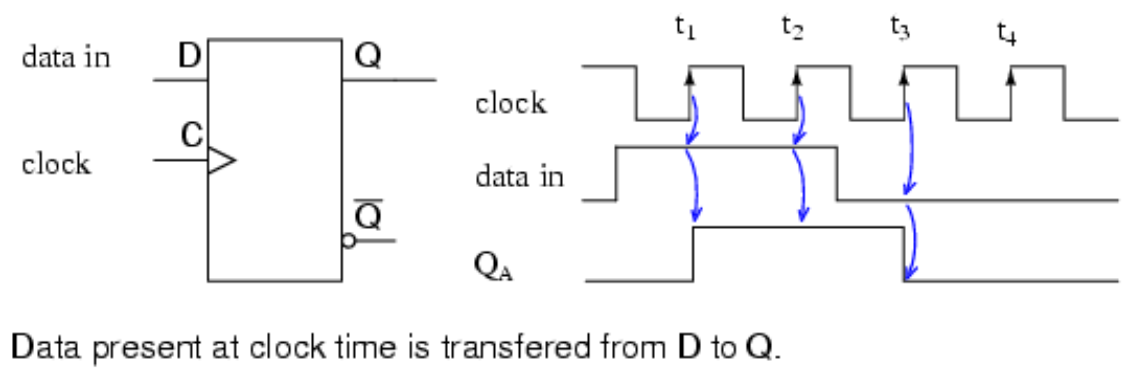
Registers



Data present at clock time is transfered from D to Q.

- Flip-flops consume current during transitions $0 \rightarrow 1$ and $1 \rightarrow 0$
- As seen with the inverter, consumption during the transitions $0 \rightarrow 1$ and $1 \rightarrow 0$ differ
- Denote consumptions: $i(0 \rightarrow 1) = \alpha$ and $i(1 \rightarrow 0) = \beta$

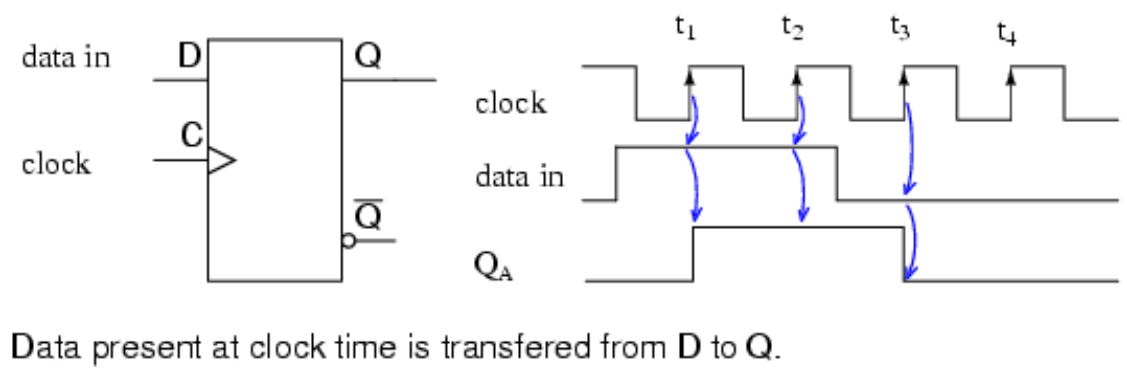
Registers



Data present at clock time is transfered from D to Q.

- Flip-flops consume current during transitions $0 \rightarrow 1$ and $1 \rightarrow 0$
- As seen with the inverter, consumption during the transitions $0 \rightarrow 1$ and $1 \rightarrow 0$ differ
- Denote consumptions: $i(0 \rightarrow 1) = \alpha$ and $i(1 \rightarrow 0) = \beta$
- α and β depend on a microcontroller (technology, layout), power supply, and many other parameters

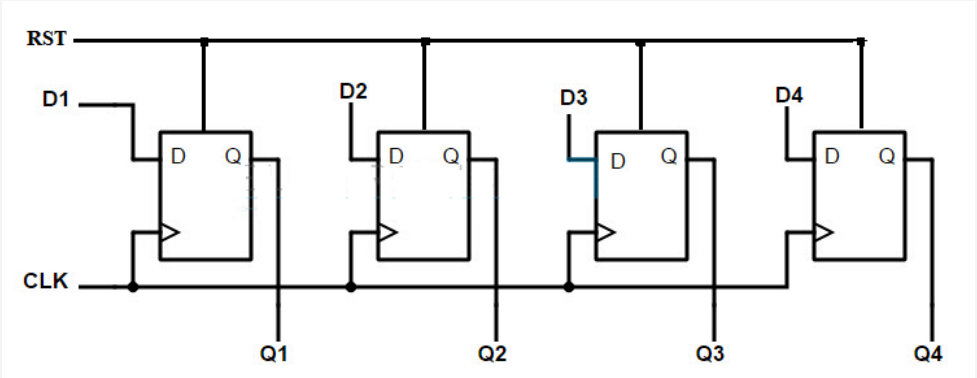
Registers



Data present at clock time is transfered from D to Q.

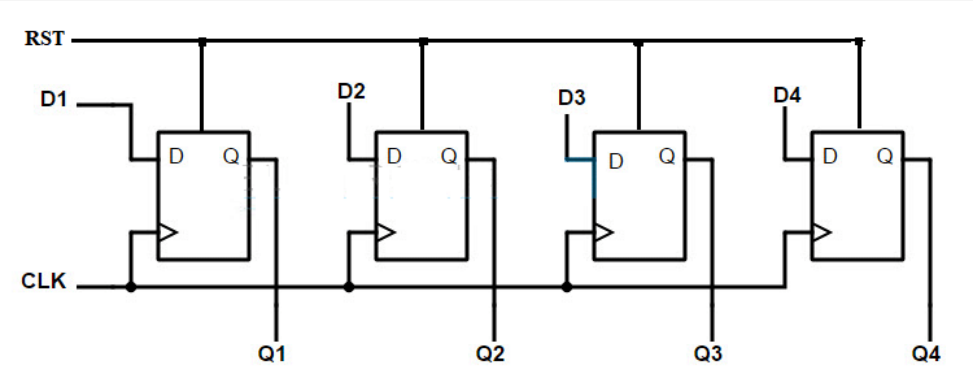
- Flip-flops consume current during transitions $0 \rightarrow 1$ and $1 \rightarrow 0$
- As seen with the inverter, consumption during the transitions $0 \rightarrow 1$ and $1 \rightarrow 0$ differ
- Denote consumptions: $i(0 \rightarrow 1) = \alpha$ and $i(1 \rightarrow 0) = \beta$
- α and β depend on a microcontroller (technology, layout), power supply, and many other parameters
- Assume that $\beta = \alpha + \delta$

Registers



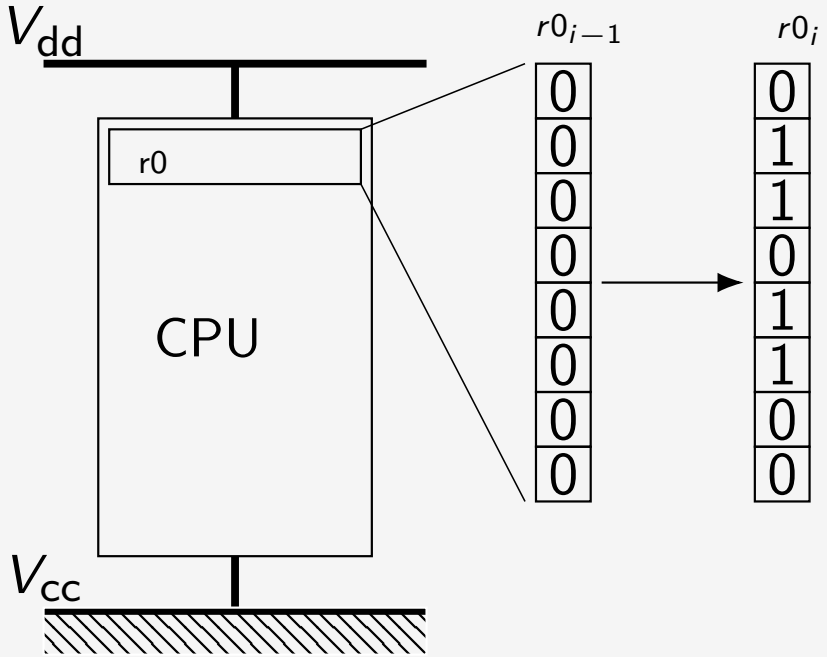
- Flip-flops are assembled into registers (CPU registers, special registers, etc.)

Registers



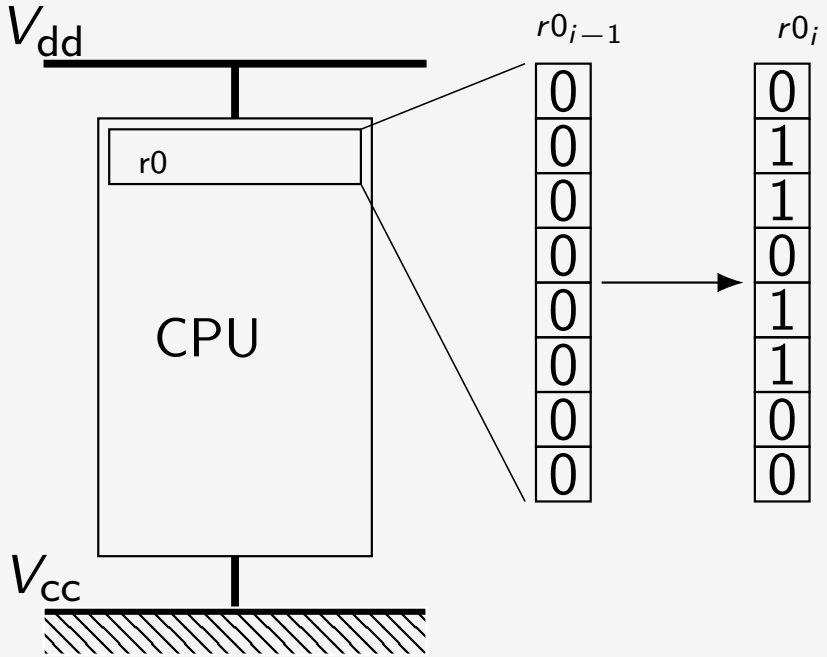
- Flip-flops are assembled into registers (CPU registers, special registers, etc.)
- Register's flip-flops have the same layout and components, so they consume equally

Power consumption of a register change



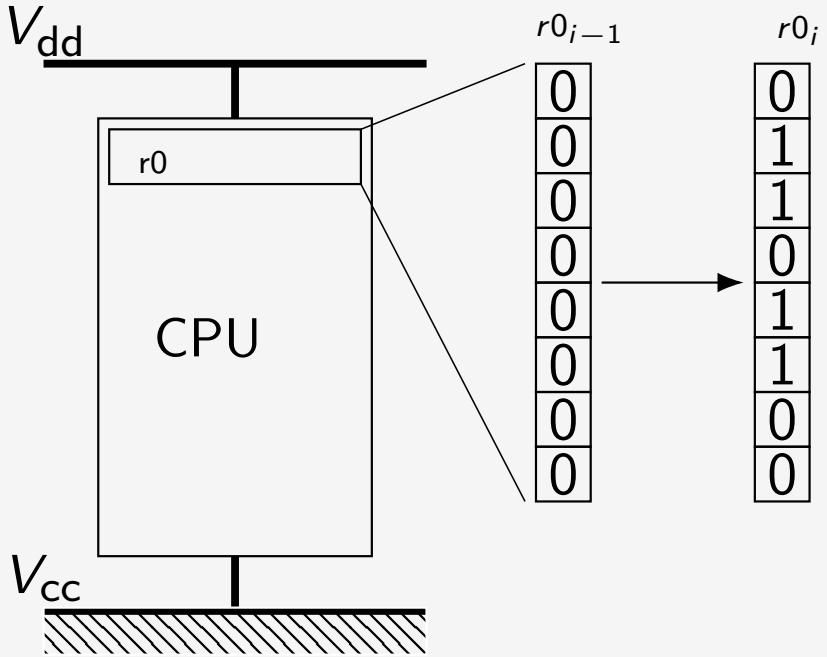
- When register switches the current is drained (power is consumed)

Power consumption of a register change



- When register switches the current is drained (power is consumed)
- The exact current consumption is not possible to model

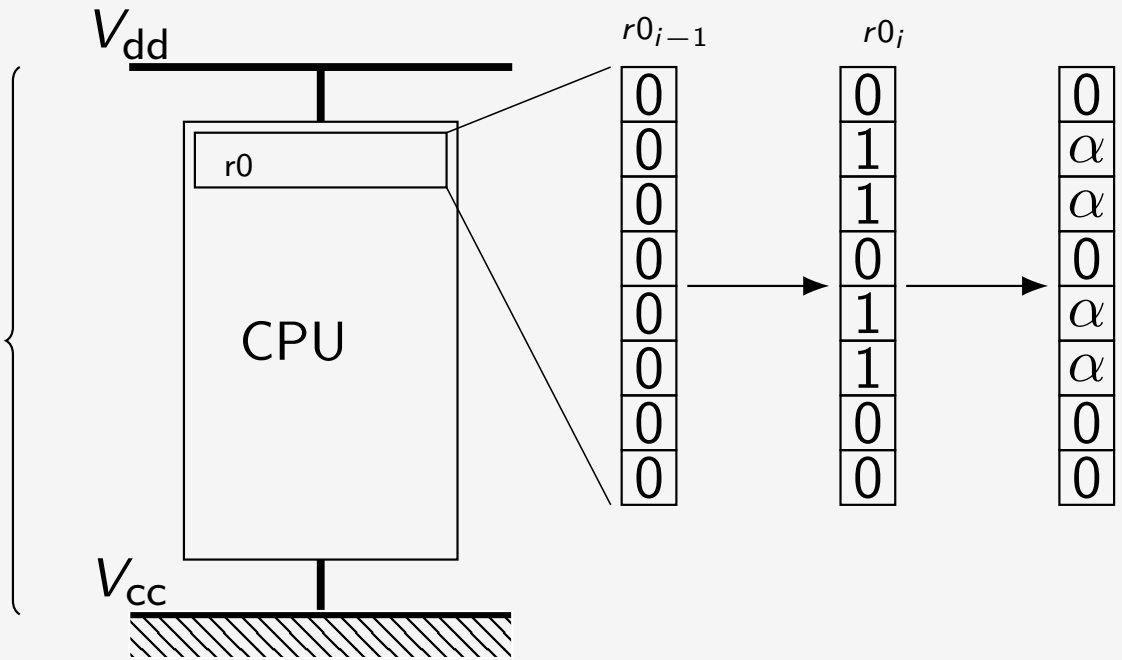
Power consumption of a register change



- When register switches the current is drained (power is consumed)
- The exact current consumption is not possible to model
- Instead, side-channel analysis studies the differences between various operations (kind of true)

Power consumption of a register change

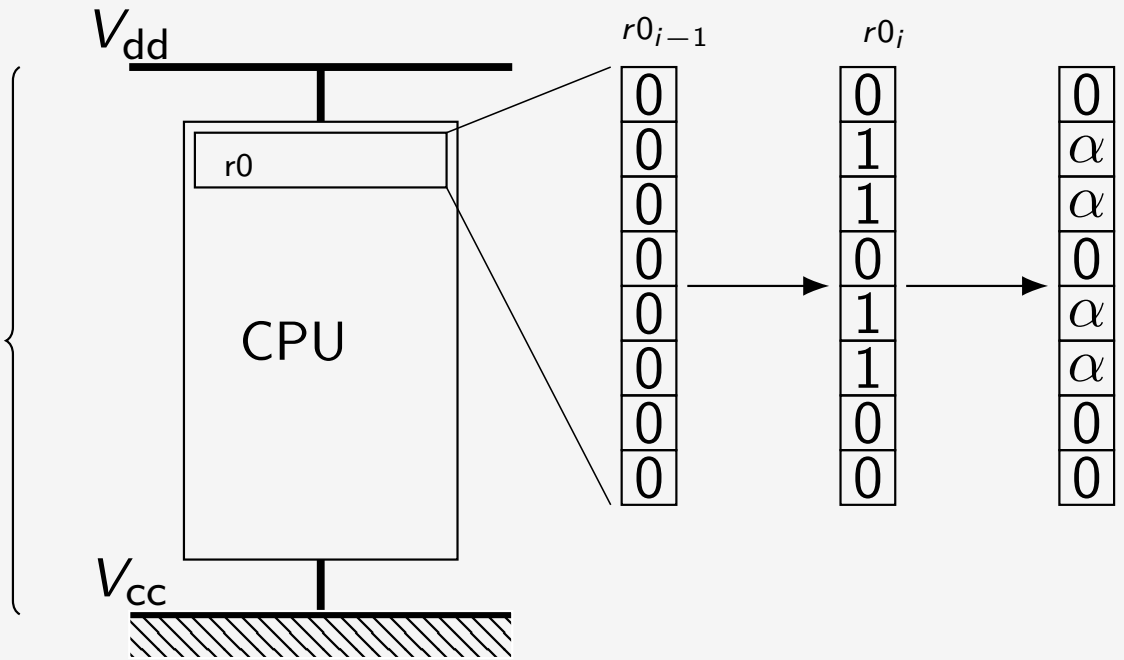
Oscilloscope



- Current drain of a transition $0x00 \rightarrow 0x6C$ shall be proportional to the number of switched bits: $i(0x00 \rightarrow 0x6C) \approx 4 \cdot i(0 \rightarrow 1) = 4 \cdot \alpha$

Power consumption of a register change

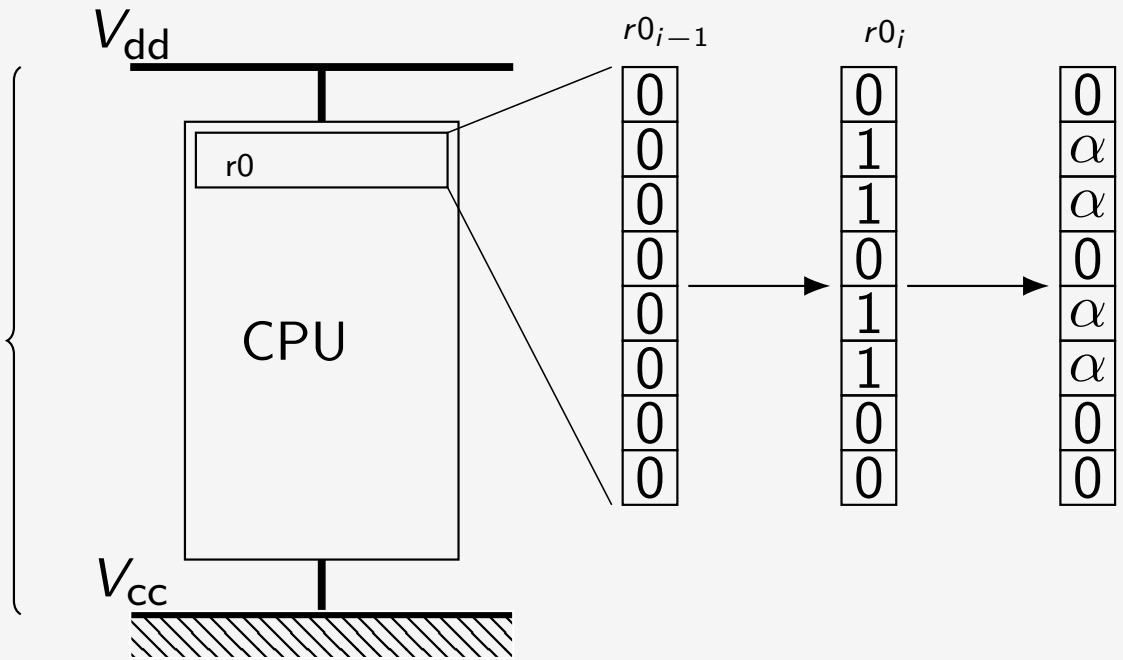
Oscilloscope



- Current drain of a transition $0x00 \rightarrow 0x6C$ shall be proportional to the number of switched bits: $i(0x00 \rightarrow 0x6C) \approx 4 \cdot i(0 \rightarrow 1) = 4 \cdot \alpha$
- In practice measurements of the transition will include noise σ: $i(0x00 \rightarrow 0x6C) = 4 \cdot \alpha + \sigma$

Power consumption of a register change

Oscilloscope



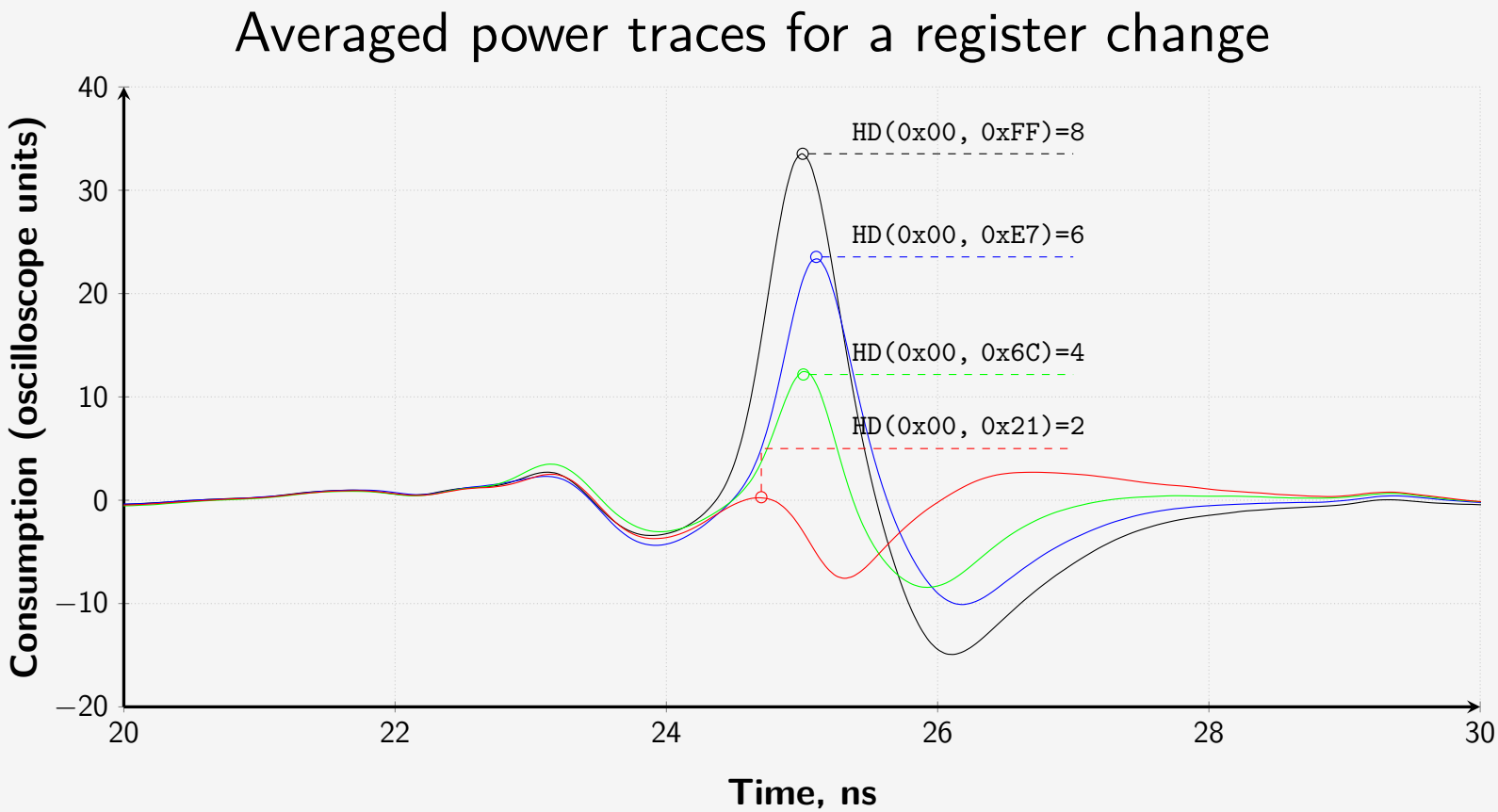
- Current drain of a transition $0x00 \rightarrow 0x6C$ shall be proportional to the number of switched bits: $i(0x00 \rightarrow 0x6C) \approx 4 \cdot i(0 \rightarrow 1) = 4 \cdot \alpha$
- In practice measurements of the transition will include noise σ: $i(0x00 \rightarrow 0x6C) = 4 \cdot \alpha + \sigma$
- The most significant noise is coming from other microcontroller blocks (video processor, modem, DMA) and environment (cross talks, EM signals around, etc.)

- Hamming distance denotes the number of different bits between the previous and the new register values: $\mathrm{HD}(r_{i-1}, r_i)$

- Hamming distance denotes the number of different bits between the previous and the new register values: $\mathrm{HD}(r_{i-1}, r_i)$
- $i(0x00 \rightarrow 0x6C) \approx \mathrm{HD}(0x00, 0x6C) \cdot i(0 \rightarrow 1) + \sigma = 4 \cdot \alpha + \sigma$

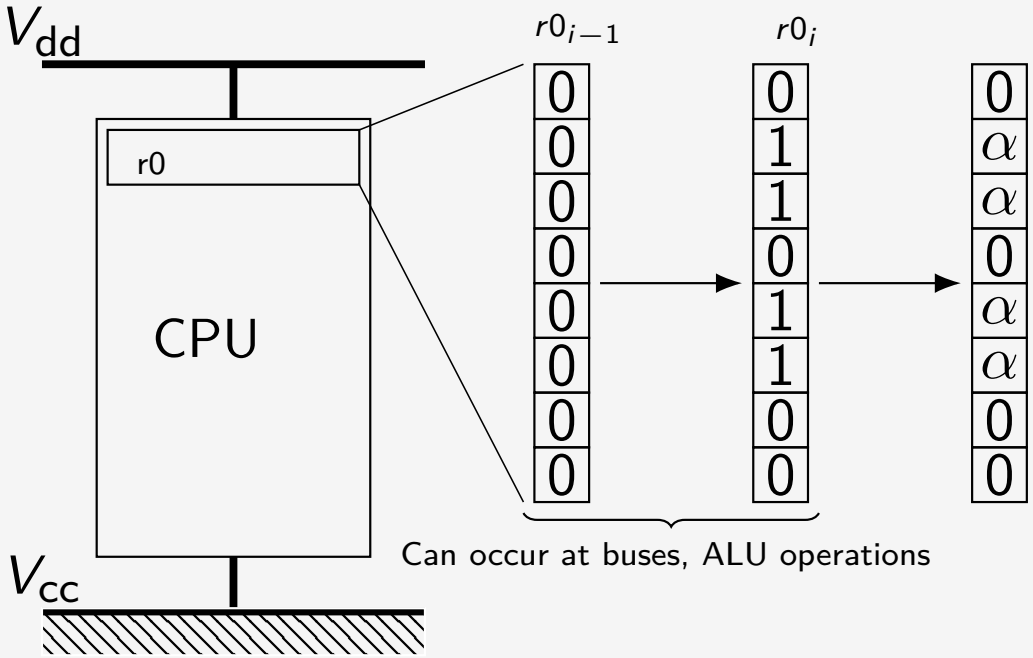
- Hamming distance denotes the number of different bits between the previous and the new register values: $\mathrm{HD}(r_{i-1}, r_i)$
- $i(0x00 \rightarrow 0x6C) \approx \mathrm{HD}(0x00, 0x6C) \cdot i(0 \rightarrow 1) + \sigma = 4 \cdot \alpha + \sigma$
- Hamming distance model is one of the best current drain approximation in side-channel attacks

- Hamming distance denotes the number of different bits between the previous and the new register values: $\mathrm{HD}(r_{i-1}, r_i)$
- $i(0x00 \rightarrow 0x6C) \approx \mathrm{HD}(0x00, 0x6C) \cdot i(0 \rightarrow 1) + \sigma = 4 \cdot \alpha + \sigma$
- Hamming distance model is one of the best current drain approximation in side-channel attacks
- Hamming distance model is practically confirmed by many side-channel attacks



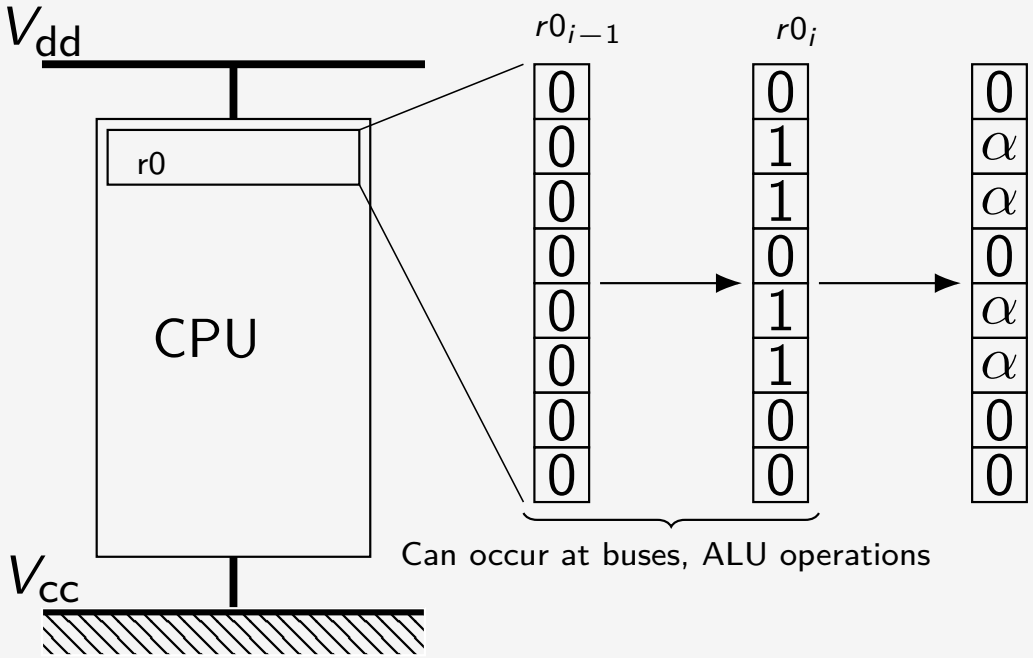
- $i(0x00 \rightarrow 0x21) = \text{HD}(0x00, 0x21) \cdot \alpha + \sigma = 2 \cdot \alpha + \sigma$
- $i(0x00 \rightarrow 0x6C) = \text{HD}(0x00, 0x6C) \cdot \alpha + \sigma = 4 \cdot \alpha + \sigma$
- $i(0x00 \rightarrow 0xE7) = \text{HD}(0x00, 0xE7) \cdot \alpha + \sigma = 6 \cdot \alpha + \sigma$
- $i(0x00 \rightarrow 0xFF) = \text{HD}(0x00, 0xFF) \cdot \alpha + \sigma = 8 \cdot \alpha + \sigma$

Power consumption of a register change



- Although previous register value can be different from $0x00$, this change $0x00 \rightarrow 0x6C$ can still be observed

Power consumption of a register change



- Although previous register value can be different from $0x00$, this change $0x00 \rightarrow 0x6C$ can still be observed
- The reason is that the register value is transmitted over buses (or written to other registers of ALU or other IPs)

Questions?

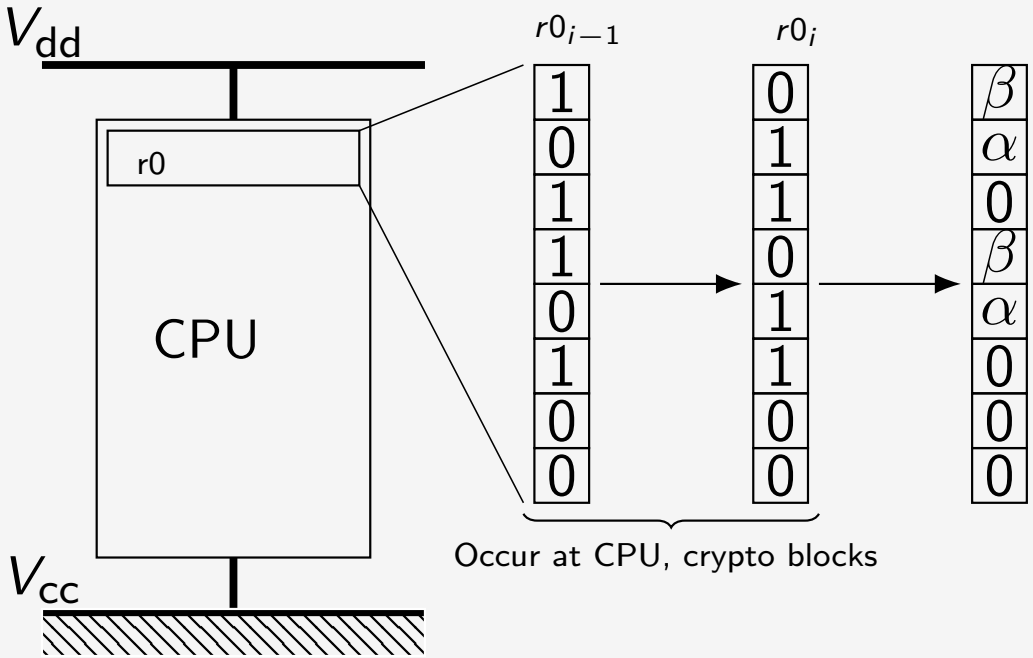
Questions?

- Consider 8-bit register. How many different values Hamming distance model can have: $\mathrm{HD}(r_{i-1}, r_i) = 0, 1, 2...$?

Questions?

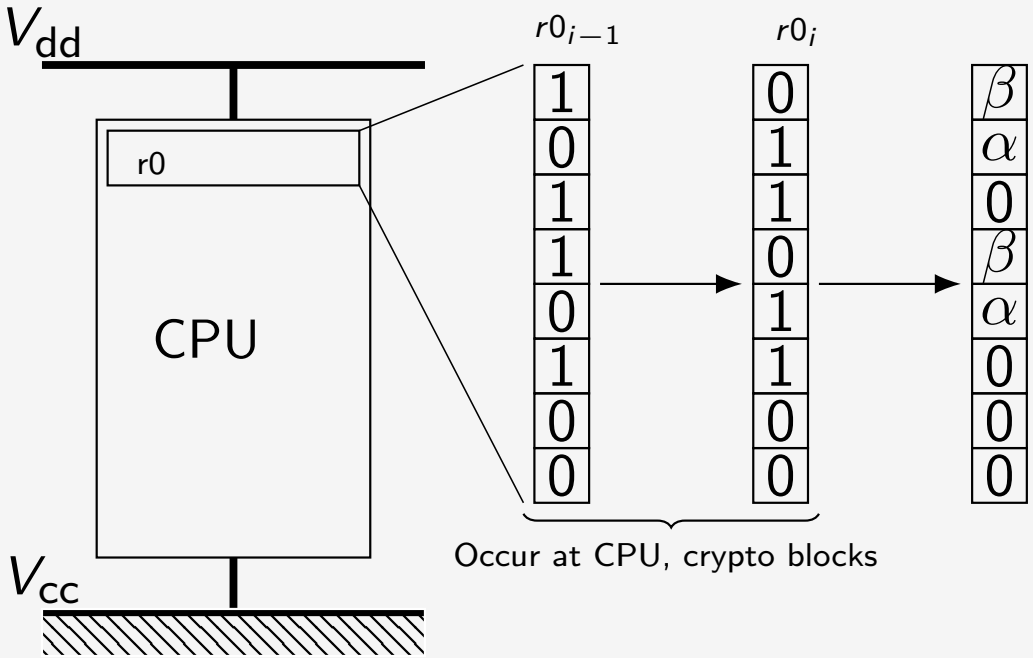
- Consider 8-bit register. How many different values Hamming distance model can have: $\mathrm{HD}(r_{i-1}, r_i) = 0, 1, 2...$?
- The same situation for 128-bit register (hardware implementation of AES): $\mathrm{HD}(r_{i-1}, r_i) = 0, 1, 2...$?

Power consumption of a register change



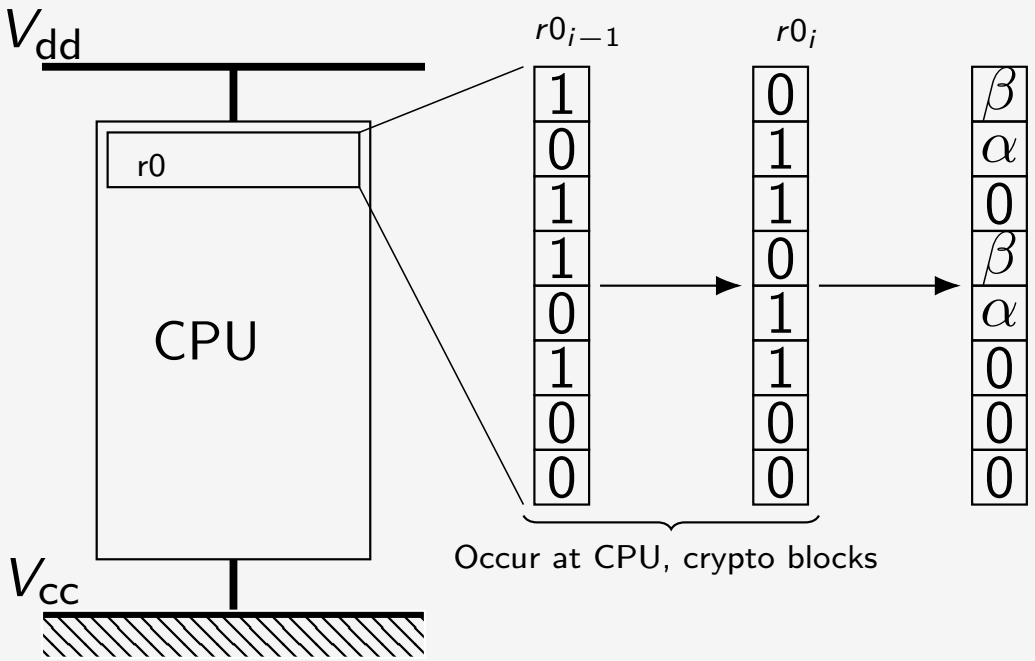
- If a previous register value is different from 0x00, the current drain can be still modelled

Power consumption of a register change



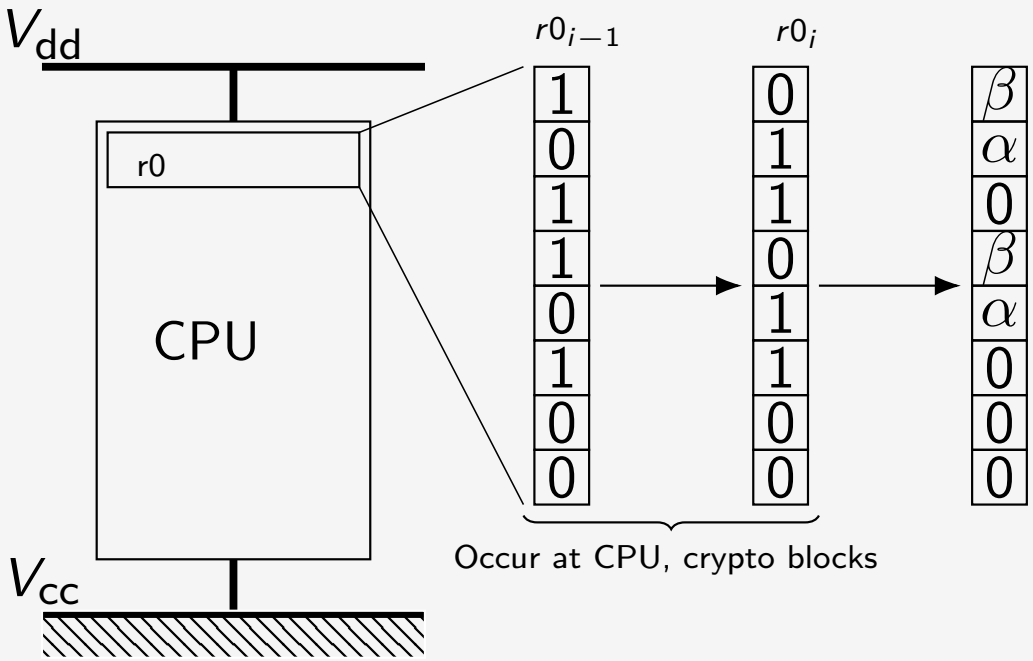
- If a previous register value is different from 0x00, the current drain can be still modelled
- Assume current drains: $i(0 \rightarrow 1) = \alpha$, and $i(1 \rightarrow 0) = \beta$
- The current is never balanced: $\alpha \neq \beta, \beta = \alpha + \delta$

Power consumption of a register change



- $i(0xD4 \rightarrow 0x6C) = 2 \cdot \alpha + 2 \cdot \beta + \sigma$

Power consumption of a register change



- $i(0xD4 \rightarrow 0x6C) = 2 \cdot \alpha + 2 \cdot \beta + \sigma$
- Side-channel attack people were lasy... so they assume that $|\alpha| >> |\delta|$ and $|\beta| >> |\delta|$, so they treat $|\delta|$ as colored noise
- $i(0xD4 \rightarrow 0x6C) = 2 \cdot \alpha + 2 \cdot \beta + \sigma = 2 \cdot \alpha + 2 \cdot (\alpha + \delta) + \sigma = 4 \cdot \alpha + (2 \cdot \delta + \sigma) = 4 \cdot \alpha + \sigma' = \mathrm{HD}(0xD4, 0x6C) \cdot \alpha + \sigma'$

- If a previous r_{i-1} and a new r_i register values are known, side-channel attacks use Hamming Distance model
- $i(r_{i-1} \rightarrow r_i) = \mathrm{HD}(r_{i-1}, r_i) \cdot \alpha + \sigma$

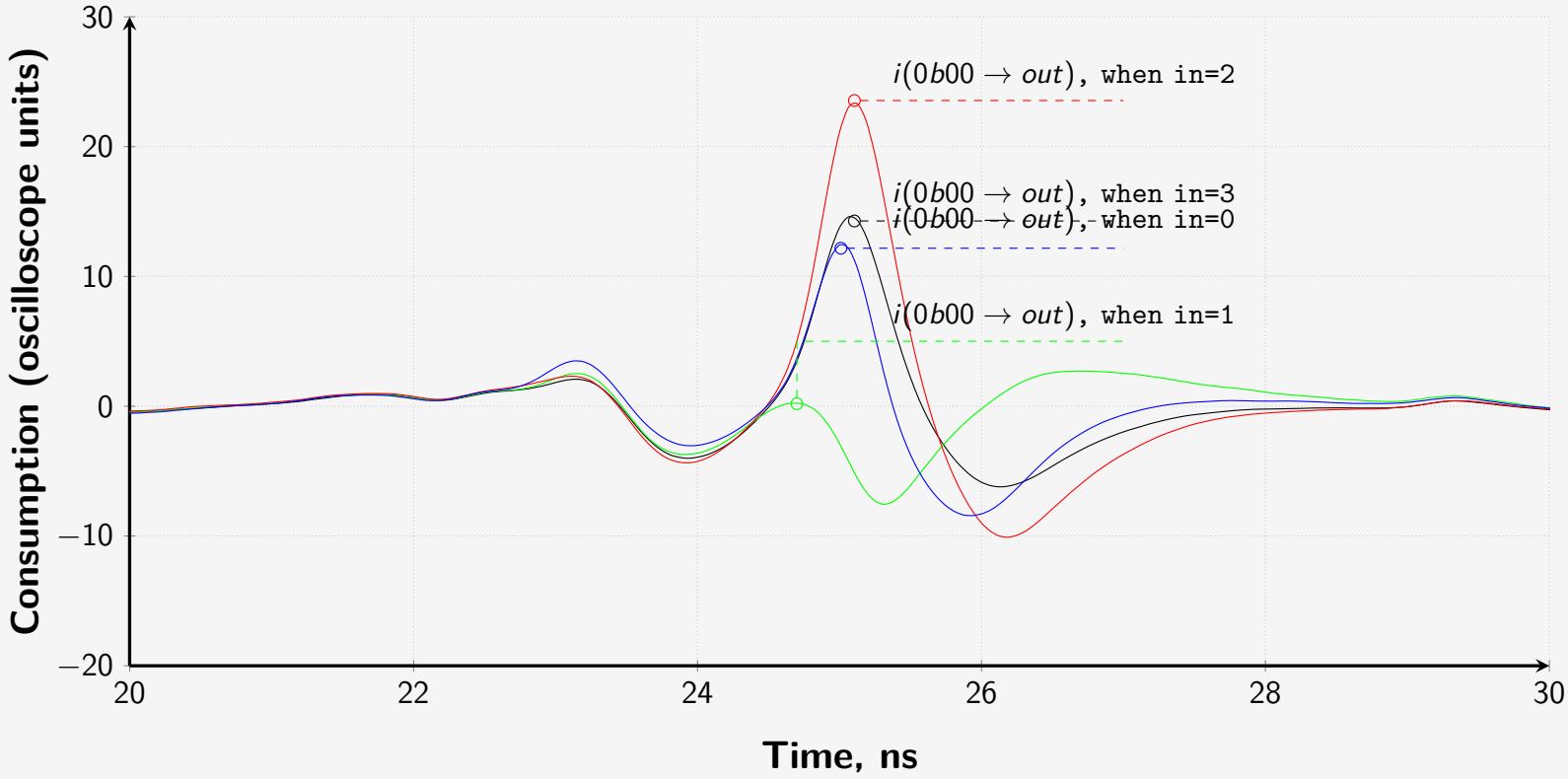
- If a previous r_{i-1} and a new r_i register values are known, side-channel attacks use Hamming Distance model
- $i(r_{i-1} \rightarrow r_i) = \mathrm{HD}(r_{i-1}, r_i) \cdot \alpha + \sigma$
- Yes, sometime people try to characterize δ and use more precise models with weights:
- $\alpha \neq \beta, \beta = 1.1 \cdot \alpha$

- If a previous r_{i-1} and a new r_i register values are known, side-channel attacks use Hamming Distance model
- $i(r_{i-1} \rightarrow r_i) = \mathrm{HD}(r_{i-1}, r_i) \cdot \alpha + \sigma$
- Yes, sometime people try to characterize δ and use more precise models with weights:
- $\alpha \neq \beta, \beta = 1.1 \cdot \alpha$
- In that case our previous Hamming distance become equal:
- $i(0xD4 \rightarrow 0x6C) = 2 \cdot \alpha + 2 \cdot \beta + \sigma = 2 \cdot \alpha + 2 \cdot 1.1 \cdot \alpha + \sigma = 4.2 \cdot \alpha + \sigma$

- If a previous r_{i-1} and a new r_i register values are known, side-channel attacks use Hamming Distance model
- $i(r_{i-1} \rightarrow r_i) = \mathrm{HD}(r_{i-1}, r_i) \cdot \alpha + \sigma$
- Yes, sometime people try to characterize δ and use more precise models with weights:
- $\alpha \neq \beta, \beta = 1.1 \cdot \alpha$
- In that case our previous Hamming distance become equal:
- $i(0xD4 \rightarrow 0x6C) = 2 \cdot \alpha + 2 \cdot \beta + \sigma = 2 \cdot \alpha + 2 \cdot 1.1 \cdot \alpha + \sigma = 4.2 \cdot \alpha + \sigma$
- Characterisation of the coefficient θ: $\beta = \theta \cdot \alpha$ takes time, plus it might depend on the current operation

- If a previous r_{i-1} and a new r_i register values are known, side-channel attacks use Hamming Distance model
- $i(r_{i-1} \rightarrow r_i) = \mathrm{HD}(r_{i-1}, r_i) \cdot \alpha + \sigma$
- Yes, sometime people try to characterize δ and use more precise models with weights:
- $\alpha \neq \beta, \beta = 1.1 \cdot \alpha$
- In that case our previous Hamming distance become equal:
- $i(0xD4 \rightarrow 0x6C) = 2 \cdot \alpha + 2 \cdot \beta + \sigma = 2 \cdot \alpha + 2 \cdot 1.1 \cdot \alpha + \sigma = 4.2 \cdot \alpha + \sigma$
- Characterisation of the coefficient θ: $\beta = \theta \cdot \alpha$ takes time, plus it might depend on the current operation
- Nevertheless, if you get the key with the Hamming distance model you don't need to care of deeper characterisation

Pen and pensil excersise

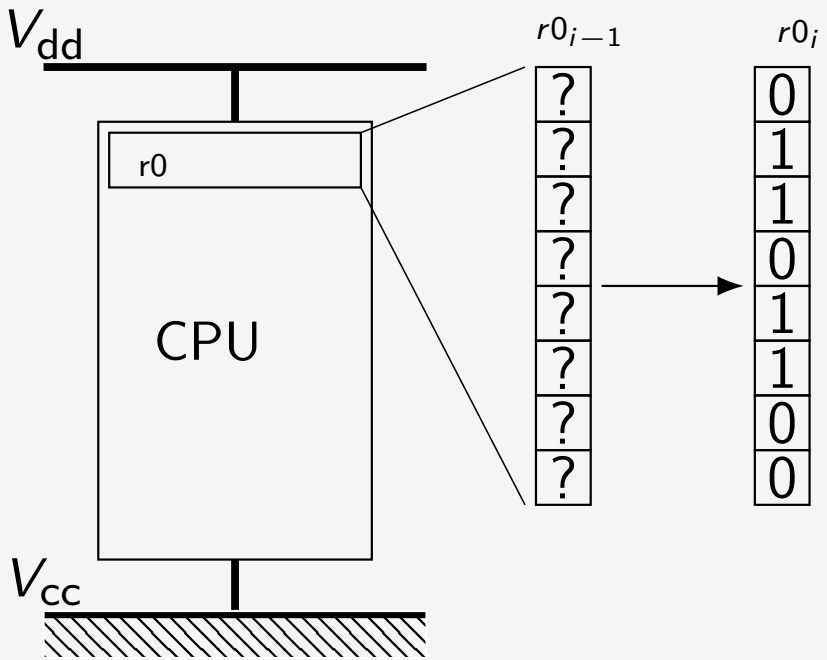


- The CPU computes $in \oplus key = out$ on 2-bit registers *in* and *out*
- Input values are known to an attacker (0,1,2,3)
- Output values are not known to an attacker; but for each *in* value he measured power consumption of the *out* register transition $0b00 \rightarrow out$
- You need to find the *key* (by hands)

Hamming distance key points

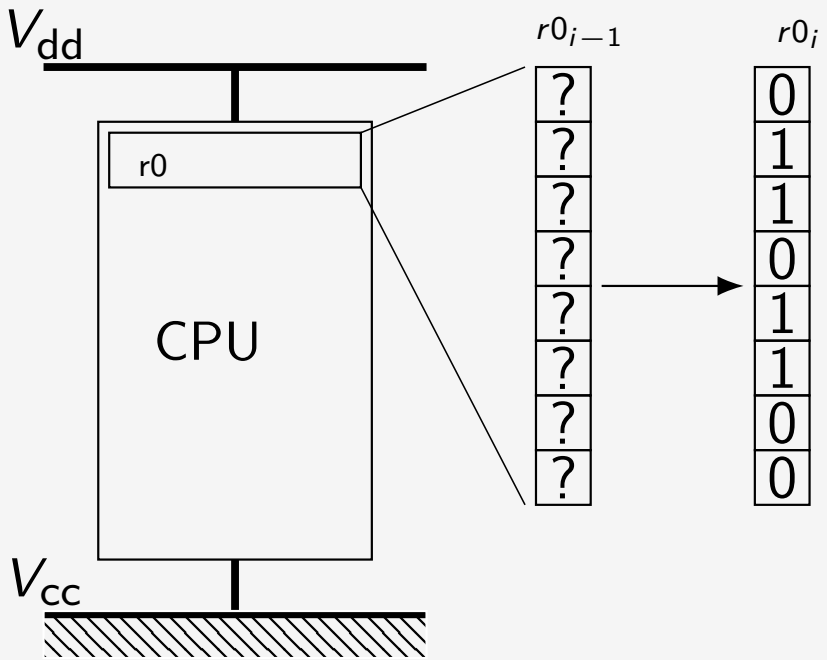
- By looking into power consumption Hamming distance model can be distinguished: $i(r_{i-1} \rightarrow r_i) = \mathrm{HD}(r_{i-1}, r_i) \cdot \alpha + \sigma$
- The "pen and pensil" attack is kind of side-channel attack
- Real side-channel attacks use slightly more advanced statistics than visual observations

Power consumption of a register change



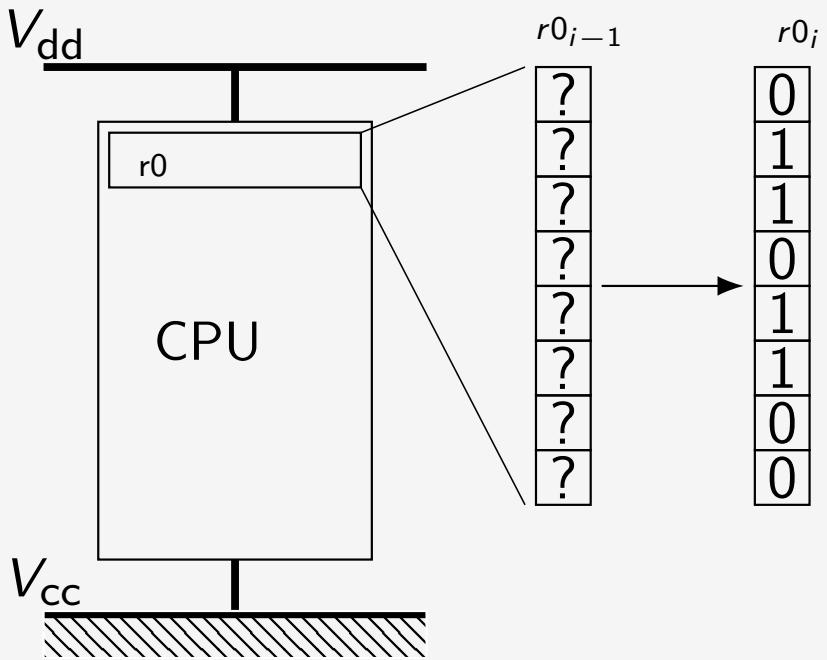
- What if you don't know the previous register value?

Power consumption of a register change



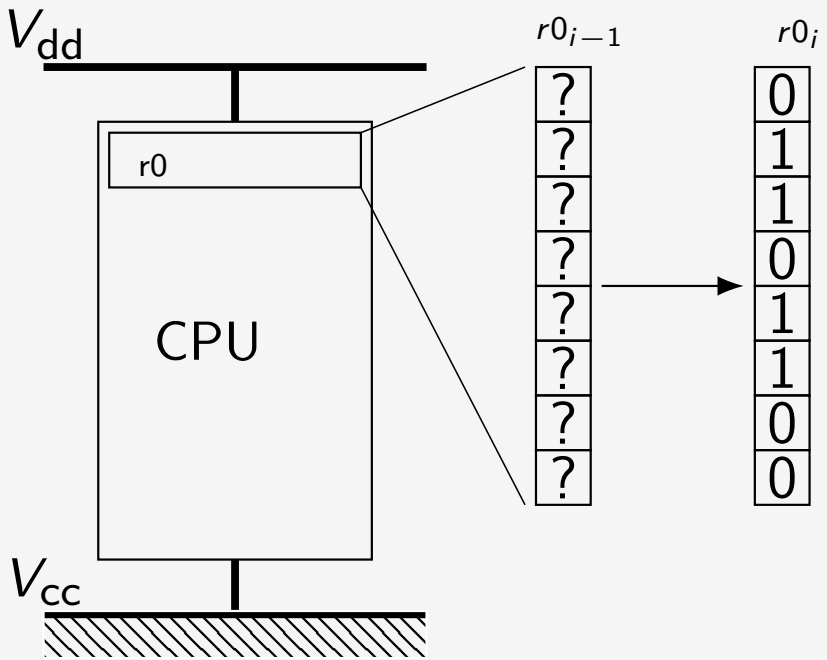
- What if you don't know the previous register value?
- One approach is to use Hamming distance model when a previous register value is 0x00:
  $i(0x00 \rightarrow r_i) = \mathrm{HD}(0x00, r_i) \cdot \alpha + \sigma = \mathrm{HW}(r_i) \cdot \alpha + \sigma$

Power consumption of a register change



- What if you don't know the previous register value?
- One approach is to use Hamming distance model when a previous register value is 0x00:
  $i(0x00 \rightarrow r_i) = \mathrm{HD}(0x00, r_i) \cdot \alpha + \sigma = \mathrm{HW}(r_i) \cdot \alpha + \sigma$
- This model is called Hamming weight model

Power consumption of a register change



- What if you don't know the previous register value?
- One approach is to use Hamming distance model when a previous register value is 0x00:
  $i(0x00 \rightarrow r_i) = \mathrm{HD}(0x00, r_i) \cdot \alpha + \sigma = \mathrm{HW}(r_i) \cdot \alpha + \sigma$
- This model is called Hamming weight model
- Hamming weight model also works when the previous register value r_{i-1} is random

- Why Hamming weight model works even if the previous register value r_{i-1} is random

- Why Hamming weight model works even if the previous register value r_{i-1} is random
- Consider a situation of 1 single flip-flop (one bit)

- Why Hamming weight model works even if the previous register value r_{i-1} is random
- Consider a situation of 1 single flip-flop (one bit)
- Many measurements of the operations $i(? \rightarrow 0)$ and $i(? \rightarrow 1)$ are performed

- Why Hamming weight model works even if the previous register value r_{i-1} is random
- Consider a situation of 1 single flip-flop (one bit)
- Many measurements of the operations $i(? \rightarrow 0)$ and $i(? \rightarrow 1)$ are performed
- An attacker only knowns that previous flip-flop value ? is random (however he knows the new register value)

- Let us average the power measurements for $i(? \to 0)$ and $i(? \to 1)$

$$\sum_0^n \frac{i(? \to 0)}{n} = \sum_0^{n/2} \frac{i(0 \to 0)}{n/2} + \sum_0^{n/2} \frac{i(1 \to 0)}{n/2} + \sum_0^n \frac{\sigma}{n} = 0 + \alpha + const$$

$$\sum_0^n \frac{i(? \to 1)}{n} = \sum_0^{n/2} \frac{i(0 \to 1)}{n/2} + \sum_0^{n/2} \frac{i(1 \to 1)}{n/2} + \sum_0^n \frac{\sigma}{n} =$$

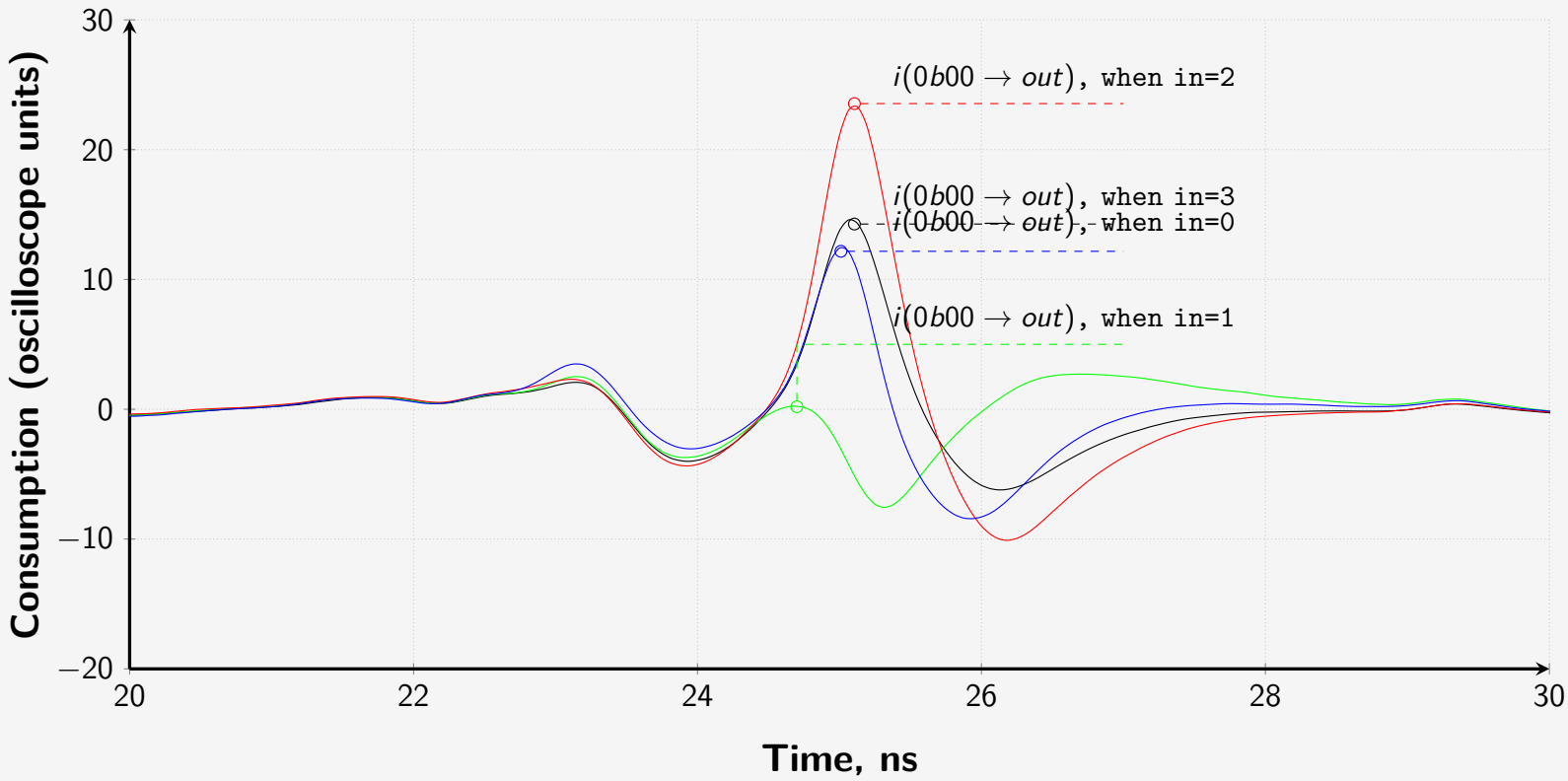
$$= \beta + 0 + const = \alpha + \delta + 0 + const$$

- Let us average the power measurements for $i(? \to 0)$ and $i(? \to 1)$

$$\sum_0^n \frac{i(? \to 0)}{n} = \sum_0^{n/2} \frac{i(0 \to 0)}{n/2} + \sum_0^{n/2} \frac{i(1 \to 0)}{n/2} + \sum_0^n \frac{\sigma}{n} = 0 + \alpha + const$$

$$\sum_0^n \frac{i(? \to 1)}{n} = \sum_0^{n/2} \frac{i(0 \to 1)}{n/2} + \sum_0^{n/2} \frac{i(1 \to 1)}{n/2} + \sum_0^n \frac{\sigma}{n} =$$

$$= \beta + 0 + const = \alpha + \delta + 0 + const$$



- Remember this excersise?

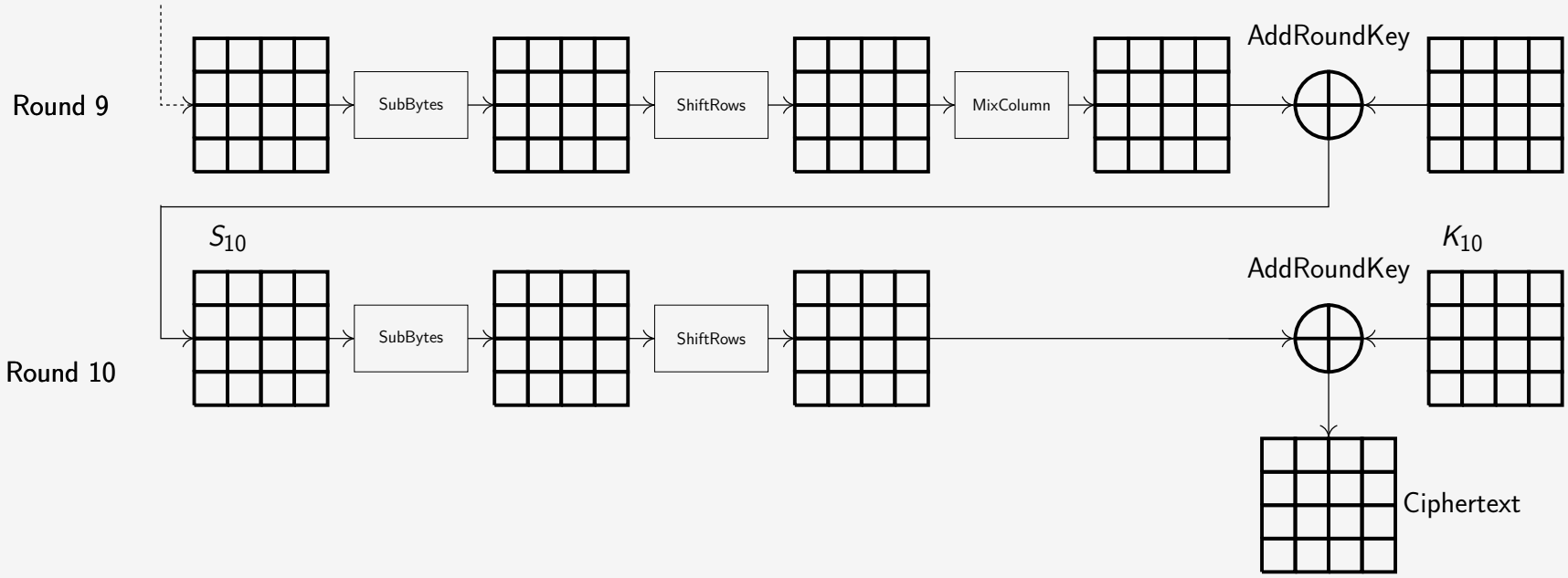
- So far we have considered three pieces of the puzzle:
  - CMOS power consumption to understand the nature of leakage
  - Law of large numbers to deal with noise
  - Power consumption models
- This is normal if you did not catch any of the topics

- So far we have considered three pieces of the puzzle:
  - CMOS power consumption to understand the nature of leakage
  - Law of large numbers to deal with noise
  - Power consumption models
- This is normal if you did not catch any of the topics
- We will consider them by excersises

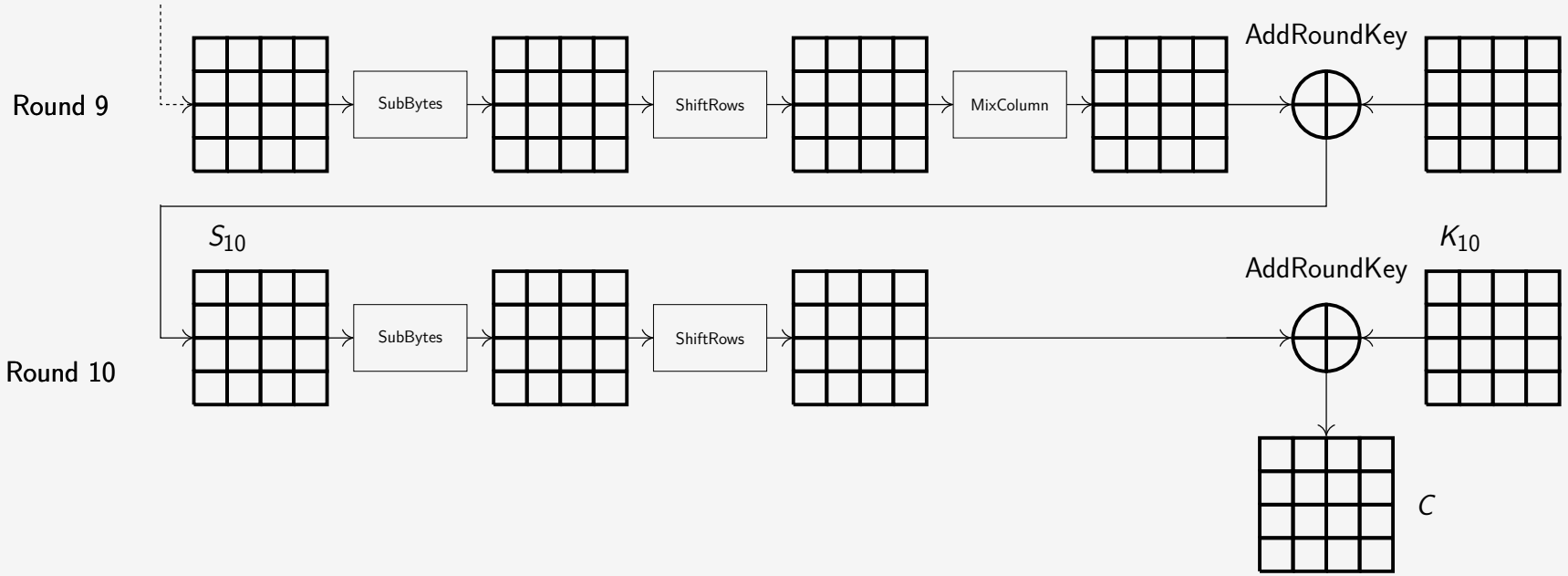
- So far we have considered three pieces of the puzzle:
  - CMOS power consumption to understand the nature of leakage
  - Law of large numbers to deal with noise
  - Power consumption models
- This is normal if you did not catch any of the topics
- We will consider them by excersises
- All the excersises will be based on AES examples

AES Example

Advanced Encryption Standard algorithm (AES-128)

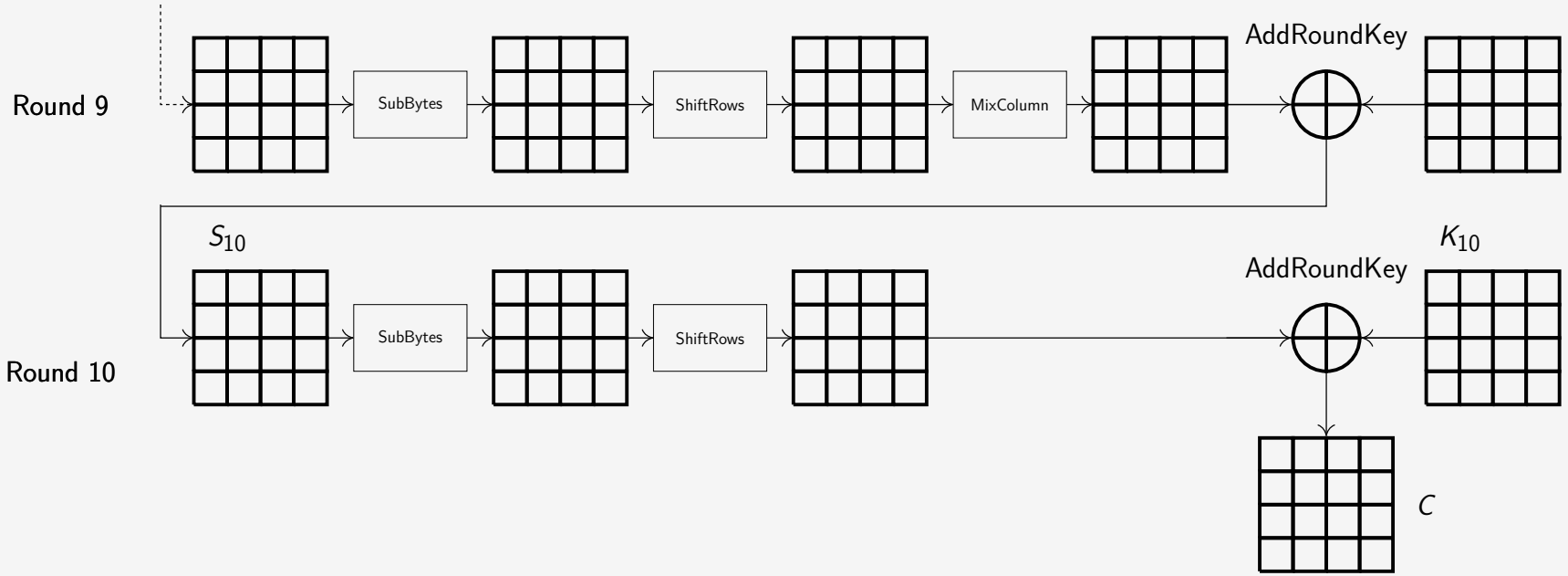


AES-128 Assignment 1



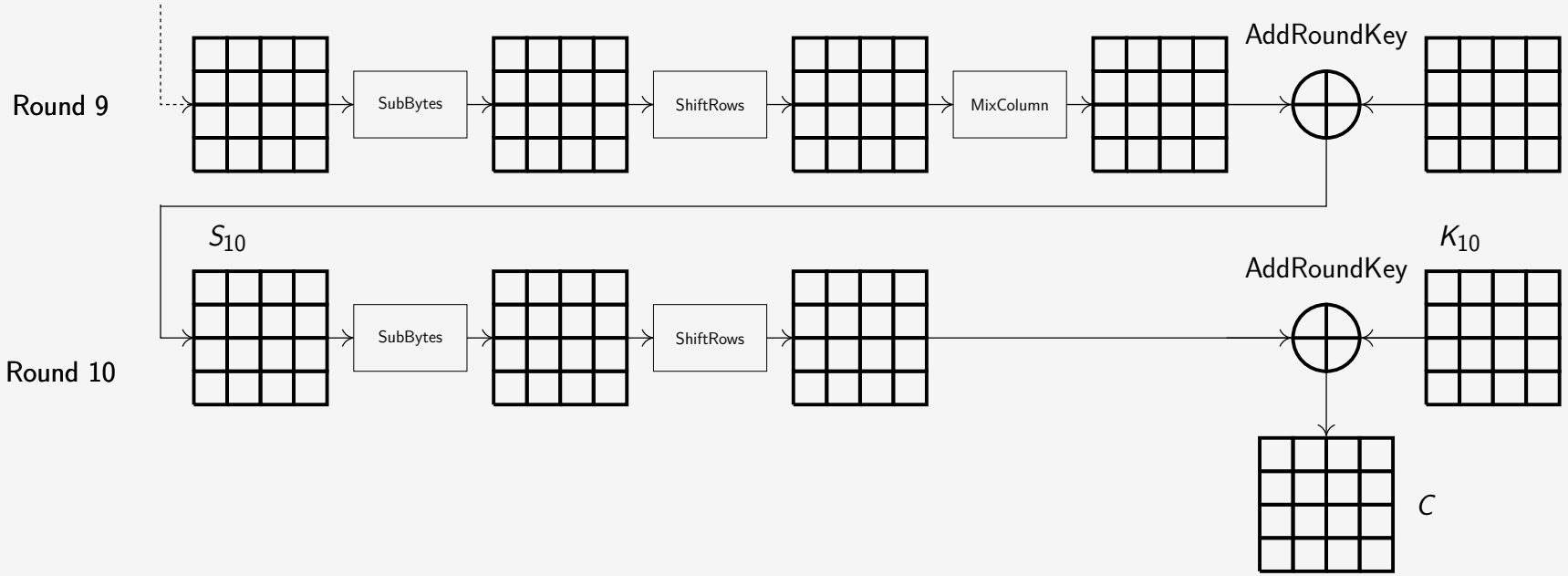
- A developer left a debug option that prints out the beginning of the state 10 and a ciphertext. What can be wrong with that?

AES-128 Assignment 1



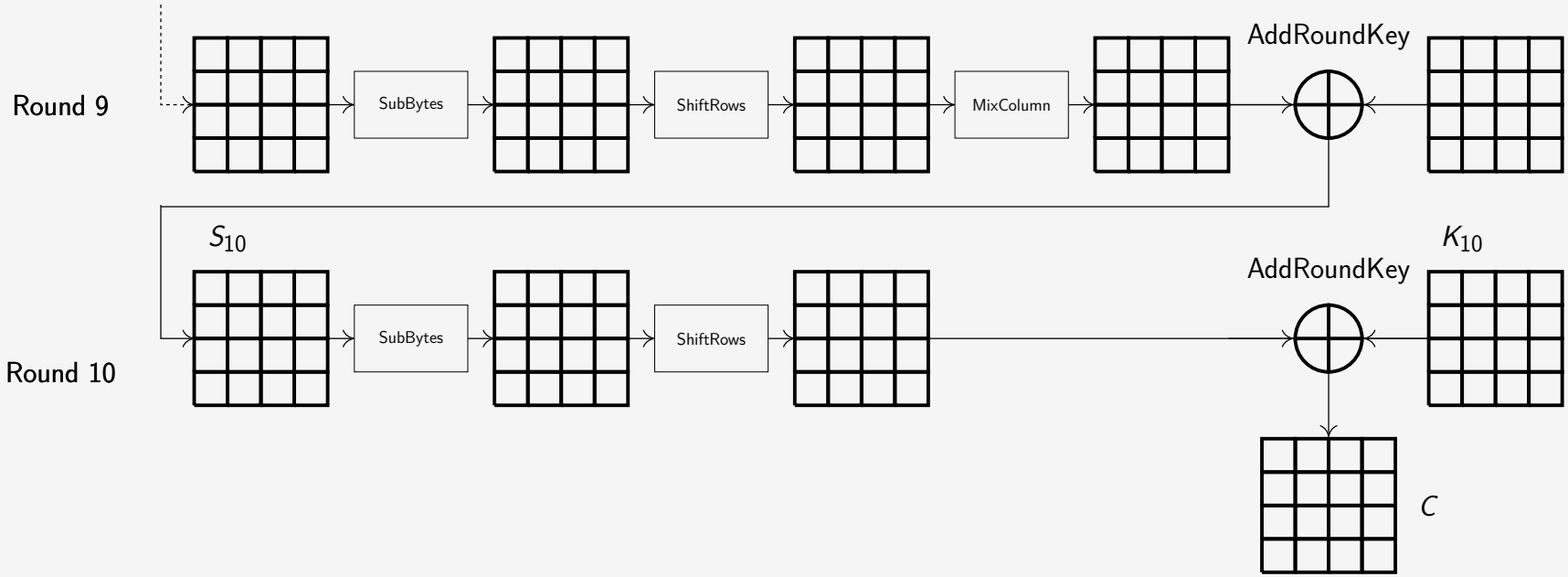
- A developer left a debug option that prints out the beginning of the state 10 and a ciphertext. What can be wrong with that?
- Find K_{10} and then get the master key using the provided code

AES-128 Assignment 1



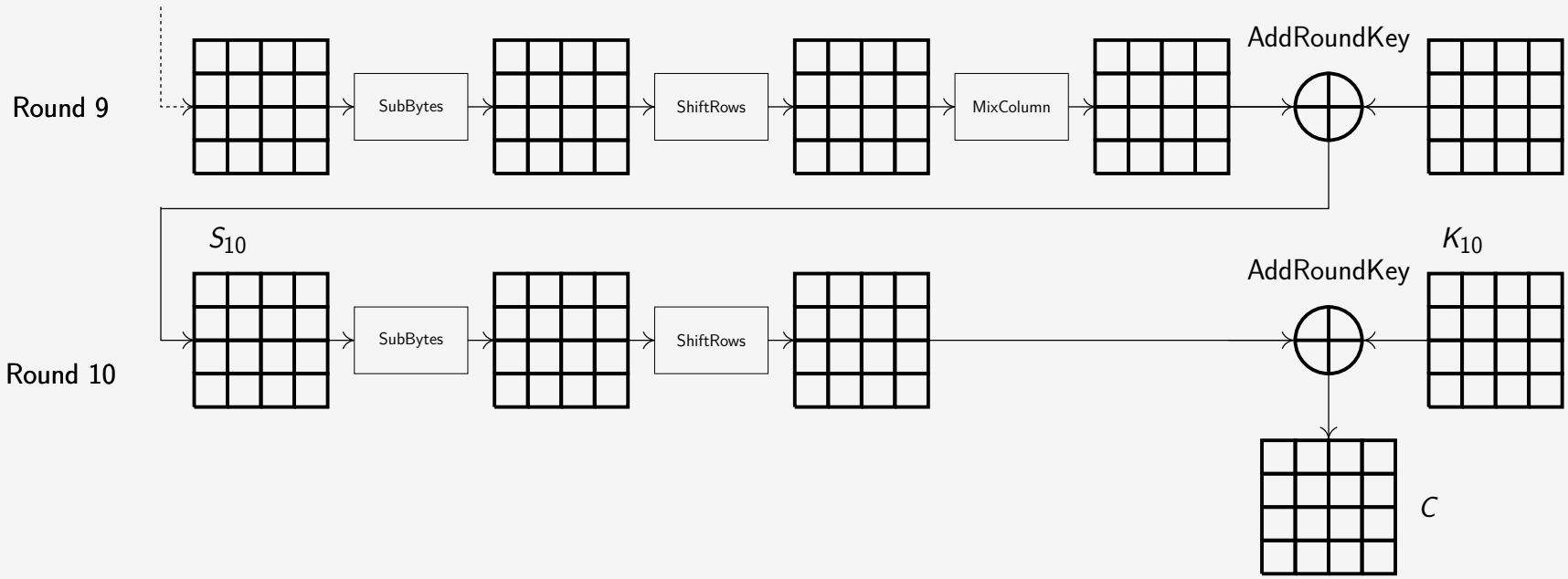
- A developer left a debug option that prints out the beginning of the state 10 and a ciphertext. What can be wrong with that?
- Find K_{10} and then get the master key using the provided code
- $C = ShiftRows(SubBytes[S_{10}]) \oplus K_{10}$

AES-128 Assignment 1



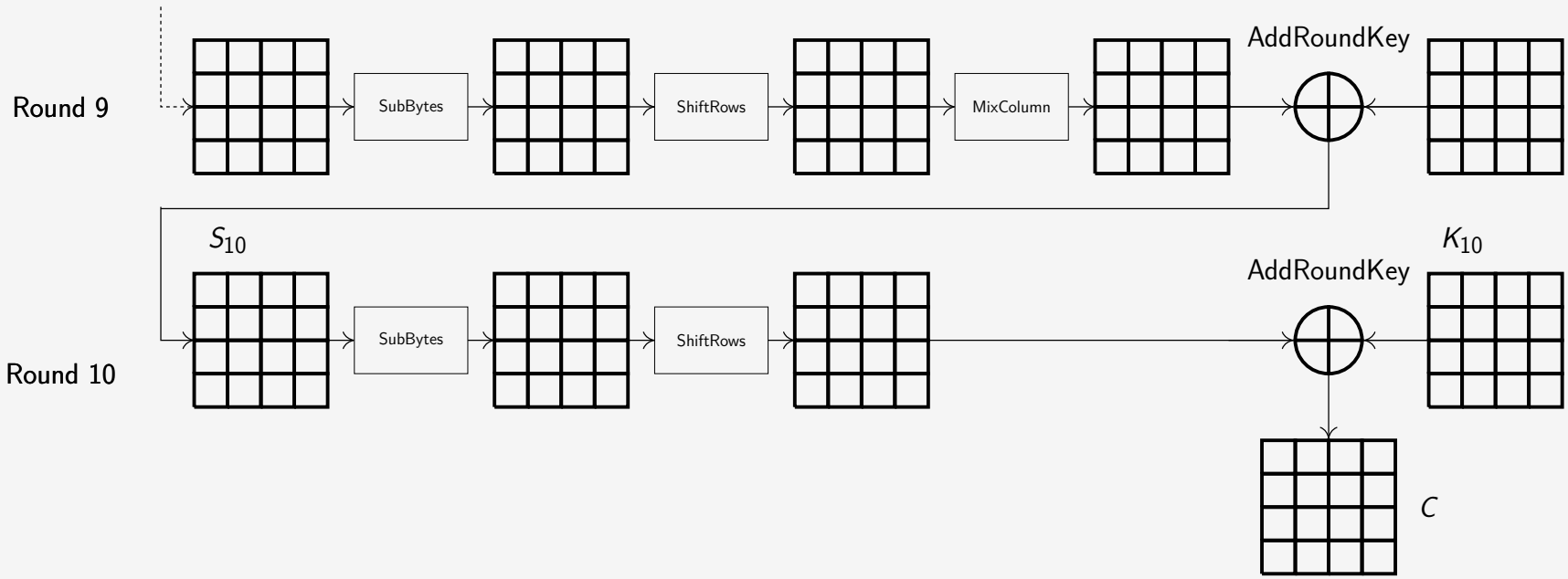
- A developer left a debug option that prints out the beginning of the state 10 and a ciphertext. What can be wrong with that?
- Find K_{10} and then get the master key using the provided code
- $C = ShiftRows(SubBytes[S_{10}]) \oplus K_{10}$
- $K_{10} = C \oplus ShiftRows(SubBytes[S_{10}])$

AES-128 Assignment 2



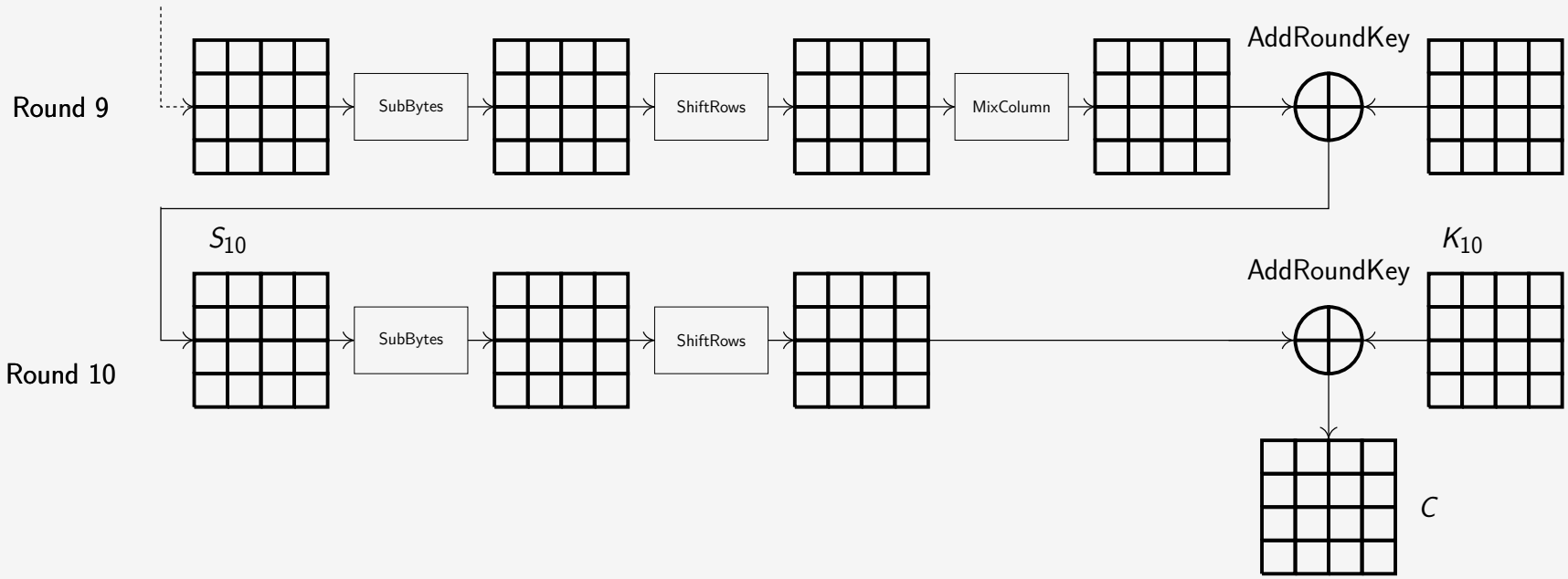
- This time a developer left a debug option that prints out Hamming weights of bytes for the beginning of the 10th round state and a ciphertext. What can be wrong with that?

AES-128 Assignment 2



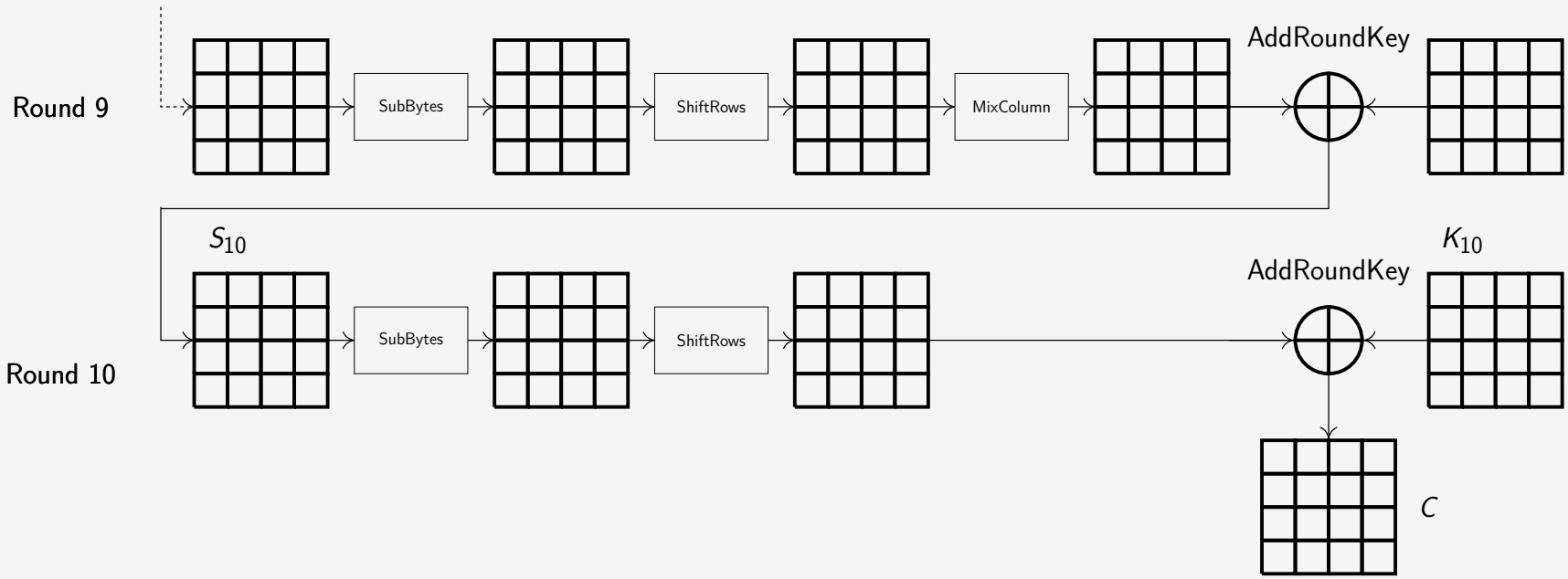
- This time a developer left a debug option that prints out Hamming weights of bytes for the beginning of the 10th round state and a ciphertext. What can be wrong with that?
- Find K_{10} and then get the master key using the provided code

AES-128 Assignment 2



- This time a developer left a debug option that prints out Hamming weights of bytes for the beginning of the 10th round state and a ciphertext. What can be wrong with that?
- Find K_{10} and then get the master key using the provided code
- $C = ShiftRows(SubBytes[S_{10}]) \oplus K_{10}$

AES-128 Assignment 2



- This time a developer left a debug option that prints out Hamming weights of bytes for the beginning of the 10th round state and a ciphertext. What can be wrong with that?
- Find K_{10} and then get the master key using the provided code
- $C = ShiftRows(SubBytes[S_{10}]) \oplus K_{10}$
- $HW(S_{10}) = HW(InvSubBytes[InvShiftRows(K_{10} \oplus C)])$

Thank you!

Roman Korkikian, Nicolas Oberli

Side-channels and Fault Attacks
February 23rd, 2023 - June 29th, 2023