Physical attacks on secure electronic devices

Brice Colombier

March 2, 2023

Who am I?

Brice Colombier
Associate professor at Université Jean Monnet

Teaching

GEII department

Research

Secure Embedded Systems and Hardware Architectures

What does the title mean?

Physical attacks on secure electronic devices

What does the title mean?

Physical attacks on secure electronic devices

Electronic devices:

- light bulb
- camera
- electric bike
- smartphone
- headphones
- computer
- credit card
- blender

What does the title mean?

Physical attacks on secure electronic devices

Electronic devices:

- light bulb ✗
- camera ✗
- electric bike ✗
- smartphone ✓
- headphones ✗
- computer ✓
- credit card ✓
- blender ✗

Only few of them handle **sensitive information**.
Only few of them must be **secure**.
However, their number is **increasing**...

What does the title mean?

Physical attacks on secure electronic devices

Electronic devices:

- light bulb ✗
- camera ✗
- electric bike ✗
- smartphone ✓
- headphones ✗
- computer ✓
- credit card ✓
- blender ✗

Only few of them handle **sensitive information**.
Only few of them must be **secure**.
However, their number is **increasing**...

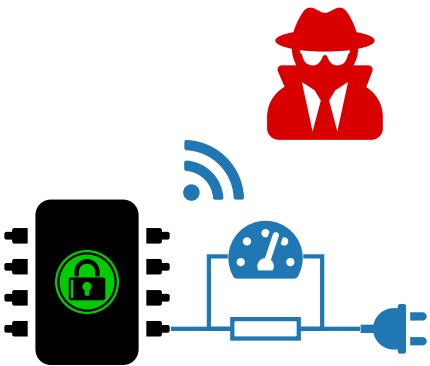
What does the title mean?

Physical attacks on secure electronic devices

Electronic devices:

- light bulb ✗
- camera ✗
- electric bike ✗
- smartphone ✓
- headphones ✗
- computer ✓
- credit card ✓
- blender ✗

Only few of them handle **sensitive information**.
Only few of them must be **secure**.
However, their number is **increasing**...



Side-channel attacks

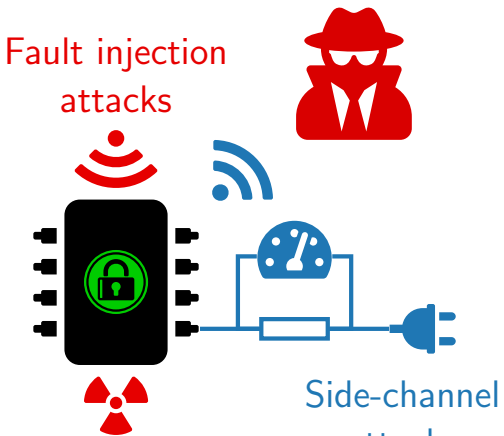
What does the title mean?

Physical attacks on secure electronic devices

Electronic devices:

- light bulb ✗
- camera ✗
- electric bike ✗
- smartphone ✓
- headphones ✗
- computer ✓
- credit card ✓
- blender ✗

Only few of them handle **sensitive information**.
Only few of them must be **secure**.
However, their number is **increasing**...



Side-channel attacks

Definition: measure a **physical quantity** while the device handles **secret information**.

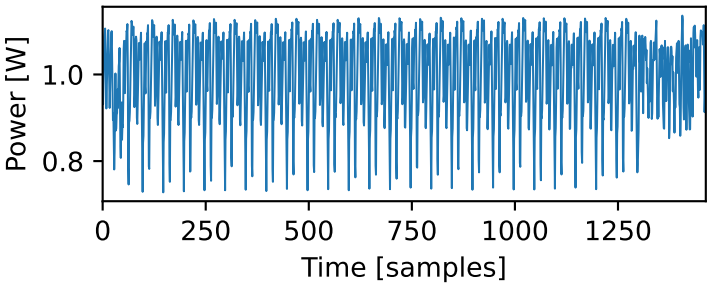
- Power consumption
- Electromagnetic radiation
- Sound
- Photons

Side-channel attacks

Definition: measure a **physical quantity** while the device handles **secret information**.

- Power consumption
- Electromagnetic radiation
- Sound
- Photons

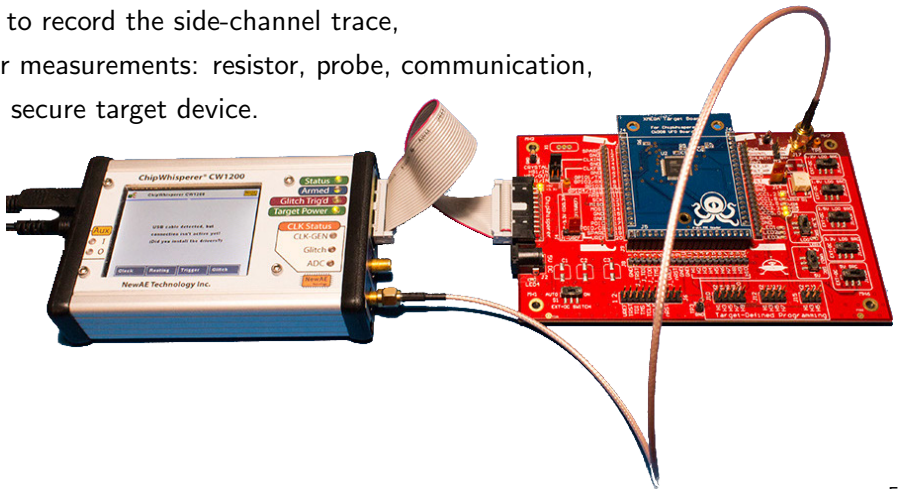
Side-channel trace:



Hardware platform

ChipWhisperer `https://www.newae.com/chipwhisperer`

- An "**oscilloscope**" to record the side-channel trace,
- A **motherboard** for measurements: resistor, probe, communication,
- A **daughterboard**: secure target device.



Target code

Credit card example: check if a 4-digit PIN entered by the user is correct.
If we want to **brute-force** it, how many trials will we need on average?

Target code

Credit card example: check if a 4-digit PIN entered by the user is correct.
If we want to **brute-force** it, how many trials will we need on average? $\frac{10^4}{2} = 5000$

Target code

Credit card example: check if a 4-digit PIN entered by the user is correct.
If we want to **brute-force** it, how many trials will we need on average? $\frac{10^4}{2} = 5000$

```
1  def verifyPIN(user_PIN):
2      reference_PIN = "1234"
3      for i in [0, 1, 2, 3]:
4          if user_PIN[i] != reference_PIN[i]:
5              return False
6      return True
7
8  print("Enter your PIN:")
9  user_PIN = input()
10 if VerifyPIN(user_PIN):
11     print("CORRECT!")
12 else:
13     print("WRONG!")
```

Target code

Credit card example: check if a 4-digit PIN entered by the user is correct.
If we want to **brute-force** it, how many trials will we need on average? $\frac{10^4}{2} = 5000$

```
1  def verifyPIN(user_PIN):
2      reference_PIN = "1234"
3      for i in [0, 1, 2, 3]:
4          if user_PIN[i] != reference_PIN[i]:
5              return False
6      return True
7
8  print("Enter your PIN:")
9  user_PIN = input()
10 if VerifyPIN(user_PIN):
11     print("CORRECT!")
12 else:
13     print("WRONG!")
```

```
> python3 verifyPIN.py
Enter your PIN:
5874
WRONG!

> python3 verifyPIN.py
Enter your PIN:
1234
CORRECT!
```

Rewritten in C to run on the target micro-controller

Demo time

Target code with countermeasure

Countermeasure: **defense** against a **known** attack.

```
1  def verifyPIN_secure(user_PIN):
2      reference_PIN = "1234"
3      authenticated = True
4      for i in [0, 1, 2, 3]:
5          if user_PIN[i] != reference_PIN[i]:
6              authenticated = False
7      return authenticated
```

We want the implementation to be **constant-time**: the duration is the same for all input values.

Demo time again

Take-away messages

- An implementation might be **correct** but still not **secure**,
- Often, security is opposed to **optimisations**,
- It is a **cat-and-mouse** game between attackers and designers,

Take-away messages

- An implementation might be **correct** but still not **secure**,
- Often, security is opposed to **optimisations**,
- It is a **cat-and-mouse** game between attackers and designers,

Want to know more? `https://bcolombier.fr`

Take-away messages

- An implementation might be **correct** but still not **secure**,
- Often, security is opposed to **optimisations**,
- It is a **cat-and-mouse** game between attackers and designers,

Want to know more? `https://bcolombier.fr`

Questions