# Cryptographic Engineering

## Lecture 3: Advanced Side-channel analysis

## February 19, 2024

Assist. Prof. Guilherme Perin| Leiden

Universiteit Leiden

Bij ons leer je de wereld kennen

# Last week…

- Introduction to SCA
- Review on RSA and AES
- SCA on RSA (simple power analysis)
- SCA on AES (DPA and CPA)

# Agenda

- Why do we need advanced side-channel analysis?
- An overview of leakage models
- Supervised learning in side-channel analysis
  - Machine learning
  - Deep learning

- Take-away messages:
  - Advanced side-channel analysis can defeat strong protections.
  - Advanced side-channel analysis is more difficult to set up (several parameters)
  - The limit to evolve side-channel analysis is the limit of AI field.

# Why do we need advanced side-channel analysis?

- Simple Power Analysis (SPA), Differential Power Analysis (DPA) and Correlation Power Analysis (CPA) are side-channel methods that work well against crypto implementations **without** specific countermeasures.

- Since first SCA methods were published (1996-2000), countermeasures started to be considered for crypto algorithms in hardware and software.

- **Questions:**
  - **How to measure/quantity if a protection is enough?**
  - **How to estimate the attack capability against specific targets? How to make sure the attack is sufficient to assess the security of an implementation?**

- **The SCA community started to apply stronger SCA methods to estimate the worst-case security ("what can the strongest attacker achieve?").**

# Two main side-channel analysis branches

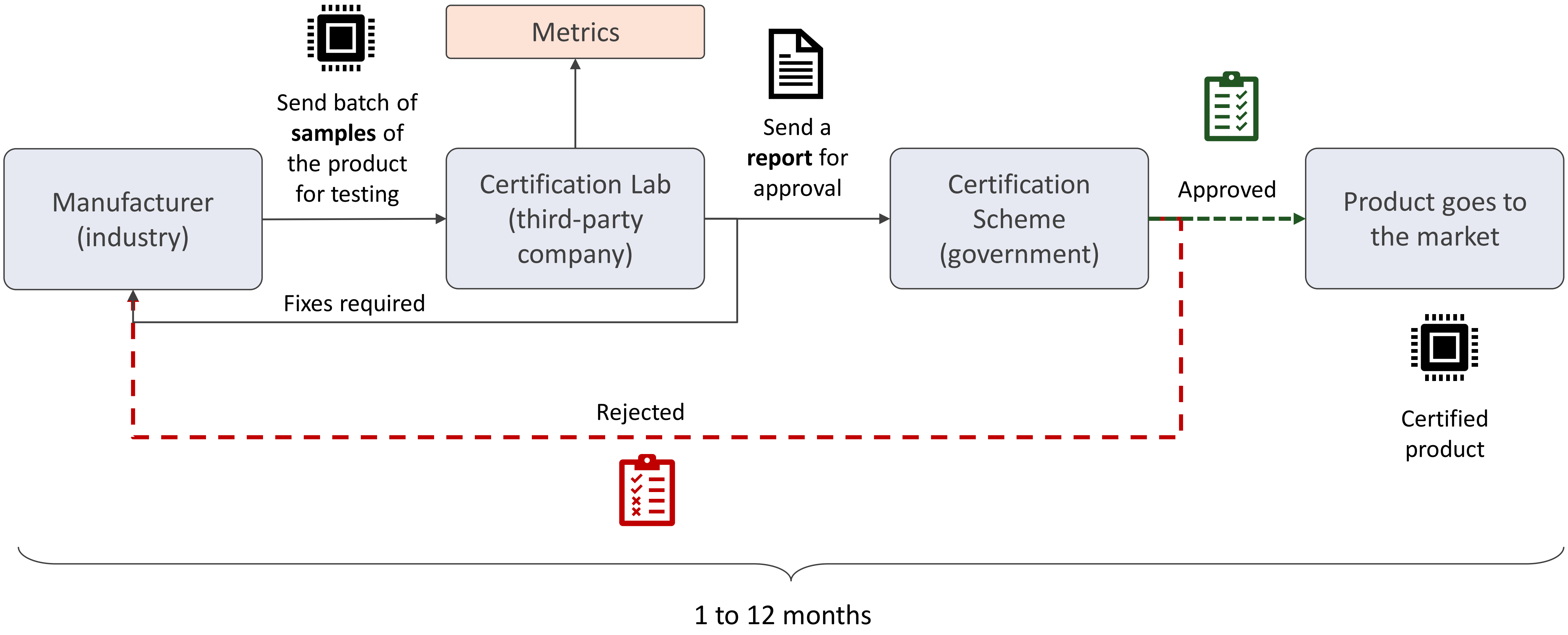
- **Direct attacks (a.k.a. non-profiled or unsupervised):**
  - **Real-world threats** (i.e., realistic)
  - Require very few assumptions about the target (basically being able to query encryptions/decryptions is enough)
  - Targets that resists these attacks (SPA, DPA, CPA) are assumed to be DPA-resistant.

- **Profiled attacks (a.k.a. supervised):**
  - **Much less realistic** (i.e. real attackers would very unlikely apply this type of method).
  - Require much more assumptions about the target (i.e., being able to set my own key).
  - Targets that resists these attacks are considered secure from the information-theoretical point-of-view.
  - Important techniques for manufacturers to answer the question: “is my implementation secure against the strongest possible adversary?”

---

- **Deep Learning attacks:**
  - Can be both: realistic and unrealistic.

# Security assessments



# Metrics

- **Number of measurements**
  - Needs to be enough to avoid **estimation errors.**
  - Some certification schemes require at least 2Million measurements.

- **Leakage model**
  - Needs to be a correct choice to avoid **assumption errors.**
  - Related to the concept of hypothetical power consumption.

- **Success rate of the attack**
  - Given the number of measurements and leakage model, how many key bits can be recovered (at least)?
  - Every certification scheme defines a maximum success rate to pass the test.

# Side-channel leakages

- Side-channel information refers to physical quantities that can be measured from a device.

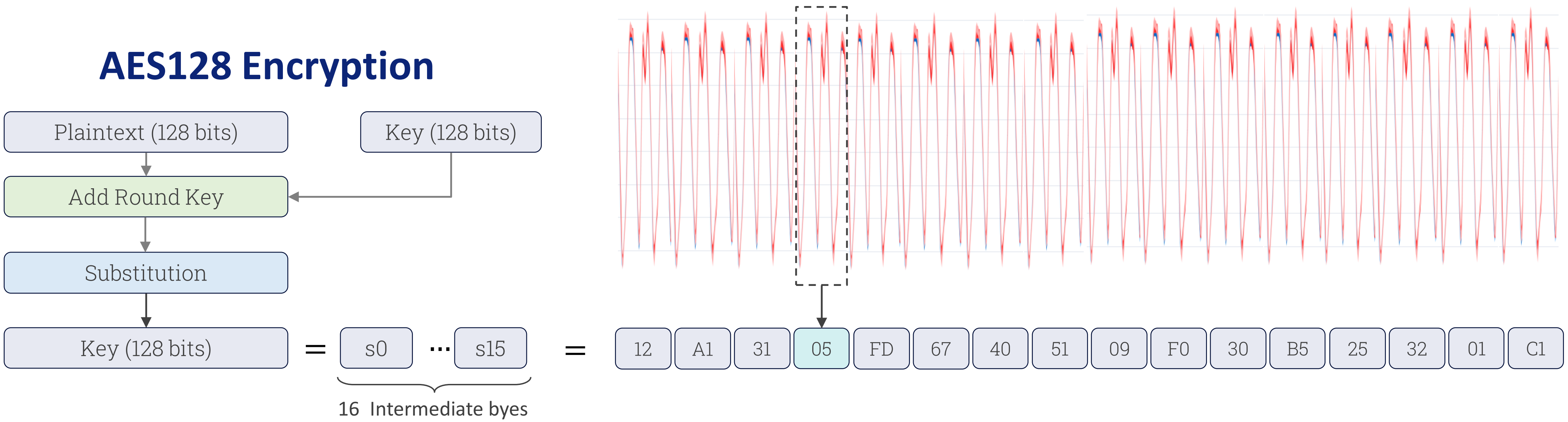
- **Leakage**, on the other hand, is part of the information conveyed through side-channels:
  - If only random noise is present: *there are no leakages*.
  - If any information can be retrieved from the measurement: *leakages exist.*

- How can we verify if there are leakages?

- **Leakage assessments** involve a statistical method that answers the question::
  - Are there any exploitable leakages?
    - If the answer is yes, tests are conducted (side-channel attacks – DPA, CPA, advanced attacks)
    - If no exploitable leakages are found, the product is considered secure and receives certification without further testing.

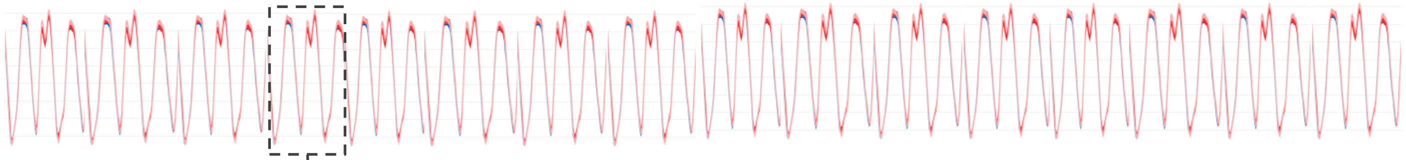
# Leakage models

- Leakage models describe how side-channel information (e.g., power consumption magnitude) correlates with the internal computations or operations of the cryptographic algorithm being executed.
- A perfect leakage model is impossible in practice because of noise.

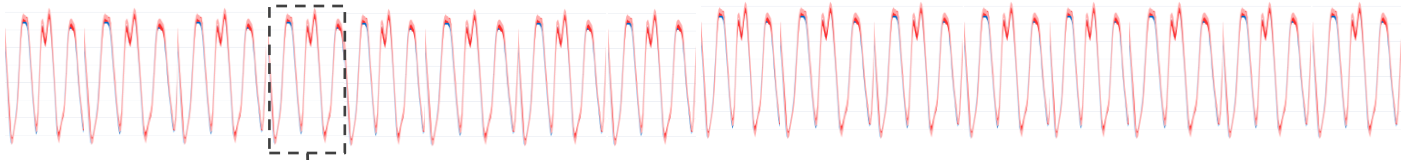


# Identity (ID) Leakage model

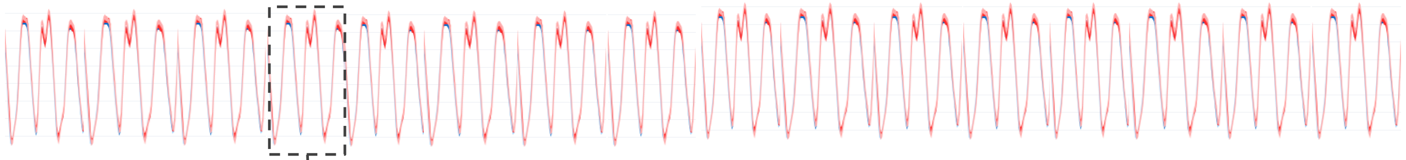
$$S(p0 \oplus k0) = s0$$



$$\underbrace{S(06 \oplus 00)}_{\substack{\text{Substitution output} \\ \text{(only 1 byte)}}} = 24$$



$$\underbrace{S(40 \oplus 00)}_{\substack{\text{Substitution output} \\ \text{(only 1 byte)}}} = B9$$



$$\underbrace{S(CA \oplus 00)}_{\substack{\text{Substitution output} \\ \text{(only 1 byte)}}} = 51$$

We are guessing that k0 is 0x00.

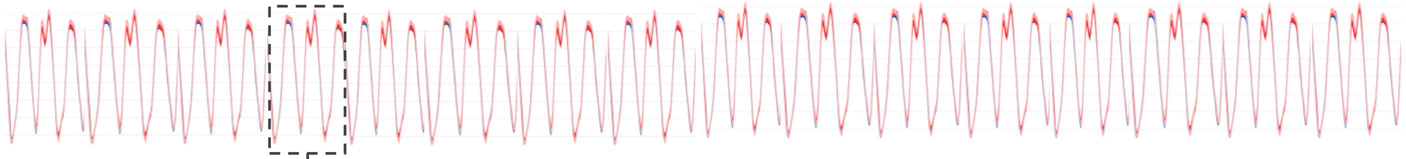
(we must guess all **256** hypothesis)

# Hamming Weight (HW) Leakage model

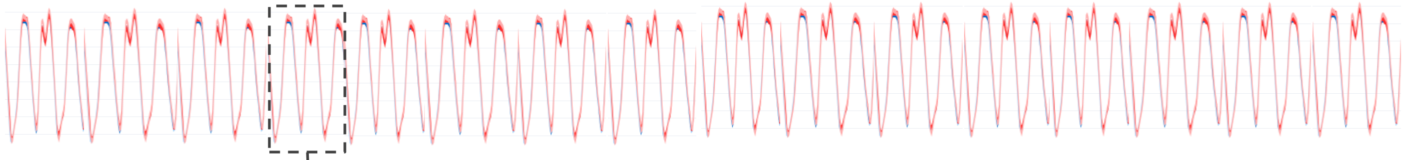
$$S(p0 \oplus k0) = s0$$

We are guessing that k0 is 0x00.

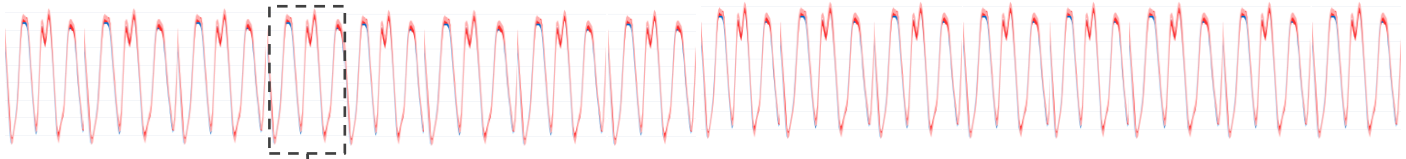
(we must guess all **256** hypothesis)



$$S(\underbrace{06 \oplus 00}_{\text{Substitution output (only 1 byte)}}) = 2 = 00100100$$

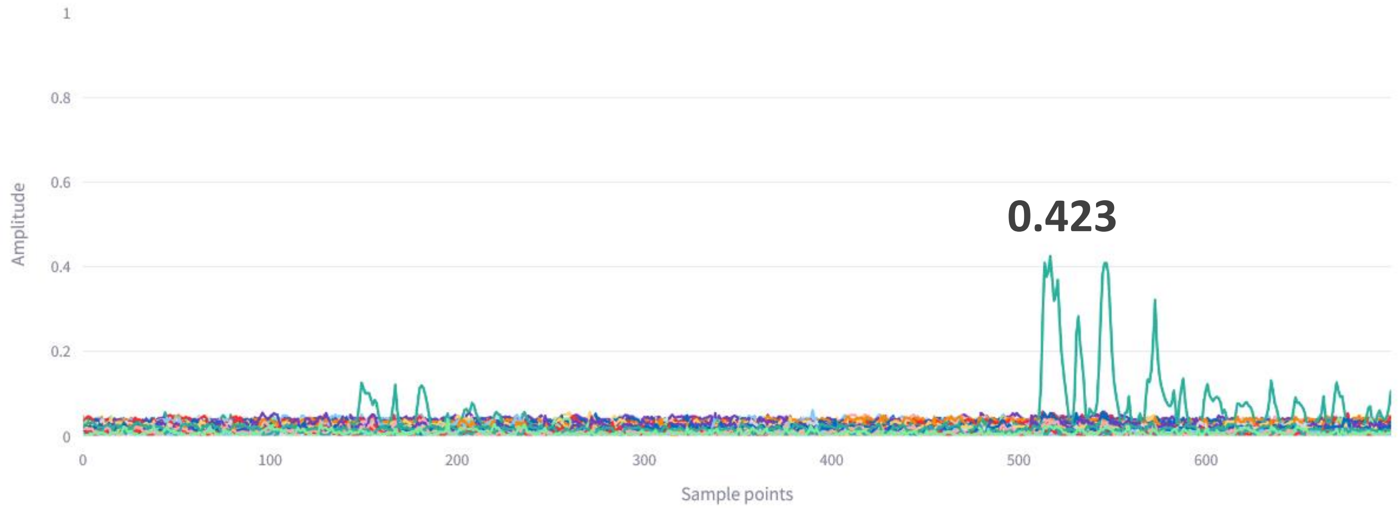


$$S(\underbrace{40 \oplus 00}_{\text{Substitution output (only 1 byte)}}) = 5 = 10111001$$

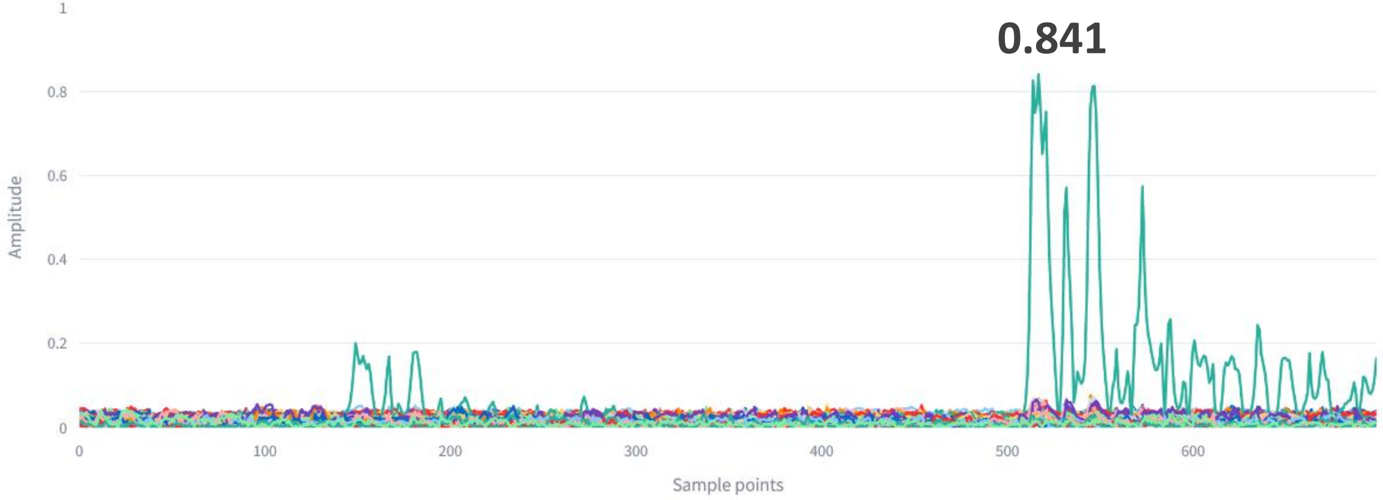


$$S(\underbrace{CA \oplus 00}_{\text{Substitution output (only 1 byte)}}) = 3 = 01010001$$

# CPA (HW vs ID)




**Identity**
leakage model




**Hamming weight**
leakage model

# Other leakage models

- Bit-level leakage model:
  - Most significant bit
  - Least significant bit
  - or any other bit…

- Hamming Distance:
  - HW($x \oplus y$)

- How to find the best leakage model?

  - A: try all the possibilities.

# Only AES?

- Obviously, not. Any computable function $f$ that depends on a portion of the key $k$ and controllable input data $d$:

$$d \longrightarrow \boxed{x = f(d, k^*)}$$

# Key ranking

- A side-channel attacks is key guessing-based attack: we need to test all key hypothesis.
- To discriminate the most likely key candidate, we need a distinguisher (difference-of-means, correlation).
- Based on a distinguisher, the key ranking informs that position of the correct key candidate among all possible key candidates.
- $c = [\rho_{key=0}, \rho_{key=1}, \dots, \rho_{key=224}, \dots, \rho_{key=255}] = [0.01, 0.02, \dots, 0.42, \dots, 0.07]$

$$\downarrow \text{sort}$$

- $c = [\rho_{key=224}, \rho_{key=255}, \dots, \rho_{key=0}, \dots, \rho_{key=1}] = [\mathbf{0.42}, 0.07, \dots, 0.02, \dots, 0.01]$

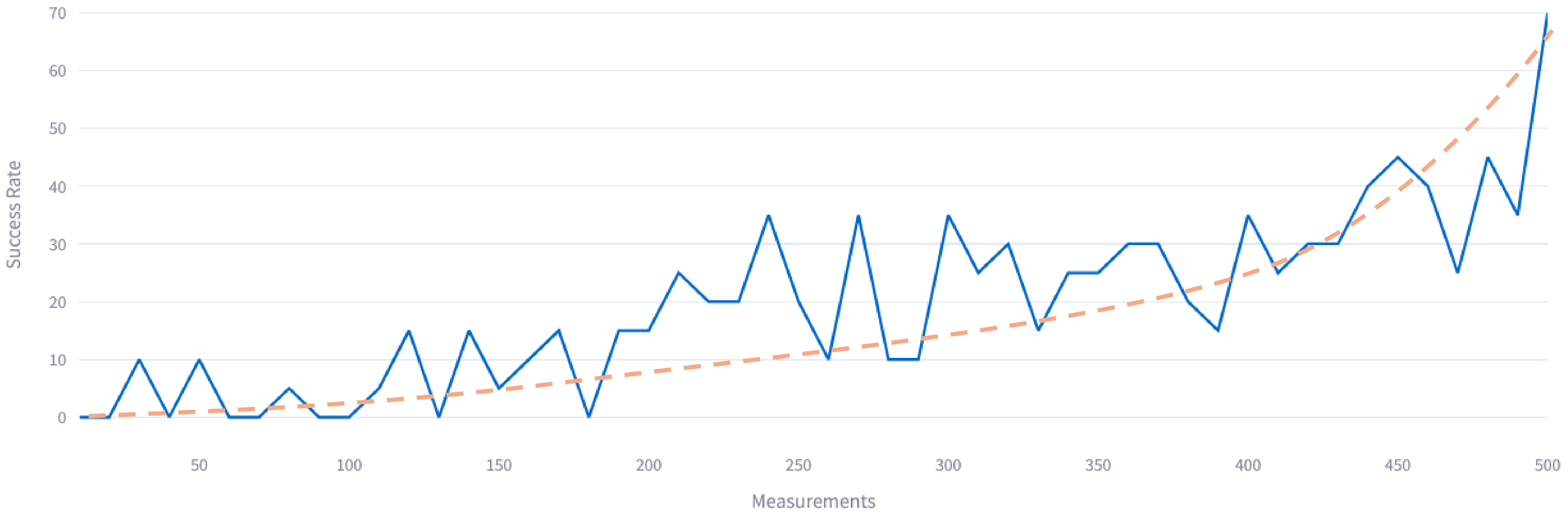
Position of the correct key candidate = **key rank**

# Guessing entropy

- Average key ranking multiple times.
- It informs about the entropy of the key (how much we still don’t know about the key).

# Success rate of an attack

- The percentage of successful key recovery (key rank = 1) if the attack process N measurements.



# How to compute the success rate?

1) Take N measurements
2) Compute the attack (CPA or DPA) for A times.
3) For all A attacks, get the percentage of times that the correct key is the most likely key candidate.
4) Increase N and return to step 1.

# Leakage assessment

- Statistical techniques to check whether the side-channel information has any (co-)relation with the intermediate data.
- Usually done with a controlled device (key can be configured).
- We are not interested in recovering the key, but only to see if **key is leaking**.

- One common method:

    - 1) collect a set of side-channel measurements with key 1
    - 2) collect a set of side-channel measurements with key 2

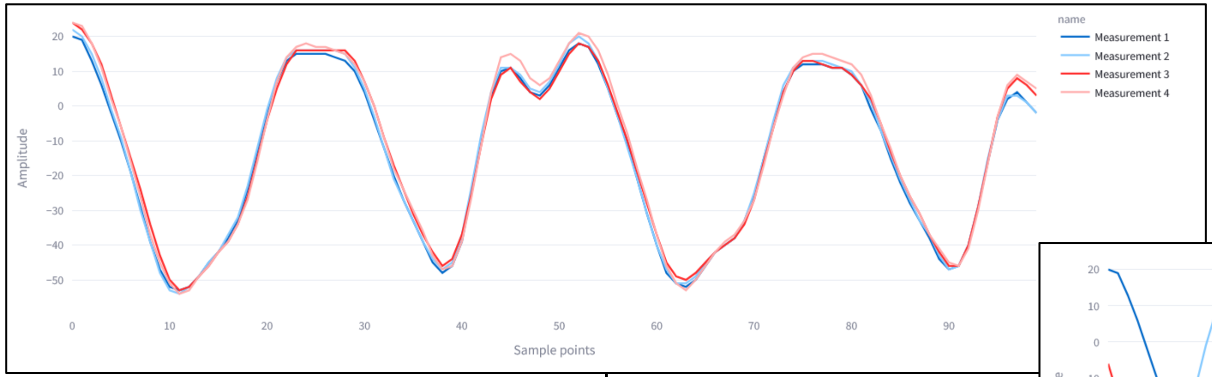
    - If both sets of measurements have statistical differences, then there are leakages.

# Countermeasures

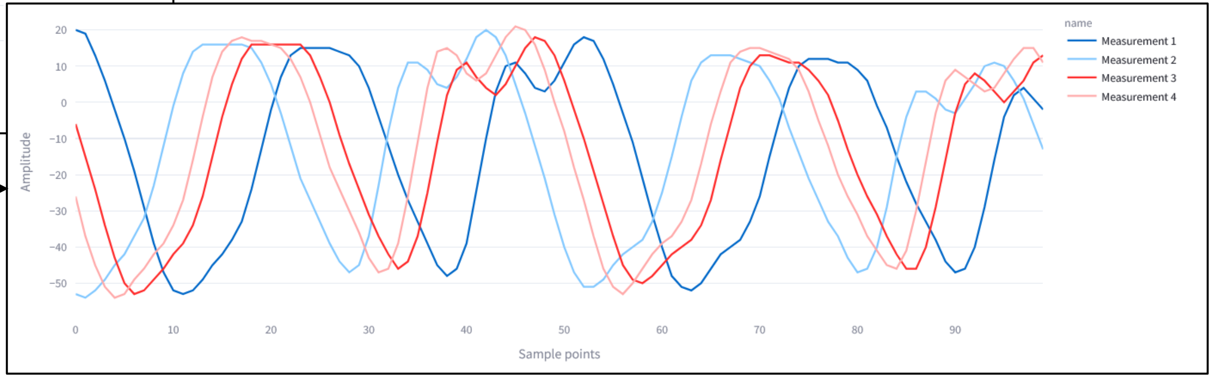
- **Hiding**: adds noise to the measurements.
- **Masking**: split the intermediate cryptographic values into several shares.

# The idea of hiding

- Add noise to the measurements.
- A side-channel analysis requires increased (sometimes unrealistic) amount of measurements.



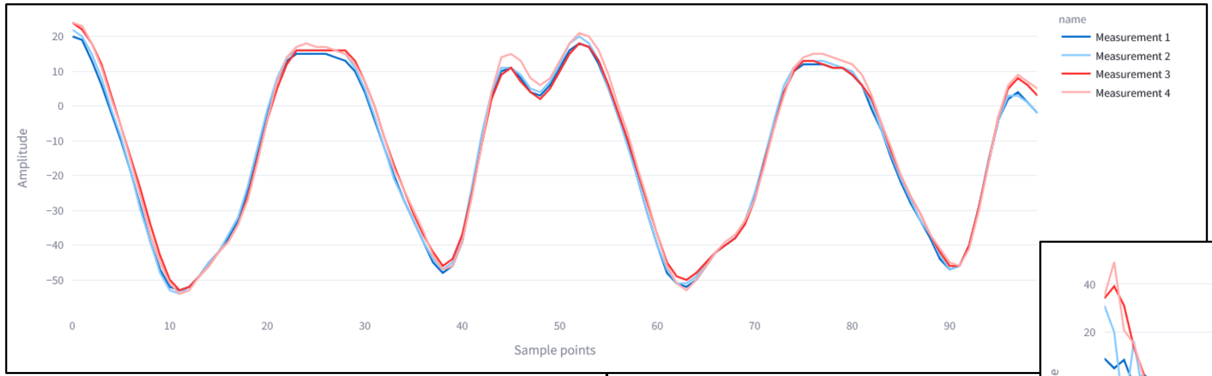

Desynchronization



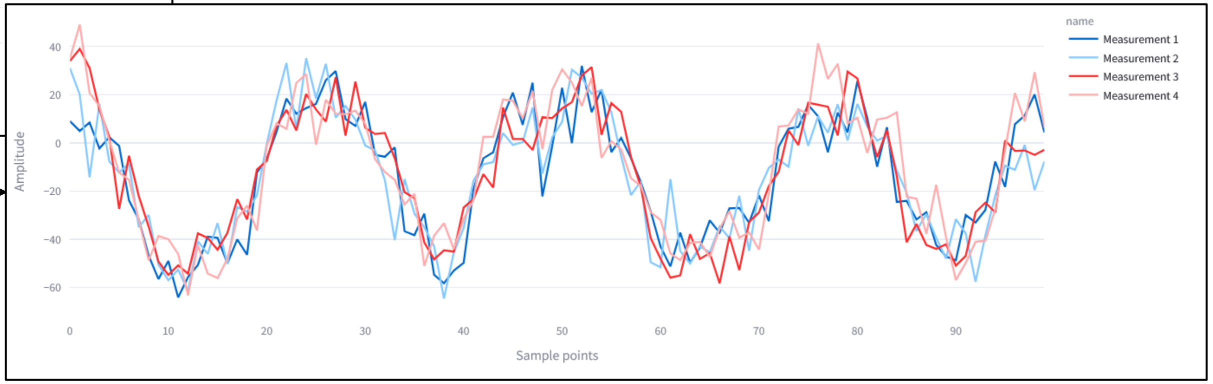

Preprocessing: trace alignment (pattern match)

# The idea of hiding

- Add noise to the measurements.
- A side-channel analysis requires increased (sometimes unrealistic) amount of measurements.



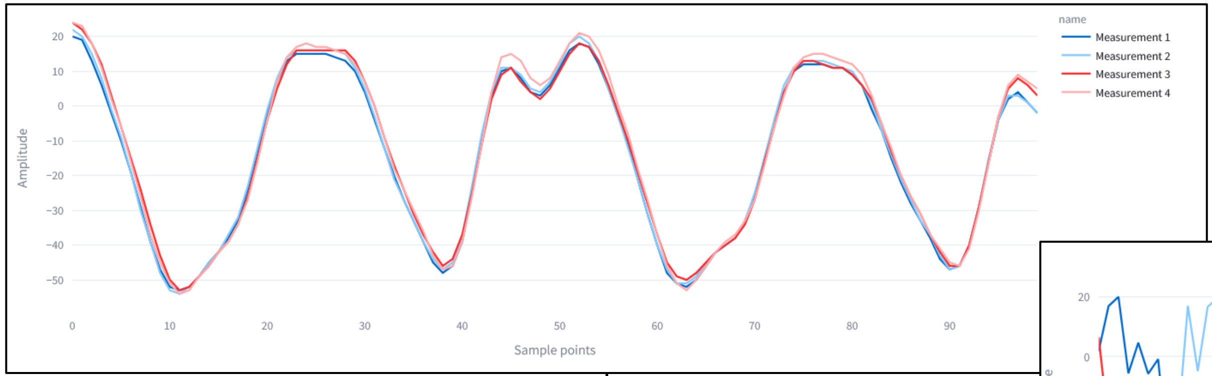

Add noise (noise generator)



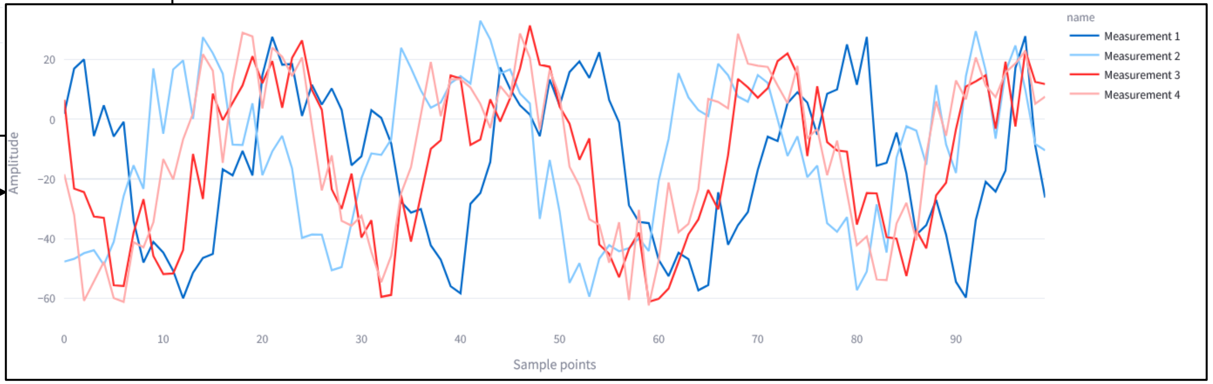

Preprocessing: trace filtering

# The idea of hiding

- Add noise to the measurements.
- A side-channel analysis requires increased (sometimes unrealistic) amount of measurements.

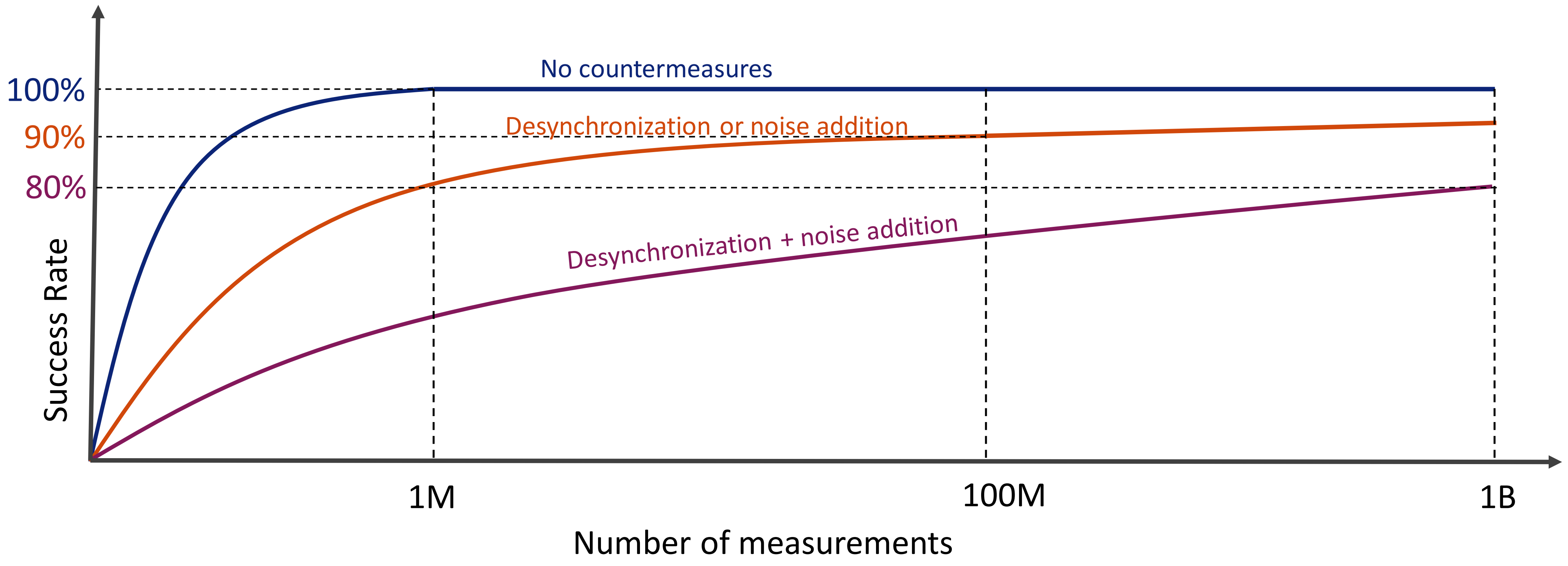



Add noise + desynchronization




Preprocessing: trace filtering + alignment (more complex)

# The effect of hiding on the success rate



# The idea of masking

- Split the intermediate cryptographic values into several shares.
- Never process intermediate values without masking them.

To mask a variable $x$:

$$x_{masked} = x \oplus mask$$

To unmask a variable $x$:

$$x = x_{masked} \oplus mask$$

First-order masking

To mask a variable $x$:

$$x_{masked} = x \oplus mask_1 \oplus mask_2$$

To unmask a variable $x$:

$$x = x_{masked} \oplus mask_1 \oplus mask_2$$

Second-order masking

# The idea of masking

- Split the intermediate cryptographic values into several shares.
- Never process intermediate values without masking them.

To mask a variable $x$:

$$x_{masked} = x \oplus mask$$

To unmask a variable $x$:

$$x = x_{masked} \oplus mask$$

To mask a variable $x$:

$$x_{masked} = x \oplus mask_1 \oplus mask_2 \oplus mask_2 \oplus \cdots \oplus mask_n$$

To unmask a variable $x$:

$$\underbrace{x = x_{masked} \oplus mask_1 \oplus mask_2 \oplus mask_2 \oplus \cdots \oplus mask_n}_{\text{n-order masking}}$$

# Side-channel analysis order

N measurements

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | … | 696 | 697 | 698 | 699 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 19 | 13 | 6 | -2 | -10 | -19 | -29 | -39 | -47 | -52 | -53 | -52 | -49 | -45 | | -11 | 0 | 9 | 15 |
| 22 | 20 | 15 | 8 | -1 | -9 | -19 | -30 | -39 | -48 | -53 | -54 | -52 | -49 | -45 | | -10 | 1 | 10 | 16 |
| 24 | 22 | 18 | 12 | 3 | -6 | -15 | -24 | -34 | -43 | -50 | -53 | -52 | -49 | -46 | … | -15 | -3 | 7 | 13 |
| 24 | 23 | 18 | 11 | 2 | -6 | -16 | -26 | -37 | -45 | -51 | -54 | -53 | -49 | -46 | | -14 | -3 | 6 | 13 |
| 21 | 20 | 16 | 9 | 1 | -6 | -16 | -24 | -34 | -43 | -48 | -51 | -50 | -47 | -44 | | -12 | -2 | 7 | 13 |
| ⋮ | | | | | | | ⋮ | | | | | | | | ⋱ | | ⋮ | | |
| 25 | 24 | 20 | 14 | 5 | -4 | -13 | -23 | -33 | -44 | -50 | -53 | -52 | -50 | -47 | … | -15 | -4 | 6 | 13 |
| 21 | 20 | 15 | 9 | 0 | -8 | -17 | -26 | -35 | -43 | -49 | -51 | -50 | -46 | -43 | | -14 | -3 | 5 | 12 |

n dimensions

# First-order Correlation Analysis

- Process 1 dimension at a time (single dimension analysis).

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Measurement 1 | 20 | 19 | 13 | 6 | -2 | -10 | -19 | -29 | -39 | -47 |
| Measurement 2 | 22 | 20 | 15 | 8 | -1 | -9 | -19 | -30 | -39 | -48 |
| Measurement 3 | 24 | 22 | 18 | 12 | 3 | -6 | -15 | -24 | -34 | -43 |
| Measurement 4 | 24 | 23 | 18 | 11 | 2 | -6 | -16 | -26 | -37 | -45 |
| Measurement 5 | 21 | 20 | 16 | 9 | 1 | -6 | -16 | -24 | -34 | -43 |
| Measurement 6 | 17 | 18 | 15 | 9 | 1 | -6 | -14 | -23 | -32 | -40 |
| Measurement 7 | 24 | 26 | 22 | 16 | 8 | -2 | -12 | -22 | -32 | -42 |
| Measurement 8 | 19 | 18 | 14 | 7 | 0 | -8 | -17 | -27 | -36 | -43 |
| Measurement 9 | 21 | 21 | 16 | 10 | 2 | -6 | -15 | -25 | -35 | -44 |
| Measurement 10 | 21 | 22 | 17 | 12 | 4 | -4 | -13 | -23 | -32 | -42 |

$k_i$

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 36 | 249 | 225 | 41 | 12 | 192 | 253 | 188 | 111 |
| 185 | 242 | 181 | 69 | 164 | 200 | 179 | 228 | 29 |
| 81 | 175 | 109 | 230 | 242 | 114 | 165 | 127 | 15 |
| 85 | 109 | 131 | 209 | 94 | 106 | 118 | 152 | 181 |
| 108 | 17 | 21 | 60 | 148 | 130 | 192 | 214 | 135 |
| 162 | 99 | 82 | 128 | 20 | 99 | 139 | 249 | 140 |
| 2 | 113 | 74 | 123 | 65 | 173 | 205 | 6 | 209 |
| 19 | 203 | 201 | 6 | 145 | 62 | 53 | 92 | 19 |
| 228 | 200 | 234 | 58 | 42 | 60 | 246 | 84 | 35 |
| 156 | 163 | 164 | 52 | 203 | 188 | 162 | 75 | 34 |

$x$

$y$

Pearson Correlation Coefficient

$$\rho(x,y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

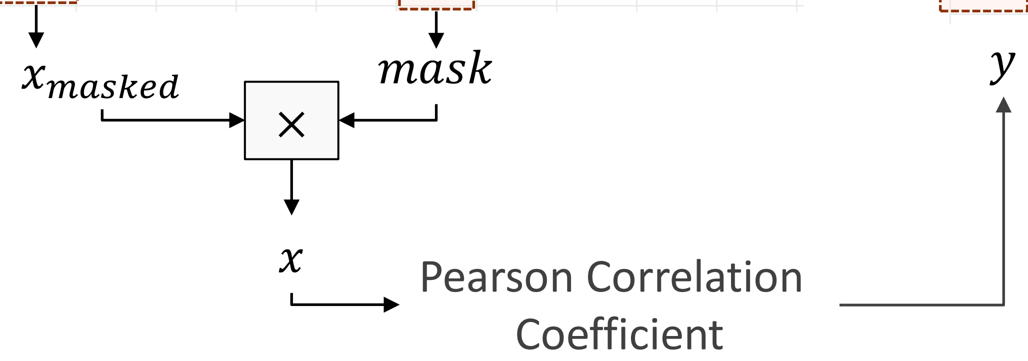
# Second-order Correlation Analysis

- Process 2 dimensions at a time.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Measurement 1 | 20 | 19 | 13 | 6 | -2 | -10 | -19 | -29 | -39 | -47 |
| Measurement 2 | 22 | 20 | 15 | 8 | -1 | -9 | -19 | -30 | -39 | -48 |
| Measurement 3 | 24 | 22 | 18 | 12 | 3 | -6 | -15 | -24 | -34 | -43 |
| Measurement 4 | 24 | 23 | 18 | 11 | 2 | -6 | -16 | -26 | -37 | -45 |
| Measurement 5 | 21 | 20 | 16 | 9 | 1 | -6 | -16 | -24 | -34 | -43 |
| Measurement 6 | 17 | 18 | 15 | 9 | 1 | -6 | -14 | -23 | -32 | -40 |
| Measurement 7 | 24 | 26 | 22 | 16 | 8 | -2 | -12 | -22 | -32 | -42 |
| Measurement 8 | 19 | 18 | 14 | 7 | 0 | -8 | -17 | -27 | -36 | -43 |
| Measurement 9 | 21 | 21 | 16 | 10 | 2 | -6 | -15 | -25 | -35 | -44 |
| Measurement 10 | 21 | 22 | 17 | 12 | 4 | -4 | -13 | -23 | -32 | -42 |

$k_i$

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 36 | 249 | 225 | 41 | 12 | 192 | 253 | 188 | 111 |
| 185 | 242 | 181 | 69 | 164 | 200 | 179 | 228 | 29 |
| 81 | 175 | 109 | 230 | 242 | 114 | 165 | 127 | 15 |
| 85 | 109 | 131 | 209 | 94 | 106 | 118 | 152 | 181 |
| 108 | 17 | 21 | 60 | 148 | 130 | 192 | 214 | 135 |
| 162 | 99 | 82 | 128 | 20 | 99 | 139 | 249 | 140 |
| 2 | 113 | 74 | 123 | 65 | 173 | 205 | 6 | 209 |
| 19 | 203 | 201 | 6 | 145 | 62 | 53 | 92 | 19 |
| 228 | 200 | 234 | 58 | 42 | 60 | 246 | 84 | 35 |
| 156 | 163 | 164 | 52 | 203 | 188 | 162 | 75 | 34 |

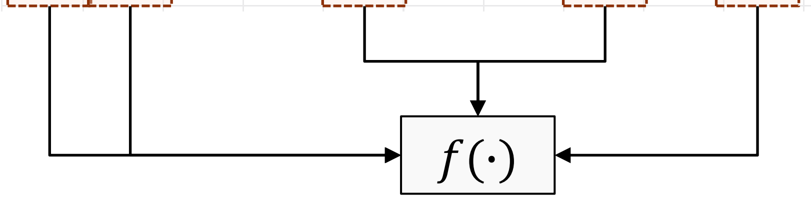



$$\rho(x,y) = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2 \sum(y_i-\bar{y})^2}} 9$$

# Multi-dimensional Analysis

- Multi-dimensional side-channel analysis: process n dimensions at a time.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Measurement 1 | 20 | 19 | 13 | 6 | -2 | -10 | -19 | -29 | -39 | -47 |
| Measurement 2 | 22 | 20 | 15 | 8 | -1 | -9 | -19 | -30 | -39 | -48 |
| Measurement 3 | 24 | 22 | 18 | 12 | 3 | -6 | -15 | -24 | -34 | -43 |
| Measurement 4 | 24 | 23 | 18 | 11 | 2 | -6 | -16 | -26 | -37 | -45 |
| Measurement 5 | 21 | 20 | 16 | 9 | 1 | -6 | -16 | -24 | -34 | -43 |
| Measurement 6 | 17 | 18 | 15 | 9 | 1 | -6 | -14 | -23 | -32 | -40 |
| Measurement 7 | 24 | 26 | 22 | 16 | 8 | -2 | -12 | -22 | -32 | -42 |
| Measurement 8 | 19 | 18 | 14 | 7 | 0 | -8 | -17 | -27 | -36 | -43 |
| Measurement 9 | 21 | 21 | 16 | 10 | 2 | -6 | -15 | -25 | -35 | -44 |
| Measurement 10 | 21 | 22 | 17 | 12 | 4 | -4 | -13 | -23 | -32 | -42 |




$k_i$

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 36 | 249 | 225 | 41 | 12 | 192 | 253 | 188 | 111 |
| 185 | 242 | 181 | 69 | 164 | 200 | 179 | 228 | 29 |
| 81 | 175 | 109 | 230 | 242 | 114 | 165 | 127 | 15 |
| 85 | 109 | 131 | 209 | 94 | 106 | 118 | 152 | 181 |
| 108 | 17 | 21 | 60 | 148 | 130 | 192 | 214 | 135 |
| 162 | 99 | 82 | 128 | 20 | 99 | 139 | 249 | 140 |
| 2 | 113 | 74 | 123 | 65 | 173 | 205 | 6 | 209 |
| 19 | 203 | 201 | 6 | 145 | 62 | 53 | 92 | 19 |
| 228 | 200 | 234 | 58 | 42 | 60 | 246 | 84 | 35 |
| 156 | 163 | 164 | 52 | 203 | 188 | 162 | 75 | 34 |

$y$

# Multi-dimensional Analysis

- Multi-dimensional side-channel analysis: process n dimensions at a time.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Measurement 1 | 20 | 19 | 13 | 6 | -2 | -10 | -19 | -29 | -39 | -47 |
| Measurement 2 | 22 | 20 | 15 | 8 | -1 | -9 | -19 | -30 | -39 | -48 |
| Measurement 3 | 24 | 22 | 18 | 12 | 3 | -6 | -15 | -24 | -34 | -43 |
| Measurement 4 | 24 | 23 | 18 | 11 | 2 | -6 | -16 | -26 | -37 | -45 |
| Measurement 5 | 21 | 20 | 16 | 9 | 1 | -6 | -16 | -24 | -34 | -43 |
| Measurement 6 | 17 | 18 | 15 | 9 | 1 | -6 | -14 | -23 | -32 | -40 |
| Measurement 7 | 24 | 26 | 22 | 16 | 8 | -2 | -12 | -22 | -32 | -42 |
| Measurement 8 | 19 | 18 | 14 | 7 | 0 | -8 | -17 | -27 | -36 | -43 |
| Measurement 9 | 21 | 21 | 16 | 10 | 2 | -6 | -15 | -25 | -35 | -44 |
| Measurement 10 | 21 | 22 | 17 | 12 | 4 | -4 | -13 | -23 | -32 | -42 |

$$f(\cdot)$$

$k_i$

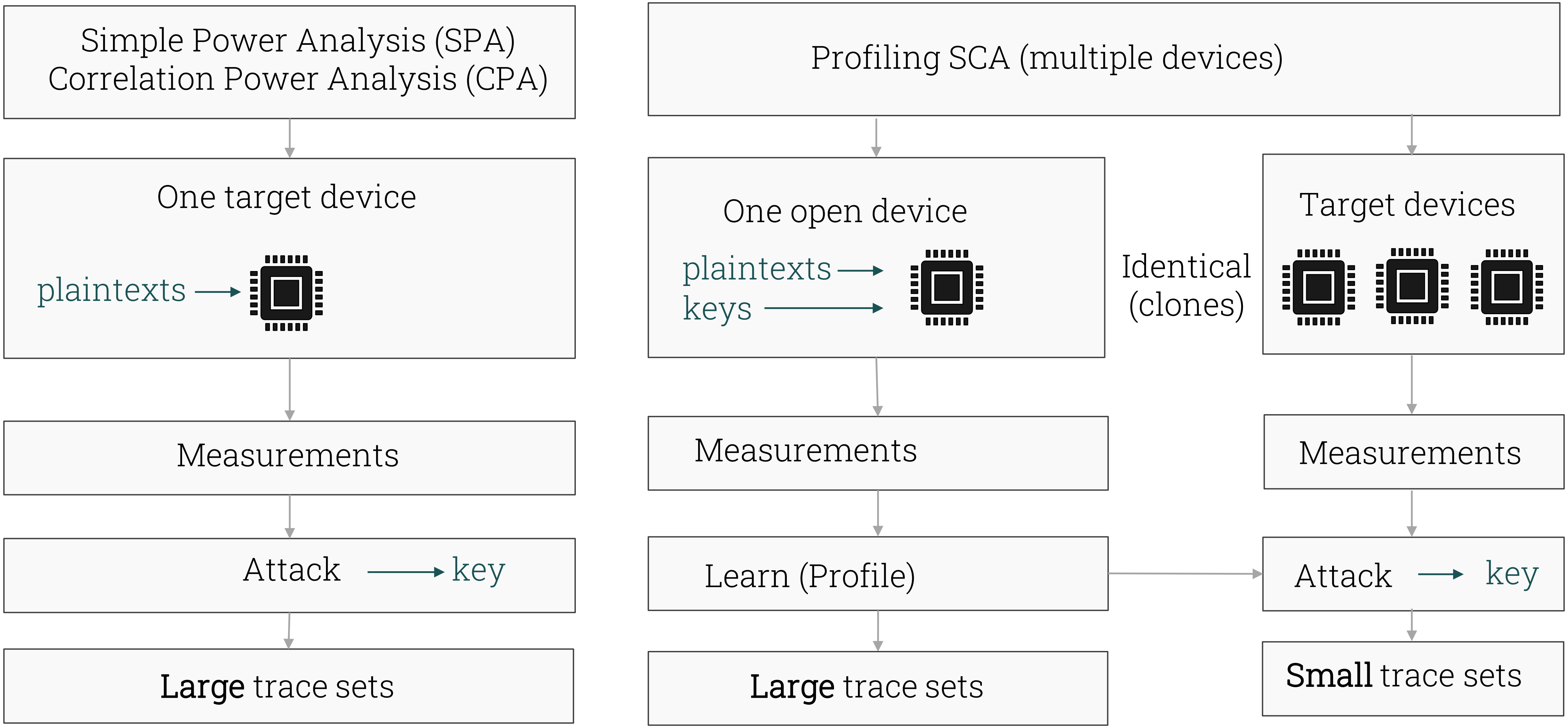
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 36 | 249 | 225 | 41 | 12 | 192 | 253 | 188 | 111 |
| 185 | 242 | 181 | 69 | 164 | 200 | 179 | 228 | 29 |
| 81 | 175 | 109 | 230 | 242 | 114 | 165 | 127 | 15 |
| 85 | 109 | 131 | 209 | 94 | 106 | 118 | 152 | 181 |
| 108 | 17 | 21 | 60 | 148 | 130 | 192 | 214 | 135 |
| 162 | 99 | 82 | 128 | 20 | 99 | 139 | 249 | 140 |
| 2 | 113 | 74 | 123 | 65 | 173 | 205 | 6 | 209 |
| 19 | 203 | 201 | 6 | 145 | 62 | 53 | 92 | 19 |
| 228 | 200 | 234 | 58 | 42 | 60 | 246 | 84 | 35 |
| 156 | 163 | 164 | 52 | 203 | 188 | 162 | 75 | 34 |

$y$

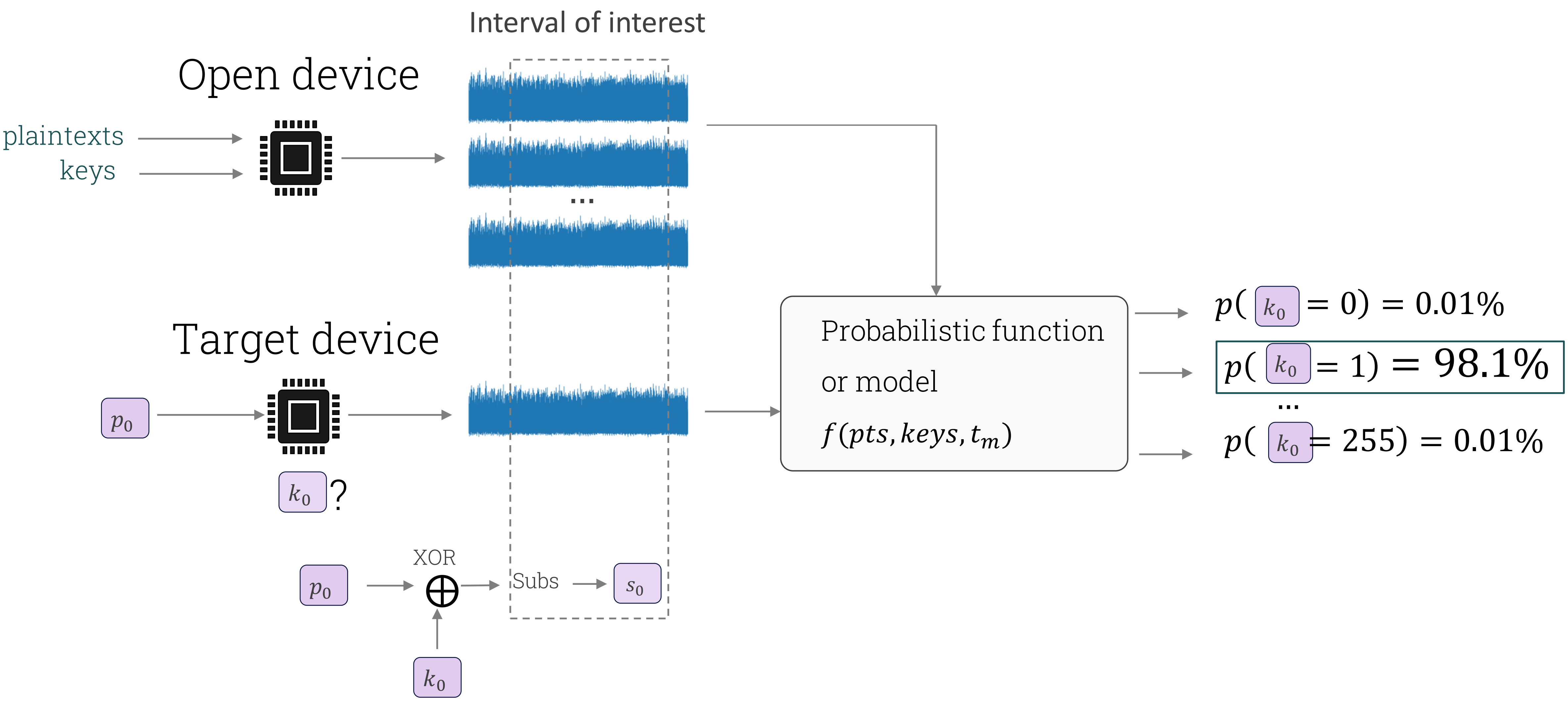
# Summary

- xxx

# Direct (CPA, DPA) vs Profiling SCA



# Profiling SCA



# Template attacks

- Two-phase attack (**learn** and **match**).
- Theoretically, the strongest attack (no model hypeparameters).

$x = f(d, k^*)$ → leakage model $(x)$ → check how many possible values → number of templates

Cryptographic algorithm (intermediate value) ex: SubBytes output: $s_i = S(p_i \oplus k_i)$

How $s_i$ is represented by side-channel information.

If leakage model is:

- HW: **9** possible values.
- ID: **256** possible values
- MSb: **2** possible values

For each possible intermediate value (after the leakage model), a template is calculated.

# Template attacks

- Feature selection (feature selection requires knowledge about the key and, sometimes, the countermeasures (values of the masks).
- Measurements need to be as synchronized as possible (i.e., aligned).

**Learning (or profiling) phase**

- Key is known
- Plaintexts are known
- *Masks are known

- Large set of measurements

**Matching (or attack) phase**

- Key is unknown
- Plaintexts are known
- *Masks are unknown

- Small set of measurements

**Identical devices**

# Template attacks – Learning Phase

Given the array of measurements:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | … | 696 | 697 | 698 | 699 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 19 | 13 | 6 | -2 | -10 | -19 | -29 | -39 | -47 | -52 | -53 | -52 | -49 | -45 | | -11 | 0 | 9 | 15 | Measurement 1 |
| 22 | 20 | 15 | 8 | -1 | -9 | -19 | -30 | -39 | -48 | -53 | -54 | -52 | -49 | -45 | | -10 | 1 | 10 | 16 | Measurement 2 |
| 24 | 22 | 18 | 12 | 3 | -6 | -15 | -24 | -34 | -43 | -50 | -53 | -52 | -49 | -46 | … | -15 | -3 | 7 | 13 | Measurement 3 |
| 24 | 23 | 18 | 11 | 2 | -6 | -16 | -26 | -37 | -45 | -51 | -54 | -53 | -49 | -46 | | -14 | -3 | 6 | 13 | Measurement 4 |
| 21 | 20 | 16 | 9 | 1 | -6 | -16 | -24 | -34 | -43 | -48 | -51 | -50 | -47 | -44 | | -12 | -2 | 7 | 13 | Measurement 5 |
| ⋮ | | | | | | | ⋮ | | | | | | | | ⋱ | | ⋮ | | | |
| 25 | 24 | 20 | 14 | 5 | -4 | -13 | -23 | -33 | -44 | -50 | -53 | -52 | -50 | -47 | … | -15 | -4 | 6 | 13 | Measurement 49999 |
| 21 | 20 | 15 | 9 | 0 | -8 | -17 | -26 | -35 | -43 | -49 | -51 | -50 | -46 | -43 | | -14 | -3 | 5 | 12 | Measurement 50000 |

Select a few columns (features) that contain the more information about the intermediate $x_i$.
(key and plaintext are known for this set of measurements)

Points-of-Interest Selection

# Template attacks – Learning Phase

Given the array of measurements:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | … | 696 | 697 | 698 | 699 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 19 | 13 | 6 | -2 | -10 | -19 | -29 | -39 | -47 | -52 | -53 | -52 | -49 | -45 | | -11 | 0 | 9 | 15 | Measurement 1 |
| 22 | 20 | 15 | 8 | -1 | -9 | -19 | -30 | -39 | -48 | -53 | -54 | -52 | -49 | -45 | | -10 | 1 | 10 | 16 | Measurement 2 |
| 24 | 22 | 18 | 12 | 3 | -6 | -15 | -24 | -34 | -43 | -50 | -53 | -52 | -49 | -46 | … | -15 | -3 | 7 | 13 | Measurement 3 |
| 24 | 23 | 18 | 11 | 2 | -6 | -16 | -26 | -37 | -45 | -51 | -54 | -53 | -49 | -46 | | -14 | -3 | 6 | 13 | Measurement 4 |
| 21 | 20 | 16 | 9 | 1 | -6 | -16 | -24 | -34 | -43 | -48 | -51 | -50 | -47 | -44 | | -12 | -2 | 7 | 13 | Measurement 5 |
| ⋮ | | | | | | | ⋮ | | | | | | | | ⋱ | | ⋮ | | | |
| 25 | 24 | 20 | 14 | 5 | -4 | -13 | -23 | -33 | -44 | -50 | -53 | -52 | -50 | -47 | … | -15 | -4 | 6 | 13 | Measurement 49999 |
| 21 | 20 | 15 | 9 | 0 | -8 | -17 | -26 | -35 | -43 | -49 | -51 | -50 | -46 | -43 | | -14 | -3 | 5 | 12 | Measurement 50000 |

**Correlation power analysis** with the known key and select the columns with highest correlation values. → Points-of-Interest Selection

# Template attacks – Learning/Profiling Phase

Points of interest selection returns only the columns with significant side-channel leakages with respect to the target intermediate variable $s_i$.

| 2 | 7 | 9 | 14 | | 698 |
|---|---|---|---|---|---|
| 13 | -29 | -47 | -45 | … | 9 |
| 15 | -30 | -48 | -45 | | 10 |
| 18 | -24 | -43 | -46 | … | 7 |
| 18 | -26 | -45 | -46 | | 6 |
| 16 | -24 | -43 | -44 | | 7 |
| | ⋮ | | | ⋱ | ⋮ |
| 20 | -23 | -44 | -47 | | 6 |
| 15 | -26 | -43 | -43 | … | 5 |

10 dimensions (example)

$\mu, \sigma^2, \Sigma$ = template

- Mean $\mu$ (vector with 10 values)
- Variance $\sigma^2$ (vector with 10 values)
- Co-variance $\Sigma$ (array with 10 x 10 values)

In Python:

- `np.mean(t, axis=0)`
- `np.var(t, axis=0)`
- `np.cov(t)`

$\mu_0, {\sigma^2}_0, \Sigma_0$ when $s_i = 0$

$\mu_1, {\sigma^2}_1, \Sigma_1$ when $s_i = 1$

$\mu_2, {\sigma^2}_2, \Sigma_2$ when $s_i = 2$

⋮

$\mu_{255}, {\sigma^2}_{255}, \Sigma_{255}$ when $s_i = 255$

**256 templates**

This is the output of the learning phase

# Template attacks – Learning/Profiling Phase

Points of interest selection returns only the columns with significant side-channel leakages with respect to the target intermediate variable $s_i$.

| 2 | 7 | 9 | 14 | | 698 |
|---|---|---|---|---|---|
| 13 | -29 | -47 | -45 | … | 9 |
| 15 | -30 | -48 | -45 | | 10 |
| 18 | -24 | -43 | -46 | … | 7 |
| 18 | -26 | -45 | -46 | | 6 |
| 16 | -24 | -43 | -44 | | 7 |
| | ⋮ | | | ⋱ | ⋮ |
| 20 | -23 | -44 | -47 | … | 6 |
| 15 | -26 | -43 | -43 | … | 5 |

10 dimensions
(example)

- Mean $\mu$ (vector with 10 values)
- Variance $\sigma^2$ (vector with 10 values)
- Co-variance $\Sigma$ (array with 10 x 10 values)

$\mu, \sigma^2, \Sigma$ = template

In Python:

```
np.mean(t, axis=0)
np.var(t, axis=0)
np.cov(t)
```

$$\begin{cases} \mu_0, {\sigma^2}_0, \Sigma_0 \text{ when } s_i = 0 \\ \mu_1, {\sigma^2}_1, \Sigma_1 \text{ when } s_i = 1 \\ \mu_2, {\sigma^2}_2, \Sigma_2 \text{ when } s_i = 2 \\ \vdots \\ \mu_8, {\sigma^2}_8, \Sigma_8 \text{ when } s_i = 8 \end{cases}$$

**9 templates**
**(for Hamming weight leakage model)**

This is the output of the learning phase

# Template attacks – Learning/Profiling Phase

What is a template?

The values $\mu, \sigma^2, \Sigma$ are statistical properties about the leakages. These values can model the side-channel information with respect to the key.

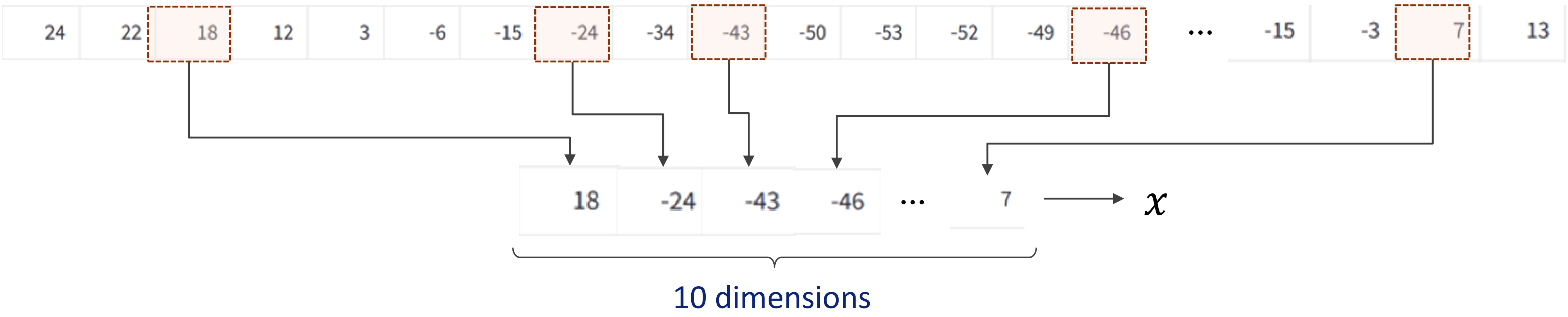
If $\mu, \sigma^2, \Sigma$ are optimal (the best possible values we can get), then these values can be used in a probability distribution function (pdf).

Pdf? A function that returns a probability: $p(x) = f(x, \mu, \sigma^2, \Sigma)$

The value $p(x)$ is the probability that $x$ is drawn from the same distribution defined by $\mu, \sigma^2, \Sigma$.

# Template attacks – Matching/Attack Phase

- Collect a new measurement from the target device (the one with unknown key $k_i$).
- Select the same columns (points-of-interests) obtained with learning phase.




$$p_0(x) = f(\boldsymbol{x}, \mu_0, {\sigma^2}_0, \Sigma_0) = 0.05$$
$$p_1(x) = f(\boldsymbol{x}, \mu_1, {\sigma^2}_1, \Sigma_1) = \mathbf{0.87}$$
$$\vdots$$
$$p_{255}(x) = f(\boldsymbol{x}, \mu_{255}, {\sigma^2}_{255}, \Sigma_{255}) = 0.01$$

The highest probability indicates what intermediate value $s_i$ is being processed by the measurement $x$, which can lead to the key $k_i$ :

$$s_i = S(p_i \oplus k_i) \rightarrow \boxed{k_i = S^{-1}(s_i) \oplus p_i}$$

# Template attacks – What is the $f$ function?

($d$ = number of dimensions in $\boldsymbol{x}$)

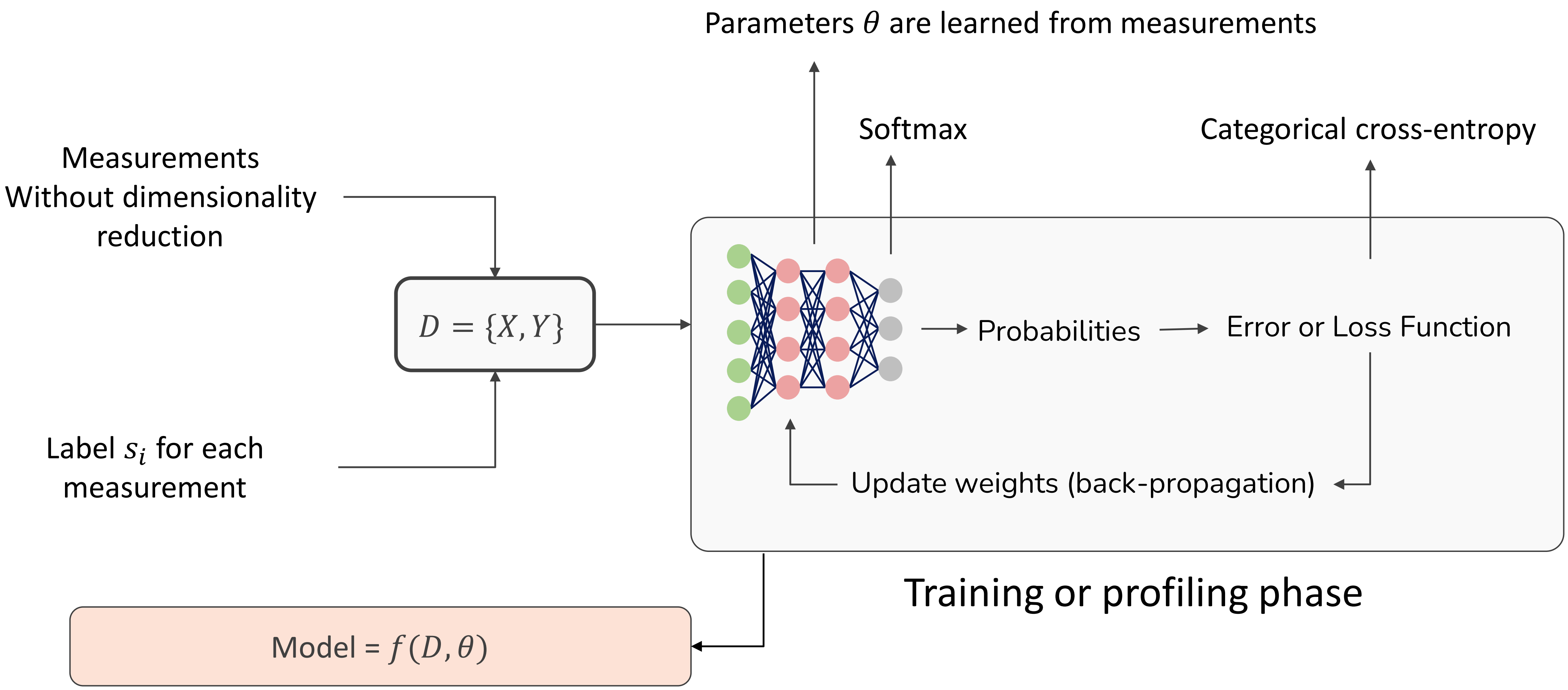
$$\underbrace{p(x, \boldsymbol{\mu_c}, \boldsymbol{\sigma^2}_c, \boldsymbol{\Sigma}_c) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\boldsymbol{\Sigma_c}|}} \exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_c})^T \boldsymbol{\Sigma_c}^{-1}(\boldsymbol{x} - \boldsymbol{\mu_c}))}_{\text{Multi-variate Gaussian Probability Distribution Function}}$$

# Deep learning-based attacks

- Supervised and unsupervised
- Automate several steps from a profiling (template) attack
- Deep neural networks as classifiers:
  - Discriminative models (to classify the intermediate values such as SubBytes output).
  - Generative models (preprocessing).

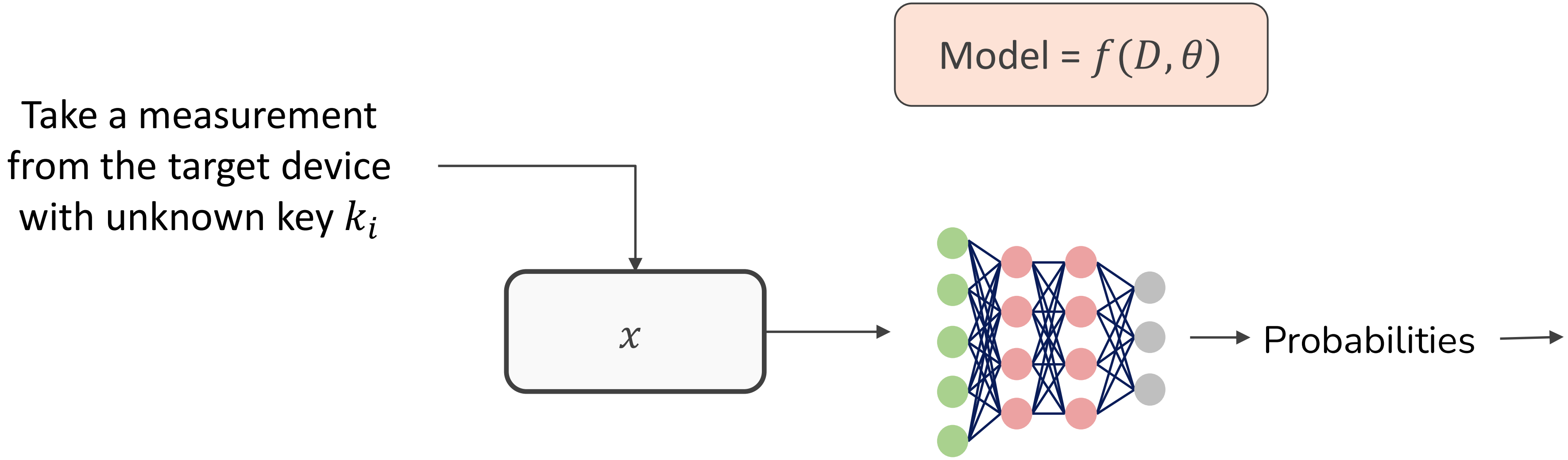
# Deep learning-based attacks

- Does not require feature selection.



# Deep learning-based attacks

- Does not require feature selection.




$$\begin{cases} p_0(x) = f(D, \theta) = 0.05 \\ p_1(x) = f(D, \theta) = \mathbf{0.32} \\ \vdots \\ p_{255}(x) = f(D, \theta) = 0.02 \end{cases}$$

Attack phase

# Template vs deep learning-based attacks

Template attacks:

- 2-phase process (learning and matching)
- Feature selection (knowledge about the implementation is necessary (countermeasures)
- One template is calculated for each possible intermediate value in the cryptographic algorithm (e.g., each possible value for the SubBytes output operation)
- Function f is calculated from data (which makes the process deterministic).

Deep learning attacks:

- 2-phase process (training and attack)
- Feature selection is not necessary (can work with high-dimensional data)
- One DNN is trained for all possible intermediate value in the cryptographic algorithm (e.g., each possible value for the SubBytes output operation)
- Function f is learned from data (which makes the process stochastic).

# Deep learning-based attacks – main challenges

- Hyperparameter tuning (difficult and time-consuming).
- Lack of data: DL requires lot of training data and SCA datasets are usually small.
- Side-channel data is very noisy and DL metrics (loss, accuracy) are not correlated with SCA metrics (key ranking, guessing entropy, success rate).

# Some recommended databases/frameworks

- **ASCAD:** https://github.com/ANSSI-FR/ASCAD
- **AISY Framework:** https://github.com/AISyLab/AISY_Framework

# State-of-the-art research papers in SCA

- **TCHES:** https://tches.iacr.org/index.php/TCHES/issue/archive
- **CARDIS:** https://sbd-research.nl/cardis-2023/
- **COSADE:** https://www.cosade.org/cosade24/

Thank you!