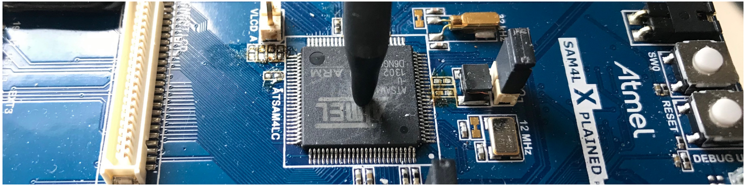


Side-Channel Countermeasures' Dissection

and the Limits of Closed Source Security Evaluations

Olivier Bronchain François-Xavier Standaert

CHES 2020, Online

European Research Council
Established by the European Commission

GROUPE CRYPTO

UCLouvain

Content

Side-Channels: How to Design Security ?

How to reach high security levels ?

Side-Channels: How to Design Security ?

How to reach high security levels ?

- Side-channel attacks are a physical problem

Side-Channels: How to Design Security ?

How to reach high security levels ?

- Side-channel attacks are a physical problem
- Let's solve it based on physical solutions

Side-Channels: How to Design Security ?

How to reach high security levels ?

- Side-channel attacks are a physical problem
- Let's solve it based on physical solutions
  - Noise addition

Side-Channels: How to Design Security ?

How to reach high security levels ?

- Side-channel attacks are a physical problem
- Let's solve it based on physical solutions
  - Noise addition
  - Signal reduction

Side-Channels: How to Design Security ?

How to reach high security levels ?

- Side-channel attacks are a physical problem
- Let's solve it based on physical solutions
  - Noise addition
  - Signal reduction
- However it may not be enough to provide high protection

Side-Channels: How to Design Security ?

How to reach high security levels ?

- Side-channel attacks are a physical problem
- Let's solve it based on physical solutions
  - Noise addition
  - Signal reduction
- However it may not be enough to provide high protection
  - Noise is not a parameter giving exponential security

Side-Channels: How to Design Security ?

How to reach high security levels ?

- Side-channel attacks are a physical problem
- Let's solve it based on physical solutions
  - Noise addition
  - Signal reduction
- However it may not be enough to provide high protection
  - Noise is not a parameter giving exponential security
- Exploit ”*noise amplification*” based on mathematical analysis

Side-Channels: How to Design Security ?

How to reach high security levels ?

- Side-channel attacks are a physical problem
- Let's solve it based on physical solutions
  - Noise addition
  - Signal reduction
- However it may not be enough to provide high protection
  - Noise is not a parameter giving exponential security
- Exploit "*noise amplification*" based on mathematical analysis
  - Requires additional hypothesis (e.g., independence for masking)

Open vs. Closed Approaches For Evaluations

What approaches exist in embedded security evaluation ?

Open vs. Closed Approaches For Evaluations

What approaches exist in embedded security evaluation ?

- Open approach 

Open vs. Closed Approaches For Evaluations

What approaches exist in embedded security evaluation ?

- Open approach
  - Evaluator gets all knowledge/control of the target

Open vs. Closed Approaches For Evaluations

What approaches exist in embedded security evaluation ?

- Open approach
  - Evaluator gets all knowledge/control of the target
    - Eased verification of physical assumptions

Open vs. Closed Approaches For Evaluations

What approaches exist in embedded security evaluation ?

- Open approach
  - Evaluator gets all knowledge/control of the target
    - Eased verification of physical assumptions
  - More privileged in academic research (but no only)

Open vs. Closed Approaches For Evaluations

What approaches exist in embedded security evaluation ?

- Open approach
  - Evaluator gets <u>all</u> knowledge/control of the target
    - Eased verification of physical assumptions
  - More privileged in academic research (but no only)
- Closed approach

Open vs. Closed Approaches For Evaluations

What approaches exist in embedded security evaluation ?

- Open approach
  - Evaluator gets all knowledge/control of the target
    - Eased verification of physical assumptions
  - More privileged in academic research (but no only)
- Closed approach
  - Evaluator gets restricted knowledge/control of the target

Open vs. Closed Approaches For Evaluations

What approaches exist in embedded security evaluation ?

- Open approach
  - Evaluator gets all knowledge/control of the target
    - Eased verification of physical assumptions
  - More privileged in academic research (but no only)
- Closed approach
  - Evaluator gets restricted knowledge/control of the target
    - Harder verification of physical assumptions

Open vs. Closed Approaches For Evaluations

What approaches exist in embedded security evaluation ?

- Open approach
  - Evaluator gets <u>all</u> knowledge/control of the target
    - Eased verification of physical assumptions
  - More privileged in academic research (but no only)
- Closed approach
  - Evaluator gets <u>restricted</u> knowledge/control of the target
    - Harder verification of physical assumptions
    - In contradiction with Kerckhoff's principle

Open vs. Closed Approaches For Evaluations

What approaches exist in embedded security evaluation ?

- Open approach
  - Evaluator gets all knowledge/control of the target
    - Eased verification of physical assumptions
  - More privileged in academic research (but no only)
- Closed approach
  - Evaluator gets restricted knowledge/control of the target
    - Harder verification of physical assumptions
    - In contradiction with Kerckhoff's principle
  - In part encouraged by some certification practices (e.g., CC)

What About Real-World Targets ?

A few published attacks on real products exist:

What About Real-World Targets ?

A few published attacks on real products exist:

- Key recovery for bitstream encryption keys (Moradi *et al.*, 2011)

What About Real-World Targets ?

A few published attacks on real products exist:

- Key recovery for bitstream encryption keys (Moradi *et al.*, 2011)
- Update forgery on HP Light Bumps (Ronen *et al.*, 2016)

What About Real-World Targets ?

A few published attacks on real products exist:

- Key recovery for bitstream encryption keys (Moradi *et al.*, 2011)
- Update forgery on HP Light Bumps (Ronen *et al.*, 2016)
- Car opening against Tesla Key Fob (Wouters *et al.*, 2019)

What About Real-World Targets ?

A few published attacks on real products exist:

- Key recovery for bitstream encryption keys (Moradi *et al.*, 2011)
- Update forgery on HP Light Bumps (Ronen *et al.*, 2016)
- Car opening against Tesla Key Fob (Wouters *et al.*, 2019)

Once (huge) reverse engineering done, attacks are straightforward.

What About Real-World Targets ?

A few published attacks on real products exist:

- Key recovery for bitstream encryption keys (Moradi *et al.*, 2011)
- Update forgery on HP Light Bumps (Ronen *et al.*, 2016)
- Car opening against Tesla Key Fob (Wouters *et al.*, 2019)

Once (huge) reverse engineering done, attacks are straightforward.

- These examples are however not reflective of certified products

What About Real-World Targets ?

A few published attacks on real products exist:

- Key recovery for bitstream encryption keys (Moradi *et al.*, 2011)
- Update forgery on HP Light Bumps (Ronen *et al.*, 2016)
- Car opening against Tesla Key Fob (Wouters *et al.*, 2019)

Once (huge) reverse engineering done, attacks are straightforward.

- These examples are however not reflective of certified products
- We lack practically relevant examples of "*sound combinations of countermeasures*"

Useful step in this direction: ANSSI's Implem.

Open-source protected AES:

Hardened Library for AES-128 encryption/decryption on ARM Cortex M4 Achitecture

Authors: Ryad Benadjila, Louiza Khati, Emmanuel Prouff and Adrian Thillard

This work is linked to the H2020 funded project **REASSURE**.

Introduction

The members of ANSSI's laboratory of embedded security have developed a `C` library to perform AES-128 encryption and decryption on 32-bit Cortex-M ARM architecture while taking Side-Channel Attacks (SCA for short) into account.

The implementation codes are published for research and pedagogical purposes only.

Useful step in this direction: ANSSI's Implem.

Open-source protected AES:

- From a team of experts

Hardened Library for AES-128 encryption/decryption on ARM Cortex M4 Achitecture

Authors: Ryad Benadjila, Louiza Khati, Emmanuel Prouff and Adrian Thillard

This work is linked to the H2020 funded project **REASSURE**.

Introduction

The members of ANSSI's laboratory of embedded security have developed a `C` library to perform AES-128 encryption and decryption on 32-bit Cortex-M ARM architecture while taking Side-Channel Attacks (SCA for short) into account.

The implementation codes are published for research and pedagogical purposes only.

Useful step in this direction: ANSSI's Implem.

Open-source protected AES:

- From a team of experts
- Mixed countermeasures

Hardened Library for AES-128 encryption/decryption on ARM Cortex M4 Achitecture

Authors: Ryad Benadjila, Louiza Khati, Emmanuel Prouff and Adrian Thillard

This work is linked to the H2020 funded project **REASSURE**.

Introduction

The members of ANSSI's laboratory of embedded security have developed a `C` library to perform AES-128 encryption and decryption on 32-bit Cortex-M ARM architecture while taking Side-Channel Attacks (SCA for short) into account.

The implementation codes are published for research and pedagogical purposes only.

Useful step in this direction: ANSSI's Implem.

Open-source protected AES:

- From a team of experts
- Mixed countermeasures
- Preliminary leakage assessment

Hardened Library for AES-128 encryption/decryption on ARM Cortex M4 Achitecture

Authors: Ryad Benadjila, Louiza Khati, Emmanuel Prouff and Adrian Thillard

This work is linked to the H2020 funded project **REASSURE**.

Introduction

The members of ANSSI's laboratory of embedded security have developed a `C` library to perform AES-128 encryption and decryption on 32-bit Cortex-M ARM architecture while taking Side-Channel Attacks (SCA for short) into account.

The implementation codes are published for research and pedagogical purposes only.

Useful step in this direction: ANSSI's Implem.

Open-source protected AES:

- From a team of experts
- Mixed countermeasures
- Preliminary leakage assessment

!! Educational purpose only !!

Hardened Library for AES-128 encryption/decryption on ARM Cortex M4 Achitecture

Authors: Ryad Benadjila, Louiza Khati, Emmanuel Prouff and Adrian Thillard

This work is linked to the H2020 funded project **REASSURE**.

Introduction

The members of ANSSI's laboratory of embedded security have developed a `C` library to perform AES-128 encryption and decryption on 32-bit Cortex-M ARM architecture while taking Side-Channel Attacks (SCA for short) into account.

The implementation codes are published for research and pedagogical purposes only.

Useful step in this direction: ANSSI's Implem.

Open-source protected AES:

- From a team of experts
- Mixed countermeasures
- Preliminary leakage assessment

!! Educational purpose only !!

It could be used to study:

Hardened Library for AES-128 encryption/decryption on ARM Cortex M4 Achitecture

Authors: Ryad Benadjila, Louiza Khati, Emmanuel Prouff and Adrian Thillard

This work is linked to the H2020 funded project **REASSURE**.

Introduction

The members of ANSSI's laboratory of embedded security have developed a `C` library to perform AES-128 encryption and decryption on 32-bit Cortex-M ARM architecture while taking Side-Channel Attacks (SCA for short) into account.

The implementation codes are published for research and pedagogical purposes only.

Useful step in this direction: ANSSI's Implem.

Open-source protected AES:

- From a team of experts
- Mixed countermeasures
- Preliminary leakage assessment

!! Educational purpose only !!

Hardened Library for AES-128 encryption/decryption on ARM Cortex M4 Achitecture

Authors: Ryad Benadjila, Louiza Khati, Emmanuel Prouff and Adrian Thillard

This work is linked to the H2020 funded project **REASSURE**.

Introduction

The members of ANSSI's laboratory of embedded security have developed a `C` library to perform AES-128 encryption and decryption on 32-bit Cortex-M ARM architecture while taking Side-Channel Attacks (SCA for short) into account.

The implementation codes are published for research and pedagogical purposes only.

It could be used to study:

1. Effectiveness of mixed countermeasures

Useful step in this direction: ANSSI's Implem.

Open-source protected AES:

- From a team of experts
- Mixed countermeasures
- Preliminary leakage assessment

!! Educational purpose only !!

Hardened Library for AES-128 encryption/decryption on ARM Cortex M4 Achitecture

Authors: Ryad Benadjila, Louiza Khati, Emmanuel Prouff and Adrian Thillard

This work is linked to the H2020 funded project **REASSURE**.

Introduction

The members of ANSSI's laboratory of embedded security have developed a `C` library to perform AES-128 encryption and decryption on 32-bit Cortex-M ARM architecture while taking Side-Channel Attacks (SCA for short) into account.

The implementation codes are published for research and pedagogical purposes only.

It could be used to study:

1. Effectiveness of mixed countermeasures
2. Security on popular 32-bit MCU's

Useful step in this direction: ANSSI's Implem.

Open-source protected AES:

- From a team of experts
- Mixed countermeasures
- Preliminary leakage assessment

!! Educational purpose only !!

Hardened Library for AES-128 encryption/decryption on ARM Cortex M4 Achitecture

Authors: Ryad Benadjila, Louiza Khati, Emmanuel Prouff and Adrian Thillard

This work is linked to the H2020 funded project **REASSURE**.

Introduction

The members of ANSSI's laboratory of embedded security have developed a `C` library to perform AES-128 encryption and decryption on 32-bit Cortex-M ARM architecture while taking Side-Channel Attacks (SCA for short) into account.

The implementation codes are published for research and pedagogical purposes only.

It could be used to study:

1. Effectiveness of mixed countermeasures
2. Security on popular 32-bit MCU's
3. Impact of open designs for worst-case security evaluations

Profiled Side-Channel Attacks in 

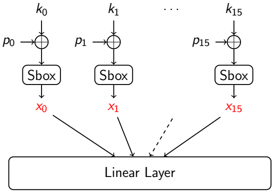
Worst-case analysis in two phases:

Profiled Side-Channel Attacks in

Worst-case analysis in two phases:

1. Profiling / Learning target behavior

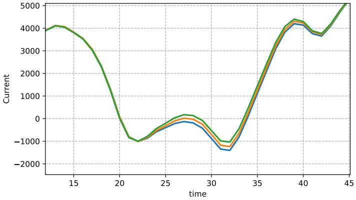
Profiled Side-Channel Attacks in 

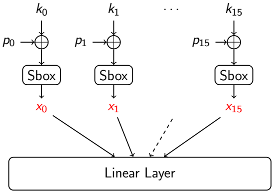


Worst-case analysis in two phases:

1. Profiling / Learning target behavior
   - Algorithm/Implementation knowledge

Profiled Side-Channel Attacks in

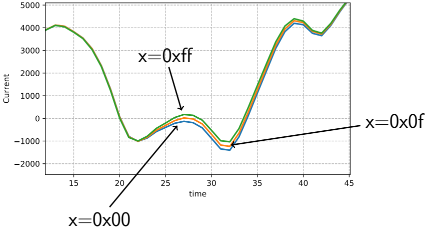


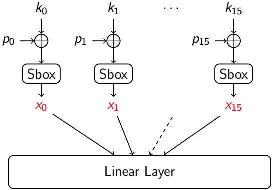


Worst-case analysis in two phases:

1. Profiling / Learning target behavior
   - Algorithm/Implementation knowledge
   - Leakage examples in controlled settings (i.e. known randomness)

Profiled Side-Channel Attacks in 

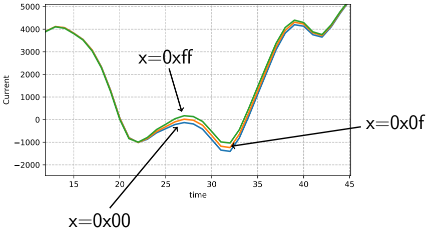


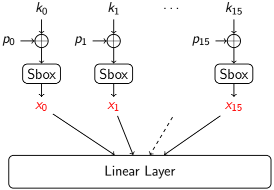


Worst-case analysis in two phases:

1. Profiling / Learning target behavior
   - Algorithm/Implementation knowledge
   - Leakage examples in controlled settings (i.e. known randomness)

Profiled Side-Channel Attacks in

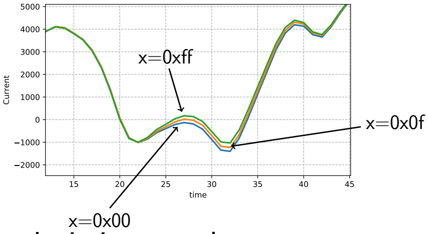


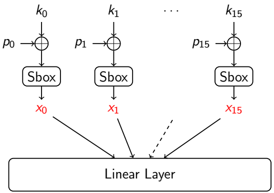


Worst-case analysis in two phases:

1. Profiling / Learning target behavior
   - Algorithm/Implementation knowledge
   - Leakage examples in controlled settings (i.e. known randomness)
2. Attack

Profiled Side-Channel Attacks in 

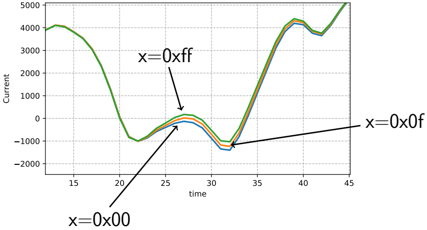


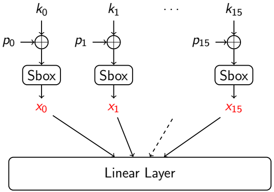


Worst-case analysis in two phases:

1. Profiling / Learning target behavior
   - Algorithm/Implementation knowledge
   - Leakage examples in controlled settings (i.e. known randomness)
2. Attack
   - Extract information from leakage

Profiled Side-Channel Attacks in 





Worst-case analysis in two phases:

1. Profiling / Learning target behavior
   - Algorithm/Implementation knowledge
   - Leakage examples in controlled settings (i.e. known randomness)
2. Attack
   - Extract information from leakage
   - Processing for secret recovery

Content

Countermeasures

At a high level:

Countermeasures

At a high level:

- Affine masking on bytes

Countermeasures

At a high level:

- Affine masking on bytes
  - Multiplicative mask r_m (same for all the 16-bytes)

Countermeasures

At a high level:

- Affine masking on bytes
  - Multiplicative mask r_m (same for all the 16-bytes)
  - Additive mask r_a

Countermeasures

At a high level:

- Affine masking on bytes
  - Multiplicative mask r_m (same for all the 16-bytes)
  - Additive mask r_a
  - Requires alternative Sbox table pre-computation

Countermeasures

At a high level:

- Affine masking on bytes
  - Multiplicative mask r_m (same for all the 16-bytes)
  - Additive mask r_a
  - Requires alternative Sbox table pre-computation
- Shuffled execution

Countermeasures

At a high level:

- Affine masking on bytes
  - Multiplicative mask r_m (same for all the 16-bytes)
  - Additive mask r_a
  - Requires alternative Sbox table pre-computation
- Shuffled execution
  - One permutation for the 16 Sboxes

Countermeasures

At a high level:

- Affine masking on bytes
  - Multiplicative mask r_m (same for all the 16-bytes)
  - Additive mask r_a
  - Requires alternative Sbox table pre-computation
- Shuffled execution
  - One permutation for the 16 Sboxes
  - Another permutation for the 4 MixColumns

Countermeasures

At a high level:

- Affine masking on bytes
  - Multiplicative mask r_m (same for all the 16-bytes)
  - Additive mask r_a
  - Requires alternative Sbox table pre-computation
- Shuffled execution
  - One permutation for the 16 Sboxes
  - Another permutation for the 4 MixColumns
  - Both are pre-computed

Countermeasures

Inputs

Pre-computation

Encryption

Countermeasures

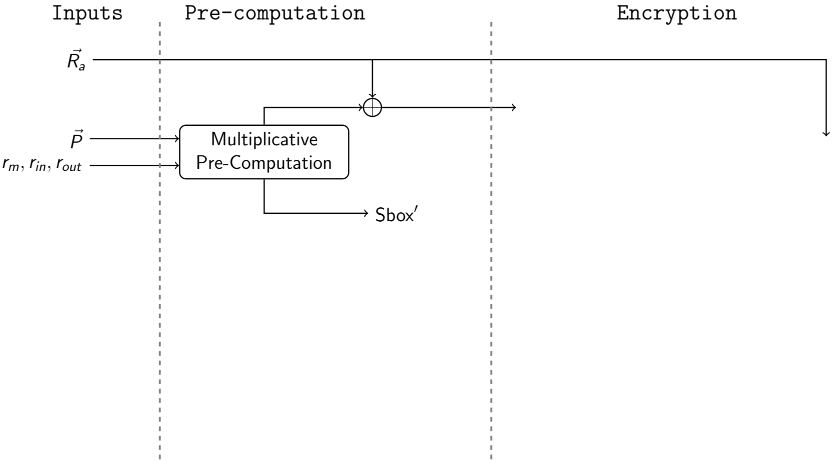
Inputs | Pre-computation | Encryption

$\vec{R}_a$

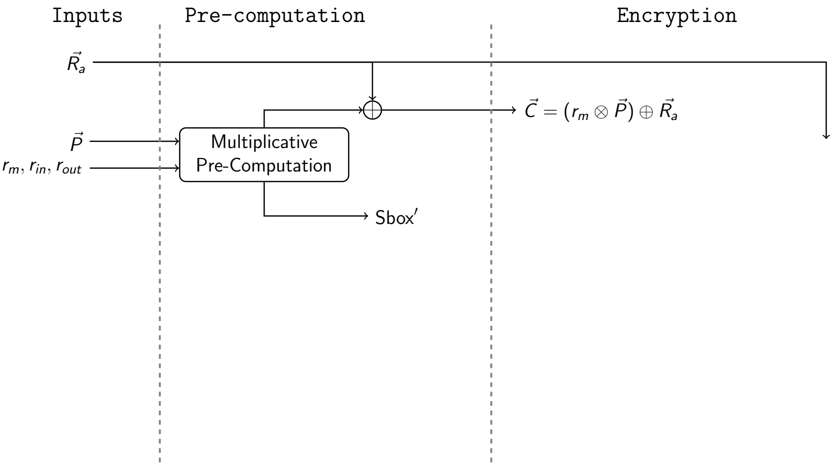
$\vec{P}$

r_m, r_{in}, r_{out}

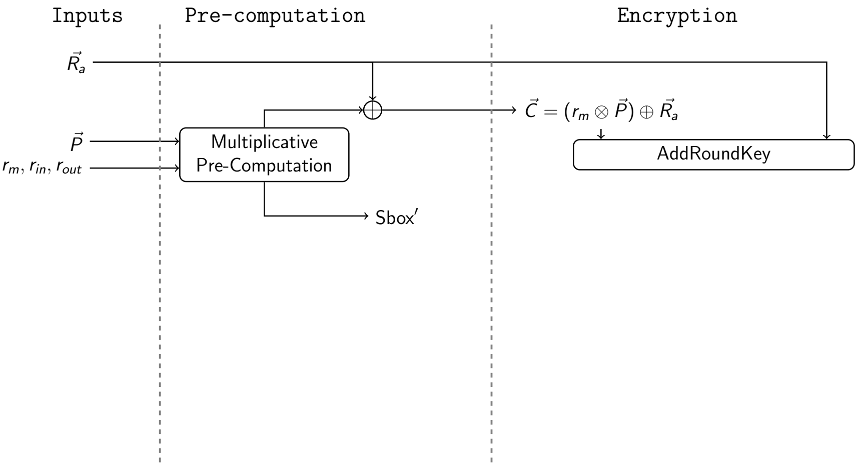
Countermeasures



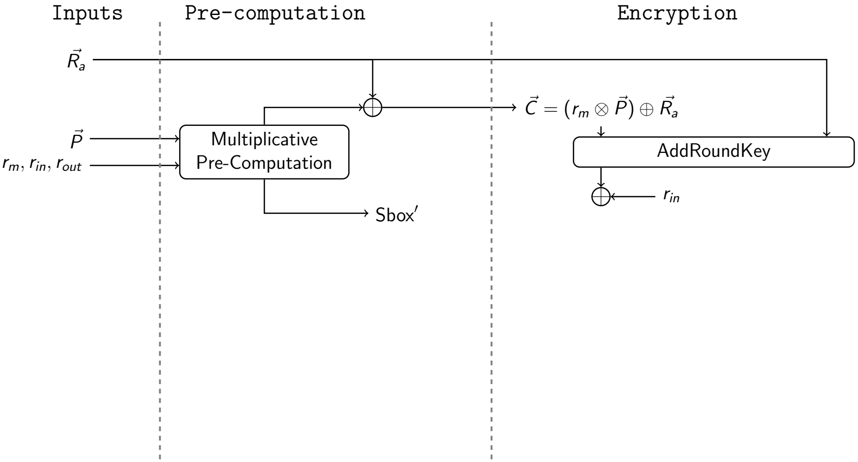
Countermeasures



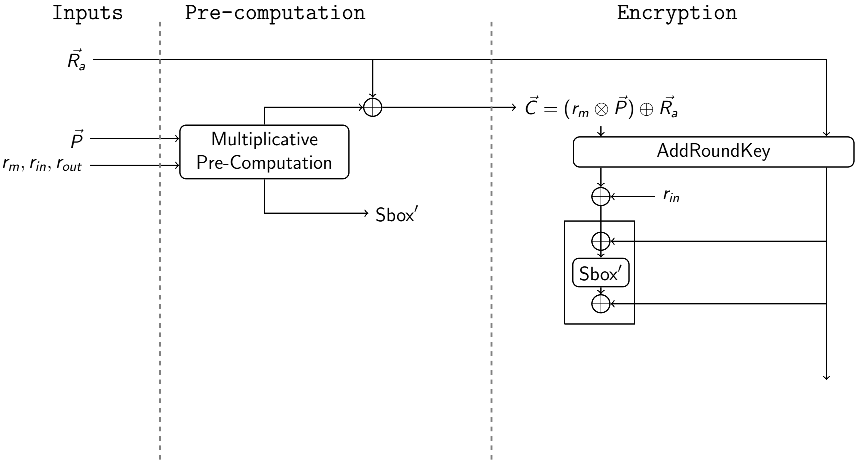
Countermeasures



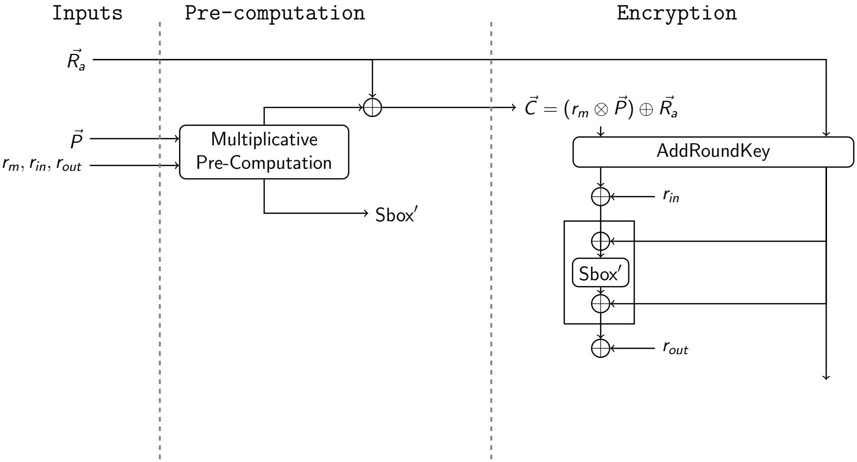
Countermeasures



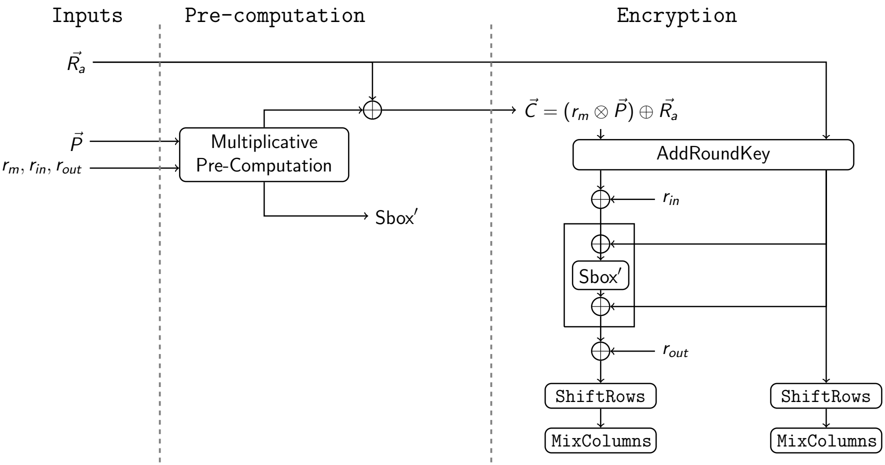
Countermeasures



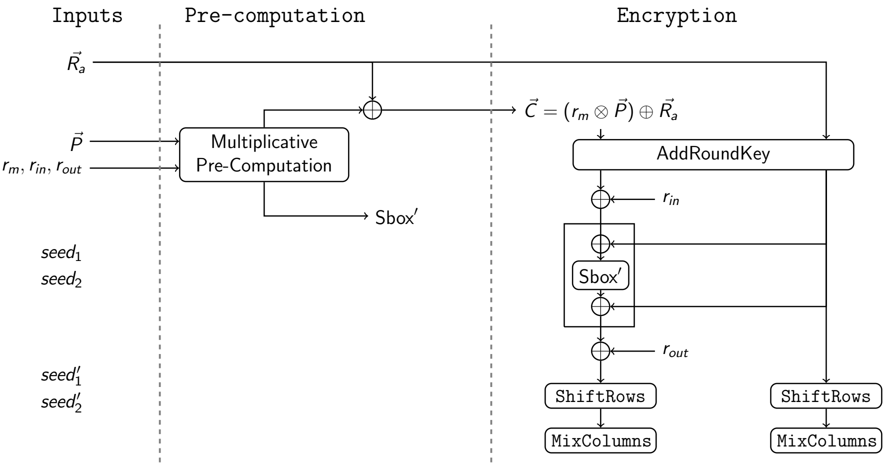
Countermeasures



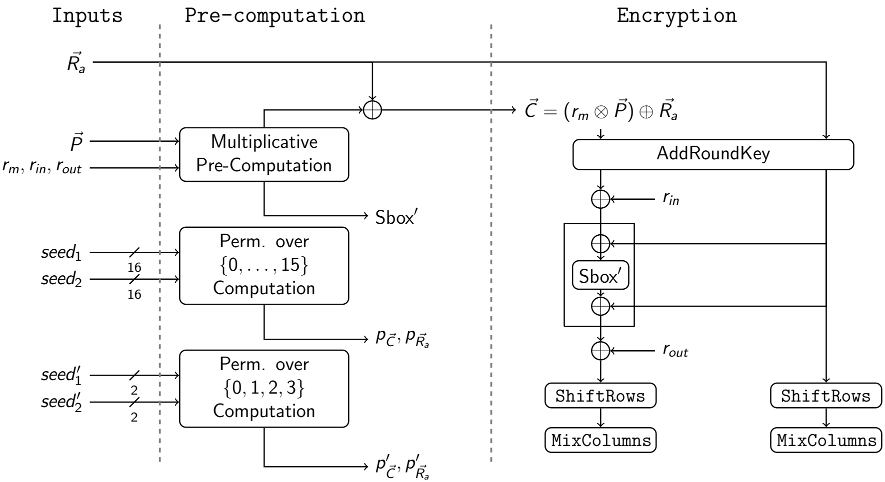
Countermeasures



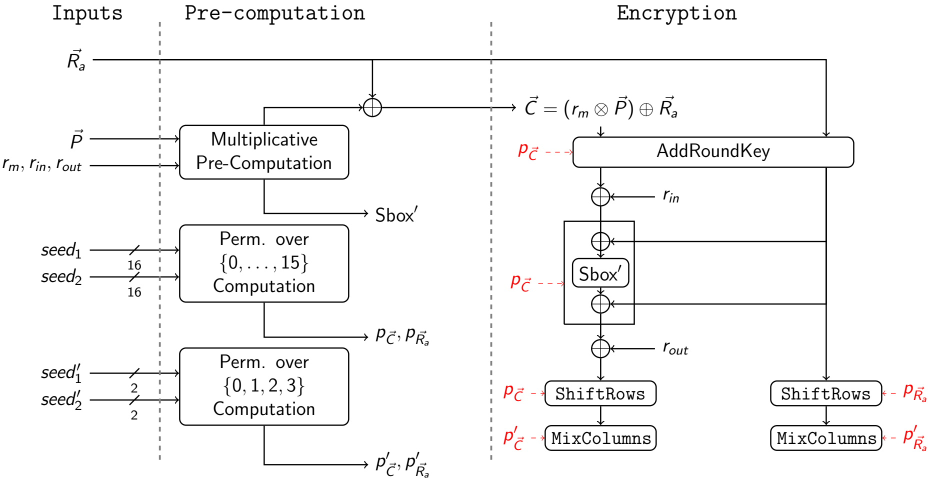
Countermeasures



Countermeasures



Countermeasures



Optimal Distinguisher

Profiled attacks are based on secret conditional distribution which depends on the countermeasures.

Full expression is written as

$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \sum_{r_a} \sum_{o_1} \sum_{o_2} \mathrm{f}[\vec{l}|r_m, r_a, c, o_1, o_2]$$

Optimal Distinguisher

Profiled attacks are based on secret conditional distribution which depends on the countermeasures.

Full expression is written as

$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \sum_{r_a} \sum_{o_1} \sum_{o_2} \mathrm{f}[\vec{l}|r_m, r_a, c, o_1, o_2]$$

Mult. mask

Optimal Distinguisher

Profiled attacks are based on secret conditional distribution which depends on the countermeasures.

Full expression is written as

Add. mask

$$f[\vec{l}|x] \propto \sum_{r_m} \sum_{r_a} \sum_{o_1} \sum_{o_2} f[\vec{l}|r_m, r_a, c, o_1, o_2]$$

Mult. mask

Optimal Distinguisher

Profiled attacks are based on secret conditional distribution which depends on the countermeasures.

Full expression is written as

Add. mask

$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \sum_{r_a} \sum_{o_1} \sum_{o_2} \mathrm{f}[\vec{l}|r_m, r_a, c, o_1, o_2]$$

Mult. mask

Perm. on shares

Optimal Distinguisher

Profiled attacks are based on secret conditional distribution which depends on the countermeasures.

Full expression is written as

Add. mask

Template

$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \sum_{r_a} \sum_{o_1} \sum_{o_2} \mathrm{f}[\vec{l}|r_m, r_a, c, o_1, o_2]$$

Mult. mask

Perm. on shares

Optimal Distinguisher

Profiled attacks are based on secret conditional distribution which depends on the countermeasures.

Full expression is written as

Add. mask   Template

$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \sum_{r_a} \sum_{o_1} \sum_{o_2} \mathrm{f}[\vec{l}|r_m, r_a, c, o_1, o_2]$$

Mult. mask

Perm. on shares

Optimal but rapidly out of reach:

Optimal Distinguisher

Profiled attacks are based on secret conditional distribution which depends on the countermeasures.

Full expression is written as

Add. mask

Template

$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \sum_{r_a} \sum_{o_1} \sum_{o_2} \mathrm{f}[\vec{l}|r_m, r_a, c, o_1, o_2]$$

Mult. mask

Perm. on shares

Optimal but rapidly out of reach:

- One template per randomness combination

Optimal Distinguisher

Profiled attacks are based on secret conditional distribution which depends on the countermeasures.

Full expression is written as

Add. mask — Template

$$f[\vec{l}|x] \propto \sum_{r_m} \sum_{r_a} \sum_{o_1} \sum_{o_2} f[\vec{l}|r_m, r_a, c, o_1, o_2]$$

Mult. mask — Perm. on shares

Optimal but rapidly out of reach:

- One template per randomness combination
- Sum over all the possible randomness

Optimal Distinguisher

Profiled attacks are based on secret conditional distribution which depends on the countermeasures.

Full expression is written as

Add. mask — Template

$$f[\vec{l}|x] \propto \sum_{r_m} \sum_{r_a} \sum_{o_1} \sum_{o_2} f[\vec{l}|r_m, r_a, c, o_1, o_2]$$

Mult. mask — Perm. on shares

Optimal but rapidly out of reach:

- One template per randomness combination
- Sum over all the possible randomness

$\Longrightarrow$ Hypotheses needed

Countermeasures Dissection

Assuming $\perp$ leakages on secret:

$$
\begin{aligned}
\mathrm{f}[\vec{l}|x] \propto \textstyle\sum_{r_m} \Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \\
\cdot \left(\textstyle\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot \Pr[o_1|\vec{l}_{o_1}]\right) \\
\cdot \left(\textstyle\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot \Pr[o_2|\vec{l}_{o_2}]\right)
\end{aligned}
$$

Countermeasures Dissection

Mult. mask ↙

Assuming $\perp$ leakages on secret:

$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \cdot \left(\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot \Pr[o_1|\vec{l}_{o_1}]\right) \cdot \left(\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot \Pr[o_2|\vec{l}_{o_2}]\right)$$

Countermeasures Dissection

Mult. mask

Add. mask + Perm

Assuming $\perp$ leakages on secret:

$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \cdot \left(\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot \Pr[o_1|\vec{l}_{o_1}]\right) \cdot \left(\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot \Pr[o_2|\vec{l}_{o_2}]\right)$$

Countermeasures Dissection

Mult. mask

Add. mask + Perm

Assuming $\perp$ leakages on secret:

$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \cdot \left(\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot \Pr[o_1|\vec{l}_{o_1}]\right) \cdot \left(\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot \Pr[o_2|\vec{l}_{o_2}]\right)$$

Enc. + Perm

Countermeasures Dissection

Assuming $\perp$ leakages on secret:

Mult. mask

Add. mask + Perm

$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \cdot \left(\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot \Pr[o_1|\vec{l}_{o_1}]\right) \cdot \left(\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot \Pr[o_2|\vec{l}_{o_2}]\right)$$

Enc. + Perm

Countermeasures' Dissection:

Countermeasures Dissection

Mult. mask

Add. mask + Perm

Assuming $\perp$ leakages on secret:

$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \cdot \left(\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot \Pr[o_1|\vec{l}_{o_1}]\right) \cdot \left(\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot \Pr[o_2|\vec{l}_{o_2}]\right)$$

Enc. + Perm

Countermeasures' Dissection:

- What: From combined countermeasures, expected multiplicative effect

Countermeasures Dissection

Mult. mask

Add. mask + Perm

Assuming $\perp$ leakages on secret:

$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \cdot \left(\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot \Pr[o_1|\vec{l}_{o_1}]\right) \cdot \left(\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot \Pr[o_2|\vec{l}_{o_2}]\right)$$

Enc. + Perm

Countermeasures' Dissection:

- What: From combined countermeasures, expected multiplicative effect
  - Reduce it to a small factor, ideally of 1.

Countermeasures Dissection

Mult. mask

Add. mask + Perm

Assuming $\perp$ leakages on secret:

$$f[\vec{l}|x] \propto \sum_{r_m} \Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \cdot \left(\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot \Pr[o_1|\vec{l}_{o_1}]\right) \cdot \left(\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot \Pr[o_2|\vec{l}_{o_2}]\right)$$

Enc. + Perm

Countermeasures' Dissection:

- What: From combined countermeasures, expected multiplicative effect
  - Reduce it to a small factor, ideally of 1.
- How: Bias the sums by independent partial attacks on secrets (i.e. shares)

Countermeasures Dissection

Mult. mask

Add. mask + Perm

Assuming $\perp$ leakages on secret:

$$f[\vec{l}|x] \propto \sum_{r_m} \Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \cdot \left(\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot \Pr[o_1|\vec{l}_{o_1}]\right) \cdot \left(\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot \Pr[o_2|\vec{l}_{o_2}]\right)$$

Enc. + Perm

Countermeasures' Dissection:

- What: From combined countermeasures, expected multiplicative effect
  - Reduce it to a small factor, ideally of 1.
- How: Bias the sums by independent partial attacks on secrets (i.e. shares)
  - $\searrow$ attack time complexity because terms are removed

Countermeasures Dissection

Mult. mask

Add. mask + Perm

Assuming $\perp$ leakages on secret:

$$f[\vec{l}|x] \propto \sum_{r_m} \Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \cdot \left(\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot \Pr[o_1|\vec{l}_{o_1}]\right) \cdot \left(\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot \Pr[o_2|\vec{l}_{o_2}]\right)$$

Enc. + Perm

Countermeasures' Dissection:

- What: From combined countermeasures, expected multiplicative effect
  - Reduce it to a small factor, ideally of 1.
- How: Bias the sums by independent partial attacks on secrets (i.e. shares)
  - $\searrow$ attack time complexity because terms are removed
  - $\searrow$ number of templates because not joint on all randomness

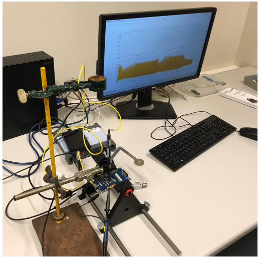
Content

Measurement Setup

Composed of

- `Cortex-M4 Atmel`
- High end EM Probe
- `PicoScope 5000` series sampling at 1GHz

Measurement Setup



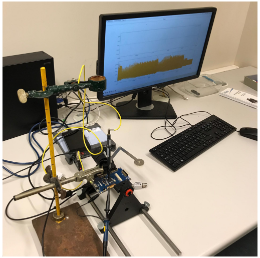
Composed of

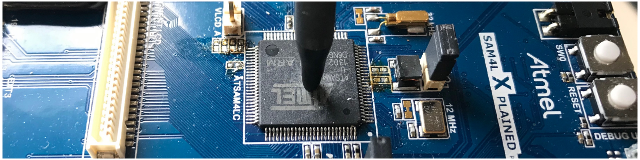
- `Cortex-M4 Atmel`
- High end EM Probe
- `PicoScope 5000` series sampling at 1GHz

Measurement Setup

Composed of

- Cortex-M4 Atmel
- High end EM Probe
- PicoScope 5000 series sampling at 1GHz

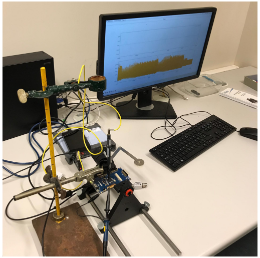


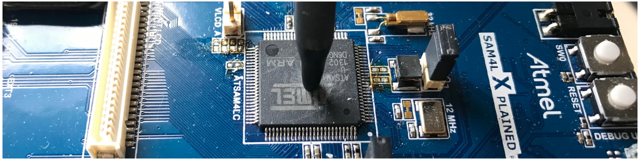


Measurement Setup

Composed of

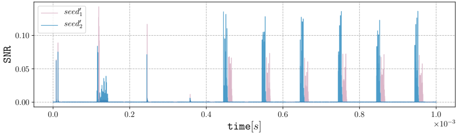
- `Cortex-M4 Atmel`
- High end EM Probe
- `PicoScope 5000` series sampling at 1GHz





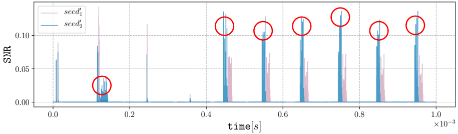
How to extract information in  ?

Profiling (e.g., permutation)



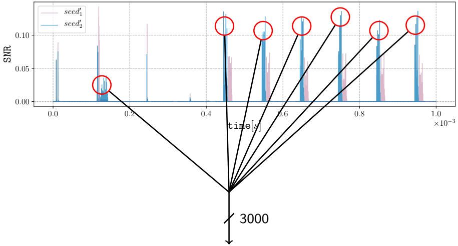
1. Compute SNR

Profiling (e.g., permutation)



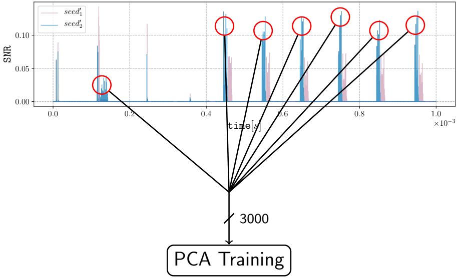
1. Compute SNR
2. Select points of interest

Profiling (e.g., permutation)



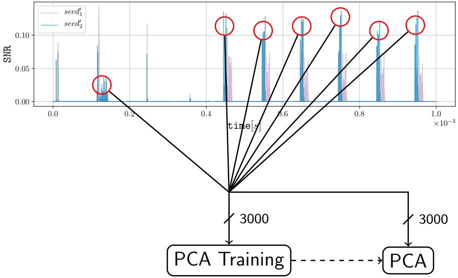
1. Compute SNR
2. Select points of interest

Profiling (e.g., permutation)



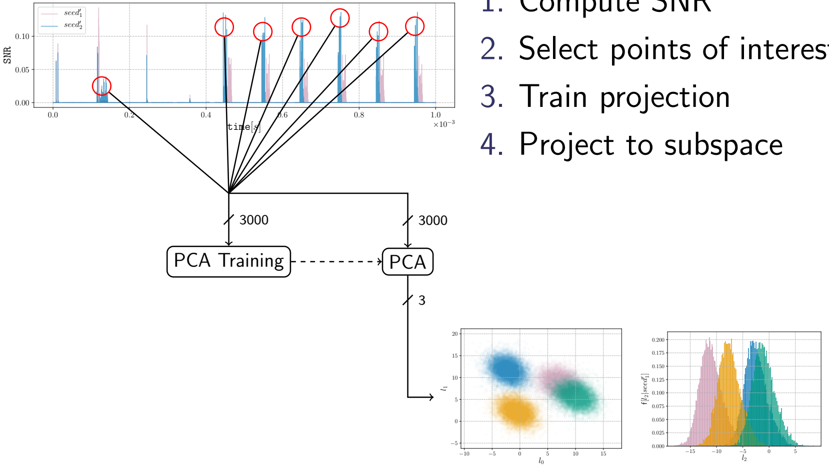
1. Compute SNR
2. Select points of interest
3. Train projection

Profiling (e.g., permutation)



1. Compute SNR
2. Select points of interest
3. Train projection

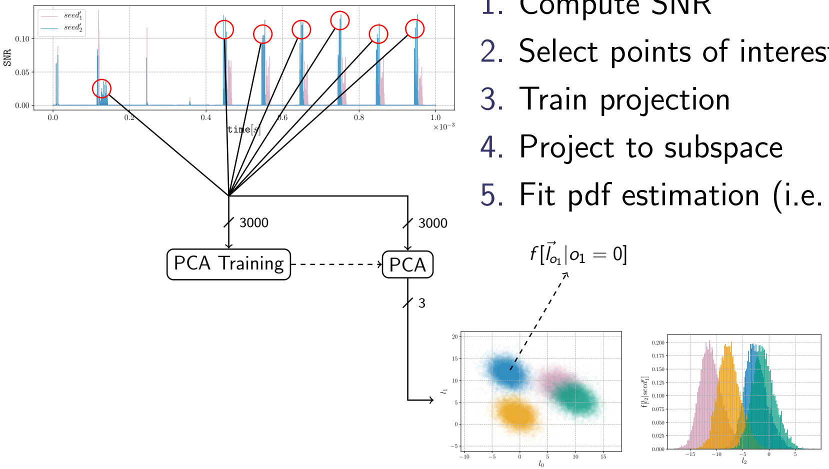
Profiling (e.g., permutation)



1. Compute SNR
2. Select points of interest
3. Train projection
4. Project to subspace

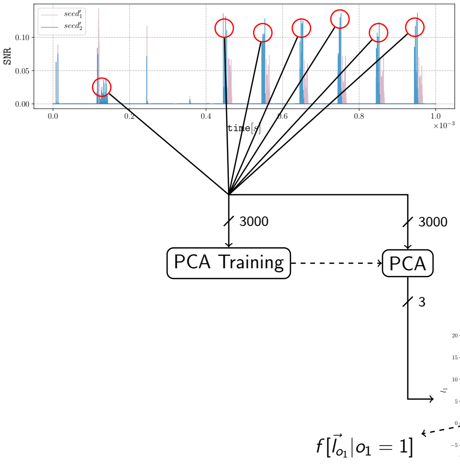
Profiling (e.g., permutation)

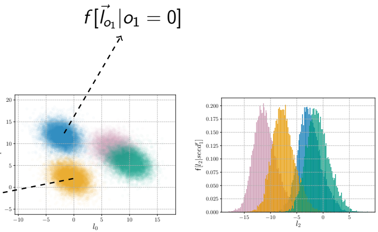
1. Compute SNR
2. Select points of interest
3. Train projection
4. Project to subspace
5. Fit pdf estimation (i.e. gauss.)



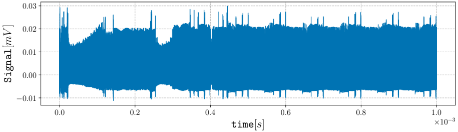
Profiling (e.g., permutation)

1. Compute SNR
2. Select points of interest
3. Train projection
4. Project to subspace
5. Fit pdf estimation (i.e. gauss.)



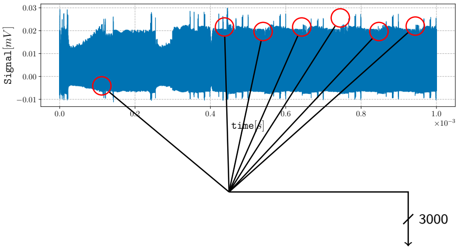


Partial Attacks



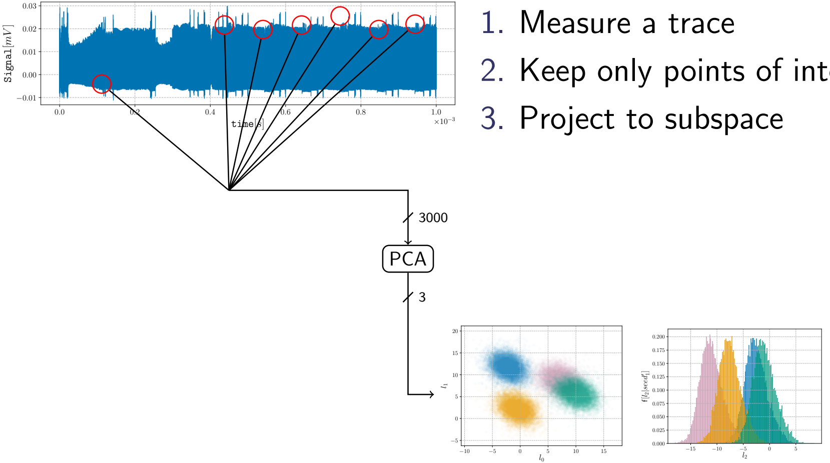
1. Measure a trace

Partial Attacks



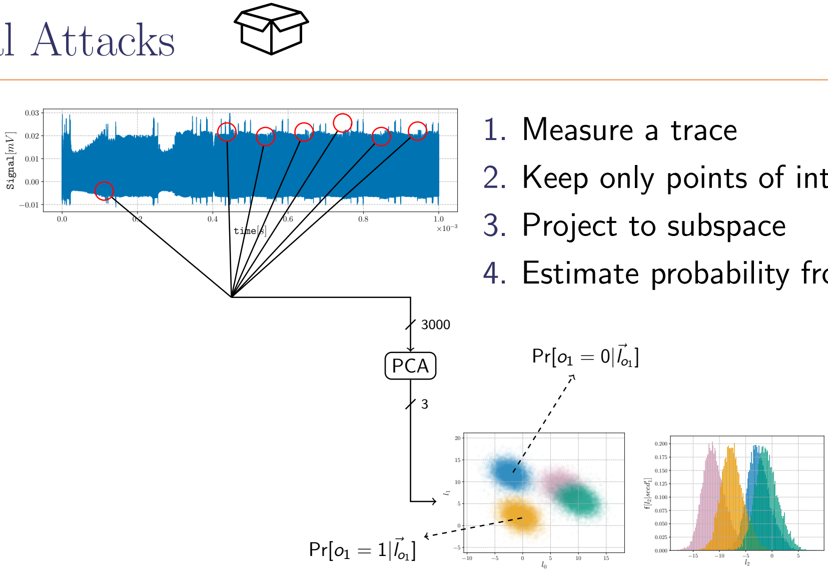
1. Measure a trace
2. Keep only points of interest

Partial Attacks



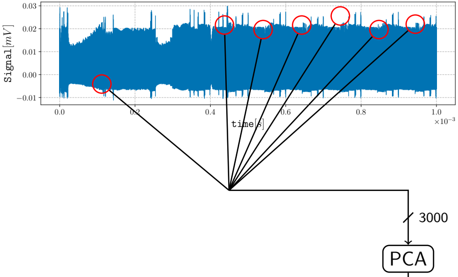
1. Measure a trace
2. Keep only points of interest
3. Project to subspace

Partial Attacks



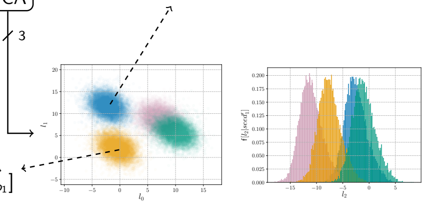
1. Measure a trace
2. Keep only points of interest
3. Project to subspace
4. Estimate probability from pdf

Partial Attacks

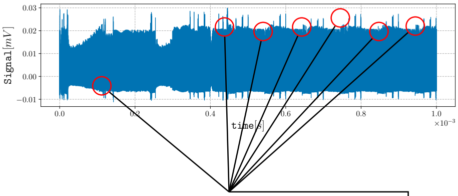


1. Measure a trace
2. Keep only points of interest
3. Project to subspace
4. Estimate probability from pdf

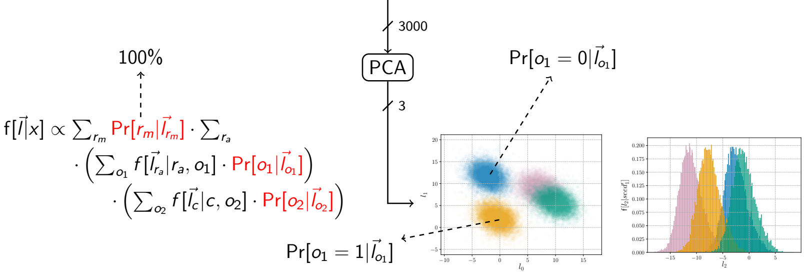
$$\mathrm{f}[\vec{l}|x] \propto \sum_{r_m} \Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \cdot \left(\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot \Pr[o_1|\vec{l}_{o_1}]\right) \cdot \left(\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot \Pr[o_2|\vec{l}_{o_2}]\right)$$



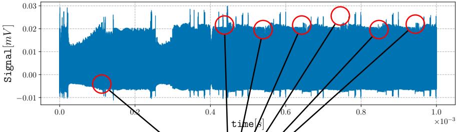
Partial Attacks



1. Measure a trace
2. Keep only points of interest
3. Project to subspace
4. Estimate probability from pdf



Partial Attacks



1. Measure a trace
2. Keep only points of interest
3. Project to subspace
4. Estimate probability from pdf

3000

PCA

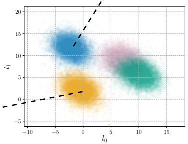
3

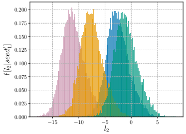
100%

98%

$$f[\vec{l}|x] \propto \sum_{r_m} Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \cdot \left(\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot Pr[o_1|\vec{l}_{o_1}]\right) \cdot \left(\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot Pr[o_2|\vec{l}_{o_2}]\right)$$

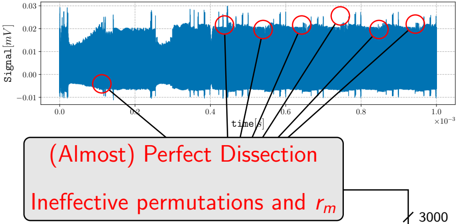
$Pr[o_1 = 0|\vec{l}_{o_1}]$





$Pr[o_1 = 1|\vec{l}_{o_1}]$

Partial Attacks

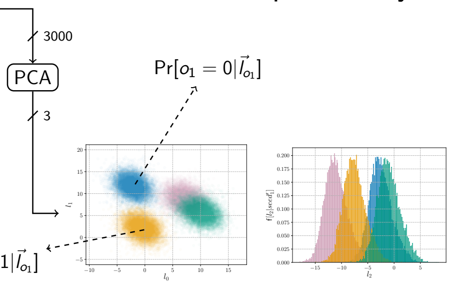


1. Measure a trace
2. Keep only points of interest
3. Project to subspace
4. Estimate probability from pdf

100%

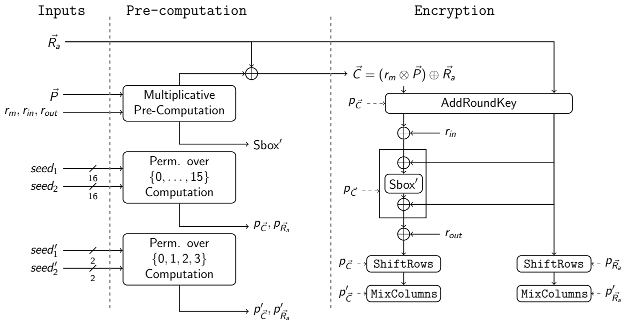
98%

$$f[\vec{l}|x] \propto \sum_{r_m} \Pr[r_m|\vec{l}_{r_m}] \cdot \sum_{r_a} \cdot \left(\sum_{o_1} f[\vec{l}_{r_a}|r_a, o_1] \cdot \Pr[o_1|\vec{l}_{o_1}]\right) \cdot \left(\sum_{o_2} f[\vec{l}_c|c, o_2] \cdot \Pr[o_2|\vec{l}_{o_2}]\right)$$

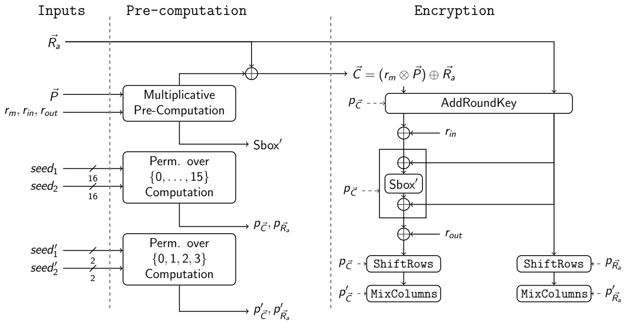


Content

Attack Path's

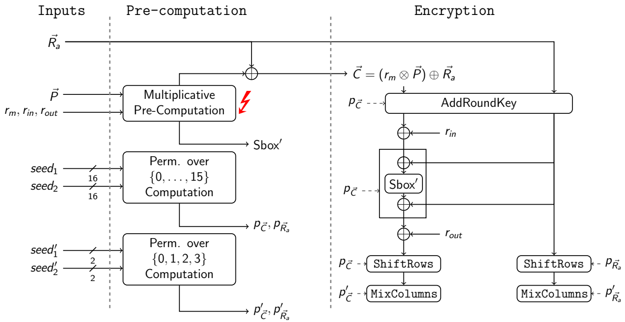


Attack Path's



Attacker should at least:

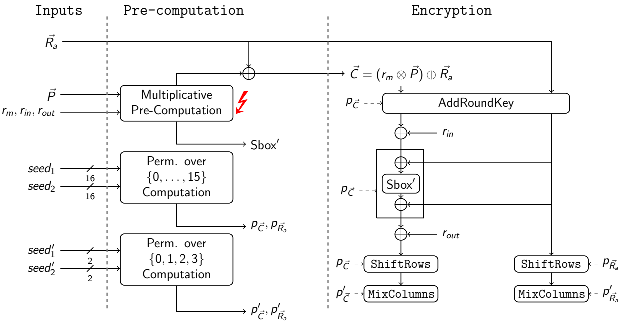
Attack Path's



Attacker should at least:

- Get information r_m

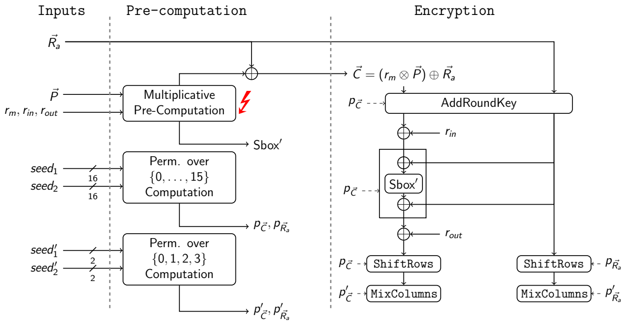
Attack Path's



Attacker should at least:

- Get information r_m
- Get information r_a and c

Attack Path's

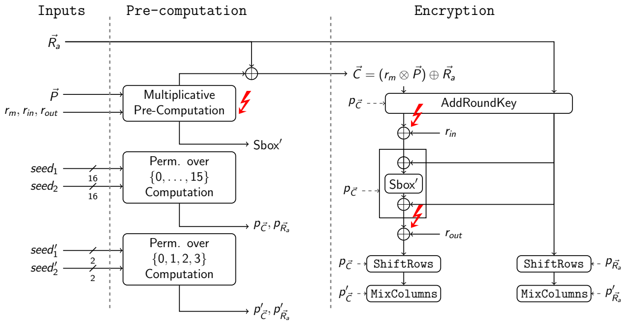


Attacker should at least:

- Get information r_m
- Get information r_a and c

Uneven shuffling:

Attack Path's



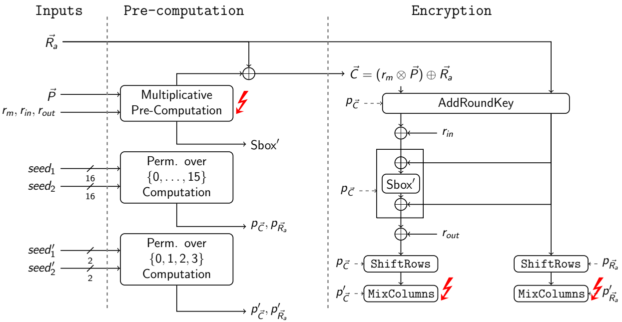
Attacker should at least:

- Get information r_m
- Get information r_a and c

Uneven shuffling:

- No permutation

Attack Path's



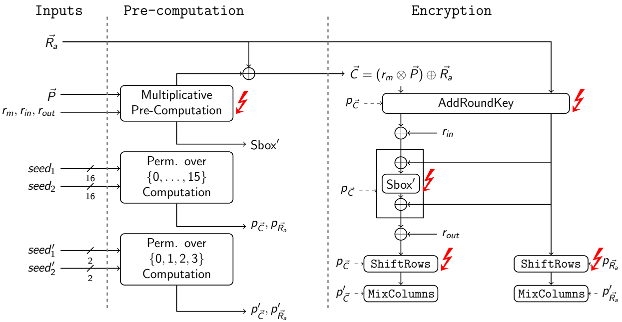
Attacker should at least:

- Get information r_m
- Get information r_a and c

Uneven shuffling:

- No permutation
- 2-bit seeded permutations

Attack Path's



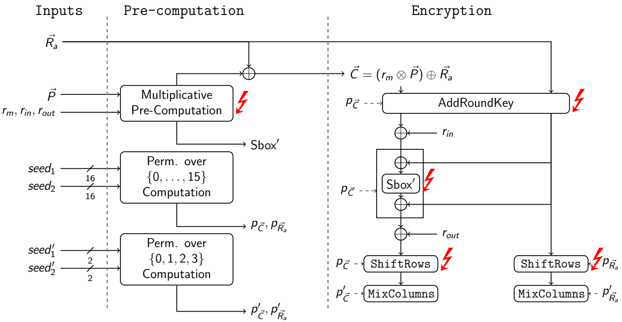
Attacker should at least:

- Get information r_m
- Get information r_a and c

Uneven shuffling:

- No permutation
- 2-bit seeded permutations
- 16-bit seeded permutations

Attack Path's



Attacker should at least:

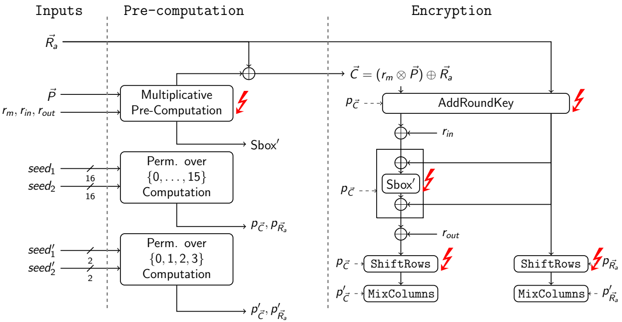
- Get information r_m
- Get information r_a and c

Uneven shuffling:

- No permutation
- 2-bit seeded permutations
- 16-bit seeded permutations

- All permutations can be enumerated

Attack Path's



Attacker should at least:

- Get information r_m
- Get information r_a and c

Uneven shuffling:

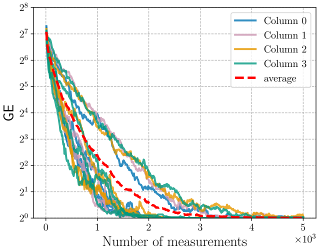
- No permutation
- 2-bit seeded permutations
- 16-bit seeded permutations

- All permutations can be enumerated
- We focus on the 2-bit seeded permutation

Attack Results

Divide & Conquer:

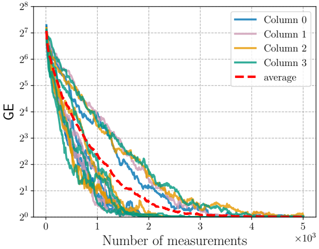
Attack Results



Divide & Conquer:

1. On each 16 bytes:

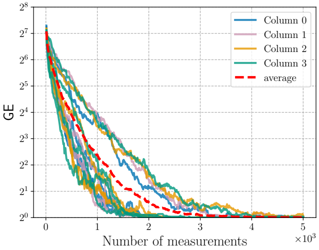
Attack Results



Divide & Conquer:

1. On each 16 bytes:
   - Entropy ↘ with measurements

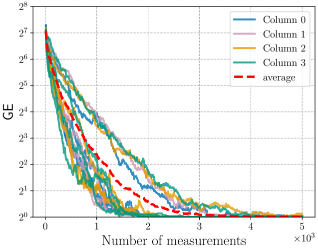
Attack Results



Divide & Conquer:

1. On each 16 bytes:
   - Entropy $\searrow$ with measurements
   - Less than a bit with $3,000$ traces

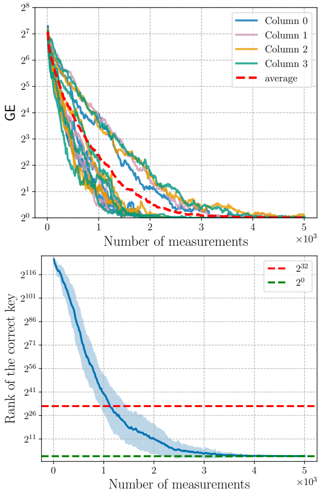
Attack Results



Divide & Conquer:

1. On each 16 bytes:
   - Entropy ↘ with measurements
   - Less than a bit with $3,000$ traces
   - One "*harder*" byte per column

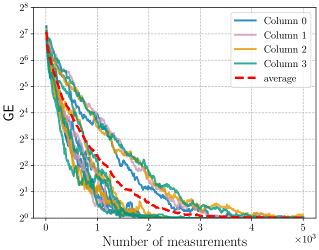
Attack Results

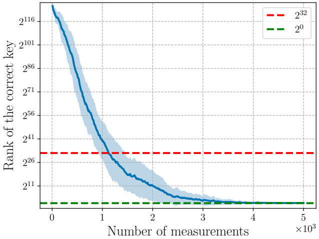


Divide & Conquer:

1. On each 16 bytes:
   - Entropy ↘ with measurements
   - Less than a bit with 3, 000 traces
   - One "*harder*" byte per column
2. On full key:

Attack Results

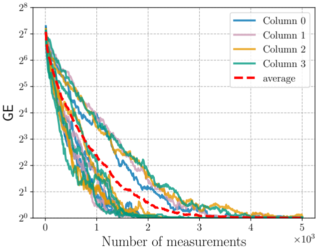


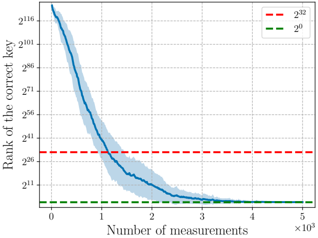


Divide & Conquer:

1. On each 16 bytes:
   - Entropy ↘ with measurements
   - Less than a bit with 3,000 traces
   - One "*harder*" byte per column
2. On full key:
   - Entropy ↘ with measurements

Attack Results

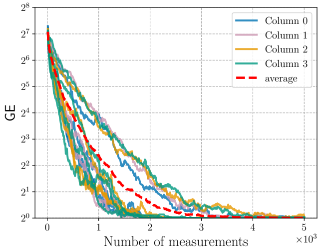


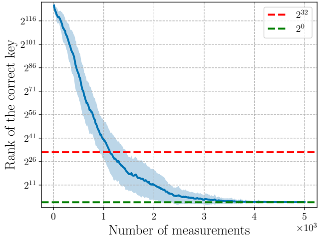


Divide & Conquer:

1. On each 16 bytes:
   - Entropy ↘ with measurements
   - Less than a bit with $3,000$ traces
   - One "*harder*" byte per column
2. On full key:
   - Entropy ↘ with measurements
   - Less than a bit with $4,000$ traces

Attack Results

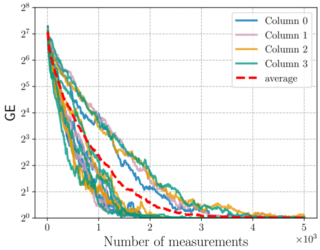


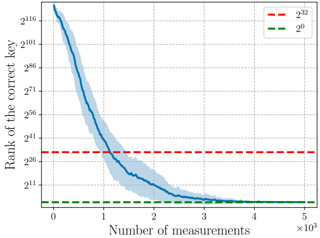


Divide & Conquer:

1. On each 16 bytes:
   - Entropy $\searrow$ with measurements
   - Less than a bit with $3,000$ traces
   - One "*harder*" byte per column
2. On full key:
   - Entropy $\searrow$ with measurements
   - Less than a bit with $4,000$ traces
   - About $1,100$ with post-processing

Attack Results





Divide & Conquer:

1. On each 16 bytes:
   - Entropy ↘ with measurements
   - Less than a bit with $3,000$ traces
   - One "*harder*" byte per column
2. On full key:
   - Entropy ↘ with measurements
   - Less than a bit with $4,000$ traces
   - About $1,100$ with post-processing

Full key in 1 minute of measurement

Content

Can this be automated in  ?

How the knowledge of the target helps in a worst-case evaluation ?

Can this be automated in ?

How the knowledge of the target helps in a worst-case evaluation ?

- Evaluators do not always have full control on the target

Can this be automated in ?

How the knowledge of the target helps in a worst-case evaluation ?

- Evaluators do not always have full control on the target
- If it helps, worrying for long term security:

Can this be automated in  ?

How the knowledge of the target helps in a worst-case evaluation ?

- Evaluators do not always have full control on the target
- If it helps, worrying for long term security:
  - Adversary with a better strategy can be more powerful than the evaluator

Can this be automated in  ?

How the knowledge of the target helps in a worst-case evaluation ?

- Evaluators do not always have full control on the target
- If it helps, worrying for long term security:
  - Adversary with a better strategy can be more powerful than the evaluator

Experiments with machine learning:

Can this be automated in  ?

How the knowledge of the target helps in a worst-case evaluation ?

- Evaluators do not always have full control on the target
- If it helps, worrying for long term security:
  - Adversary with a better strategy can be more powerful than the evaluator

Experiments with machine learning:

- Representative of closed approach since able to deal with unknown countermeasures

Can this be automated in ?

How the knowledge of the target helps in a worst-case evaluation ?

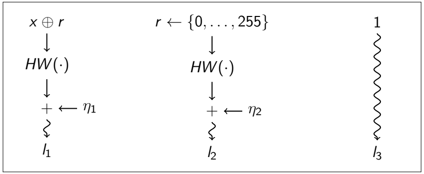
- Evaluators do not always have full control on the target
- If it helps, worrying for long term security:
  - Adversary with a better strategy can be more powerful than the evaluator

Experiments with machine learning:

- Representative of closed approach since able to deal with unknown countermeasures
- We instantiate MLP classifiers in simulated settings

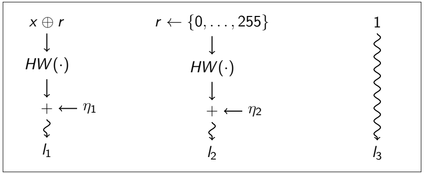
Simulated Experimental Setting

Simulated Experimental Setting



Boolean Masking with leakage on:

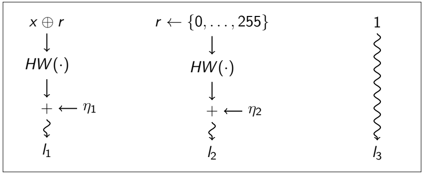
Simulated Experimental Setting



Boolean Masking with leakage on:

- Two shares

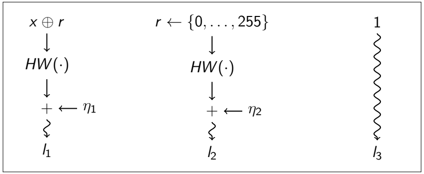
Simulated Experimental Setting



Boolean Masking with leakage on:

- Two shares
- Hamming weight + Gaussian noise

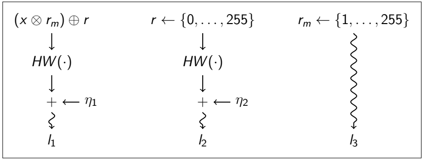
Simulated Experimental Setting



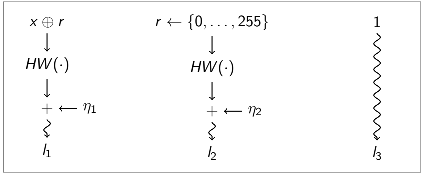
Boolean Masking with leakage on:

- Two shares
- Hamming weight + Gaussian noise

Affine Masking with leakage on:



Simulated Experimental Setting

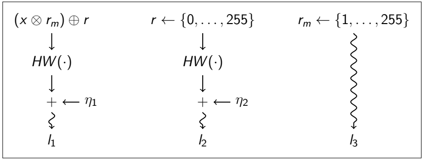


Boolean Masking with leakage on:

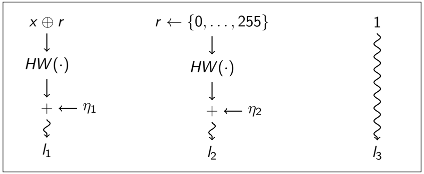
- Two shares
- Hamming weight + Gaussian noise

Affine Masking with leakage on:

- Two shares + Multiplicative mask



Simulated Experimental Setting

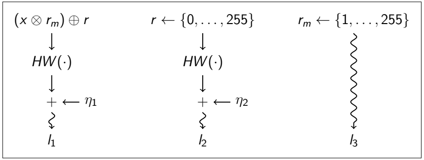


Boolean Masking with leakage on:

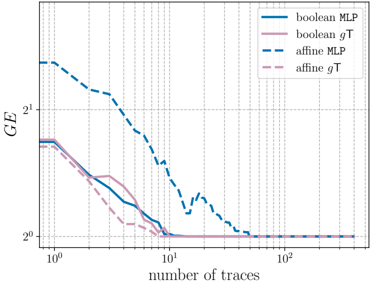
- Two shares
- Hamming weight + Gaussian noise

Affine Masking with leakage on:

- Two shares + Multiplicative mask
- Hamming weight + Gaussian noise



Comparison Open vs. Closed Approaches

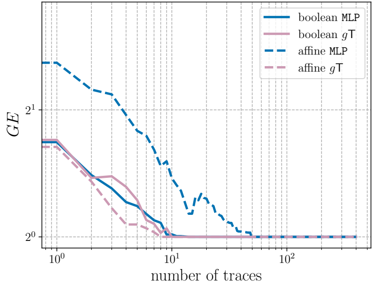


3-bit

Comparison Open vs. Closed Approaches

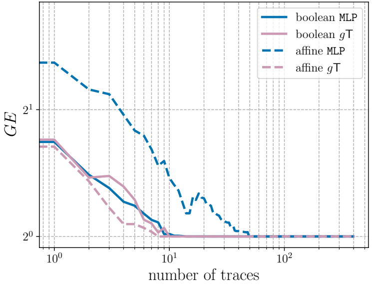
For :

For :



3-bit

Comparison Open vs. Closed Approaches



3-bit

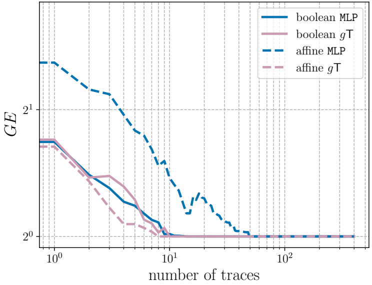
For :

- Schemes are equivalent

For :

- Schemes are not equivalent

Comparison Open vs. Closed Approaches



3-bit

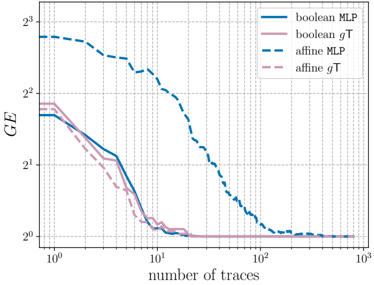
For  :

- Schemes are equivalent
- No need to learn multiplications

For  :

- Schemes are not equivalent
- Need to learn multiplications based on leakage

Comparison Open vs. Closed Approaches



4-bit

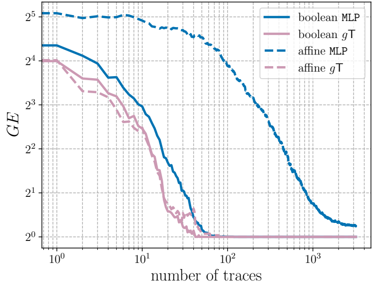
For :

- Schemes are equivalent
- No need to learn multiplications

For :

- Schemes are not equivalent
- Need to learn multiplications based on leakage

Comparison Open vs. Closed Approaches



6-bit

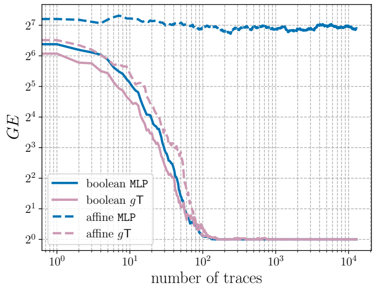
For :

- Schemes are equivalent
- No need to learn multiplications

For :

- Schemes are not equivalent
- Need to learn multiplications based on leakage

Comparison Open vs. Closed Approaches



8-bit

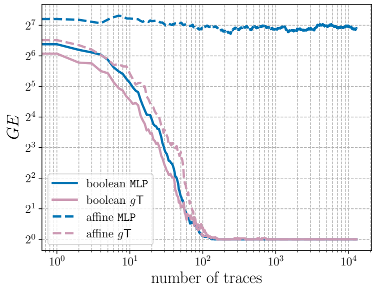
For :

- Schemes are equivalent
- No need to learn multiplications

For :

- Schemes are not equivalent
- Need to learn multiplications based on leakage
- Harder with ↗ field size

Comparison Open vs. Closed Approaches



8-bit

For :

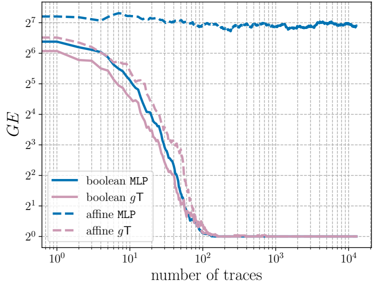
- Schemes are equivalent
- No need to learn multiplications

For :

- Schemes are not equivalent
- Need to learn multiplications based on leakage
- Harder with ↗ field size

- Profiling cost of such a closed evaluation will be prohibitive

Comparison Open vs. Closed Approaches



8-bit

For :

- Schemes are equivalent
- No need to learn multiplications

For :

- Schemes are not equivalent
- Need to learn multiplications based on leakage
- Harder with ↗ field size

- Profiling cost of such a closed evaluation will be prohibitive
- While comes for free in withe box

Content

Technical Summary

This analysis of mixed countermeasures shows:

Technical Summary

This analysis of mixed countermeasures shows:

- Online attack in less than a minute with:

Technical Summary

This analysis of mixed countermeasures shows:

- Online attack in less than a minute with:
  - With old state-of-the-art pdf estimation tools

Technical Summary

This analysis of mixed countermeasures shows:

- Online attack in less than a minute with:
  - With old state-of-the-art pdf estimation tools
  - Some equations depending on the countermeasures

Technical Summary

This analysis of mixed countermeasures shows:

- Online attack in less than a minute with:
  - With old state-of-the-art pdf estimation tools
  - Some equations depending on the countermeasures
  - Sounded hypotheses

Technical Summary

This analysis of mixed countermeasures shows:

- Online attack in less than a minute with:
  - With old state-of-the-art pdf estimation tools
  - Some equations depending on the countermeasures
  - Sounded hypotheses
- Preliminary leakage assessment found no weakness with 100,000 traces

Technical Summary

This analysis of mixed countermeasures shows:

- Online attack in less than a minute with:
  - With old state-of-the-art pdf estimation tools
  - Some equations depending on the countermeasures
  - Sounded hypotheses
- Preliminary leakage assessment found no weakness with 100,000 traces
- Difficulty to protect 32-bit software:

Technical Summary

This analysis of mixed countermeasures shows:

- Online attack in less than a minute with:
  - With old state-of-the-art pdf estimation tools
  - Some equations depending on the countermeasures
  - Sounded hypotheses
- Preliminary leakage assessment found no weakness with 100,000 traces
- Difficulty to protect 32-bit software:
  - Inherent to low noise on the platform and not to optimized shuffling

Technical Summary

This analysis of mixed countermeasures shows:

- Online attack in less than a minute with:
  - With old state-of-the-art pdf estimation tools
  - Some equations depending on the countermeasures
  - Sounded hypotheses
- Preliminary leakage assessment found no weakness with 100,000 traces
- Difficulty to protect 32-bit software:
  - Inherent to low noise on the platform and not to optimized shuffling

Knowledge needed to reproduce on other targets :

Technical Summary

This analysis of mixed countermeasures shows:

- Online attack in less than a minute with:
  - With old state-of-the-art pdf estimation tools
  - Some equations depending on the countermeasures
  - Sounded hypotheses
- Preliminary leakage assessment found no weakness with 100,000 traces
- Difficulty to protect 32-bit software:
  - Inherent to low noise on the platform and not to optimized shuffling

Knowledge needed to reproduce on other targets :

- Source code and randomness knowledge during profiling

Technical Summary

This analysis of mixed countermeasures shows:

- Online attack in less than a minute with:
  - With old state-of-the-art pdf estimation tools
  - Some equations depending on the countermeasures
  - Sounded hypotheses
- Preliminary leakage assessment found no weakness with 100,000 traces
- Difficulty to protect 32-bit software:
  - Inherent to low noise on the platform and not to optimized shuffling

Knowledge needed to reproduce on other targets :

- Source code and randomness knowledge during profiling
- Sufficient understanding of countermeasures

Technical Summary

This analysis of mixed countermeasures shows:

- Online attack in less than a minute with:
  - With old state-of-the-art pdf estimation tools
  - Some equations depending on the countermeasures
  - Sounded hypotheses
- Preliminary leakage assessment found no weakness with 100,000 traces
- Difficulty to protect 32-bit software:
  - Inherent to low noise on the platform and not to optimized shuffling

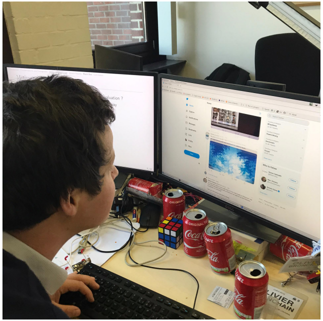
Knowledge needed to reproduce on other targets :

- Source code and randomness knowledge during profiling
- Sufficient understanding of countermeasures
- Not so much time !

Time Line

Time Line

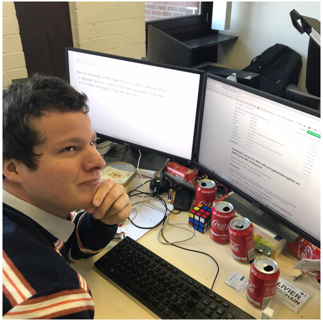
Day 0: Code is Online



Scrolling Twitter

Time Line

Day 0: Code is Online

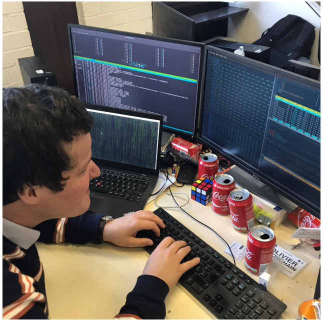


Code Available

Time Line

Day 0: Code is Online

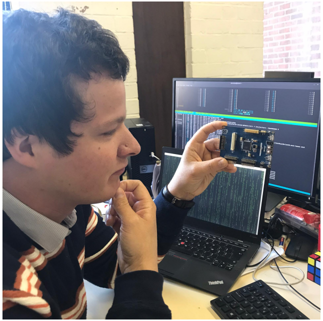
Day 1: Start looking at it



Entering Hacker Mode

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it



Finding MCU

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it

Removing Capacitors

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it

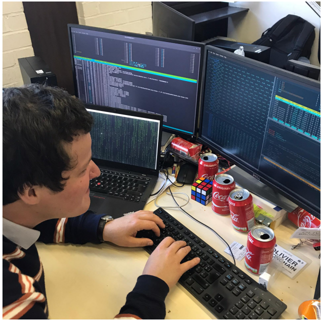
Engraving EM Probe

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready



Entering Hacker Mode

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready
- **Day 6**: Multiplicative mask recovery

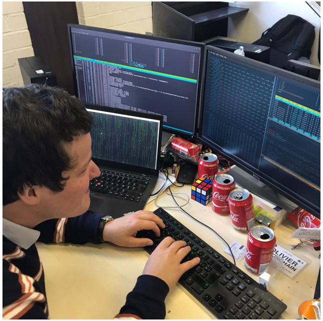
Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready
- **Day 6**: Multiplicative mask recovery

Really Happy

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready
- **Day 6**: Multiplicative mask recovery



Entering Hacker Mode

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready
- **Day 6**: Multiplicative mask recovery
- **Day 10**: First attacks

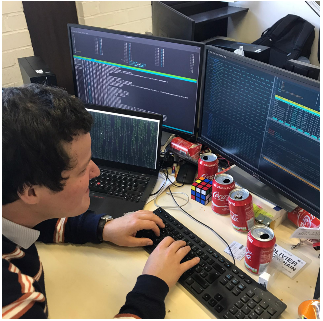
Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready
- **Day 6**: Multiplicative mask recovery
- **Day 10**: First attacks

Really Happy

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready
- **Day 6**: Multiplicative mask recovery
- **Day 10**: First attacks



Entering Hacker Mode

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready
- **Day 6**: Multiplicative mask recovery
- **Day 10**: First attacks
- **Day 11**: Key enumeration

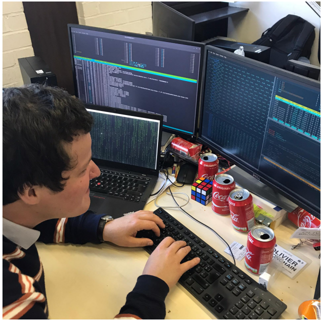
Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready
- **Day 6**: Multiplicative mask recovery
- **Day 10**: First attacks
- **Day 11**: Key enumeration

Really Happy

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready
- **Day 6**: Multiplicative mask recovery
- **Day 10**: First attacks
- **Day 11**: Key enumeration



Entering Hacker Mode

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready
- **Day 6**: Multiplicative mask recovery
- **Day 10**: First attacks
- **Day 11**: Key enumeration
- **Day 15**: Full attack

Time Line

- **Day 0**: Code is Online
- **Day 1**: Start looking at it
- **Day 5**: Setup ready
- **Day 6**: Multiplicative mask recovery
- **Day 10**: First attacks
- **Day 11**: Key enumeration
- **Day 15**: Full attack

Really Happy

Take Home Message

ANSSI's implementation was a stimulating first step:

Take Home Message

ANSSI's implementation was a stimulating first step:

- Nice research challenge to design/evaluate more secure implementations

Take Home Message

ANSSI's implementation was a stimulating first step:

- Nice research challenge to design/evaluate more secure implementations
- Possibly dealing with limited physical noise

Take Home Message

ANSSI's implementation was a stimulating first step:

- Nice research challenge to design/evaluate more secure implementations
- Possibly dealing with limited physical noise

Thanks !

Twitter: @BronchainO
email: olivier.bronchain@uclouvain.be