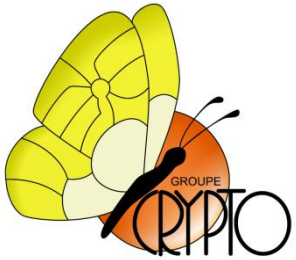
A Systematic Approach to the Side-Channel Analysis of ECC Implementations with Worst-Case Horizontal Attacks

Romain Poussier, François-Xavier Standaert: Université catholique de Louvain
Yuanyuan Zhou: Université catholique de Louvain & Brightsight BV



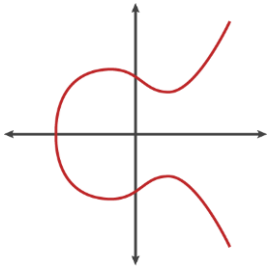
Outline

- Context and motivation
- Horizontal differential power attack: systematic framework
- Practical experiments
  - Setup
  - Points of interest
  - Result on Cortex-M4
  - Result on Cortex-A8
- Conclusion and future works

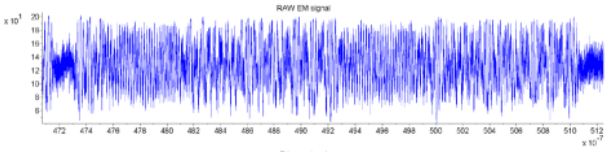
SCA on ECC: many options

Elliptic curve cryptography (ECC)

Scalar multiplication
$[k]P$



Side-channel attacks (SCA)



Many attack classes	Different tools
• DPA • Horizontal DPA • Template • Bit manipulation • Horizontal Collision • …	• Difference of mean • Correlation • Likelihood • Machine learning • …

Which attack to use for evaluation

Many attack classes	Different tools
• DPA • Horizontal DPA • Template • Bit manipulation • Horizontal Collision • …	• Difference of mean • Correlation • Likelihood • Machine learning • …

Which attack to use for a fixed time security evaluation?

Which attack to use for evaluation

Many attack classes	Different tools
• DPA • Horizontal DPA • Template • Bit manipulation • Horizontal Collision • …	• Difference of mean • Correlation • Likelihood • Machine learning • …

Which attack to use for a fixed time security evaluation?

Our general goal: approaching worst-case security

How: use most of the available side-channel information

State of the art

	# needed traces	Input point	Attacker's assumptions	# Information used
DPA	N	A posteriori known	Strong	Small
Template	1	A priori known	Strong	Customizable (first bits only)
Online template	1	A priori known	Very strong	Customizable
H-DPA	1	A posteriori known	Strong	Customizable
H-Collision	1	Not needed	Weak	Small
Bit manipulation	1	Not needed	Weak	Small

This study: contribution on H-DPA

Complex **framework**

- Systematic approach
- Close to worst-case with leakage characterization

Few practical experiments for H-DPA

- A to Z application
- Cortex-M4 (easy)
- Cortex-A8 (more challenging)

Teaser: promising future work shown at the end of the talk!

Outline

- Context and motivations
- Horizontal differential power attack: systematic framework
- Practical experiments
  - Setup
  - Points of interest
  - Result on Cortex-M4
  - Result on Cortex-A8
- Conclusion and future works

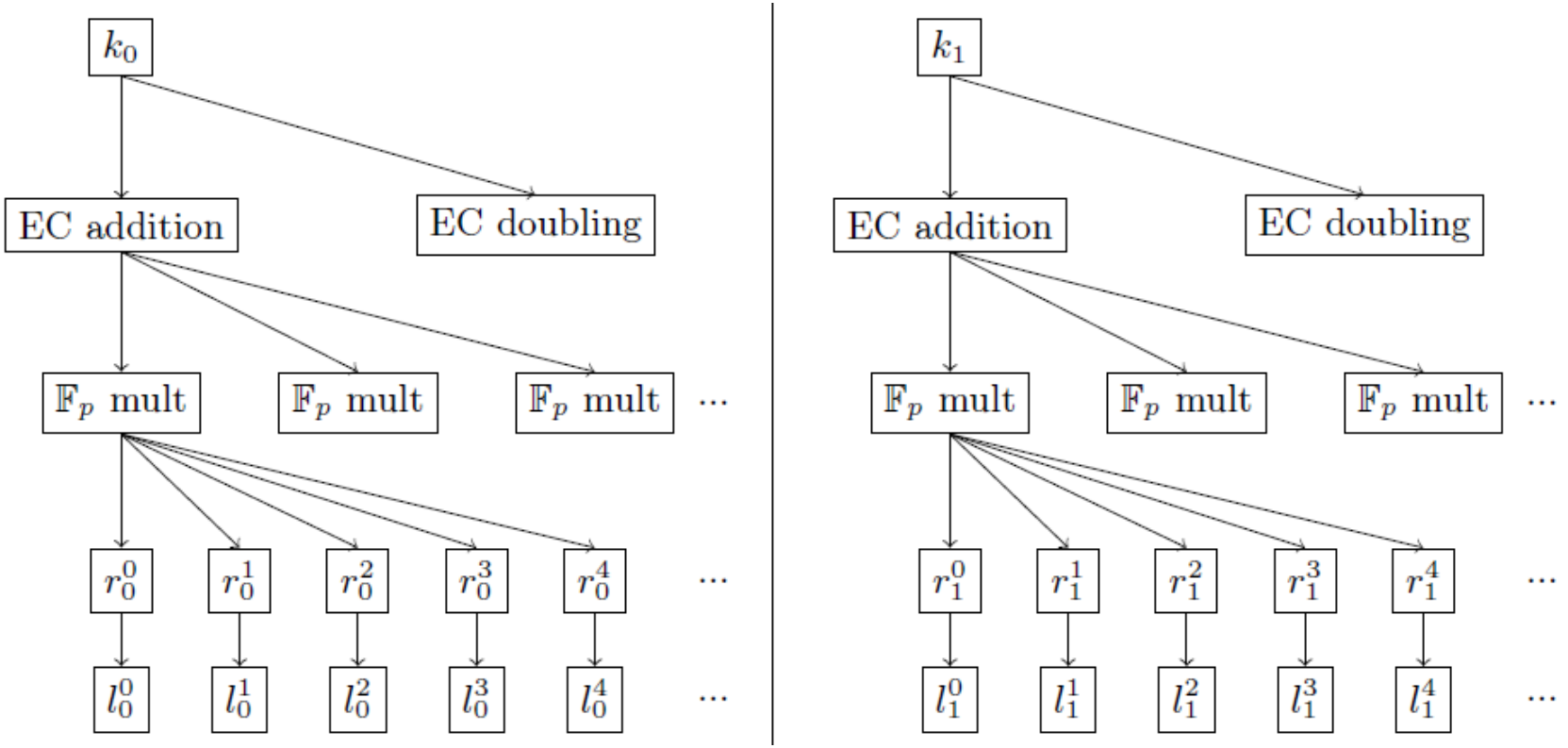
Elliptic curve scalar multiplication (ECSM)

Algorithm 1 Montgomery ladder.

Input: $P, \mathbf{k} = (k_0, ..., k_{n-1})$
Output: $[k]P$
  $R_0 \leftarrow \mathcal{O}$
  $R_1 \leftarrow P$
  for $i = 0$ to $n - 1$ **do**
    $R_{1-k_i} \leftarrow R_{1-k_i} + R_{k_i}$
    $R_{k_i} \leftarrow [2]R_{k_i}$
  end for
  return R_0

Note: only collision attack against this ECSM: Hanley et al. (CTRSA 2015)

Identify the information: abstract view of regular ECSM



Fixed and predictable sequence of register operations: N registers per scalar bit

Horizontal DPA: modus operandi

Algorithm 1 Montgomery ladder.

Input: $P, \mathbf{k} = (k_0, ..., k_{n-1})$
Output: $[k]P$
  $R_0 \leftarrow \mathcal{O}$
  $R_1 \leftarrow P$
  for $i = 0$ to $n - 1$ **do**
    $R_{1-k_i} \leftarrow R_{1-k_i} + R_{k_i}$
    $R_{k_i} \leftarrow [2]R_{k_i}$
  end for
  return R_0

HDPA attack on k_0:

1. Select several internal registers operations Rs that depends on P and k_0

Horizontal DPA: modus operandi

Algorithm 1 Montgomery ladder.

Input: $P, \mathbf{k} = (k_0, ..., k_{n-1})$
Output: $[k]P$
  $R_0 \leftarrow \mathcal{O}$
  $R_1 \leftarrow P$
  for $i = 0$ to $n - 1$ **do**
    $R_{1-k_i} \leftarrow R_{1-k_i} + R_{k_i}$
    $R_{k_i} \leftarrow [2]R_{k_i}$
  end for
  return R_0

HDPA attack on k_0:

1. Select several internal registers operations Rs that depends on P and k_0
2. Modelize the function L that characterizes how Rs leak: **information extraction**

Horizontal DPA: modus operandi

Algorithm 1 Montgomery ladder.

Input: $P, \mathbf{k} = (k_0, ..., k_{n-1})$
Output: $[k]P$
  $R_0 \leftarrow \mathcal{O}$
  $R_1 \leftarrow P$
  for $i = 0$ to $n - 1$ **do**
    $R_{1-k_i} \leftarrow R_{1-k_i} + R_{k_i}$
    $R_{k_i} \leftarrow [2]R_{k_i}$
  end for
  return R_0

HDPA attack on k_0:

1. Select several internal registers operations Rs that depends on P and k_0
2. Modelize the function L that characterizes how Rs leak: **information extraction**
3. Acquire 1 attack measurement

Horizontal DPA: modus operandi

Algorithm 1 Montgomery ladder.

Input: $P, \mathbf{k} = (k_0, ..., k_{n-1})$
Output: $[k]P$
  $R_0 \leftarrow \mathcal{O}$
  $R_1 \leftarrow P$
  for $i = 0$ to $n - 1$ **do**
    $R_{1-k_i} \leftarrow R_{1-k_i} + R_{k_i}$
    $R_{k_i} \leftarrow [2]R_{k_i}$
  end for
  return R_0

HDPA attack on k_0:

1. Select several internal registers operations Rs that depends on P and k_0
2. Modelize the function L that characterizes how Rs leak: **information extraction**
3. Acquire 1 attack measurement
4. Prepare two sets S_0 (resp. S_1) that contain the guesses for the values Rs_0 (resp. Rs_1) in function of P and $k_0 = 0$ (resp. $k_0 = 1$)

Horizontal DPA: modus operandi

Algorithm 1 Montgomery ladder.

Input: $P, \mathbf{k} = (k_0, ..., k_{n-1})$
Output: $[k]P$
  $R_0 \leftarrow \mathcal{O}$
  $R_1 \leftarrow P$
  for $i = 0$ to $n - 1$ **do**
    $R_{1-k_i} \leftarrow R_{1-k_i} + R_{k_i}$
    $R_{k_i} \leftarrow [2]R_{k_i}$
  end for
  return R_0

HDPA attack on k_0:

1. Select several internal registers operations Rs that depends on P and k_0
2. Modelize the function L that characterizes how Rs leak: **information extraction**
3. Acquire 1 attack measurement
4. Prepare two sets S_0 (resp. S_1) that contain the guesses for the values Rs_0 (resp. Rs_1) in function of P and $k_0 = 0$ (resp. $k_0 = 1$)
5. Compare $L(Rs_i)$ with the actual SCA leakages using a distinguisher D: **information combination**

Horizontal DPA: modus operandi

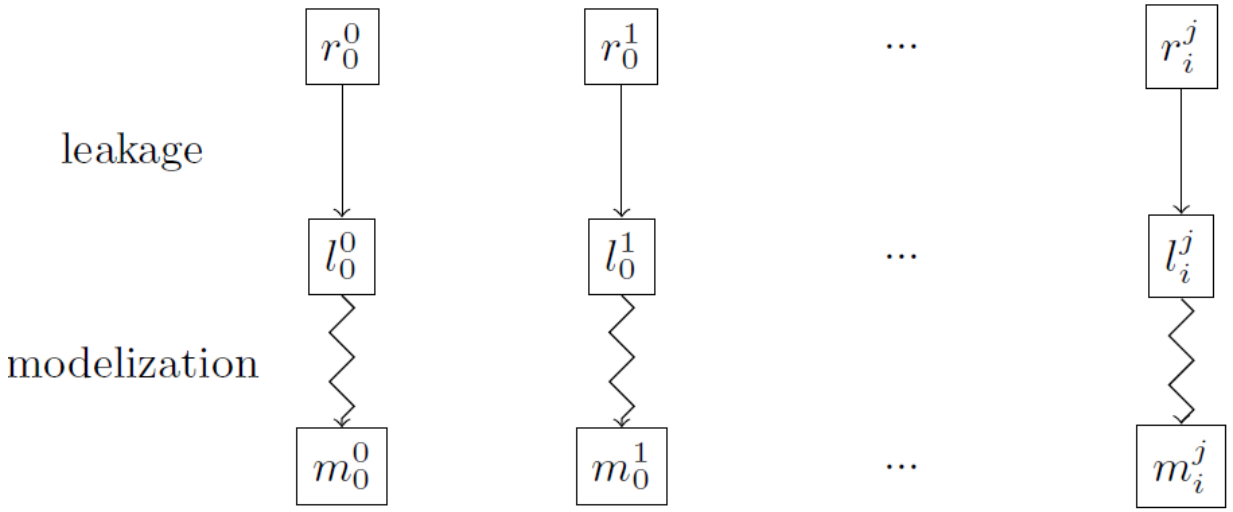
Algorithm 1 Montgomery ladder.

Input: $P, \mathbf{k} = (k_0, ..., k_{n-1})$
Output: $[k]P$
  $R_0 \leftarrow \mathcal{O}$
  $R_1 \leftarrow P$
  for $i = 0$ to $n - 1$ **do**
    $R_{1-k_i} \leftarrow R_{1-k_i} + R_{k_i}$
    $R_{k_i} \leftarrow [2]R_{k_i}$
  end for
  return R_0

HDPA attack on k_0:

1. Select several internal registers operations Rs that depends on P and k_0
2. Modelize the function L that characterizes how Rs leak: **information extraction**
3. Acquire 1 attack measurement
4. Prepare two sets S_0 (resp. S_1) that contain the guesses for the values Rs_0 (resp. Rs_1) in function of P and $k_0 = 0$ (resp. $k_0 = 1$)
5. Compare $L(Rs_i)$ with the actual SCA leakages using a distinguisher D: **information combination**
6. Select $k_0 = i$ such that $D(S_i, L(Rs_i))$ is maximised.

Extracting the information: linear regression



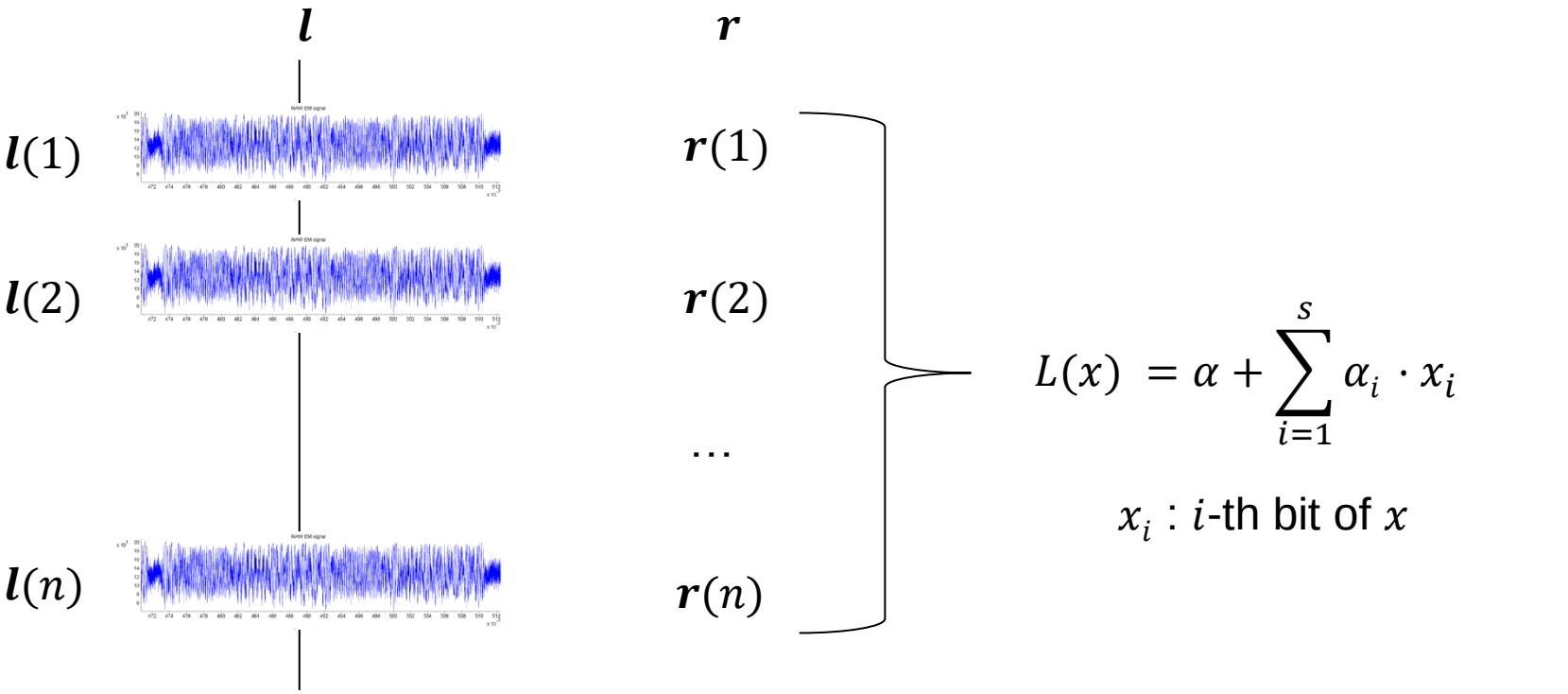
$$m_i^j(x) = L_i^j(x) + \mathcal{R}(0, \sigma^2)$$

Registers of size s bits: Classical templates: $O(2^s)$

Linear regression: $O(s)$ (or more: tradeoff)

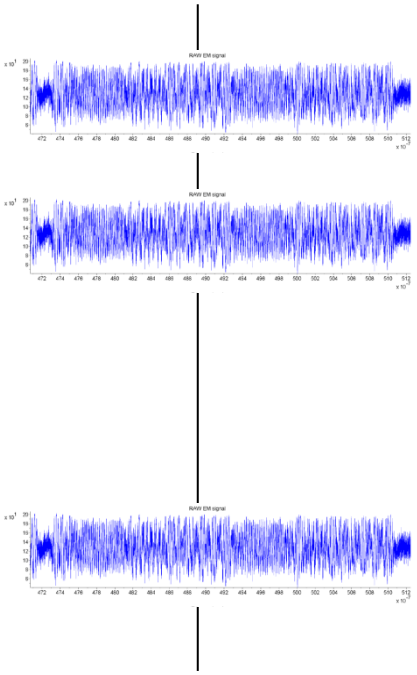
Linear regression: deterministic part

Acquire n traces with random known P and k.



Linear regression: noise

Acquire m traces with random known P and k



$\boldsymbol{l}$

$\boldsymbol{l}(1)$

$\boldsymbol{l}(2)$

$\boldsymbol{l}(m)$

Leakages

$\boldsymbol{r}$

$\boldsymbol{r}(1)$

$\boldsymbol{r}(2)$

…

$\boldsymbol{r}(m)$

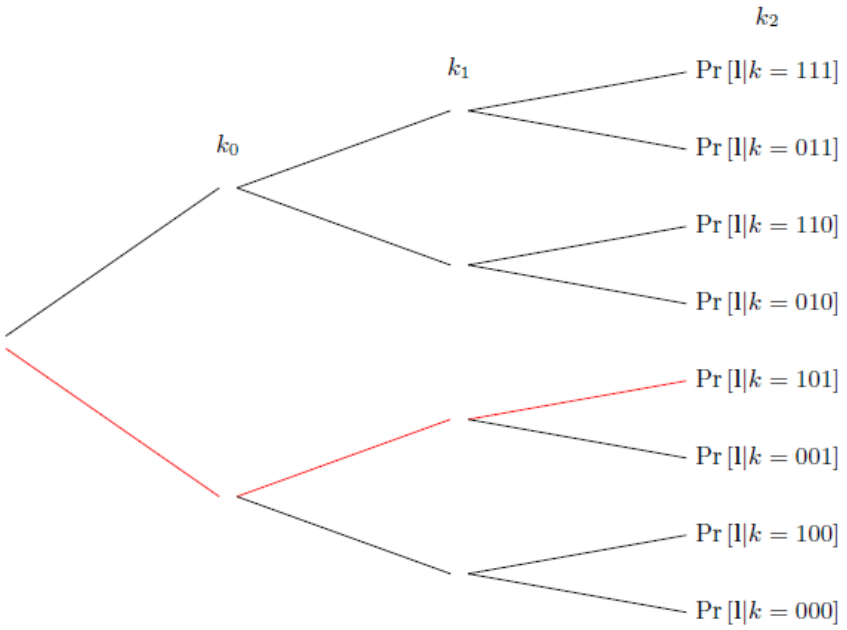
Processed value

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{l}(i) - L(\boldsymbol{r}(i))\right)^2$$

Noise approximation

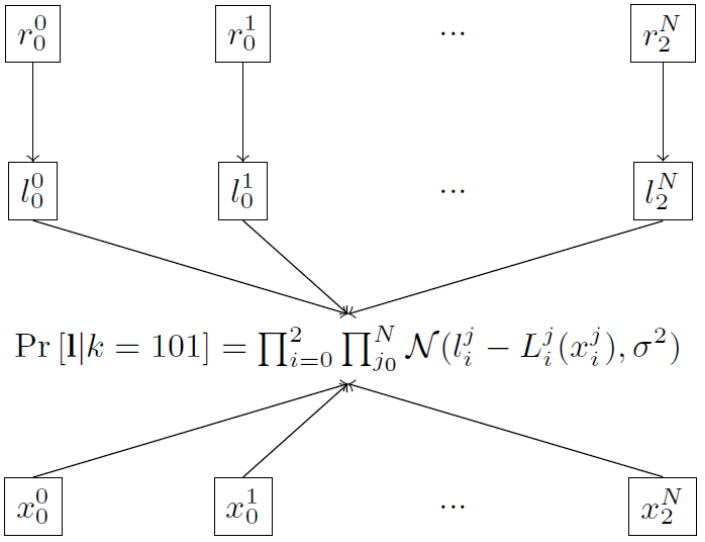
Combining the information (attack)

Parameter: d scalar bits attacked per iteration



Target

Simulator
$k = 101$



$d = 3$

Outline

- Context and motivations
- Horizontal differential power attack: systematic framework
- Practical experiments
  - Setup
  - Points of interest
  - Result on Cortex-M4
  - Result on Cortex-A8
- Conclusion and future works

Setup: target implementation/devices

Cortex-M4

- 100 MHz
- Constant time instructions (mostly)
- 32-bit registers

Cortex-A8

- 1 GHz
- Constant time instructions (mostly)
- 32-bit registers
- Ubuntu running in background

Custom constant time assembly implementation of NIST p256

256x256-bit multiplication achieved through 64 32x32-bit register multiplications (framework independent of the curve/implementation)

N=1600 target registers per scalar bit (only)

Setup: trace acquisition & scenario

Cortex-M4

- Power measurement
- Lecroy WaveRunner HRO 66
- 200 Ms/sec
- 123 scalar bits
- 40,000,000 samples per trace

Scenario: 1st order success rate on 123 bits

Cortex-A8

- EM measurement
- Lecroy WaveRunner 620Zi
- 10 GS/s
- 4 scalar bits
- 2,000,000 samples per trace
- Trace alignment

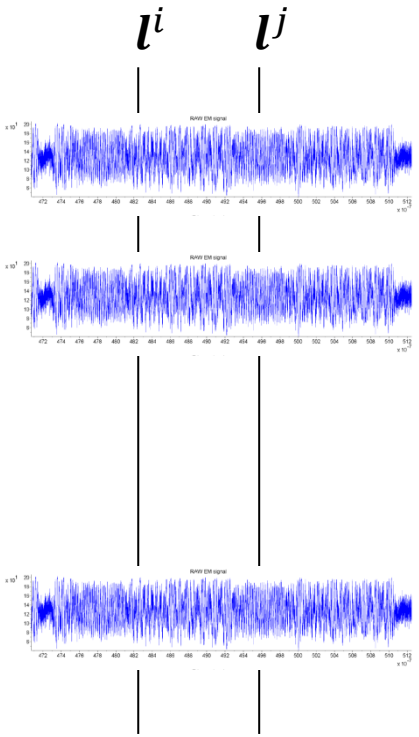
Scenario: Lattice attack (ECDSA) with several partial nonces

Outline

- Context and motivations
- Horizontal differential power attack: systematic framework
- Practical experiments
  - Setup
  - Points of interest
  - Result on Cortex-M4
  - Result on Cortex-A8
- Conclusion and future works

Points of interest: CPA and partial SNR

Acquire n traces with random known P and k



Leakages

$\boldsymbol{r}$

$\boldsymbol{r}(1)$

$\boldsymbol{r}(2)$

…

$\boldsymbol{r}(n)$

Processed value

$$t = argmax_i(\rho(HW(\boldsymbol{r}), \boldsymbol{l}^i))$$

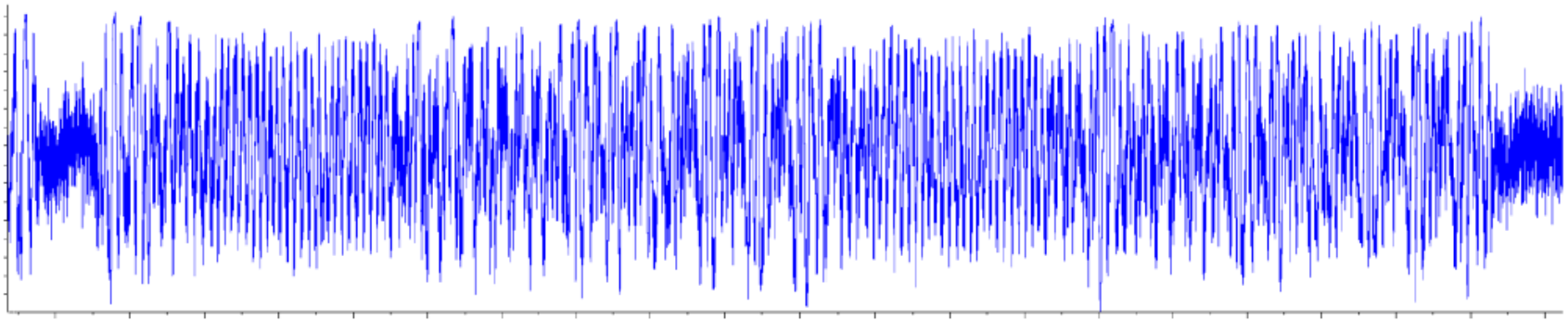
$$t = argmax_i(SNR\,(trunc_b(\boldsymbol{r}), \boldsymbol{l}^i))$$

Time sample

Points of interest: windowed mode

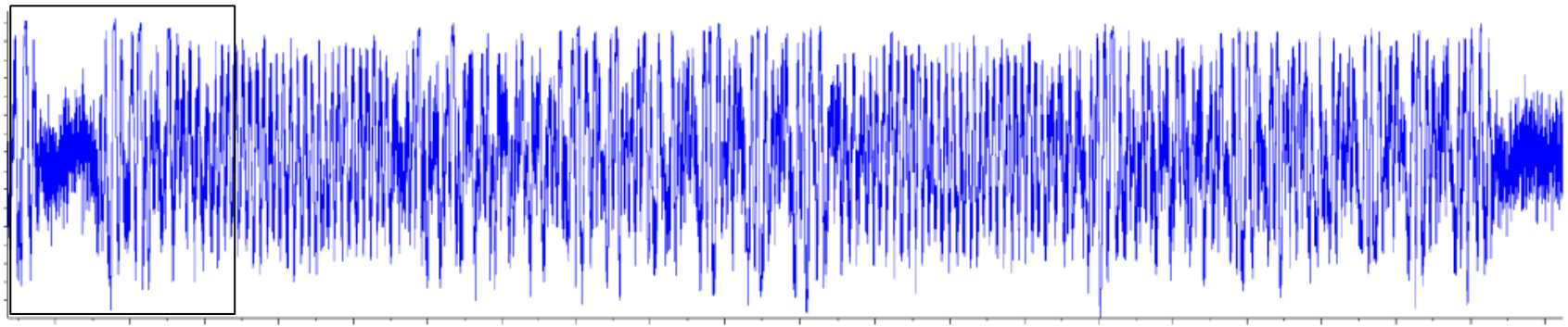
Cortex-M4: $1600 \cdot 123$ POIs ; 40,000,000 samples

r_0^0 r_0^1 $\cdots$ r_i^j



Points of interest: windowed mode

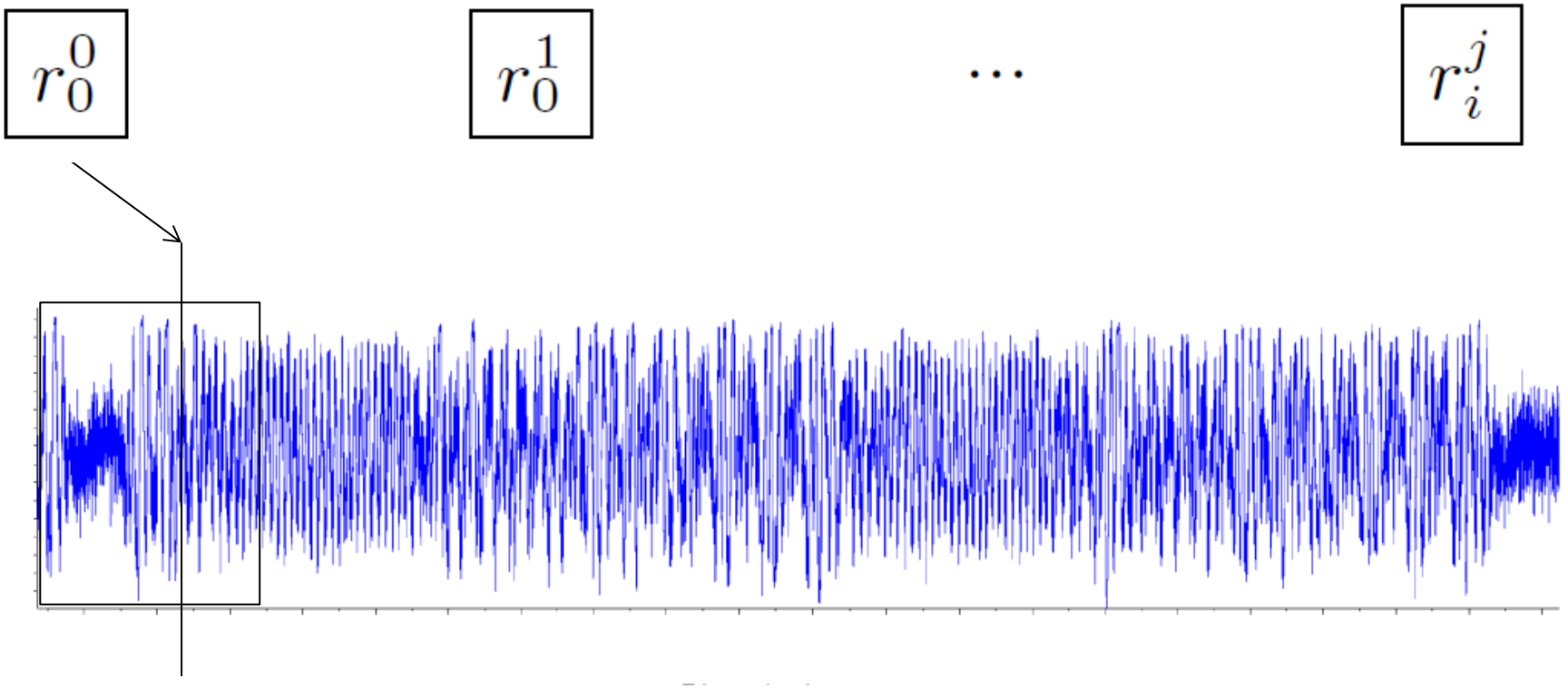
r_0^0 r_0^1 $\cdots$ r_i^j



Points of interest: windowed mode

CPA: p-value

partial SNR: heuristic threshold

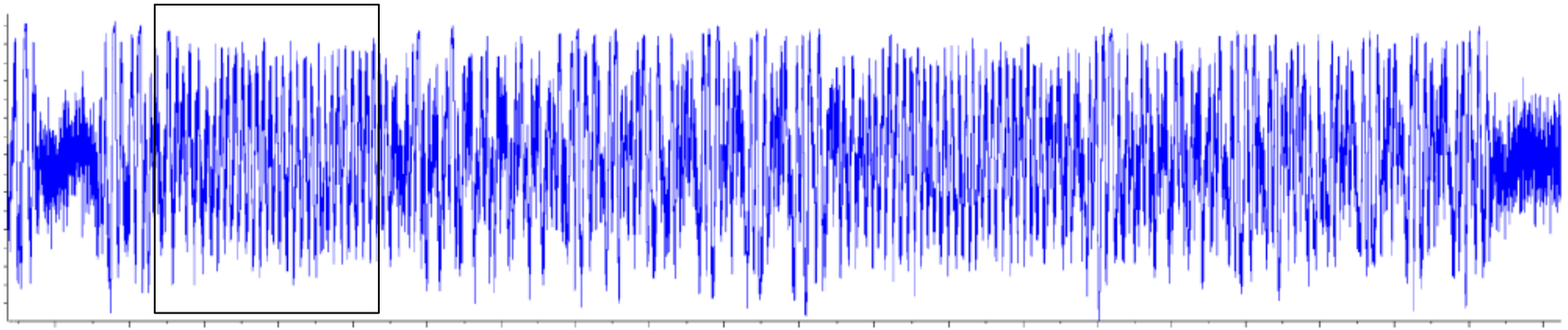


Points of interest: windowed mode

CPA: p-value

partial SNR: heuristic threshold

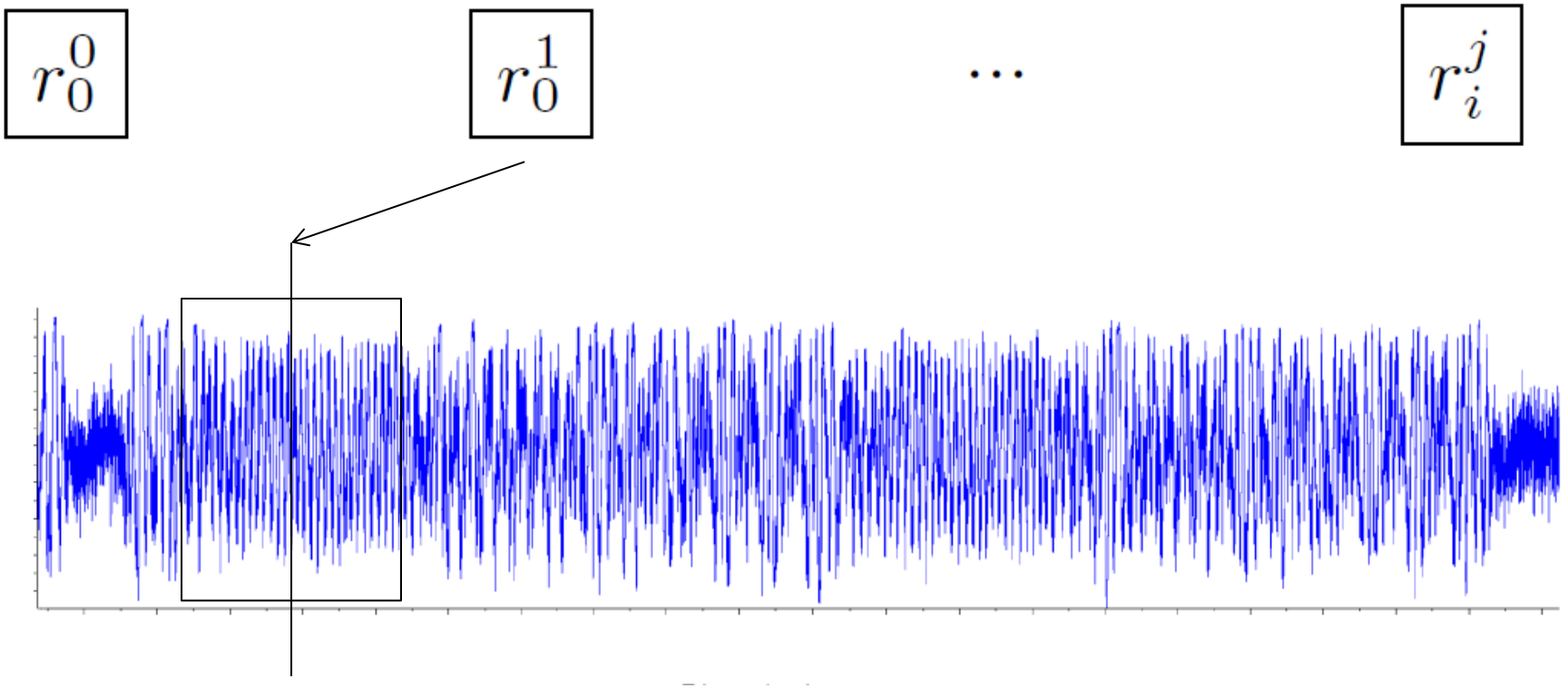
r_0^0 r_0^1 $\cdots$ r_i^j



Points of interest: windowed mode

CPA: p-value

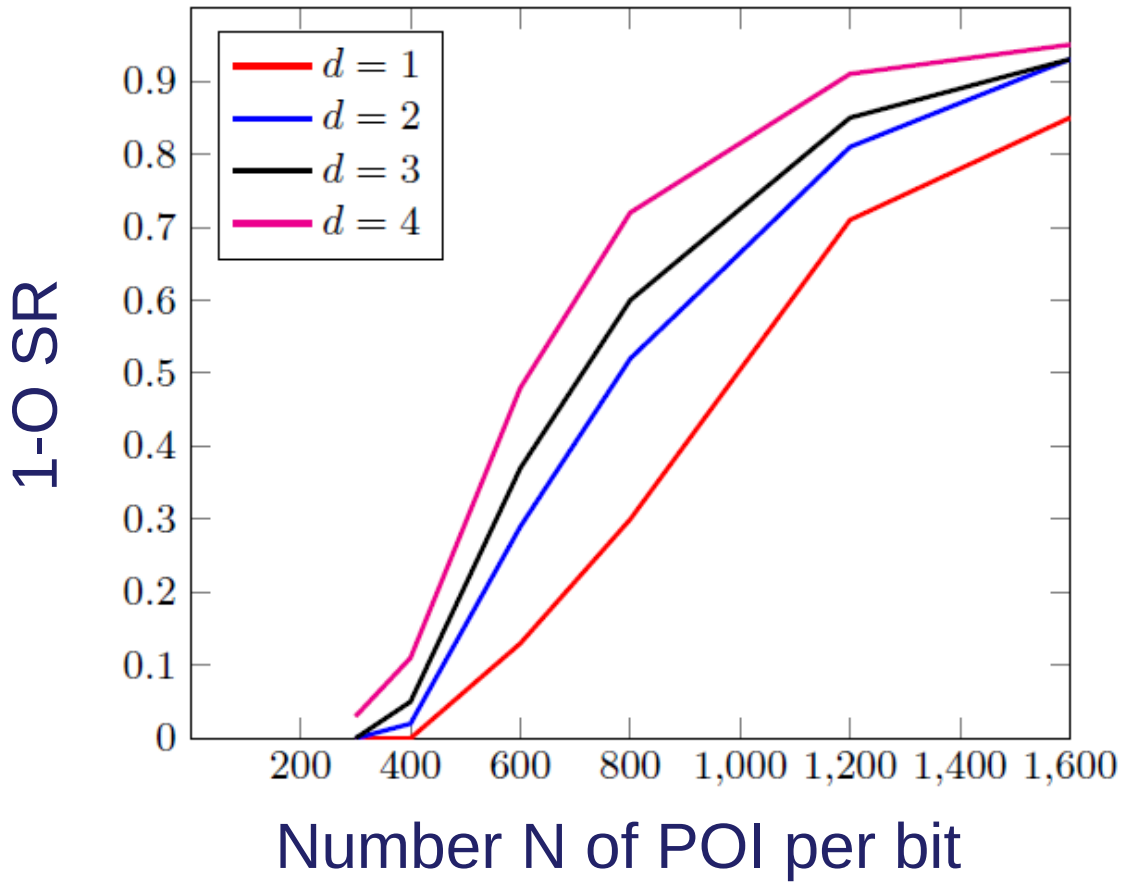
partial SNR: heuristic threshold



Outline

- Context and motivations
- Horizontal differential power attack: systematic framework
- Practical experiments
  - Setup
  - Points of interest
  - Result on Cortex-M4: first order success rate on 123 scalar bits
  - Result on Cortex-A8
- Conclusion and future works

Cortex-M4 results: 1-O SR on 123 bits



Reminder on the parameters:

- d: number of scalar bit targeted at the same time
- N: number of target register per scalar bit

Outline

- Context and motivations
- Horizontal differential power attack: systematic framework
- Practical experiments
  - Setup
  - Points of interest
  - Result on Cortex-M4
  - Result on Cortex-A8: lattice attack on 4-bit partial nonces (ECDSA)
- Conclusion and future works

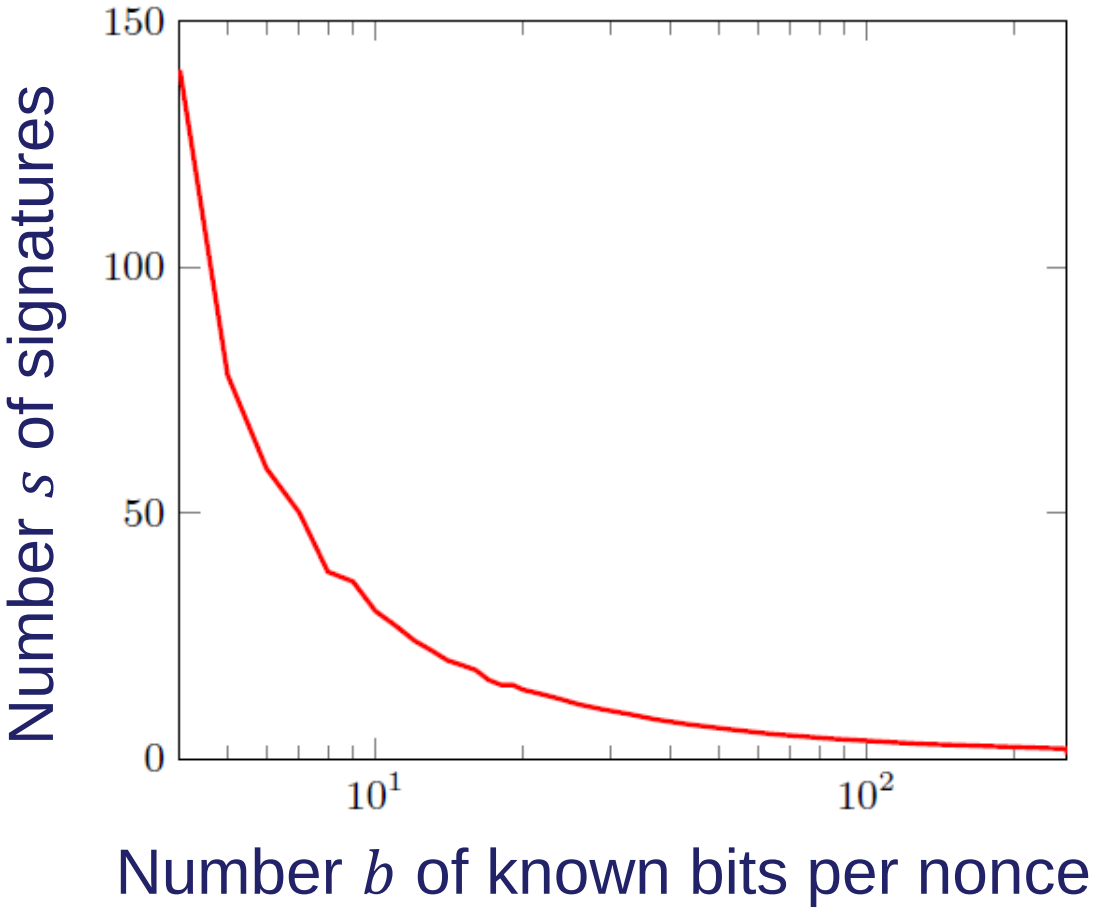
Lattice attack on ECDSA: tradeoff

Run s signatures:

- s nonces n_i

- Recover b bits of each n_i on the ECSM $[n]P$

- Recover k from the s partial nonces $trunc_b(n_i)$



Lattice attack on ECDSA: our tradeoff

Run 2200 signatures:

- 2200 nonces n_i

- Recover 4 bits of each n_i on the ECSM $[n]P$

- Recover k from the s partial nonces $trunc_4(n_i)$

- 140 correct partial nonces required

Problem: incorrect partial nonces

Answer: - Bayes for real probas
- probability threshold t

$$\Pr\left[\mathsf{trunc}_4(n) \,\middle|\, \mathbf{l}\right] = \frac{\Pr\left[\mathbf{l} \,\middle|\, \mathsf{trunc}_4(n)\right]}{\sum_{n'=0}^{15} \Pr\left[\mathbf{l} \,\middle|\, \mathsf{trunc}_4(n')\right]}$$

$$\Pr\left[\mathsf{trunc}_4(n) \,\middle|\, \mathbf{l}\right] > t$$

Cortex-A8 results: threshold for lattice attack on ECDSA

Probability threshold t	Nonce 1-O SR $= x$	Key 1-O SR $= x^{140}$	# discarded traces	#remaining traces
None	0.815	$3{,}9 \cdot 10^{-13}$	0	2200
0.9	0.868	$2{,}5 \cdot 10^{-9}$	306	1894
0.9999999	0.992	0.312	1597	613
0.99999999999	1	1	1958	$242 > 140$

Conclusion and future work

Conclusion:

1. Detailed systematic framework to apply HDPA in a close-to optimal way:
   a) Information identification
   b) Information extraction
   c) Information combination

1. Application in practice:
   a) Cortex-M4 & Cortex-A8
   b) A-Z application of the framework

A lot of noise is required to resist HDPA

Next step: breaking point randomization !!

1. Replace the information combination by a **belief probagation algorithm** (soft analytical side-channel attack)

2. Recover the random point

3. Apply the same framework as in this study to recover the key

- Cannot be done with TA or OTA
- Cannot be used with correlation H-DPA

⇨ SASCA on asym: See next talk !!!