Getting the Most Out of Leakage Detection

Statistical tools and measurement setups hand in hand

COSADE 2017 – APRIL 14, 2017
SESSION 7 : SIDE-CHANNEL TOOLS

SANTOS MERINO DEL POZO
FRANÇOIS-XAVIER STANDAERT
UCL CRYPTO GROUP

Motivation

- SCA evaluations are complex and expensive
- So, how can we reduce evaluation time?
  - Optimizing the distinguishers
    - Perhaps the hottest topic in the SCA community
  - Detecting rather than exploiting leakages
    - Cool guys are using *t*-test based tools

Motivation

- SCA evaluations are complex and expensive
- So, how can we reduce evaluation time?
  - Optimizing the distinguishers
    - Perhaps the hottest topic in the SCA community
  - Detecting rather than exploiting leakages
    - Cool guys are using *t*-test based tools
  - Obtaining good measurements
    - Mostly disregarded by academia
    - But, shall we not care about it?

Our goals

- Highlighting the impact of measurement setups to ease the detection of leakages
  - with tools available in almost any electronics retailer
  - sophisticated bespoke tools are out of scope
- Discussing about the effectiveness of state-of-the-art *t*-test based leakage detection tools
  - fair comparison using the same measurement setup
- Ultimate goal: combine the best of two worlds
  - illustrated in a highly noisy case study

t-test based leakage detection

- Determine a distinguisher
  - typically, fixed vs. random (non-specific)
  - more recently, fixed vs. fixed (improved signal)

t-test based leakage detection

- Determine a distinguisher
  - typically, fixed vs. random (non-specific)
  - more recently, fixed vs. fixed (improved signal)
- Record side-channel traces (e.g., power)

t-test based leakage detection

- Determine a distinguisher
  - typically, fixed vs. random (non-specific)
  - more recently, fixed vs. fixed (improved signal)
- Record side-channel traces (e.g., power)
- Group traces based on the distinguisher
  - resulting in two sets $\mathcal{T}_0$ and $\mathcal{T}_1$ of traces

t-test based leakage detection

- Determine a distinguisher
  - typically, fixed vs. random (non-specific)
  - more recently, fixed vs. fixed (improved signal)
- Record side-channel traces (e.g., power)
- Group traces based on the distinguisher
  - resulting in two sets $\mathcal{T}_0$ and $\mathcal{T}_1$ of traces
- Estimate sample mean and variance in a univariate fashion
  - traces must be accordingly preprocessed to detect higher-order leakages
- Compute *t*-test statistic for each time sample

$$\mathrm{t} = \frac{\mu(\mathcal{T}_0) - \mu(\mathcal{T}_1)}{\sqrt{\frac{\sigma^2(\mathcal{T}_0)}{|\mathcal{T}_0|} + \frac{\sigma^2(\mathcal{T}_1)}{|\mathcal{T}_1|}}},$$

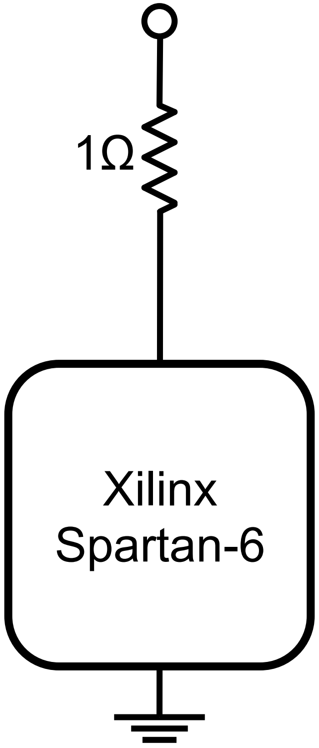
t-test based leakage detection

- Determine a distinguisher
  - typically, fixed vs. random (non-specific)
  - more recently, fixed vs. fixed (improved signal)
- Record side-channel traces (e.g., power)
- Group traces based on the distinguisher
  - resulting in two sets $\mathcal{T}_0$ and $\mathcal{T}_1$ of traces
- Estimate sample mean and variance in a univariate fashion
  - traces must be accordingly preprocessed to detect higher-order leakages
- Compute *t*-test statistic for each time sample

$$\mathrm{t} = \frac{\mu(\mathcal{T}_0) - \mu(\mathcal{T}_1)}{\sqrt{\frac{\sigma^2(\mathcal{T}_0)}{|\mathcal{T}_0|} + \frac{\sigma^2(\mathcal{T}_1)}{|\mathcal{T}_1|}}},$$

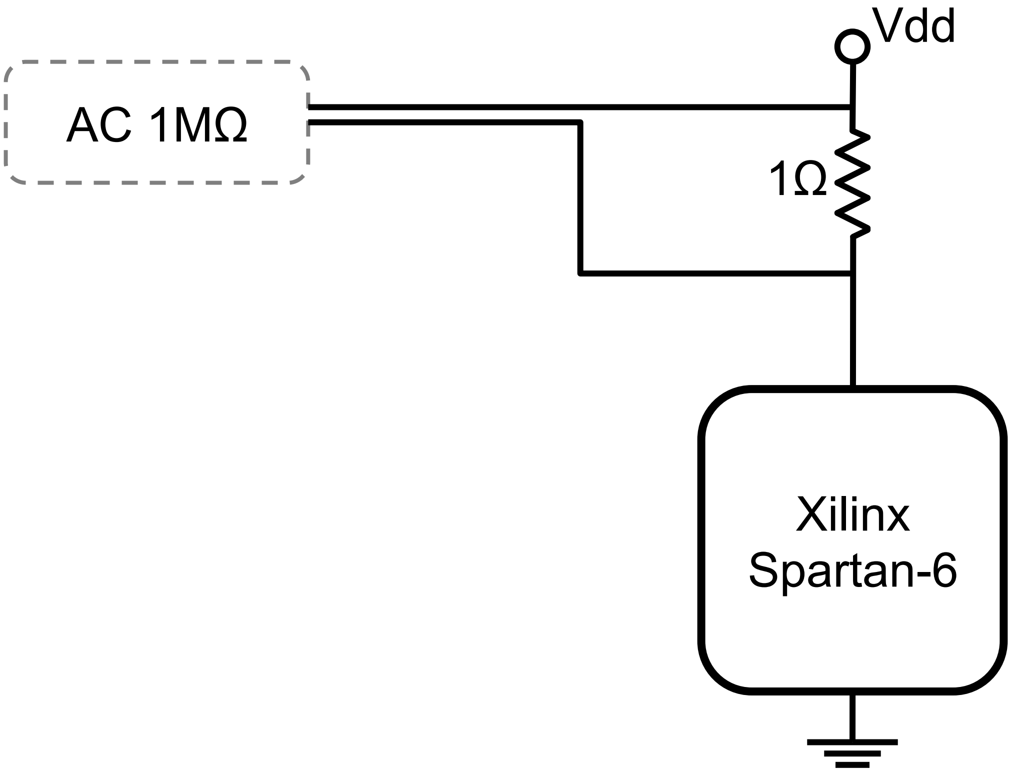
- Test fails if at any time point |t| ≥ 4.5 ➔ i.e., leakage detected

Setups

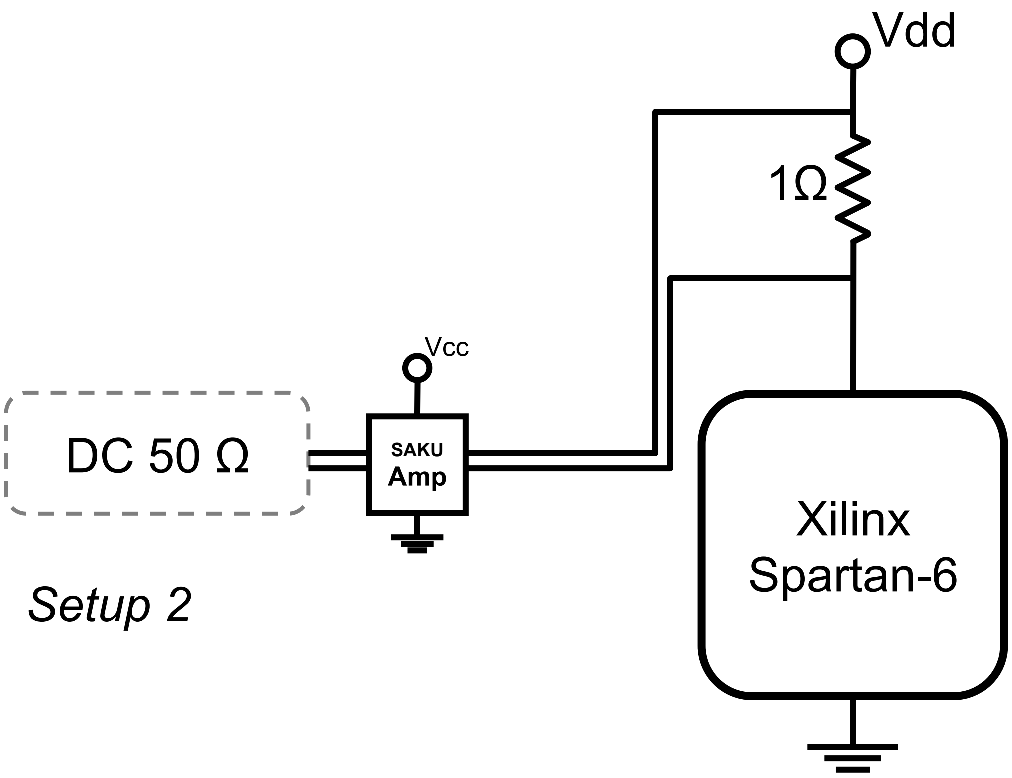


Setups

Setup 1

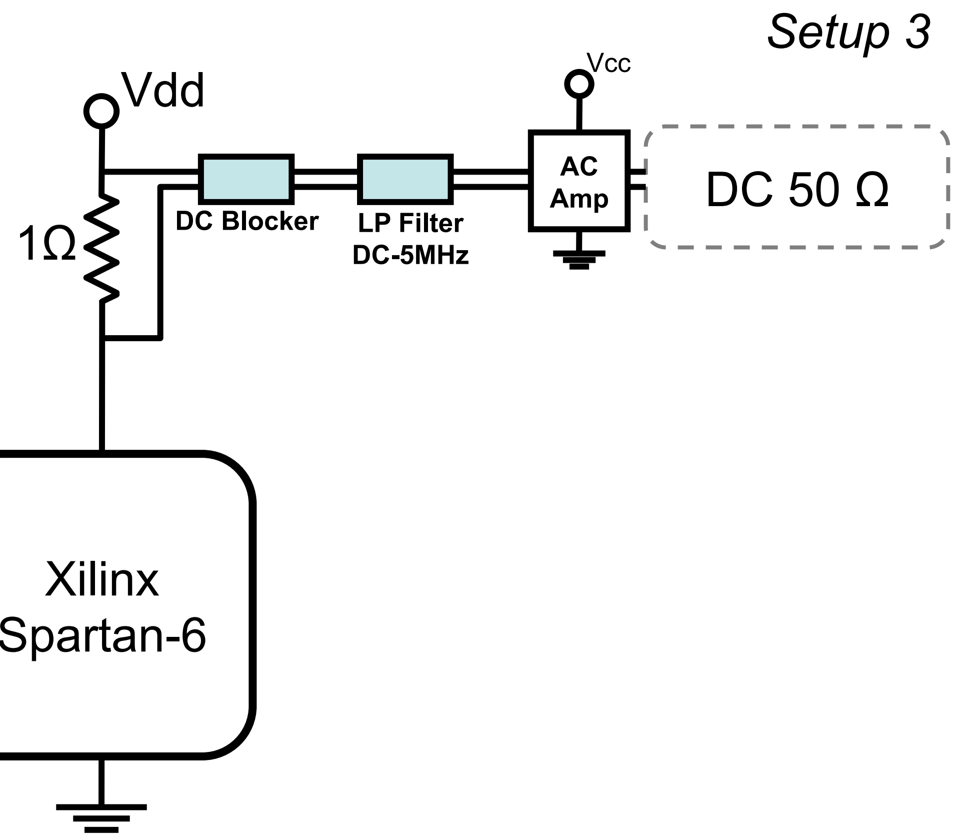


Setups



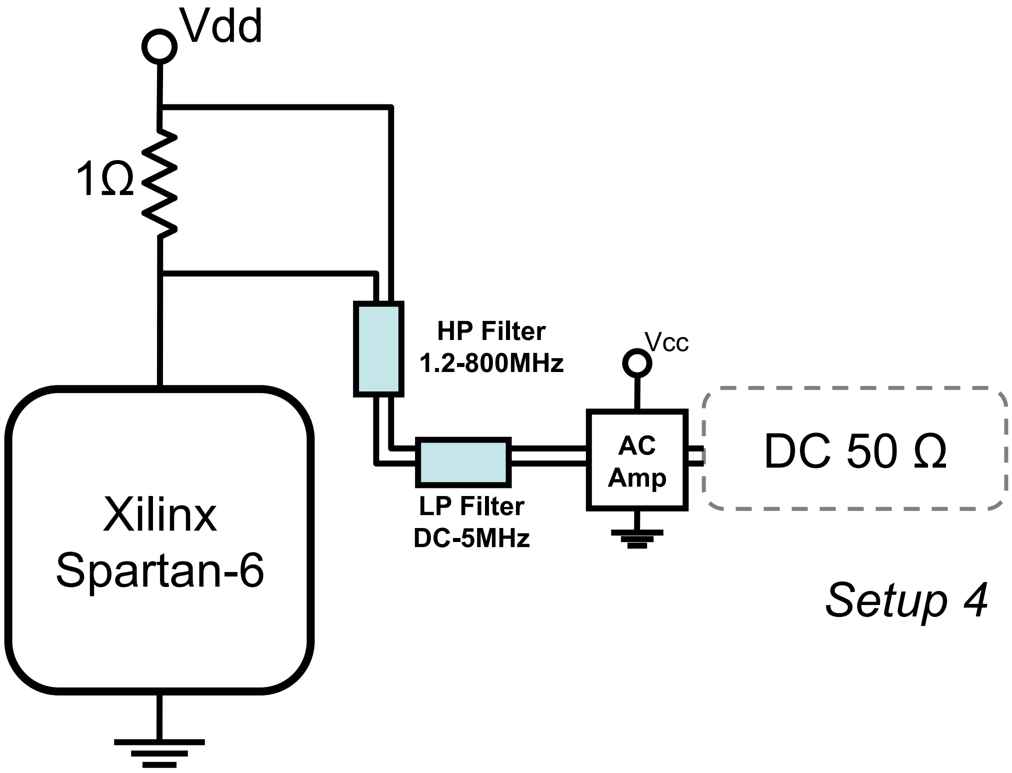
Setup 2

Setups



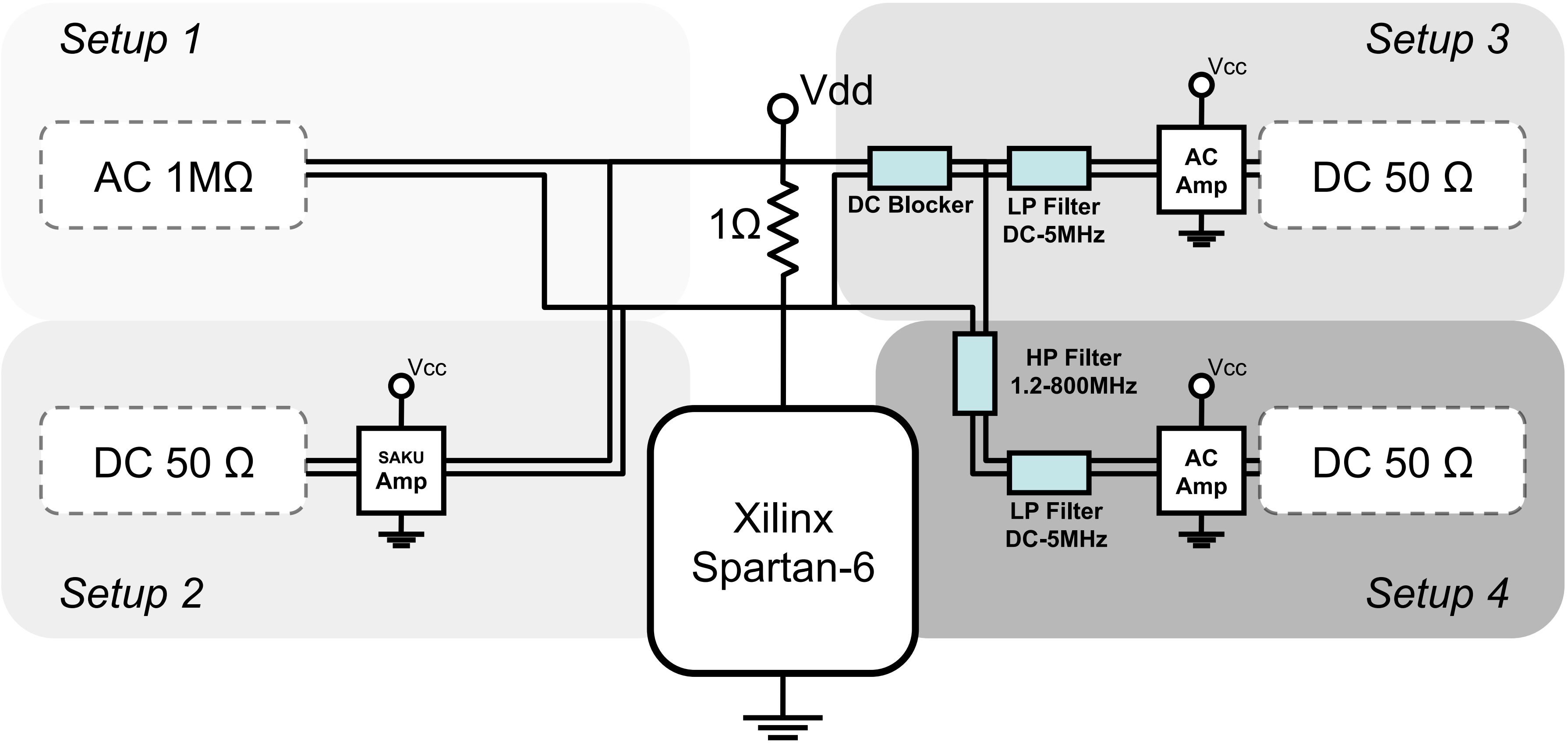
Setup 3
Vdd
Vcc
1Ω
DC Blocker
LP Filter
DC-5MHz
AC
Amp
DC 50 Ω
Xilinx
Spartan-6

Setups



Setup 4

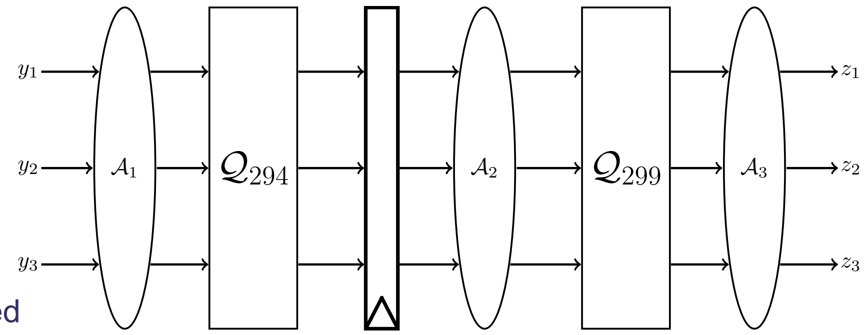
Setups



Setup 1
Setup 2
Setup 3
Setup 4
AC 1MΩ
Vdd
1Ω
DC Blocker
LP Filter
DC-5MHz
Vcc
AC
Amp
DC 50 Ω
HP Filter
1.2-800MHz
SAKU
Amp
Xilinx
Spartan-6

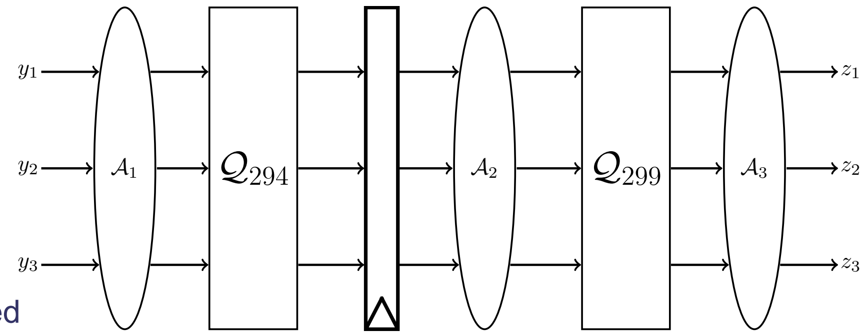
Case studies

- First-order threshold implementation of PRESENT
  - 3-share Boolean masking
    - secure against first-order SCAs
    - even in the presence of glitches
  - Negligible algorithmic noise
    - fully serialized architecture
    - small combinatorial circuits
    - random masks are externally provided
  - Design clocked @3MHz
    - so, no windowing effect

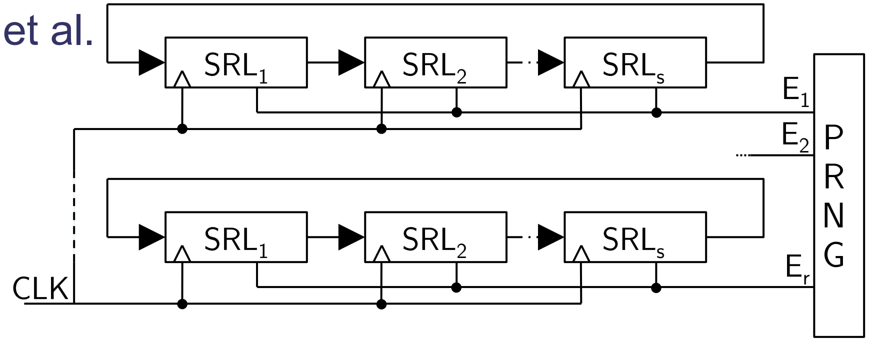


Case studies

- First-order threshold implementation of PRESENT
  - 3-share Boolean masking
    - secure against first-order SCAs
    - even in the presence of glitches
  - Negligible algorithmic noise
    - fully serialized architecture
    - small combinatorial circuits
    - random masks are externally provided
  - Design clocked @3MHz
    - so, no windowing effect



- Gaussian noise engine
  - FPGA-dedicated design by Güneysu et al.
  - Configuring unused LUTs
    - *r* cycling rings: noise variance
    - *s* LUTs per ring: noise amplitude
    - our design: *r*=16, *s*=100
  - PRNG implemented as a LFSR

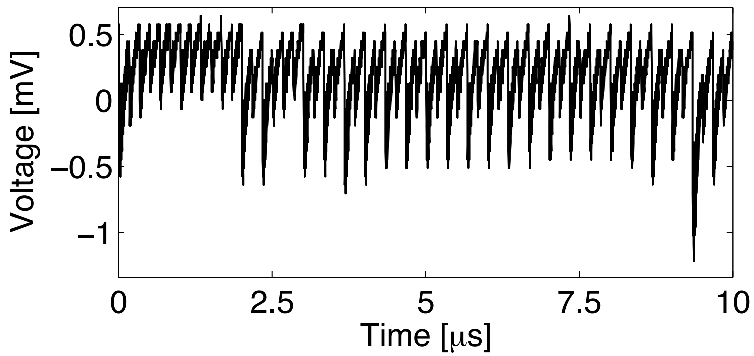


Comparing setups

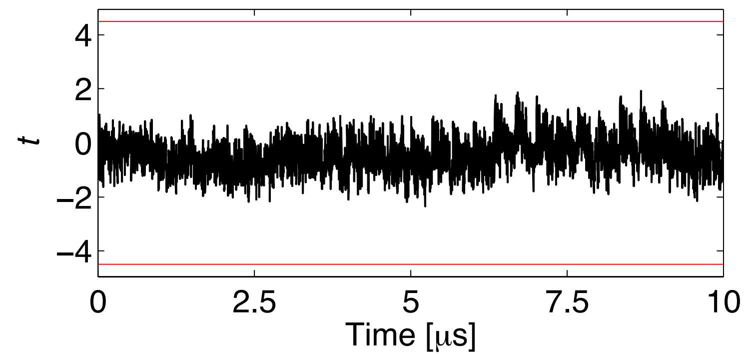
- Based on the fixed vs. random *t*-test
- Analysis up to third-orders
- Full control over the implementation
  – input data can be reproduced for each setup
- 1M traces recorded in a low noise regime
  – negligible algorithmic noise of target design
    - noise engine is not implemented yet

  – ultra-low-noise design of SAKURA-G board

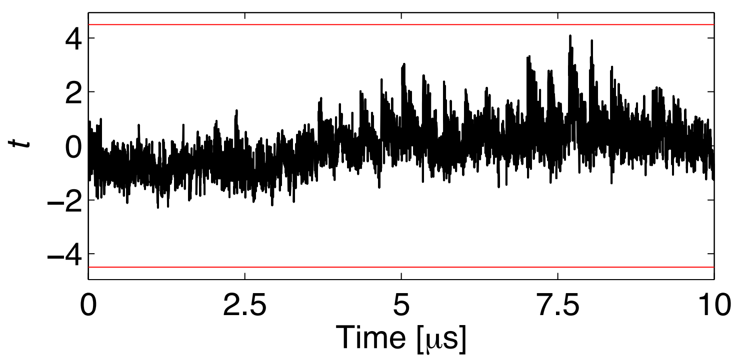
Comparing setups – *Setup 1*



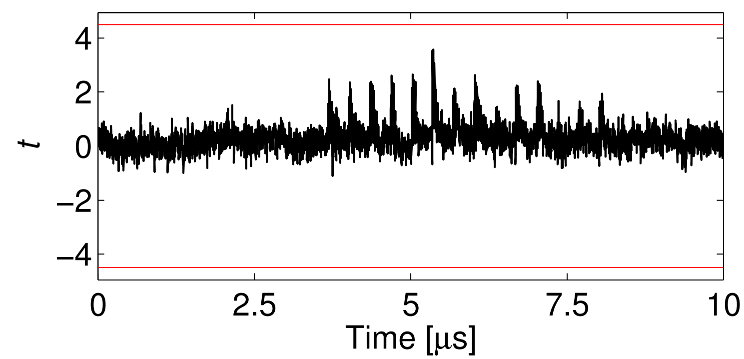
(a) Sample trace



(b) first-order

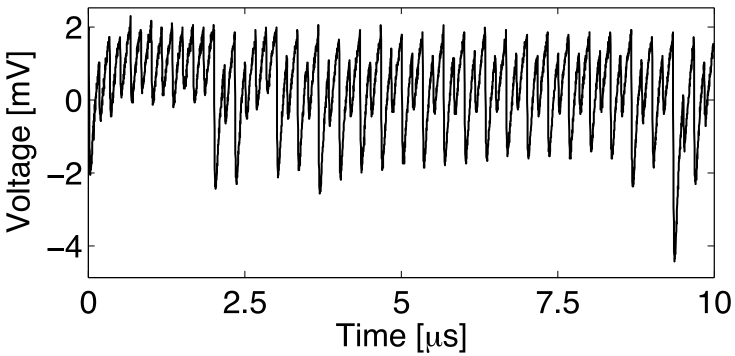


(c) second-order

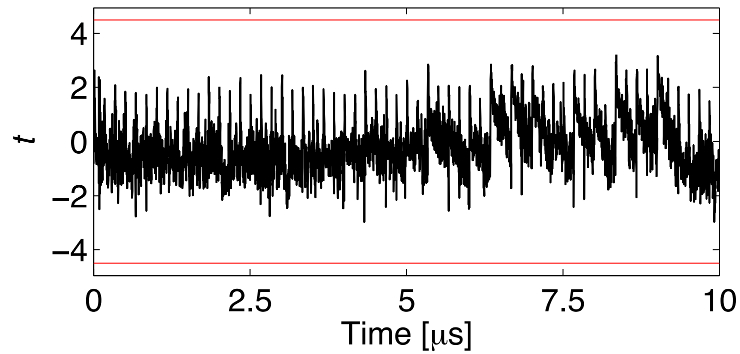


(d) third-order

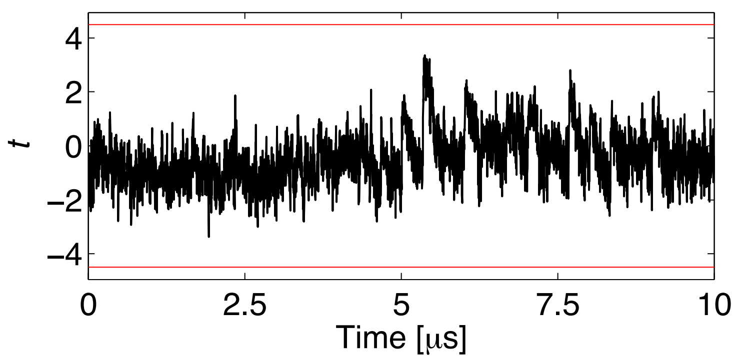
Comparing setups – *Setup 2*



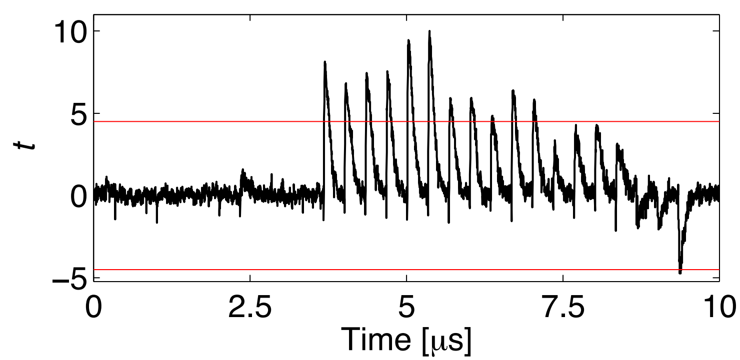
(a) Sample trace



(b) first-order

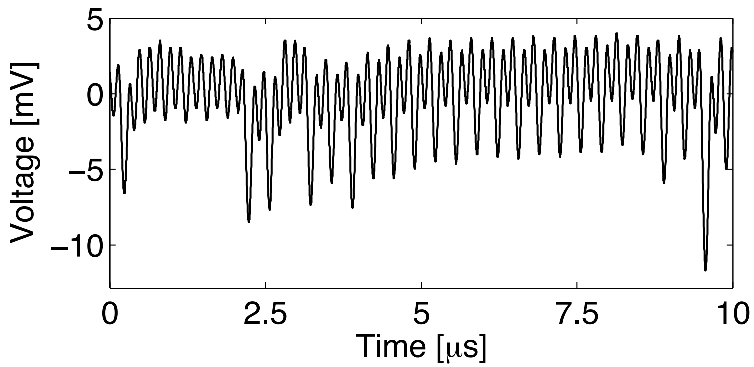


(c) second-order

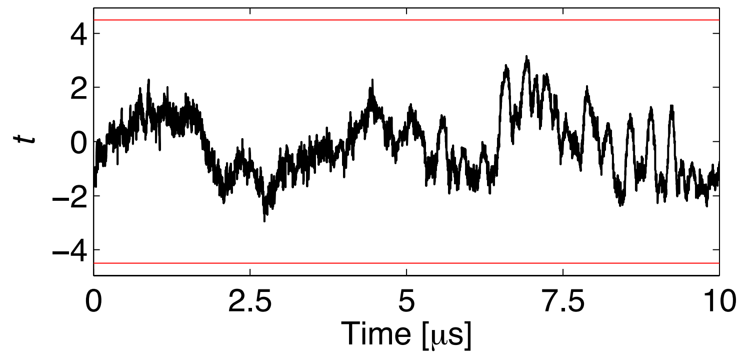


(d) third-order

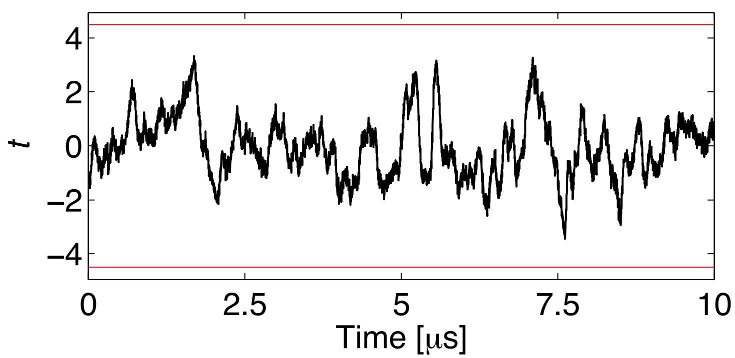
Comparing setups – *Setup 3*



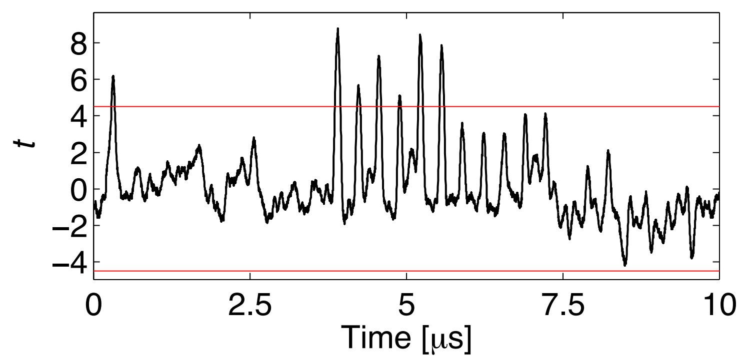
(a) Sample trace



(b) first-order

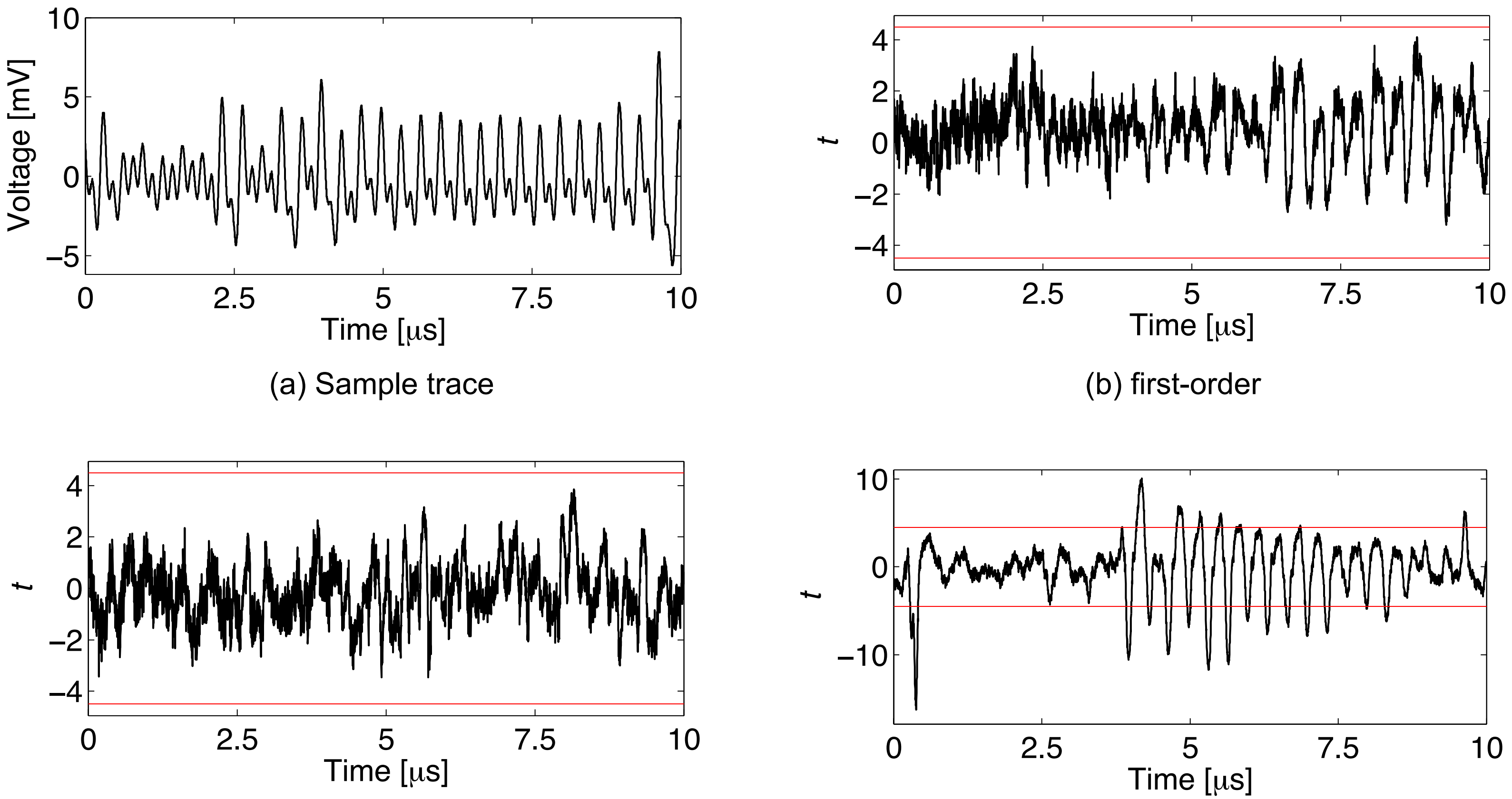


(c) second-order



(d) third-order

Comparing setups – *Setup 4*



(a) Sample trace

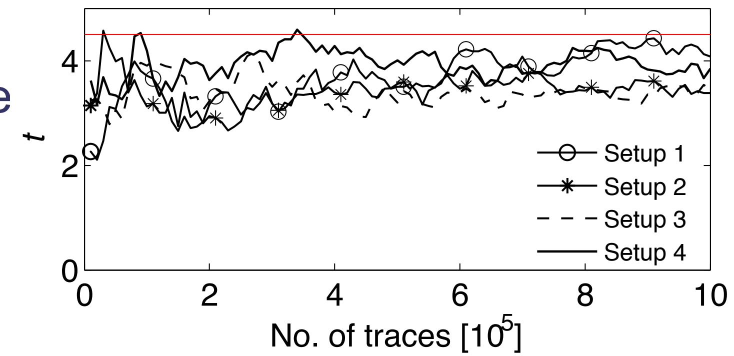
(b) first-order

(c) second-order

(d) third-order

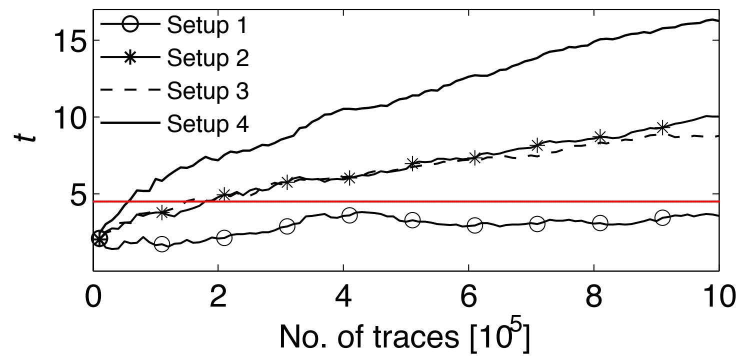
Comparing setups – *Wrapping up*

- Second-order leakages: not detected
  - due to the register-oriented architecture
  - *Setup 1* is the closest to detection
    - noise introduced by the additional hardware?



(a) second-order

- Third-order leakages: not detected by *Setup 1*
  - due to the low peak-to-peak amplitude
  - yet, is $|t| \geq 4.5$ a good criteria?
    - clear pattern in the plots *of Setup 1*



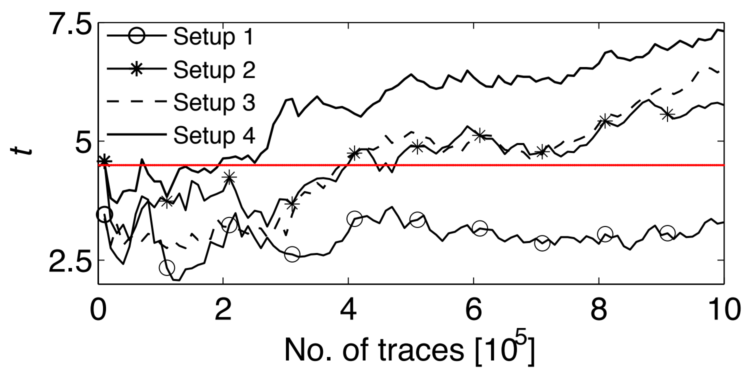
(b) third-order

Comparing distinguishers

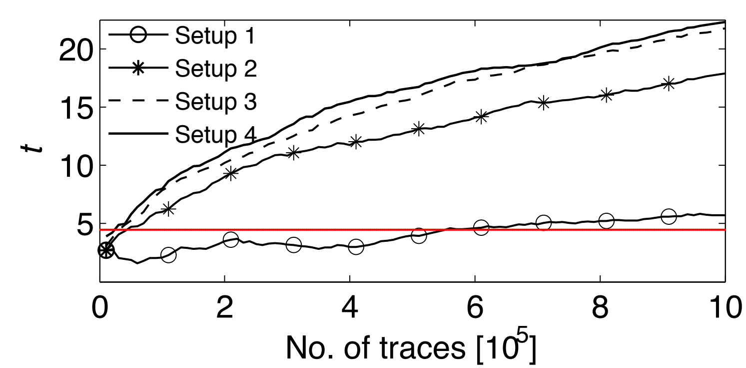
- Same methodology
- Same target
- Just, different distinguisher: fixed vs. fixed
  - goal: evaluate the improvement in convergence speed
- Assuming a powerful adversary
  - full knowledge of the target design and its implementation
  - so, inputs are carefully selected
    - e.g., to maximize HD differences
  - yet, this is not a major requirement for fixed vs. fixed to work

Comparing distinguishers – *Results*

- Second-order leakages: detected
  - *Setup 1*: still unsuccessful
  - significant improvements for the other setups
- Third-order leakages: pinpointed with higher confidence
  - *Setup 1*: becomes successful
  - *Setups 2 and 3*: reduction by a factor ≈ 4 on the number of traces



(a) second-order

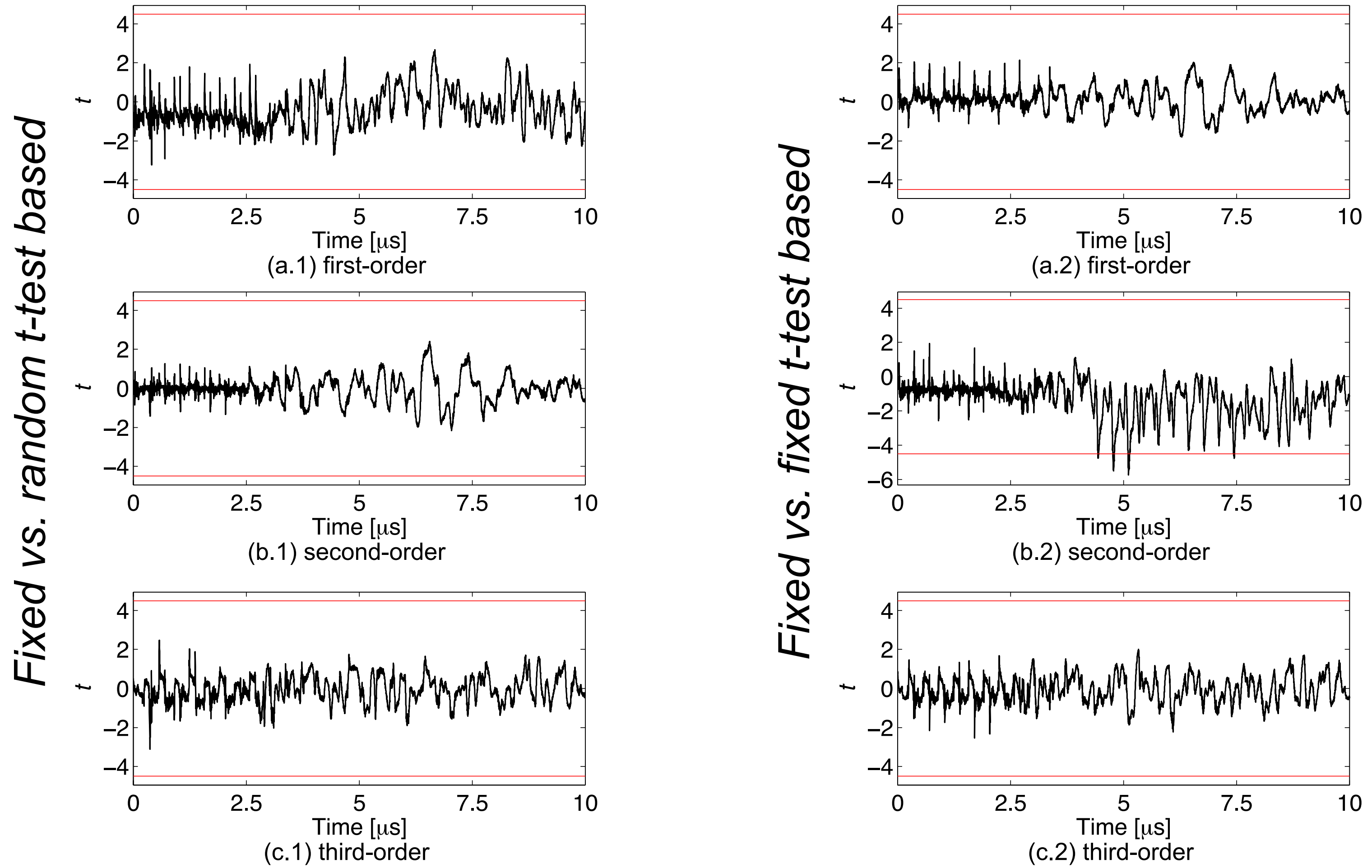


(b) third-order

Consolidating results

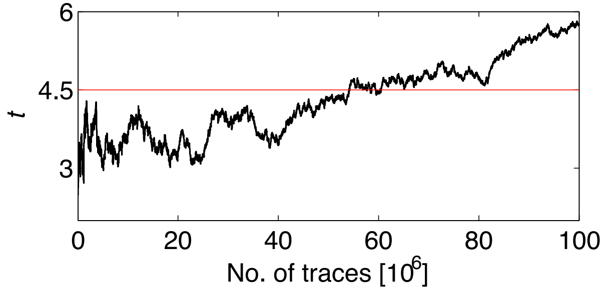
- Fact: fixed vs. fixed enables leakage detection with reduced data complexity
- Question: is it because of a greater signal or just a reduction in noise?
- Solution: scenario with hard-to-filter noise
  - if so, all gains will be due to an improved signal
  - remember the Gaussian noise engine?
    - noise synchronized with the crypto core
- 100M measurements recorded with *Setup 4*

Consolidating results – *Comparison*



Consolidating results - *Wrapping up*

- Fixed vs. random: no leakage detected
  - Different (fixed) plaintexts tested: yet, same results
- Fixed vs. fixed: second-order leakages detected
  - results are inline with theory
    - low-order moments are more informative in high noise settings
  - Indeed, 60M measurements are enough



- We answered the question:
  - increasing the signal (significantly) reduces the data complexity to detect higher-order leakages

Final message

- Take care of your measurement setup
  - small tweaks can make a huge difference
- Use the best distinguisher you can
  - for reduced acquisition time and storage requirements
    - critical factor when multi-million traces need to be recorded and then analyzed
  - a plus when the measurement hardware cannot help you
    - e.g., by increasing the signal in the presence of hard-to-filter noise

Final message

- Take care of your measurement setup
  - small tweaks can make a huge difference
- Use the best distinguisher you can
  - for reduced acquisition time and storage requirements
    - critical factor when multi-million traces need to be recorded and then analyzed
  - a plus when the measurement hardware cannot help you
    - e.g., by increasing the signal in the presence of hard-to-filter noise

THANK YOU!

DO YOU HAVE QUESTIONS?