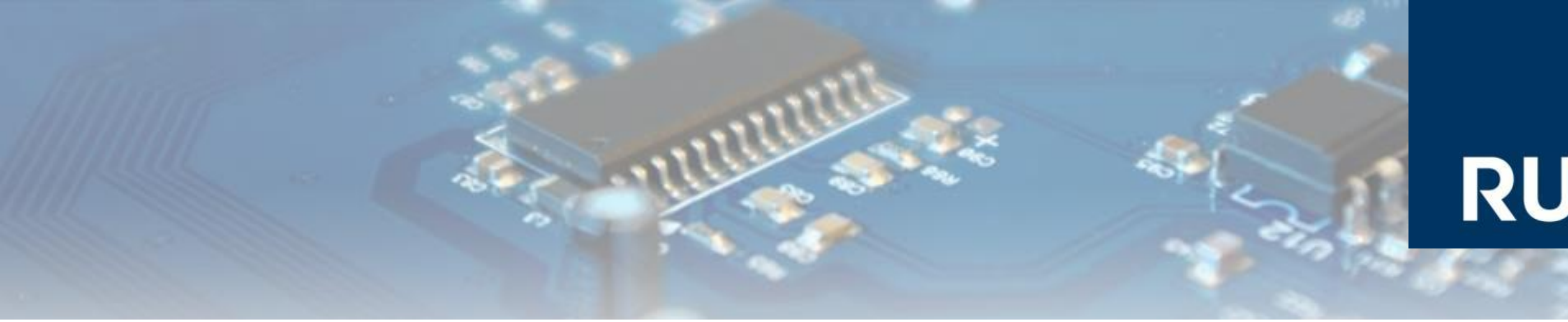


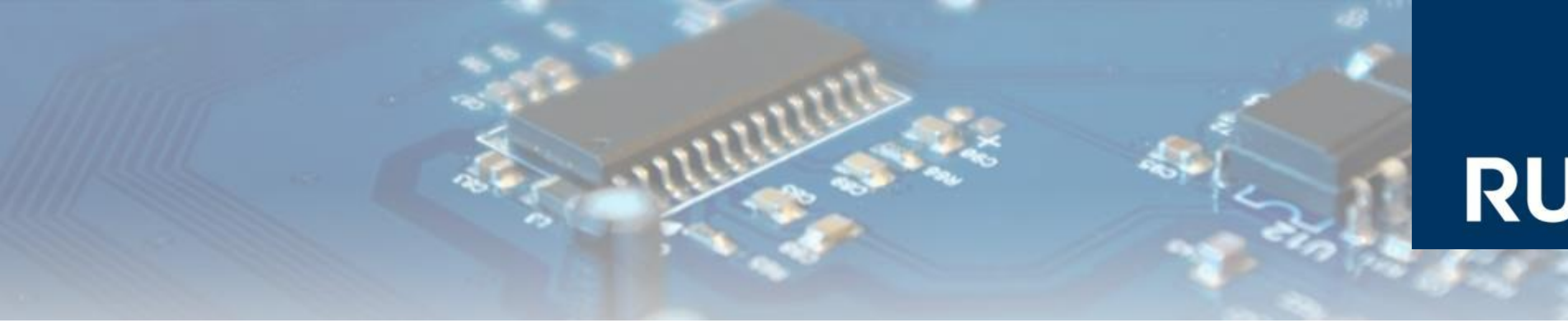
Improving Side-Channel Analysis with Optimal Pre-Processing

CARDIS 2012

David Oswald, Christof Paar

Horst Görtz Institute for IT Security

Ruhr-Universität Bochum



Improving Side-Channel Analysis with “Optimal” Pre-Processing

CARDIS 2012

David Oswald, Christof Paar

Horst Görtz Institute for IT Security

Ruhr-Universität Bochum

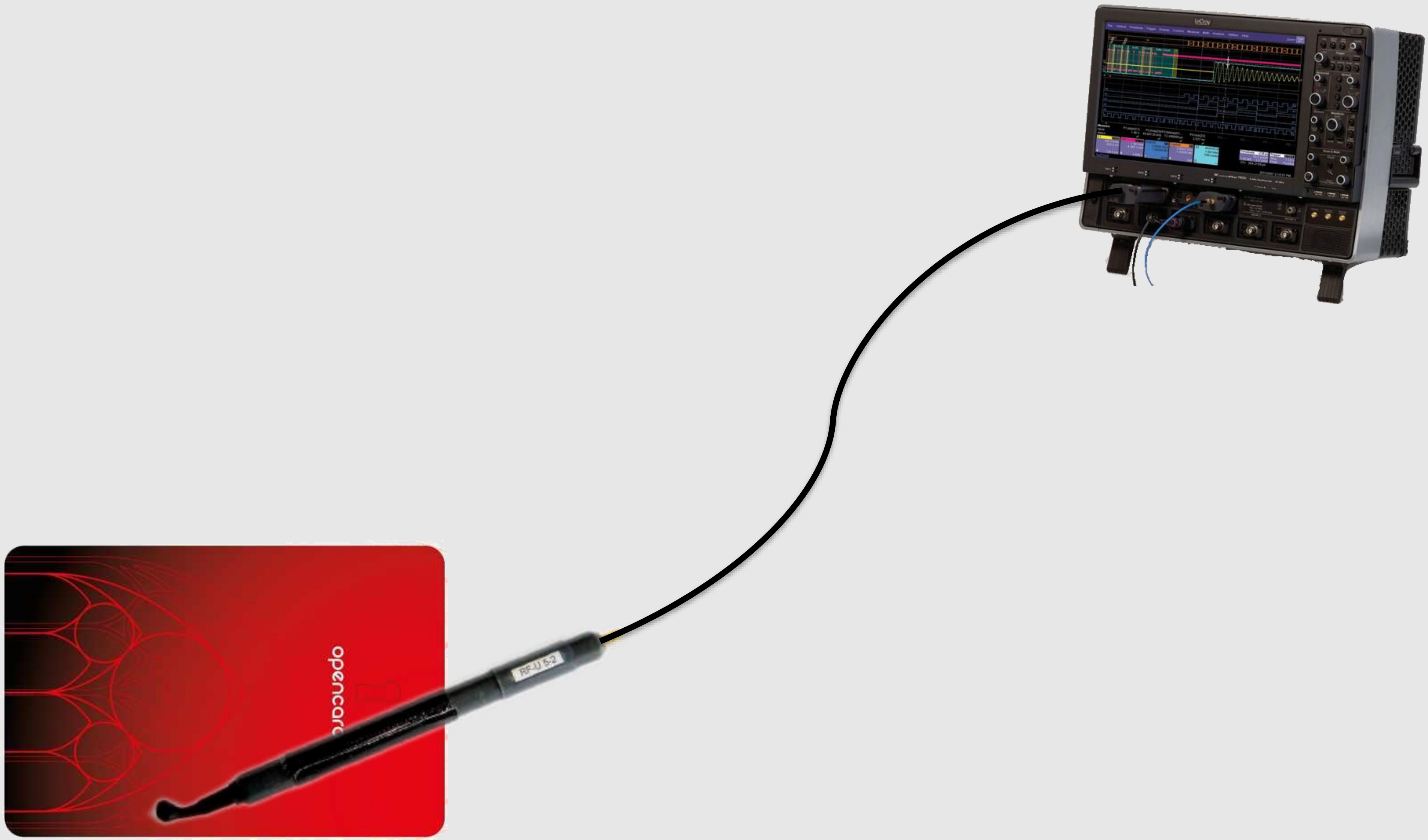
Agenda

RUB

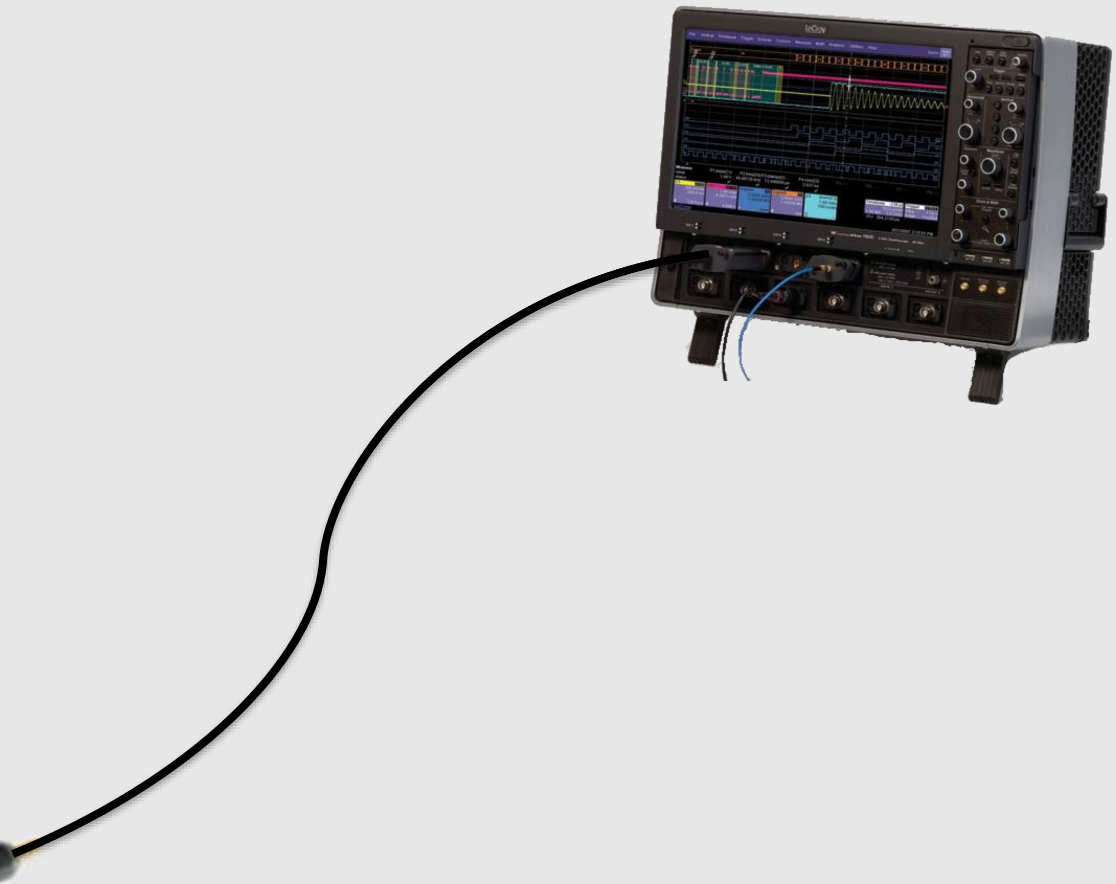
- **Intro:** SCA and (Linear) Transforms
- **Theory:** CPA and Linear Transforms
- **Optimization:** Linear Transforms
- **Results:** Practical Experiments
- **Conclusion:** Lessons Learned

Introduction:
SCA and (Linear) Transforms

Side-Channel Analysis: Basic Setup



Side-Channel Analysis: Basic Setup



Noise
Uncorrelated signals
DC shifts
...

Side-Channel Analysis: Basic Setup

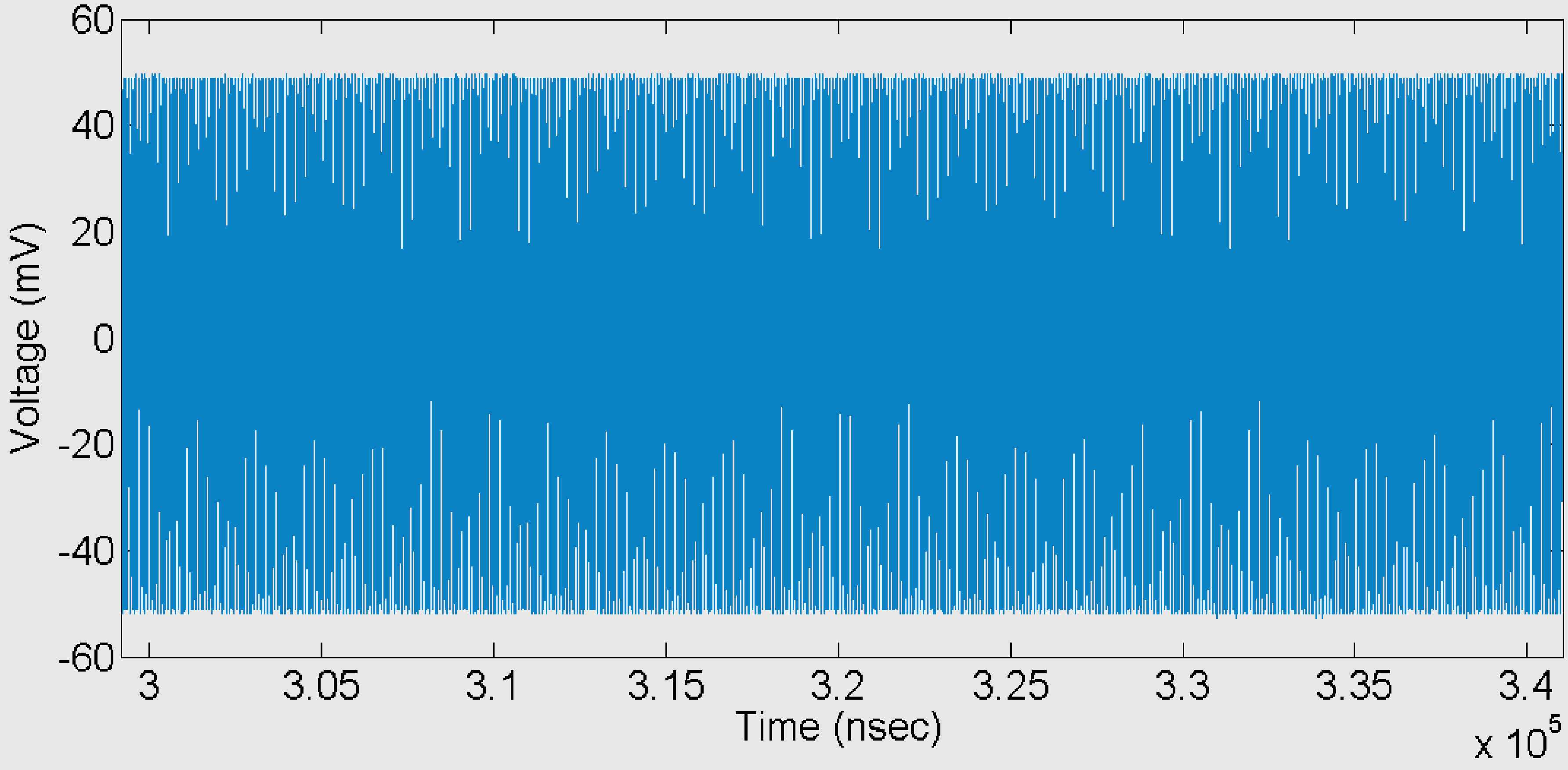
Noise
Uncorrelated signals
DC shifts
...

Filter

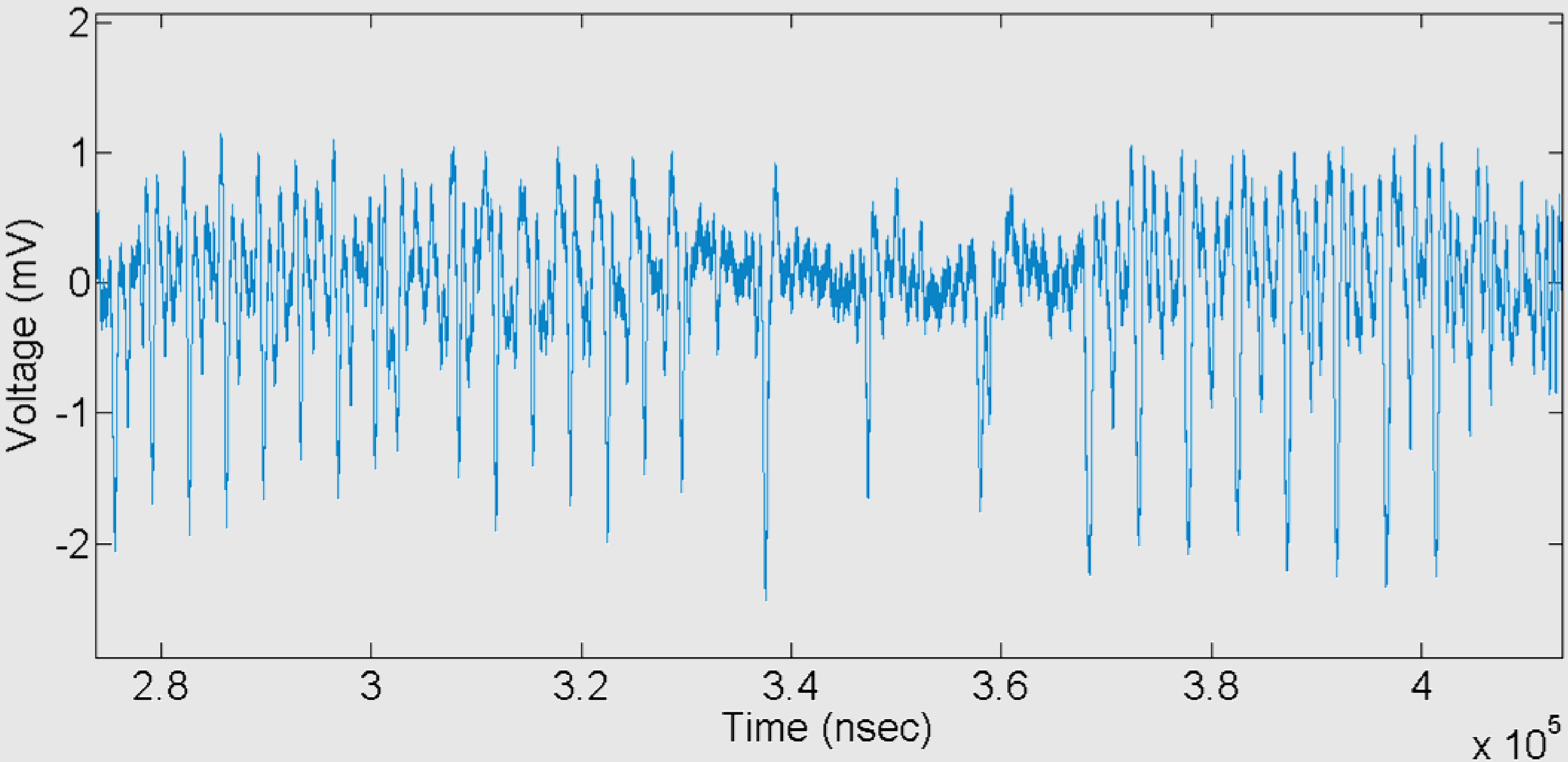
SCA and Filtering

- **Pre-processing** step more or less proposed together with DPA
- Our motivation: Practical experience
  - DESFire
  - ... many more examples

Practical Experience: DESFire



Practical Experience: DESFire



Problems

- Parameter selection is
  - Heuristic
  - Time consuming
  - Based on experience
- **Can we do better?**

How does a linear filter affect the results of Correlation Power Analysis?

Correlation Power Analysis

- **C**orrelation **P**ower **A**nalysis
- Most common distinguisher

- Trace vectors $\vec{x}_i \;\; 0 \leq i < N$
- (Key-dependent) prediction h_i

Matrix Notation

$\boldsymbol{cov(\vec{x}, h)}$

$$\vec{\rho} = \frac{\Sigma_{\vec{x}h}}{\sqrt{diag(\Sigma_{\vec{x}\vec{x}})}\sqrt{\Sigma_{hh}}}$$

$\boldsymbol{var(\vec{x})}$

$\boldsymbol{var(h)}$

Linear Transform

- Weighted sum of each trace

$$y_i = \vec{a}^T * \vec{x}_i$$

- Can realize FIR filter

$$y(t) = \sum_m a(m) \; * \; x(t-m)$$

Linear Transforms

- **Usual approach:**
  1. Compute transform on each trace
  2. Compute correlation

- **Our approach:**
  1. Compute correlation
  2. Compute transform on result

Untransformed Correlation

$$\vec{\rho} = \frac{\Sigma_{\vec{x}h}}{\sqrt{diag\left(\Sigma_{\vec{x}\vec{x}}\right)\sqrt{\Sigma_{hh}}}}$$

Transformed Correlation

$$\rho(\vec{a}) = \frac{\vec{a}^T * \Sigma_{\vec{x}h}}{\sqrt{\vec{a}^T * \Sigma_{\vec{x}\vec{x}} * \vec{a}}\sqrt{\Sigma_{hh}}}$$

Intermezzo

RUB

- **Closed form:**
  Correlation after linear transform
- Needs covariance matrix $\Sigma_{\vec{x}\vec{x}}$
- Can include coefficients for *h*
- Can be combined with non-linear transforms (e.g. frequency domain)

How can we profit from this result?

Idea

- **Usual approach:**
  Select coefficients manually (filter parameters, ...)
- **Our approach:**
  (Numerically) optimize coefficients

Numerical Optimization

Algorithm

Criterion

$$\max_{\vec{a}} f(\vec{a})$$

Optimization Criterion

- Maximize “distinguishability” of **correct** key candidate
- **Semi-profiled:** Correct key known
- Efficient evaluation of f ?

Optimization Criterion (1)

$$f(\vec{a}) = \frac{|\rho_{correct}(\vec{a})|}{mean(|\rho(\vec{a})|)}$$

Optimization Criterion (2)

$$\cdots = \frac{\left|\vec{a}^T \ * \Sigma_{\vec{x}h_{correct}}\right|}{\sum_k \left|{}^1\!/\!. \ * \vec{a}^T \ * \Sigma_{\vec{x}h_k}\right|}$$

Algorithm

- f efficient to evaluate
- MATLAB optimization toolbox
- `fminunc()`
- Avoid **overfitting**: Better choices?

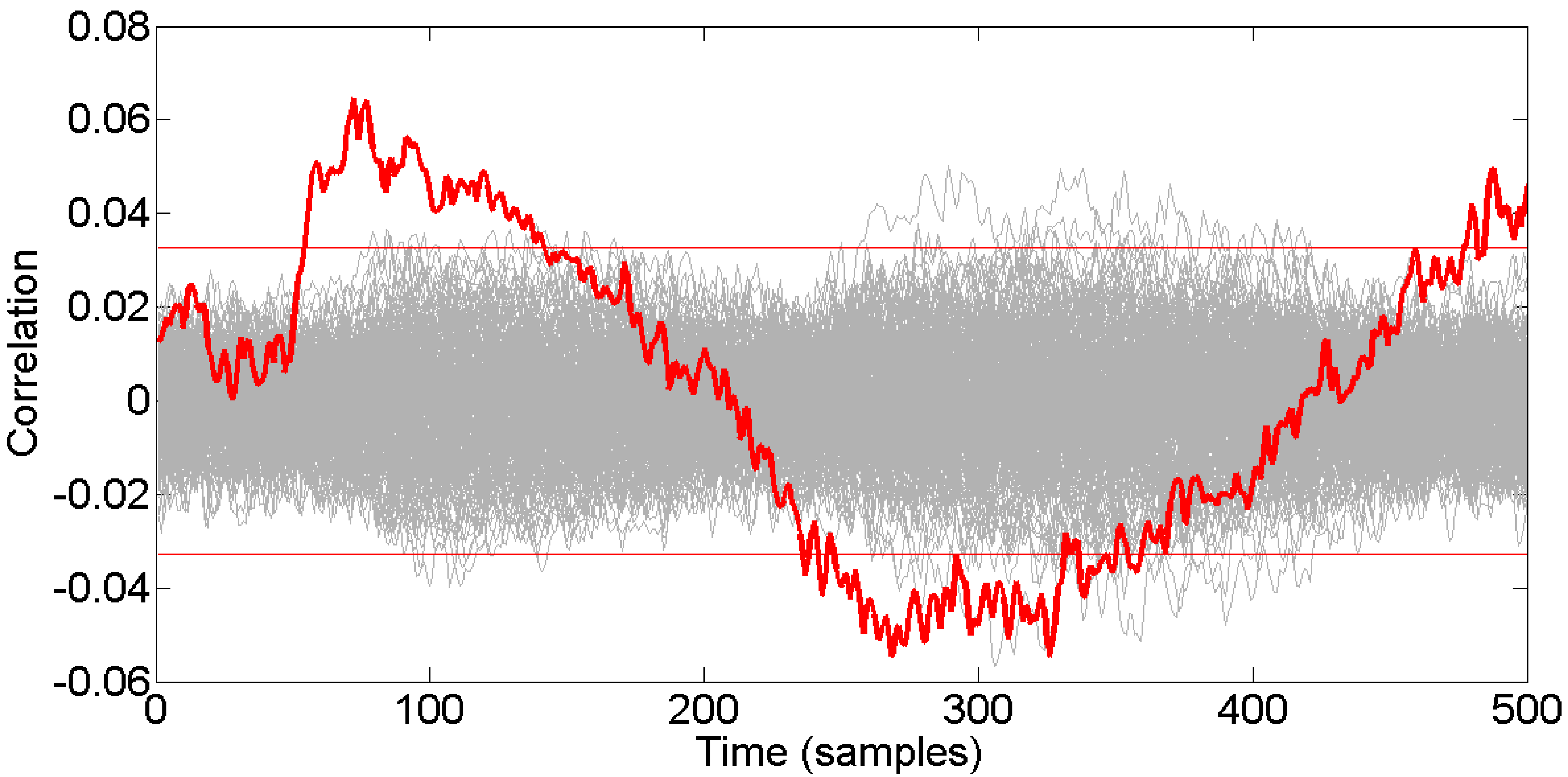
Does it work in practice?

Practical Experiments

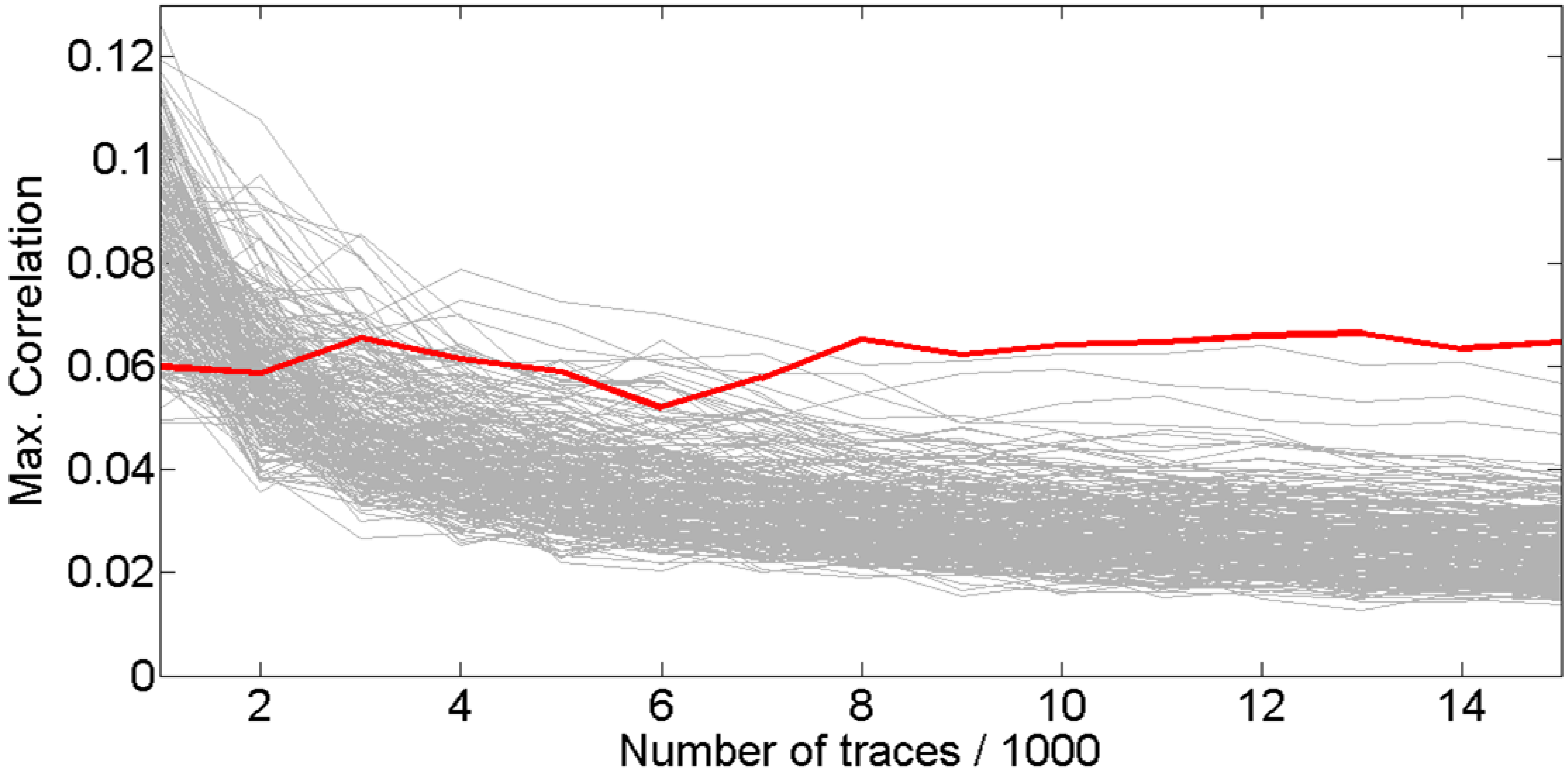
- Some simulations ...
- **Then:** DPA contest v2 traces
- AES on Sasebo G2
- Leakage characteristics **known**
- 15,000 traces

Untransformed Correlation (1)

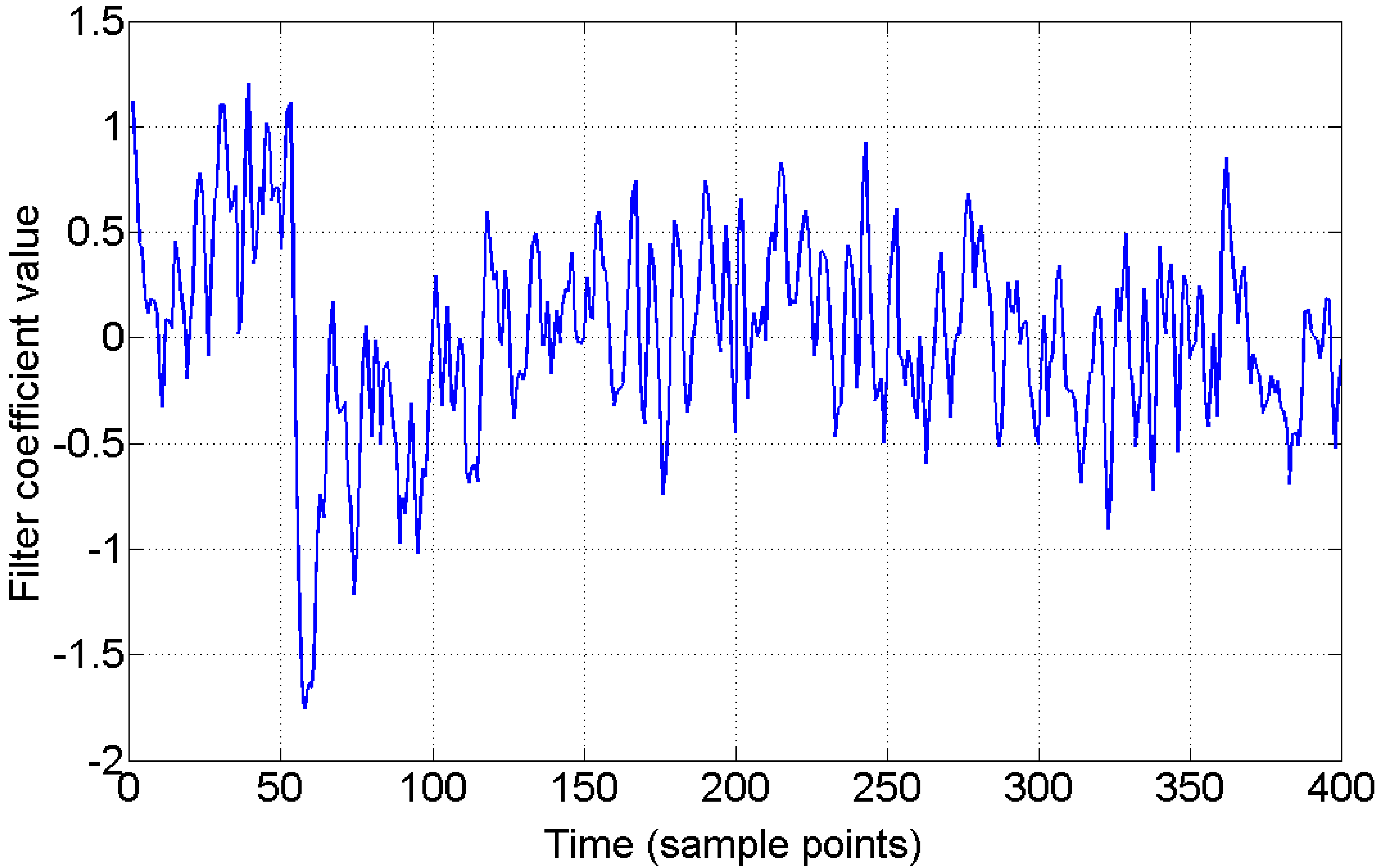
RUB



Untransformed Correlation (2)

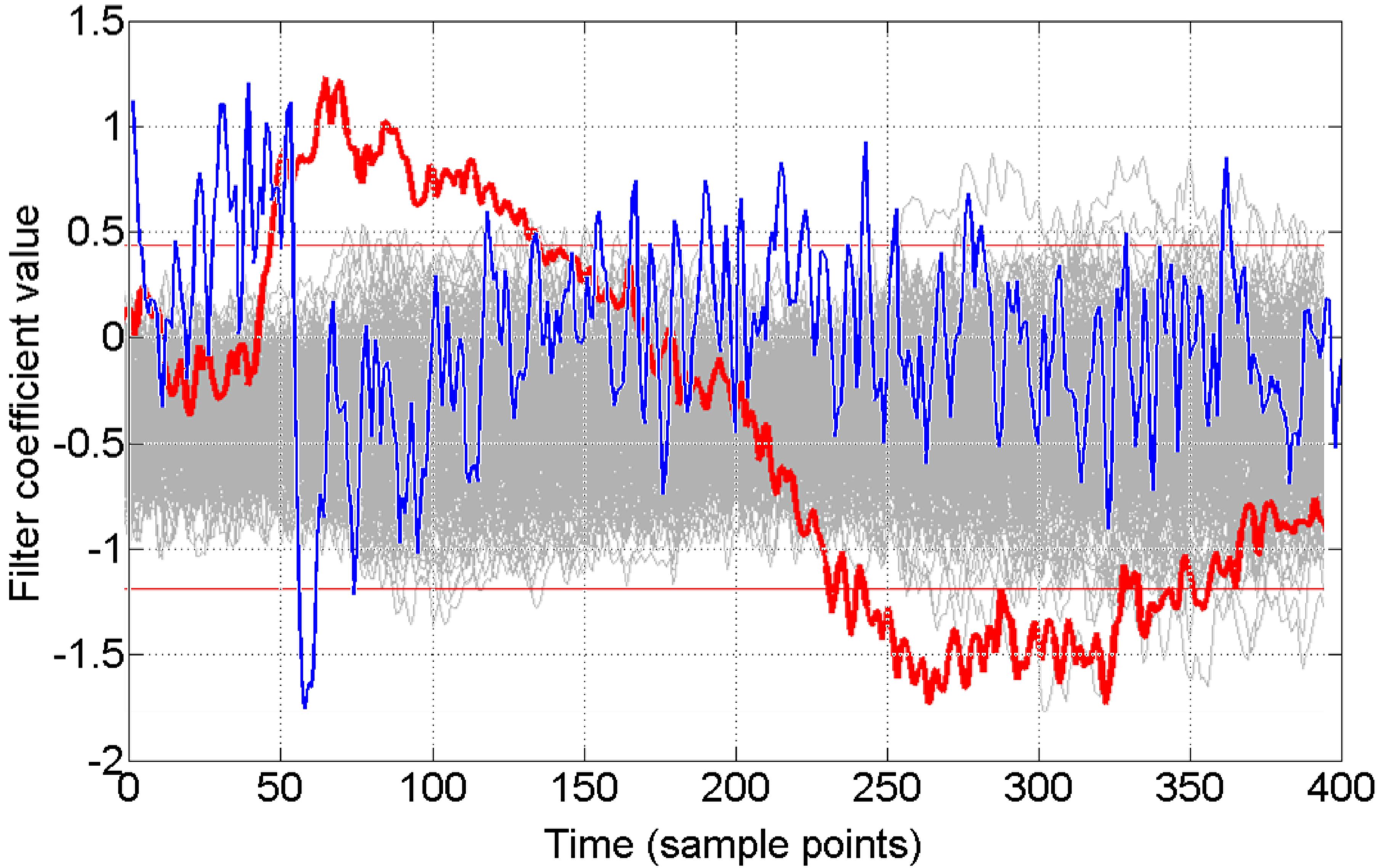


Optimized Transform (1)

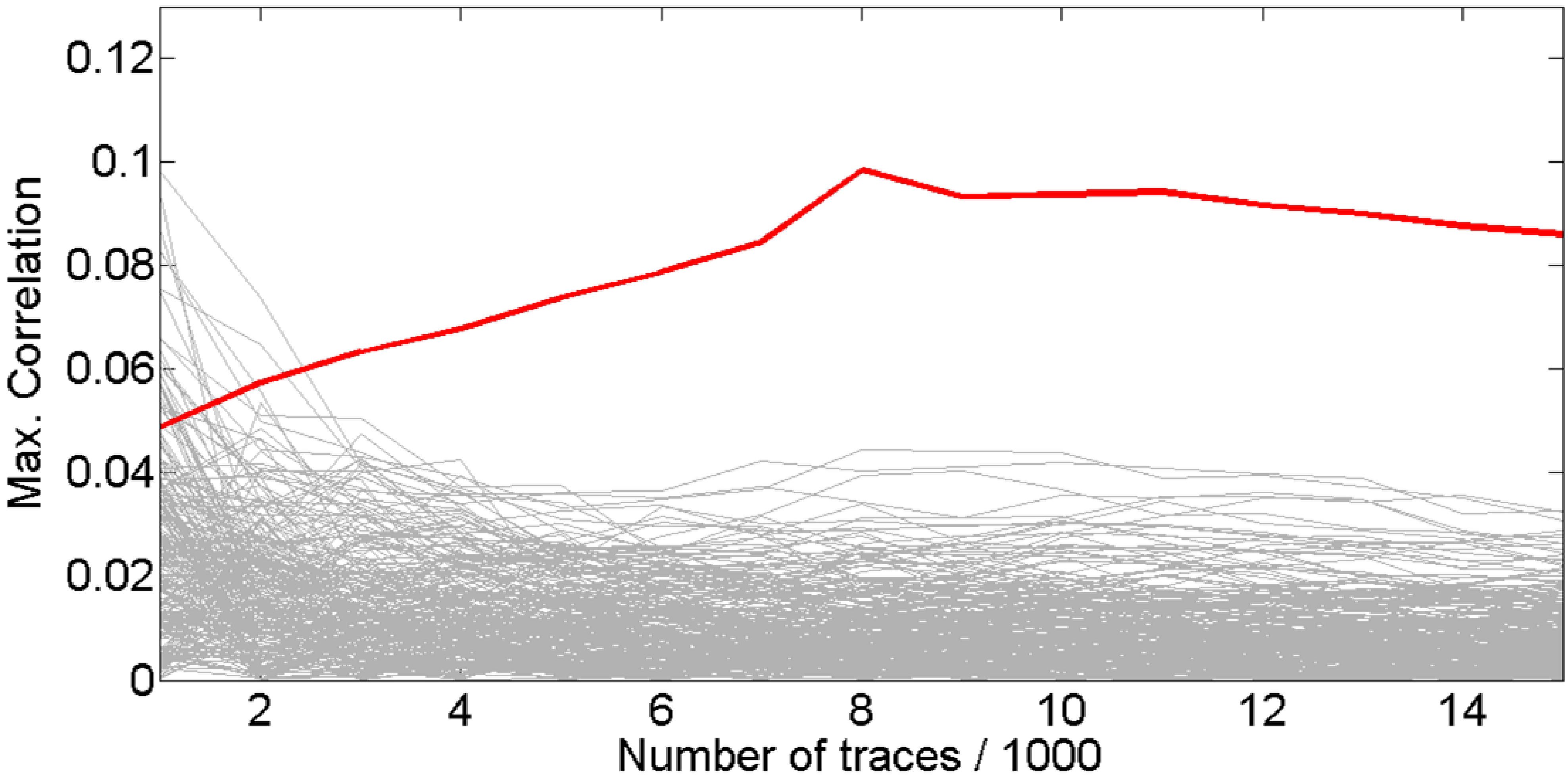


Optimized Transform (2)

RUB



Transformed Correlation



Practical Experiments: Summary

RUB

- From **0.064 → 0.087**
- Better ratio correct vs. incorrect: **1.83 vs. 2.9**
- Not: Full DPA contest framework
- But: Results similar for all S-Boxes

Conclusion:
Lessons Learned

Lessons Learned

- **More systematic way for selecting linear transforms for SCA**
- More practical applications?
- Better criterion?
- Avoid overfitting?
- Analytical solutions?

Thanks!

Questions?

...

or later:

david.oswald@rub.de