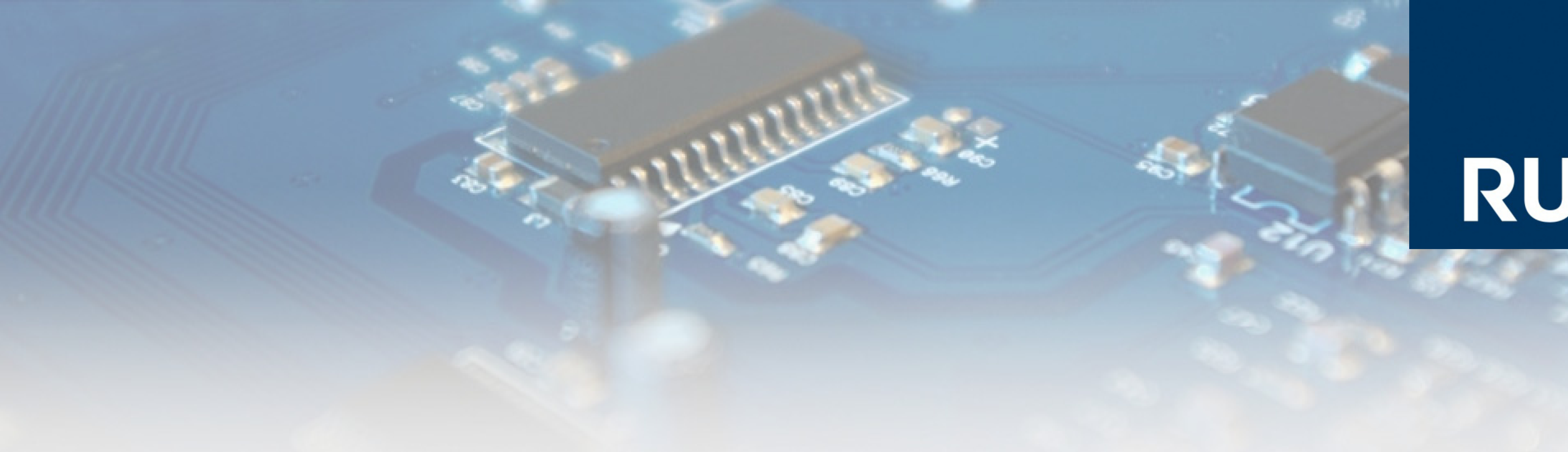


RUHR-UNIVERSITÄT BOCHUM

Correlation-Enhanced Power Analysis Collision Attack

18. August 2010

Amir Moradi, Oliver Mischke, Thomas Eisenbarth

Embedded Security Group, Ruhr University Bochum, Germany
Florida Atlantic University, USA

Horst-Görtz Institut für IT Sicherheit

EmbeddedSecurity

Outline

- How did we start it?
- Implementation
- Finding Leakages
- A new Attack
- Conclusions

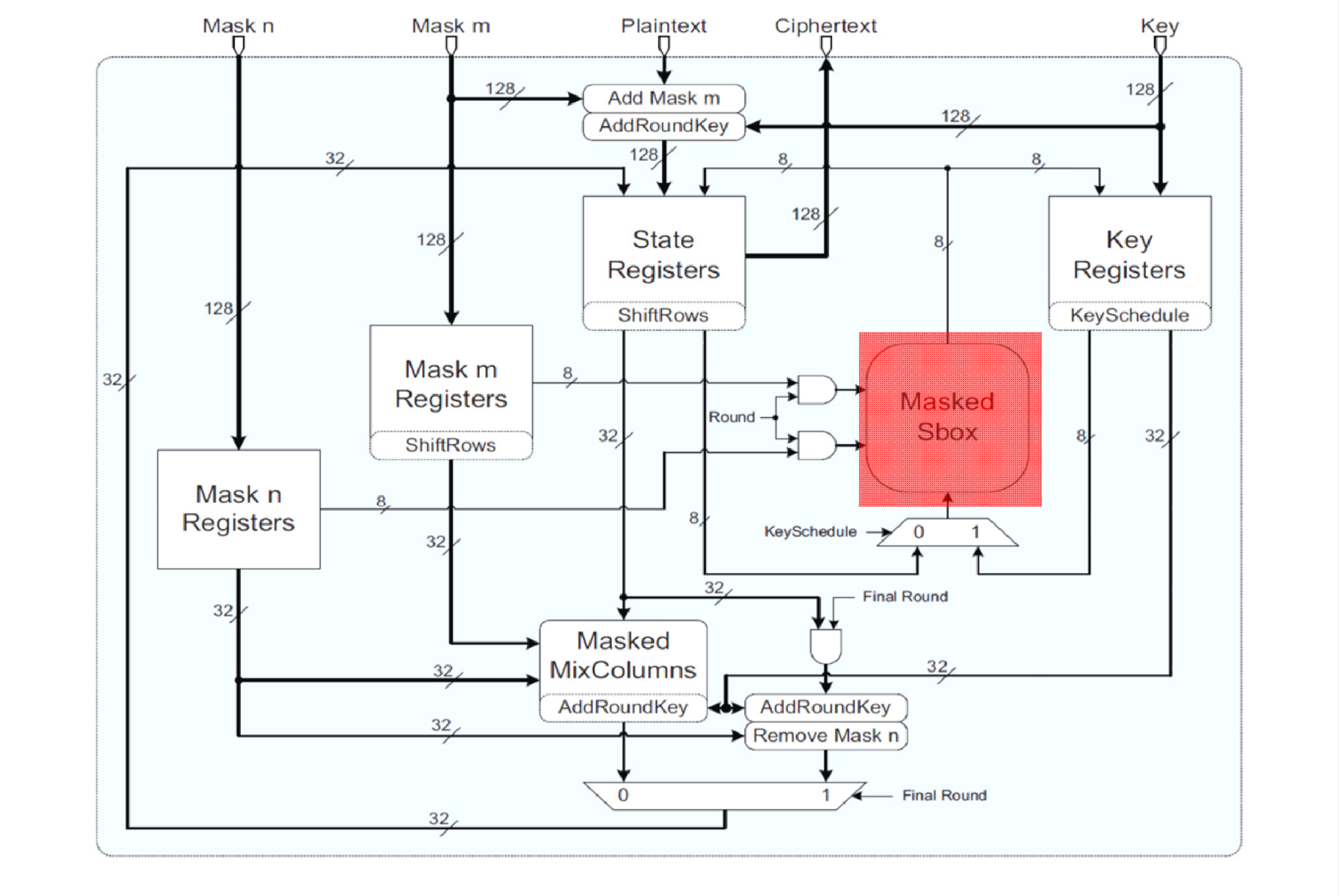
What is the story?

- It comes from a project
  - 1st order resistant lightweight implementation of AES for FPGA/ASIC
- Looking into the literatures
  - smallest masked AES S-box by Canright and Batina
- Implementation
  - Attacks
    - HD/HW models did not work (as expected)

1st Order Leakage of Masking in Hardware

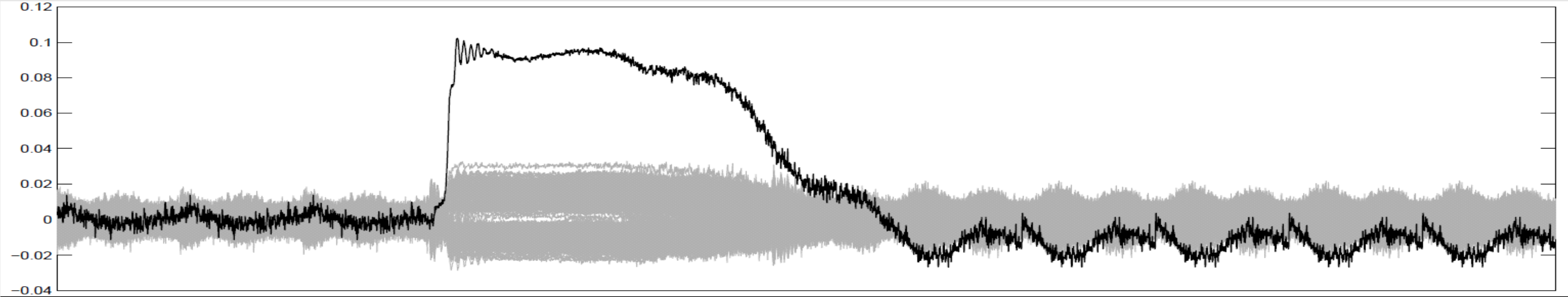
- Usually cannot be exploited by HD/HW models
- Zero-Value model sometimes works
- Toggle-Count Model
- of course, MIA
- We all know why → glitches

An Overview of the Architecture

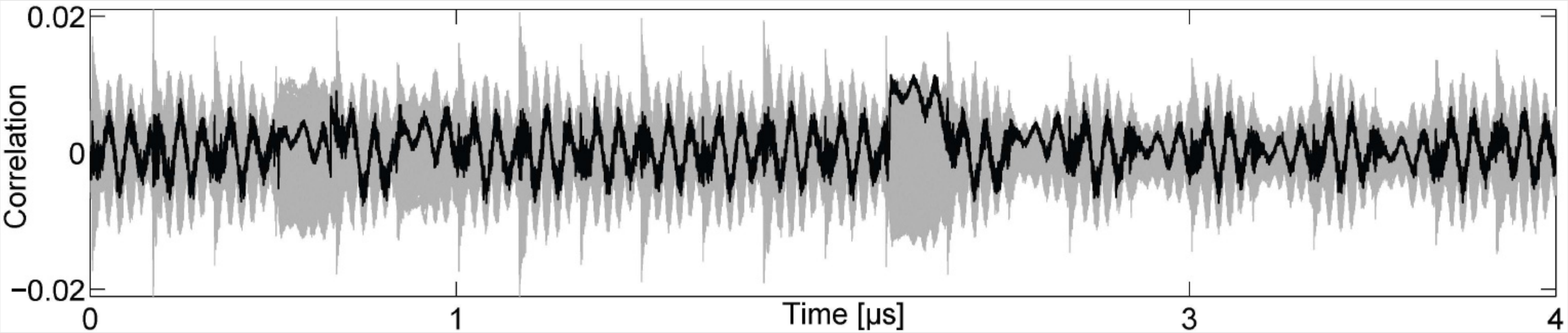


More Evaluation

- Zero-Value Attack worked very well (10K traces)

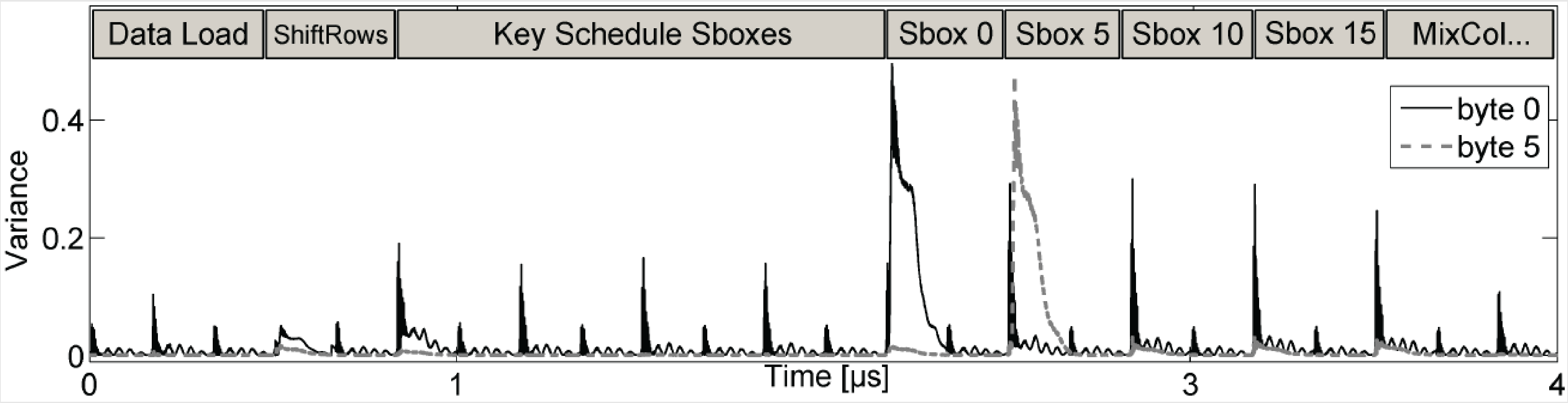


- more concentration of the masked S-box, keeping the hierarchy levels, avoiding any optimization,…(1M traces)



More Evaluation?

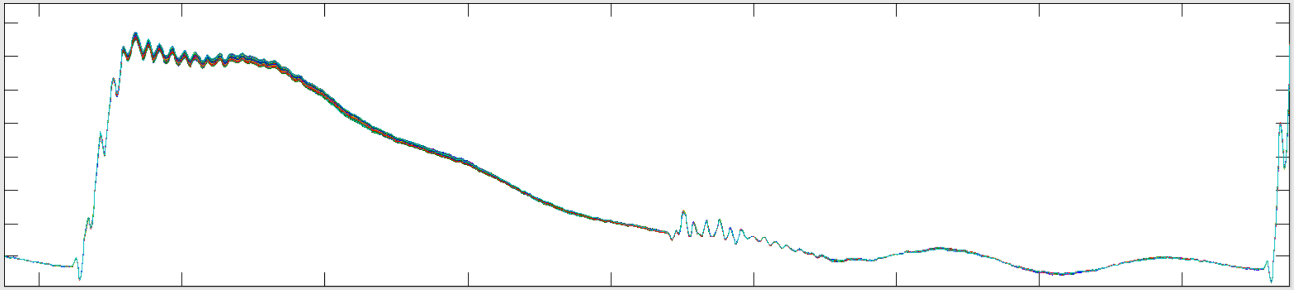
- Templates? (without knowing the key?)
  - First, averaging based on plaintext bytes
    - 256 mean traces for each plaintext byte
    - Variance over mean traces (each plaintext byte separately)



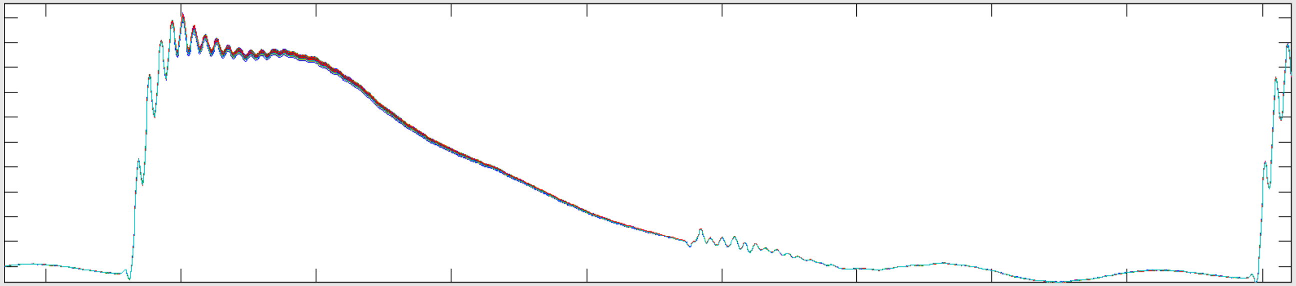
- Something depends on plaintext bytes

Designing an Attack

- Supposing knowing a key byte, we get mean traces for the corresponding plaintext byte

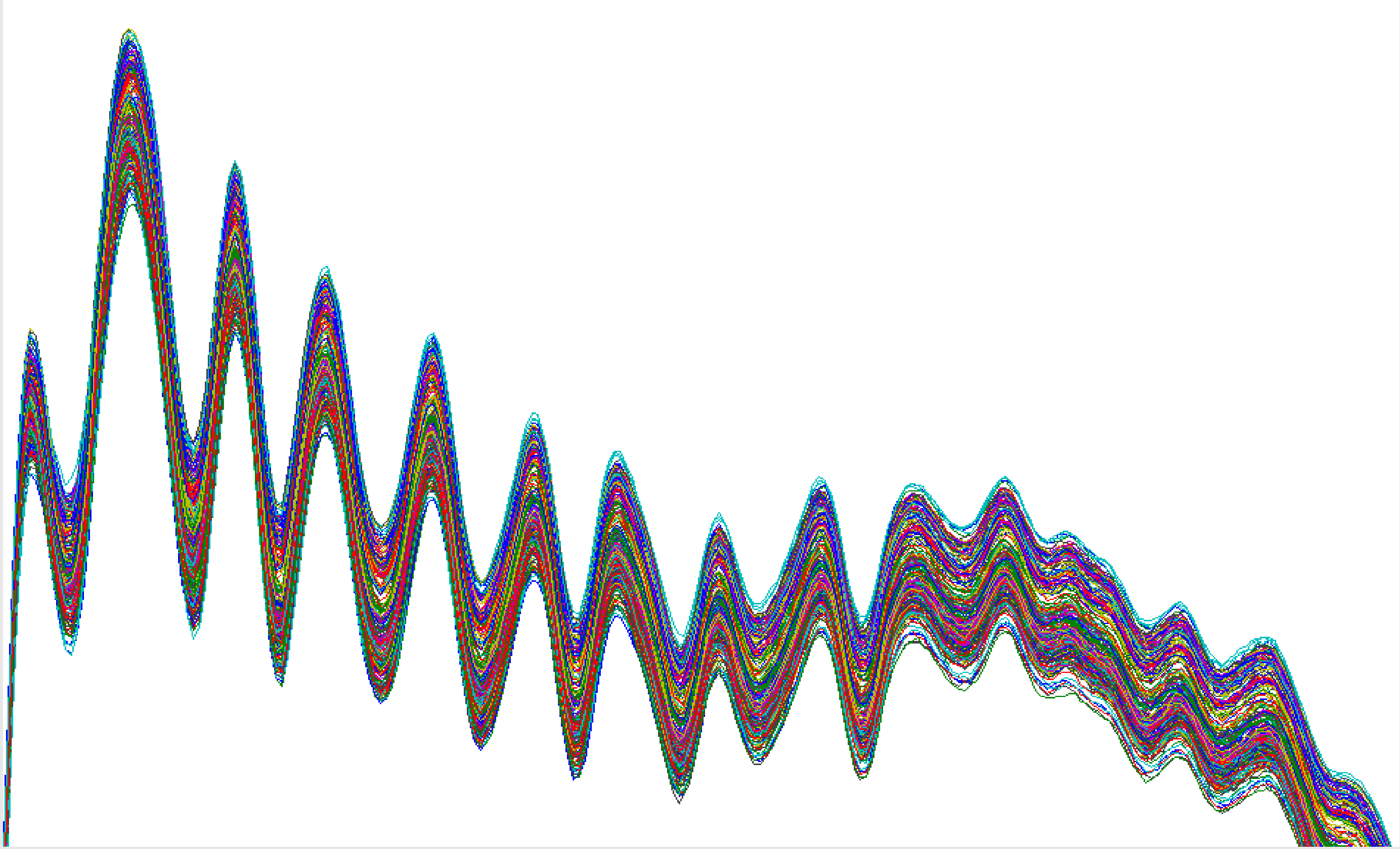


- For another plaintext byte (unknown key), we get mean traces



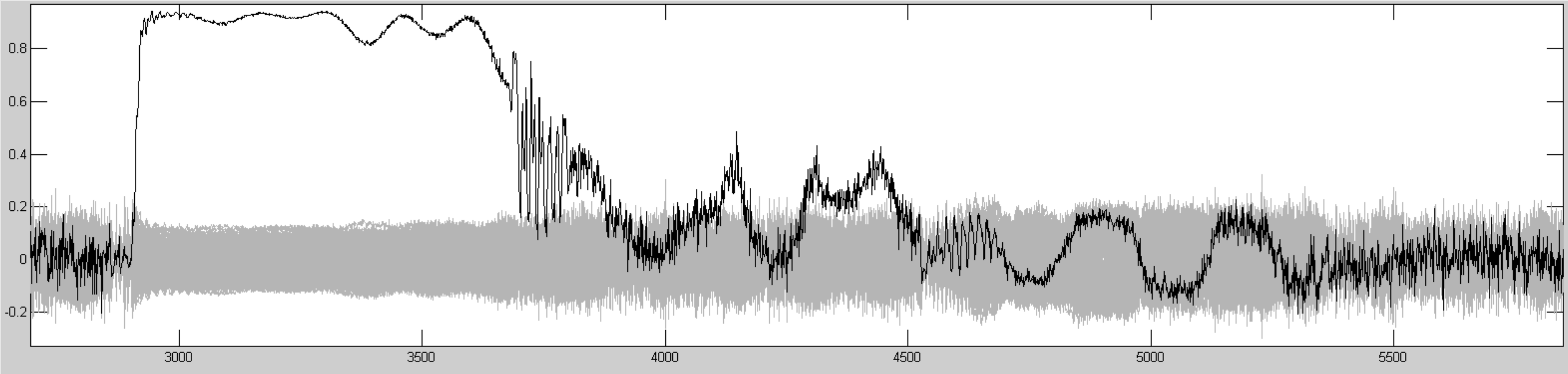
- How are these mean traces related to each other?

Designing an Attack



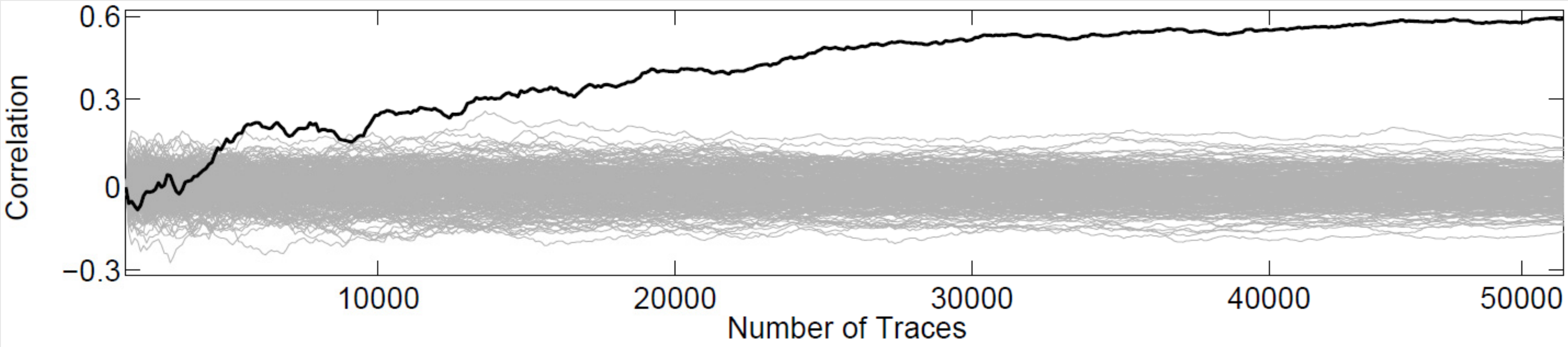
Designing an Attack

- The mean traces for the unknown key bytes can be generated for each key byte hypothesis
- The correct key byte can be found comparing the mean traces at each time instance
  - Correlation helps here!
    - Correlation of two sets of mean traces based on key hypothesis (is almost 1 for right key (due to equal power consumption))



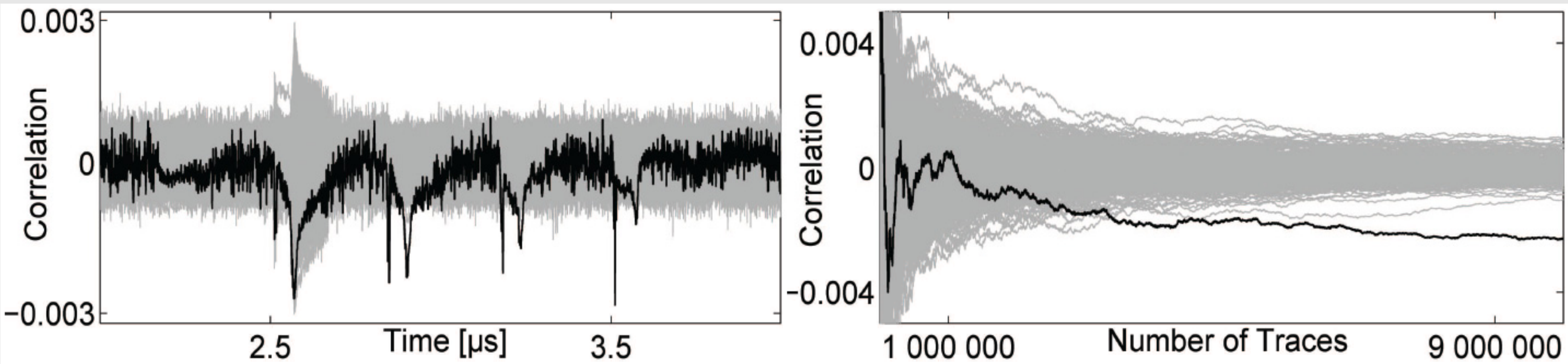
Extending the Attack

- If the first key byte (for the first mean traces) is not known, what we recover is the linear difference between two key bytes: k_1+k_2 , because of addroundkey of AES
  - Linear correlation attack on AES but using all possible collisions!
- Number of required traces?

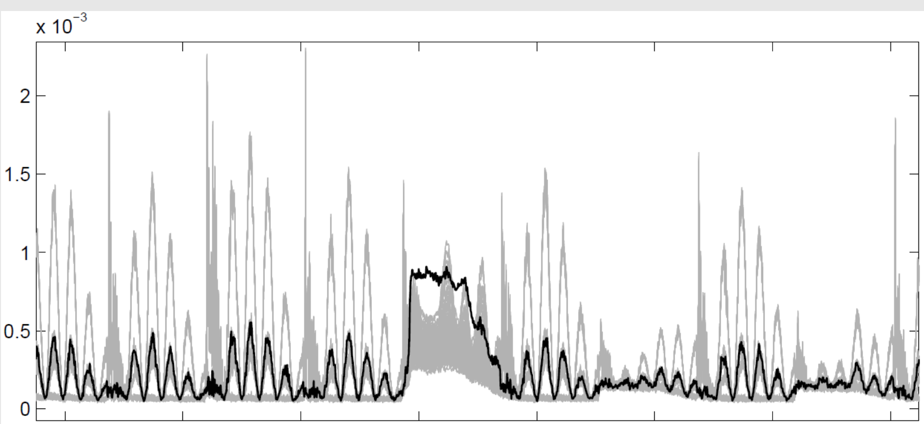


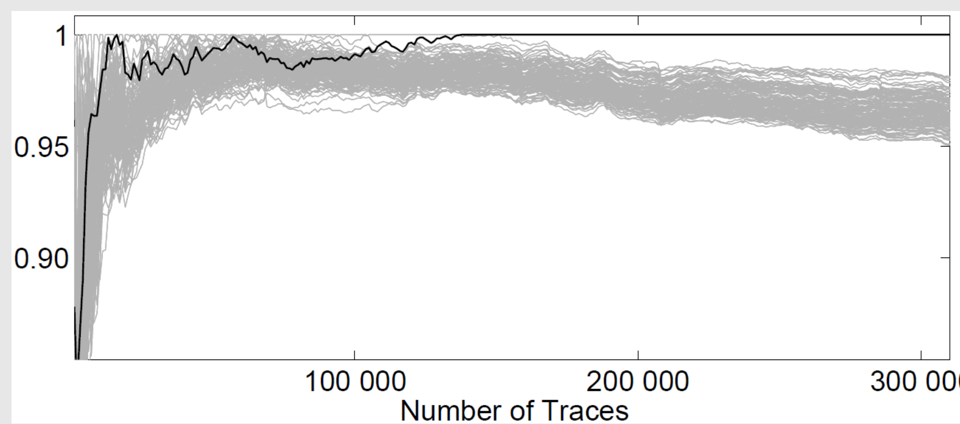
Comparisons?

- 2^{nd} order attack (zero-offset 2DPA), 8M traces required



- MIA, ~ 200K traces required





Why does it work?

- There is one instance of S-box in an 8-bit architecture
  - The power consumption characteristics of the same instance of the S-box is used in mean traces
  - Power consumption of an instance of the S-box is compared to itself in different clock cycles
- What does happen for larger architecture?
  - The same netlist for the S-boxes, even the same placement and routing, but still process variations exists
    - Small differences on power consumption characteristics of different instances of the S-box
  - The same instances of the S-box should be compared

Larger Architectures

- 32-bit architecture
  – Increased noise → more traces (in our case ~ 10 times)
- 128-bit (round-based) implementation
  – Crypto LSI by SASEBO's
  – Efficiency of the attack depends on the similarity of power consumption characteristics of different instances of the S-box

The gain of the attack

- Relation between key bytes
  - 8-bit arch. → 15 relations, 2^8 candidates for the 128-bit key
  - 32-bit arch. → 12 relations, 2^{32} candidates for the 128-bit key
- How to get the correct key?
  - A pair of plain-/ciphertext
  - Continue the attack on the second round of the AES for each key candidate

First Conclusion

- The implemented masking scheme has a 1st order leakage
  - because of an implementation error!
  - glitches should be prevented
  - HOW?
    - Control signals?
    - toward logic styles
    - algorithmically-masked AES S-box implemented by a masked logic styles, e.g., (i)MDPL
    - The circuit grows incredibly

Presence of other Countermeasures

- Shuffling?
  - noise addition → more traces
  - Combing → keeping the # of traces still low
- Logic Styles?
  - tested on an iMDPL chip (32-bit arch.)
    - around 200K traces are enough to get the full key
  - also an Crypto LSI by SASEBO's (MAO, MDPL, WDDL 128-bit arch.)
    - Something between 100K-200K traces are enough
- Any kind of 1st order leakage can be detected given enough traces to estimate the means

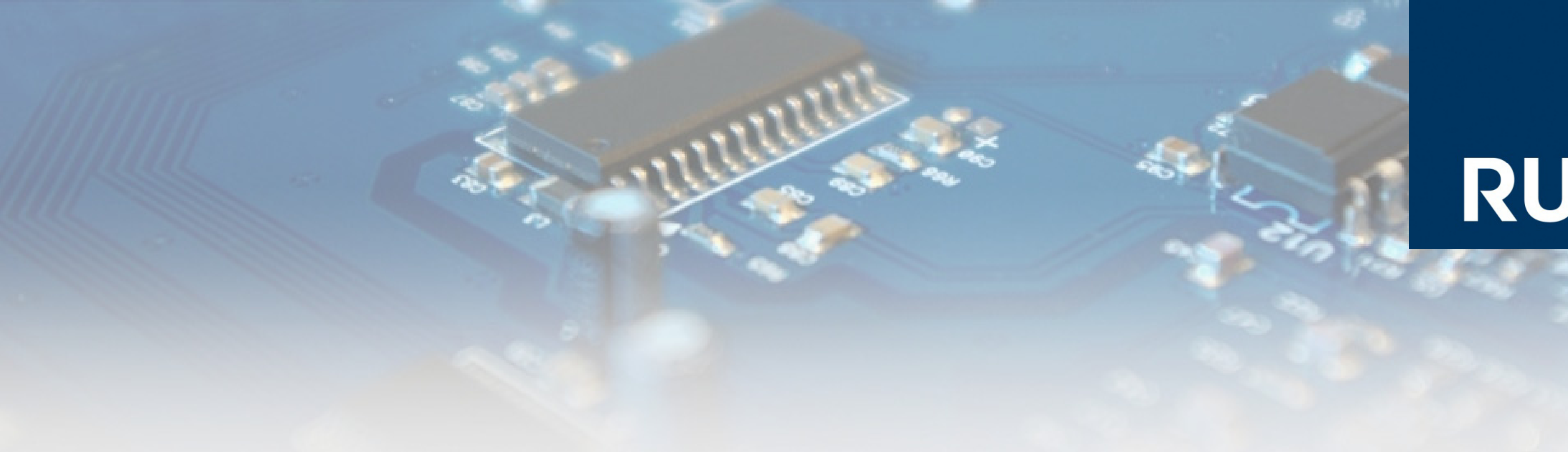
Can the attack be applied to other algorithms?

- A general scheme is presented in an eprint version

  http://eprint.iacr.org/2010/297

  – any kind of 1st order leakage can be exploited (works better if same instances of combinational function (S-box) are repeatedly used)

  – again, it can recover the relation between key portions (bytes, nibbles, etc)



Thanks!

Any questions?

Horst-Görtz Institut für IT Sicherheit

Embedded Security Group, Ruhr University Bochum, Germany
Florida Atlantic University, USA

EmbeddedSecurity