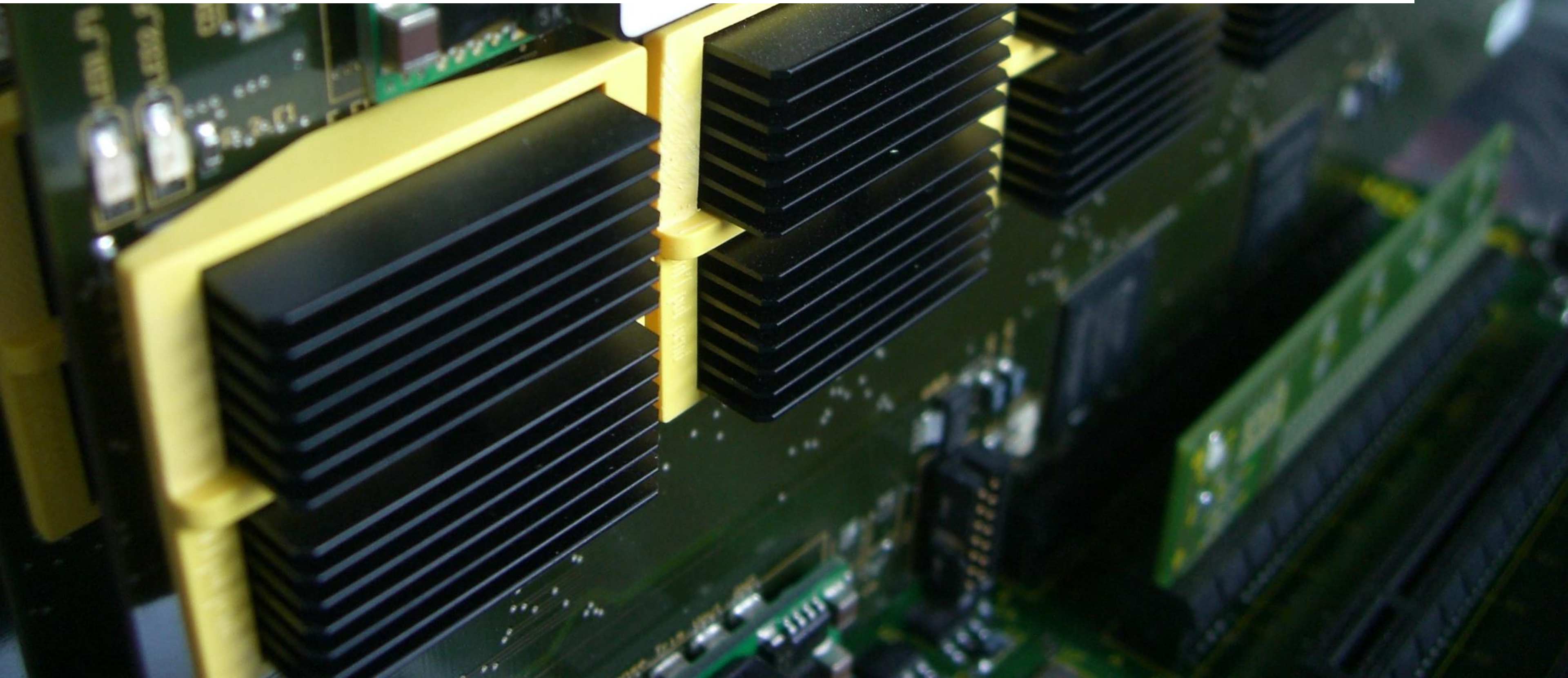
RUHR-UNIVERSITÄT BOCHUM

WHITE-BOX CRYPTOGRAPHY IN THE GRAY BOX

− A HARDWARE IMPLEMENTATION AND ITS SIDE CHANNELS −

PASCAL SASDRICH, AMIR MORADI, TIM GÜNEYSU

23RD INT. CONFERENCE ON FAST SOFTWARE ENCRYPTION, BOCHUM, GERMANY
MARCH 21, 2016



THE STORY OF THIS WORK

HOW DID THIS WORK START?

> *“The challenge that **white-box cryptography** aims to address is to implement a cryptographic algorithm **in software** in such a way that cryptographic assets remain secure even when subject to **white-box attacks**.”*

(www.whiteboxcrypto.com)

SOME QUESTIONS AROSE:

1. If an implementation is secure against white-box attacks, will it be secure against grey-box (i.e. side-channel) attacks as well?
2. Can we use white-box cryptography or adopt its ideas to build side-channel secure implementations?
3. Why do we only address software implementations? Can we implement white-box cryptography in hardware, too?

THIS IS THE STORY OF A

WHITE-BOX HARDWARE IMPLEMENTATION AND ITS SIDE CHANNELS.

CRYPTOGRAPHIC ADVERSARY MODELS

Modern cryptography differentiates between three models to estimate the capabilities of an adversary:

BLACK-BOX ADVERSARY MODEL:

- trusted environment
- secure communication endpoints
- adversary can only observe input/output behavior (black-box)

GREY-BOX ADVERSARY MODEL:

- adversary has limited access to implementation internals
- usually targets implementations rather than algorithms

WHITE-BOX ADVERSARY MODEL:

- capabilities are virtually unlimited
- full control over implementation and execution environment
- white-box secure implementation behaves as virtual black-box

GENERAL IDEA OF WHITE-BOX CRYPTOGRAPHY

An ideal white-box implementation would be a single look-up table (for a fixed secret key).

- Obviously this is impractical for modern ciphers with block and key sizes of 128 bits and more.

So, practically feasible approaches for round-based symmetric block ciphers look like:

$$\underbrace{(\mathsf{f}^{(r+1)})^{-1} \circ \mathsf{E}^{r} \circ \mathsf{f}^{r}}_{table} \circ \cdots \circ \underbrace{(\mathsf{f}^{(3)})^{-1} \circ \mathsf{E}^{2} \circ \mathsf{f}^{2}}_{table} \circ \underbrace{(\mathsf{f}^{(2)})^{-1} \circ \mathsf{E}^{1} \circ \mathsf{f}^{1}}_{table}$$

$$= (\mathsf{f}^{(r+1)})^{-1} \circ \mathsf{E}^{r} \circ \cdots \circ \mathsf{E}^{2} \circ \mathsf{E}^{1} \circ \mathsf{f}^{1} = (\mathsf{f}^{(r+1)})^{-1} \circ \mathsf{E}_{K} \circ \mathsf{f}^{1},$$

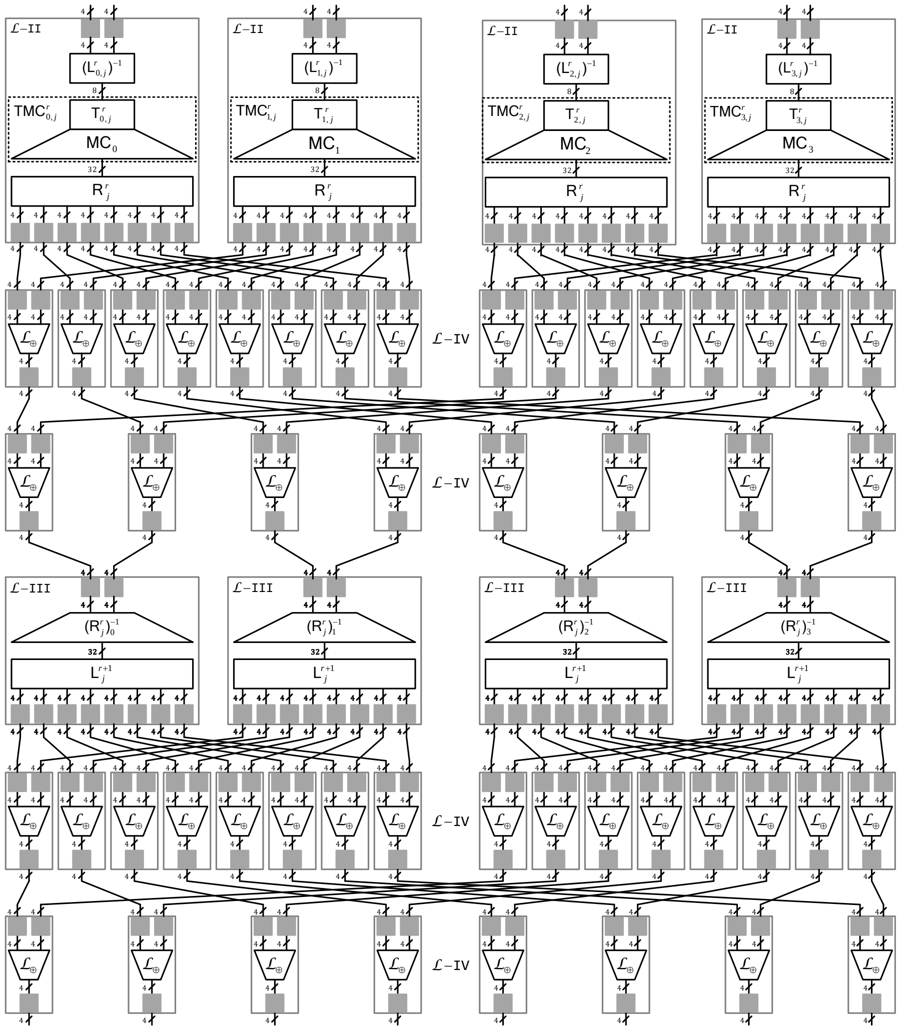
This principle was initially proposed by Chow et al. for DES [1] and AES [2] in 2002.

WHITE-BOX IMPLEMENTATIONS CAN BE SEEN AS

NETWORK OF RANDOMIZED LOOK-UP TABLES.

[1] S. Chow, P. A. Eisen, H. Johnson, and P. C. van Oorschot. A White-Box DES Implementation for DRM Applications.
[2] S. Chow, P. A. Eisen, H. Johnson, and P. C. van Oorschot. White-Box Cryptography and an AES Implementation.

HARDWARE WHITE-BOX IMPLEMENTATION OF AES



DESIGN AND CONSTRUCTION IN FOUR STEPS:

1. ***PARTIAL EVALUTATION***

   S-box and key addition are merged (T-Box)

2. ***MATRIX PARTITIONING***

   MixColumns is added to T-Box (TMC-Box)

3. ***MIXING BIJECTIONS***

   linear encodings (8-bit and 32-bit) are added

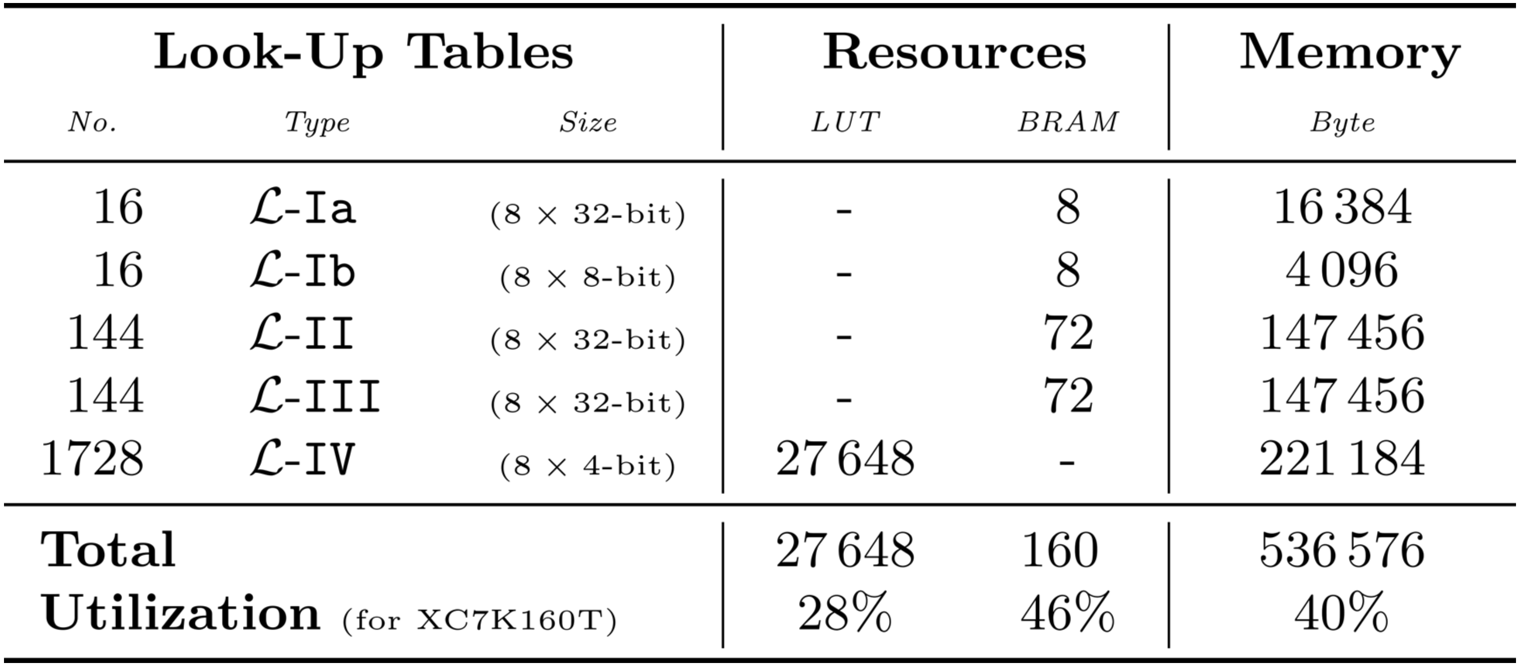
4. ***NIBBLE ENCODINGS***

   4-bit non-linear nibble encodings are applied to all tables

HARDWARE (FPGA) IMPLEMENTATION:

- $\mathcal{L}$-II and $\mathcal{L}$-III are mapped into BRAM
- $\mathcal{L}$-IV is mapped into LUTs

RESULTS FOR FPGA BASED IMPLEMENTATION



Look-Up Tables			Resources		Memory
No.	*Type*	*Size*	*LUT*	*BRAM*	*Byte*
16	$\mathcal{L}$-Ia	(8 × 32-bit)	-	8	16 384
16	$\mathcal{L}$-Ib	(8 × 8-bit)	-	8	4 096
144	$\mathcal{L}$-II	(8 × 32-bit)	-	72	147 456
144	$\mathcal{L}$-III	(8 × 32-bit)	-	72	147 456
1728	$\mathcal{L}$-IV	(8 × 4-bit)	27 648	-	221 184
Total			27 648	160	536 576
Utilization (for XC7K160T)			28%	46%	40%

SIDE-CHANNEL ANALYSIS

OUR SETUP:

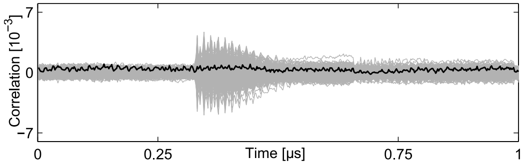
- SAKURA-X Board (Kintex-7)
- 500 MS/s, FPGA@3MHz

EVALUATION:

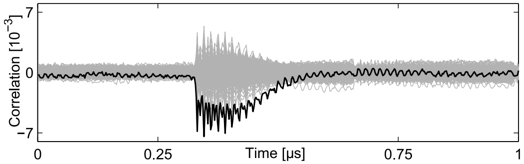
- 10,000,000 power traces
- classical (single bit) DPA

RESULTS:

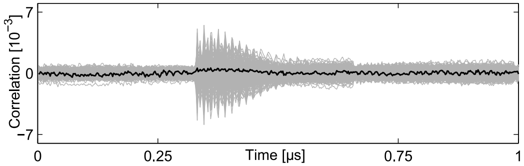
- target value: 5th S-Box output
- key hypotheses: 8-bit (256)
- one bit allowed to recover key (bit 2)



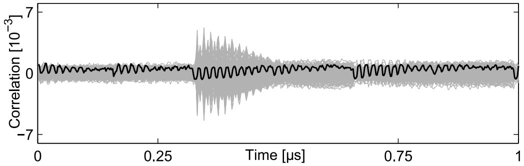
Bit 1



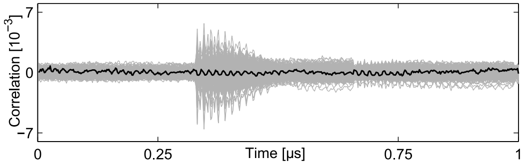
Bit 2



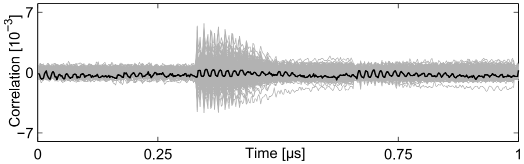
Bit 3



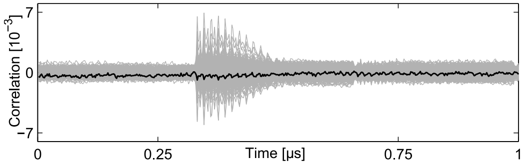
Bit 4



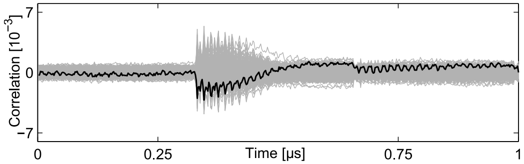
Bit 5



Bit 6



Bit 7



Bit 8

WHY IS A CLASSICAL DPA POSSIBLE?

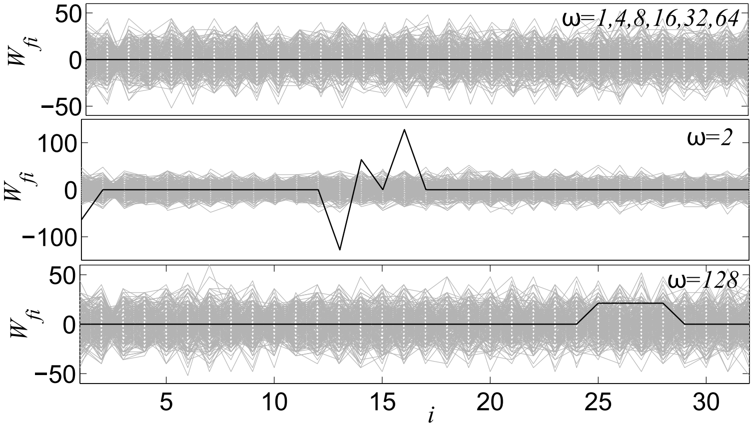
MATHEMATICAL ANALYSIS

TO UNDERSTAND THE PROBLEM, WE APPLIED A WELL KNOWN TOOL FOR BOOLEAN FUNCTIONS:

Definition 1. *Let* $x = < x_1, ..., x_n >$, $\omega = < \omega_1, ..., \omega_n >$ *be elements of* $\{0,1\}^n$ *and* $x \cdot \omega = x_1\omega_1 \oplus ... \oplus x_n\omega_n$. *Let* $f(x)$ *be a Boolean function of* n *variables. Then the Walsh transform of the function* $f(x)$ *is a real valued function over* $\{0,1\}^n$ *that can be defined as* $W_f(\omega) = \sum_{x\in\{0,1\}^n} (-1)^{f(x)\oplus x\cdot\omega}$.

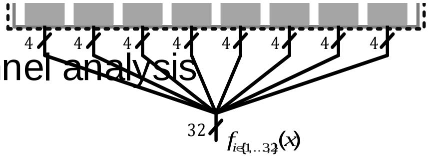
MATHEMATICAL EVALUATION OF $\mathcal{L}$-`Ia` TABLE:

- assume external encodings are known or non-existing
- consider table as 32 different Boolean functions f_i
- calculate Walsh transform for all f_i and all key candidates (for different ω)



RESULTS:

- Walsh transform for ω with HW(ω) = 1 confirm results of side-channel analysis
- directly related to single bit DPA



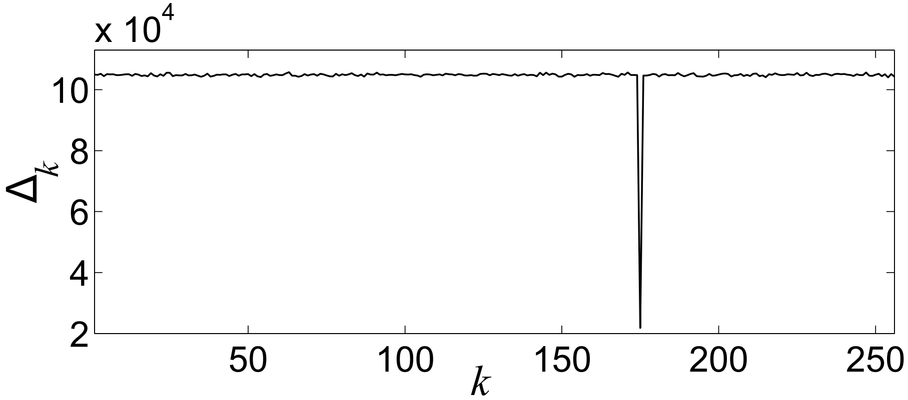
HOW TO PREVENT SUCH ATTACKS?

WE HAVE TO INTRODUCE A SECOND CONCEPT:

Definition 2. *Iff the Walsh transform W_f of a Boolean function $f(x_1, ..., x_n)$ satisfies $W_f(\omega) = 0$, for $0 \leq HW(\omega) \leq m$, it is called a balanced m-th order correlation immune (CI) function or an m-resilient function, where HW stands for Hamming weight.*

CAN WE AVOID ATTACKS BY USING 1ST-ORDER CORRELATION IMMUNE FUNCTIONS?

- all f_i will be m -th order correlation immune $(m \geq 1)$ for the correct key guess
- not necessary the case for a wrong key guess
- then, simply compute:



CONCLUSION AND FUTURE WORK

THE END OF THE STORY:

1. We presented the first AES white-box implementation realized in hardware.
2. Provided results of a practical grey-box (side-channel) analysis and revealed side channels.
3. Investigated underlying mathematical reasons for discovered vulnerabilities.

WHAT HAS TO BE DONE IN FUTURE WORK?

1. Further investigations for linear/non-linear encodings. Specify requirements to prevent analysis through imbalances in Walsh transformations.

2. Enhance white-box security by countermeasures to prevent grey-box attacks, e.g. using dynamically updated encodings.

RUHR-UNIVERSITÄT BOCHUM

hg EMSEC

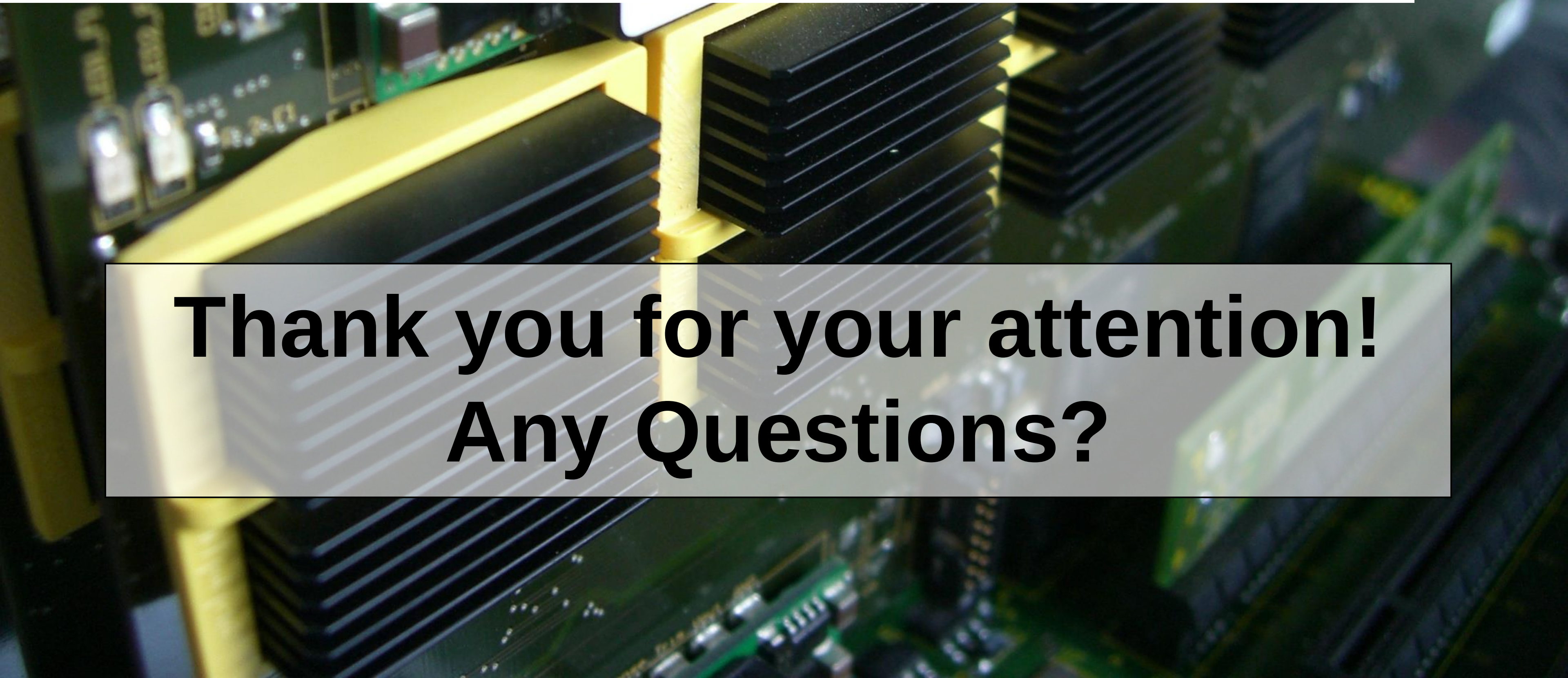
RUB

WHITE-BOX CRYPTOGRAPHY IN THE GRAY BOX

– A HARDWARE IMPLEMENTATION AND ITS SIDE CHANNELS –

pascal.sasdrich@rub.de

23RD INT. CONFERENCE ON FAST SOFTWARE ENCRYPTION, BOCHUM, GERMANY MARCH 21, 2016



Thank you for your attention!
Any Questions?