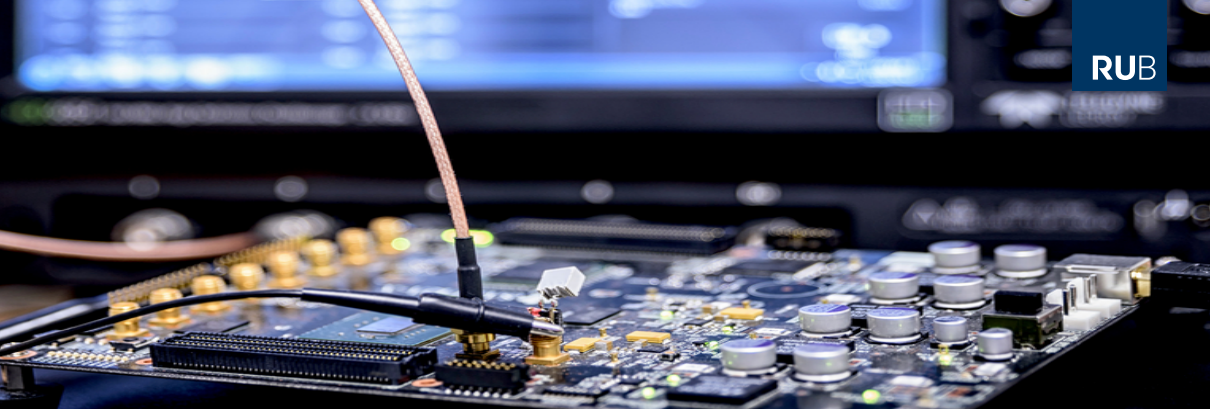


RUHR-UNIVERSITÄT BOCHUM

Breaking and Protecting the CRYSTAL

A Side-channel Analysis of Dilithium in Hardware

Hauke Steffen[1] **Georg Land**[2] **Lucie Kogelheide**[3] **Tim Güneysu**[2,4]

[1]TÜV Informationstechnik GmbH, Essen, Germany
[2]Chair for Security Engineering, Ruhr-Universität Bochum, Germany
[3]BWI GmbH, Bonn, Germany
[4]DFKI GmbH, Cyber-Physical Systems, Bremen, Germany

August 18, 2023

Motivation: NIST Signatures

RUB

Dilithium **Falcon** **SPHINCS$^+$**

Motivation: NIST Signatures

	Dilithium	Falcon	SPHINCS$^+$
Embedded use cases?	✓	✓	(✓)

Motivation: NIST Signatures

	Dilithium	Falcon	SPHINCS$^+$
Embedded use cases?	✓	✓	(✓)
Masking feasible?	✓	✗	✗

Motivation: Dilithium Side-Channel Research

Number of papers on **software** side-channel security of Dilithium: 5+.

Number of papers on **hardware** aspects: none.

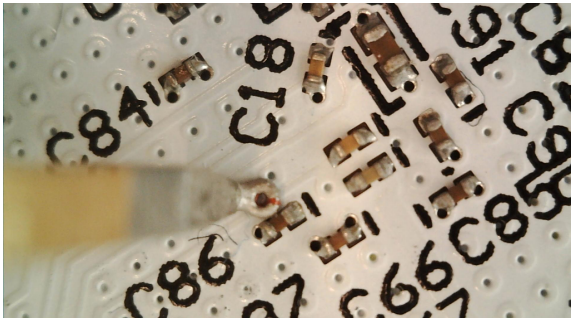
Our Work

1. First side-channel analysis of a Dilithium hardware implementation, with special emphasis on practicality.

2. Two attack strategies: SPA and CPA.

3. Efficient countermeasures.

Measurement Setup



Target:

- discovery board with Artix-7 FPGA, 100 MHz
- unaltered, no integration of measurement resistor, no opening of FPGA package

Setup:

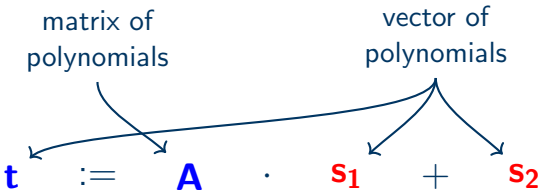
- near-field probe next to capacitor in the power path
- EM emanation is proportional to power consumption of the whole FPGA

Dilithium Keys

$$\mathbf{t} := \mathbf{A} \cdot \mathbf{s}_1 + \mathbf{s}_2$$

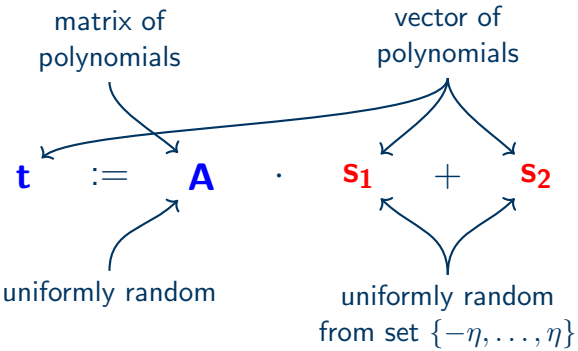
public key — secret key

Dilithium Keys



public key — secret key

Dilithium Keys



public key — secret key

$\eta \in 2, 4$

Secret Key Range

$\eta = 2$

$\overline{x}$	$x = \eta - \overline{x} \bmod q$	HW(x)
0	0x000002	1
1	0x000001	1
2	0x000000	0
3	0x7fe000	10
4	0x7fdfff	22

$\eta = 4$

$\overline{x}$	$x = \eta - \overline{x} \bmod q$	HW(x)
0	0x000004	1
1	0x000003	2
2	0x000002	1
3	0x000001	1
4	0x000000	0
5	0x7fe000	10
6	0x7fdfff	22
7	0x7fdffe	21
8	0x7fdffd	21

Secret Key Range

RUB

$\eta = 2$

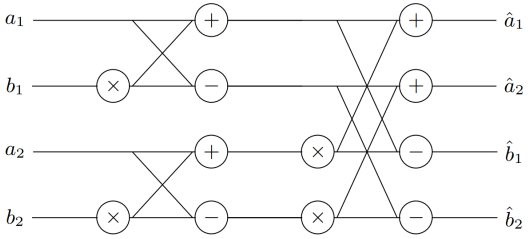
$\overline{x}$	$x = \eta - \overline{x} \bmod q$	HW(x)
0	0x000002	1
1	0x000001	1
2	0x000000	0
3	0x7fe000	10
4	0x7fdfff	22

$\eta = 4$

$\overline{x}$	$x = \eta - \overline{x} \bmod q$	HW(x)
0	0x000004	1
1	0x000003	2
2	0x000002	1
3	0x000001	1
4	0x000000	0
5	0x7fe000	10
6	0x7fdfff	22
7	0x7fdffe	21
8	0x7fdffd	21

Diverse Hamming weight! Can we classify each case when processed?

NTT Butterfly Unit



2x2 butterfly:

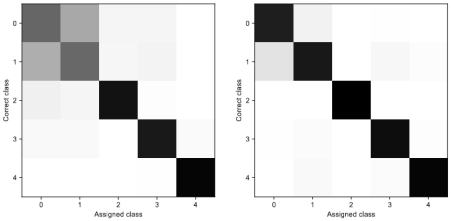
- four coefficients processed in parallel
- multiplication with constants (for b_1, a_2, b_2) yields more diverse power signature (better classification!)

Profiled SPA on First NTT Stage

First stage: Attacker has access to a profiling device with full control, also over secret key.

Second stage: Attacker obtains access to target device, aims at obtaining secret key.

Profiled SPA on First NTT Stage

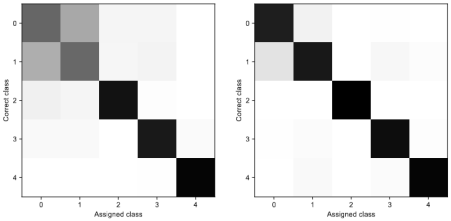


(a) a_1 (b) b_1

(c) a_2 (d) b_2

- 50 000 profiling traces
- 5 000 attack traces
- displayed: probability that a single trace is assigned a class given its known class

Profiled SPA on First NTT Stage

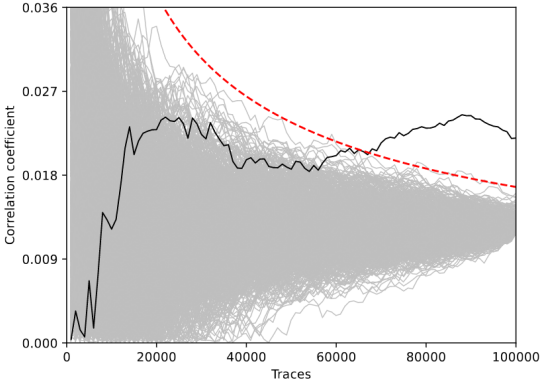


(a) a_1 (b) b_1

(c) a_2 (d) b_2

- 50 000 profiling traces
- 5 000 attack traces
- displayed: probability that a single trace is assigned a class given its known class
- multi-trace attacks: a_1 takes 34 attack traces, all others at most 4

Correlation Power Analysis on Challenge Multiplication



- stronger attack than SPA, weaker attacker model!
- target: multiplication of c (part of signature candidate) and secret key
- power oracle: single-bit
- Hamming weight/distance yielded worse results!
- more in the paper: method to halve the number of hypotheses

Countermeasures

SPA can often be countered effectively by **shuffling**.

Protection against CPA usually requires **masking**.

Countermeasures

SPA can often be countered effectively by **shuffling**.

Protection against CPA usually requires **masking**.

Problem: **storing** the key.

Arithmetic vs Boolean Masking

Boolean Masking

Pro: compared to unmasked, keys are bigger by factor d (masking order)

Contra: polynomial multiplications and NTT are complicated*

Arithmetic Masking

Pro: polynomial multiplications are easy, can be done share-wise with public values

Contra: compared to unmasked, keys are bigger by factor $7d$

Arithmetic vs Boolean Masking

Boolean Masking

Pro: compared to unmasked, keys are bigger by factor d (masking order)

Contra: polynomial multiplications and NTT are complicated*

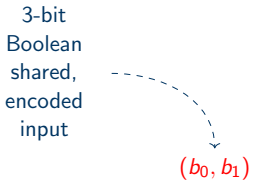
Arithmetic Masking

Pro: polynomial multiplications are easy, can be done share-wise with public values

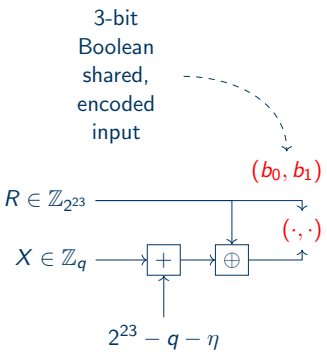
Contra: compared to unmasked, keys are bigger by factor $7d$

*Multiplication with c can be done in Boolean domain completely with Schoolbook multiplication, but is slower and requires additional randomness.

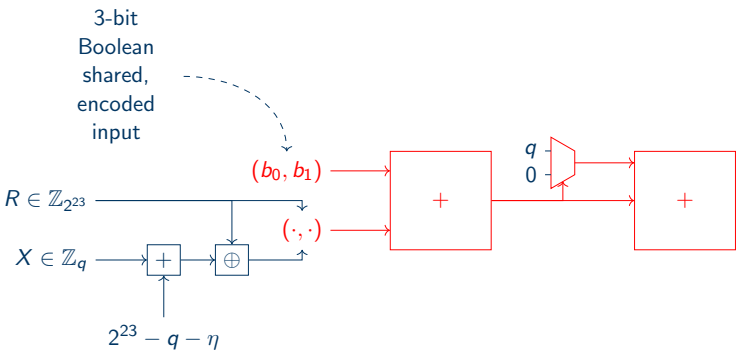
Masking Conversion with Integrated Decoding



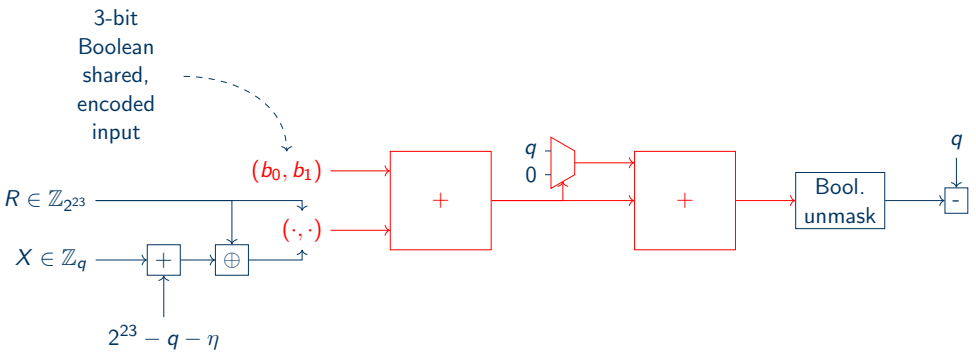
Masking Conversion with Integrated Decoding



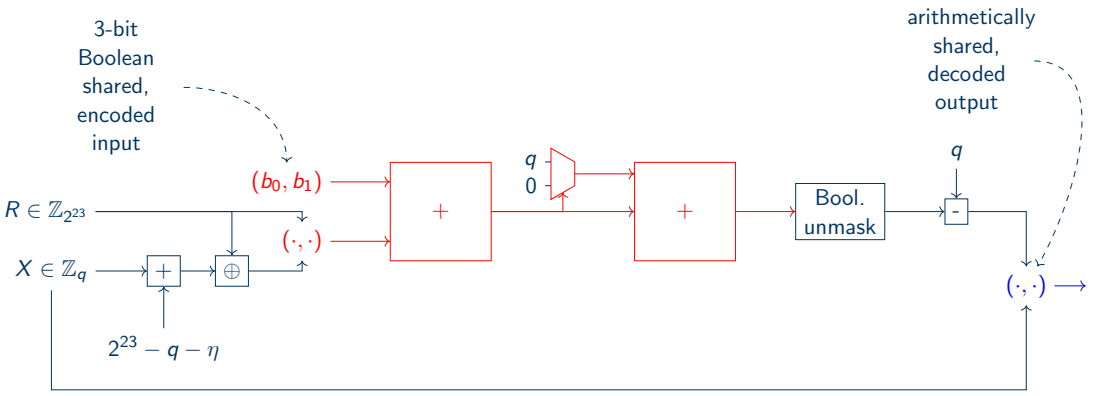
Masking Conversion with Integrated Decoding



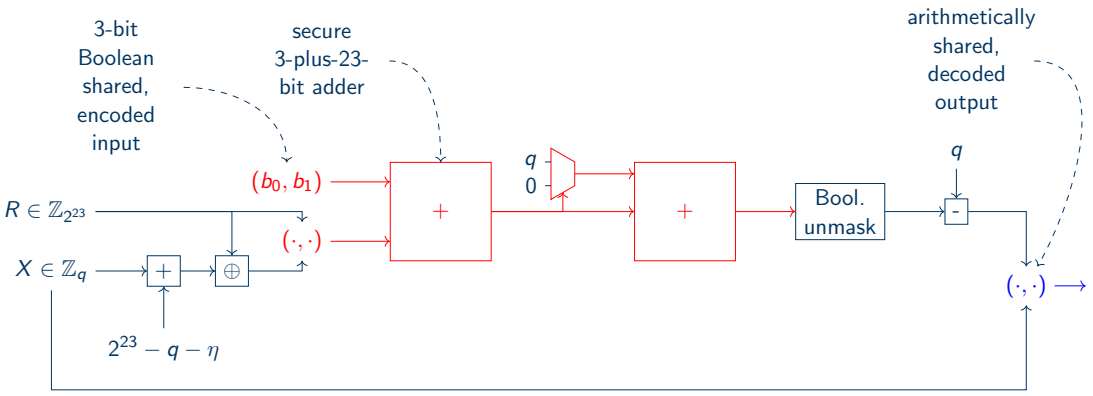
Masking Conversion with Integrated Decoding



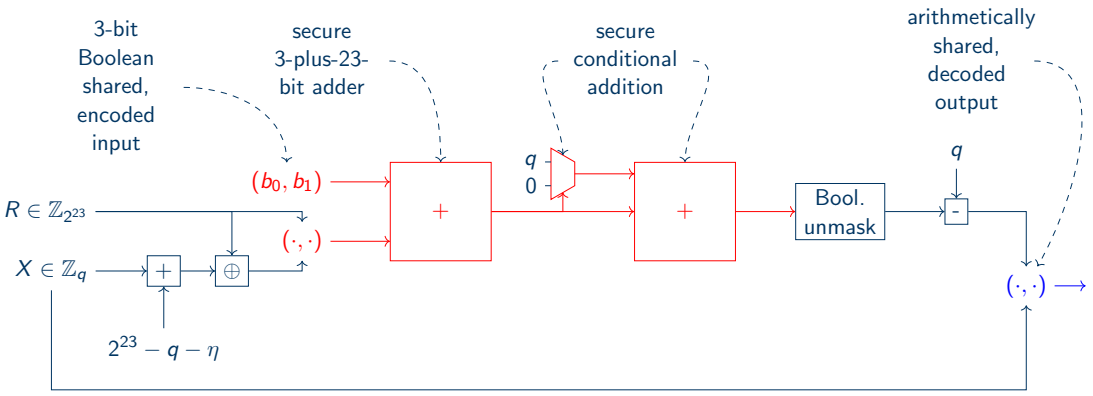
Masking Conversion with Integrated Decoding



Masking Conversion with Integrated Decoding



Masking Conversion with Integrated Decoding



Conclusion

RUB

What did we do?

- first side-channel analysis of Dilithium on hardware
- concrete SPA and CPA attack strategies
- efficient countermeasures

Why is it important?

- Dilithium will be standardized soon.
- There are no other works dedicated to security of hardware implementations of Dilithium!

What is left open?

- fully protected hardware implementation
- potential benefits of using Boolean masking only
- fault attacks and countermeasures in hardware

Link to the paper: `eprint.iacr.org/2022/1410`

Contact: `mail@georg.land`