Side Channel Analysis and Protection for McEliece Implementations

Thomas Eisenbarth

Joint work with Cong Chen, Ingo von Maurich and Rainer Steinwandt

9/27/2016

NATO Workshop- Tel Aviv University

Overview

- Motivation
- QC-MDPC McEliece
- Horizontal and Vertical Side Channel Analysis of McEliece
- Masking a QC-MDPC McEliece implementation

Motivation

Post-Quantum Cryptography?

- Internet Security rests on Public Key Cryptography
  - Digital Signatures (RSA, (EC)-DSA)
  - Key Exchange ((EC)DH)
  - Public Key Encryption (RSA)
- Security Relies on Hardness of Factoring or Discrete Logarithm Problem
- Quantum Computers:
  - Shor's Algorithm solves DL/Factoring in polynomial time
  - Prediction: 10 – 30 years from now

➢ Can You afford to disclose your current secrets in 10 years?

Timeline for PQC Standardization

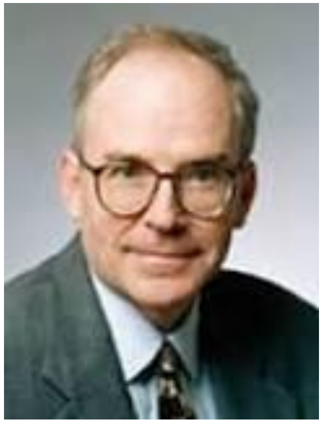
- NSA 2015: Time to switch to
  "Quantum-Secure Cryptography"

- August 2016: NIST Post Quantum Crypto Project
  NIST announces PQC Standardization Process

  Deadline: November 2017

McEliece Cryptosystem

- Code-based Cryptosystem
- PK Encryption
- Proposed by McEliece in 1978
  - Fairly efficient
  - No efficient attacks
  - Large key size



QC-MDPC McEliece

QC-MDPC as Public Key Scheme [1]

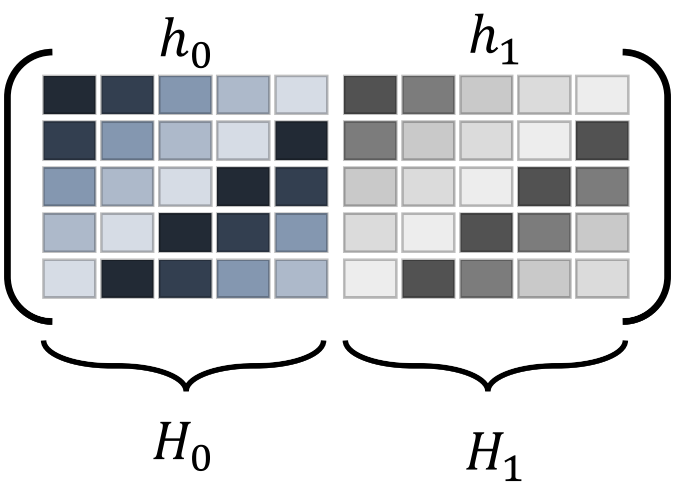
McEliece based on Quasi-Cyclic Moderate Density Parity-Check code

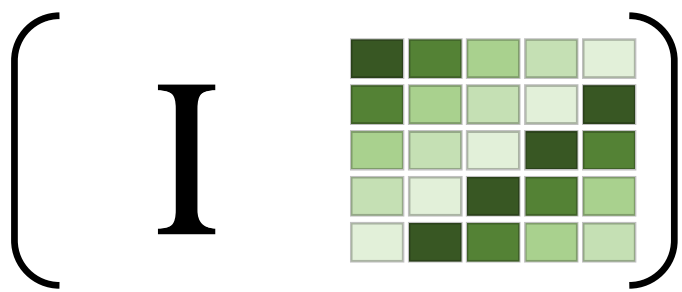
Key Generation:

- Parity Check Matrix
  $H = [H_0 | H_1], H_0, H_1 \in \mathbb{F}_2^{4801 \times 4801}$
  - $wt(h_0) = wt(h_1) = 45$

Public Key

$G = [I | (H_1^{-1} \cdot H_0)^T],\ I \in \mathbb{F}_2^{4801 \times 4801}$





1. Misoczki, R. etc, "MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes," *Information Theory Proceedings (ISIT), 2013*

QC-MDPC McEliece

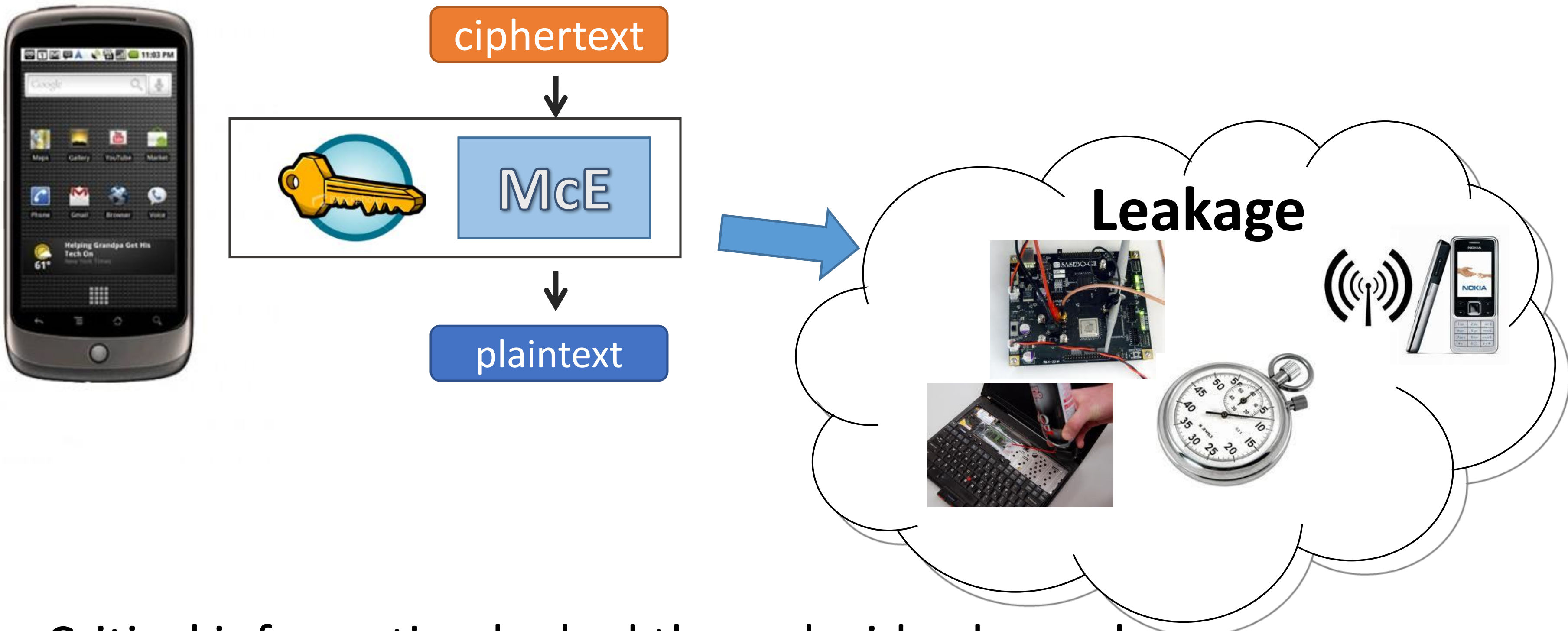
Encryption
Message $m \in \mathbb{F}_2^{4801}$, error vector $e \in_R \mathbb{F}_2^{9602}, wt(e) \leq 84$
$\mathrm{x} \leftarrow mG + e$

Decryption

1. Compute the syndrome $s = Hx^T$
2. Count $\#_{upc}$ for each ciphertext bit
   a) If $\#_{upc}$ exceeds threshold b_i, flip the ciphertext bit
   b) Add current row h_j to the syndrome
3. Repeat 2. until either $s = \mathbf{0}$ or exceeding max. iterations

Side Channel Analysis

Side Channel Attacks



- Critical information leaked through side channels
- Adversary can extract critical secrets (keys etc.)
- Usually require physical access (proximity)

Power Analysis of McEliece [HMP10]

- AVR Software implementation of classic McEliece
  - SPA based approaches on various key parts

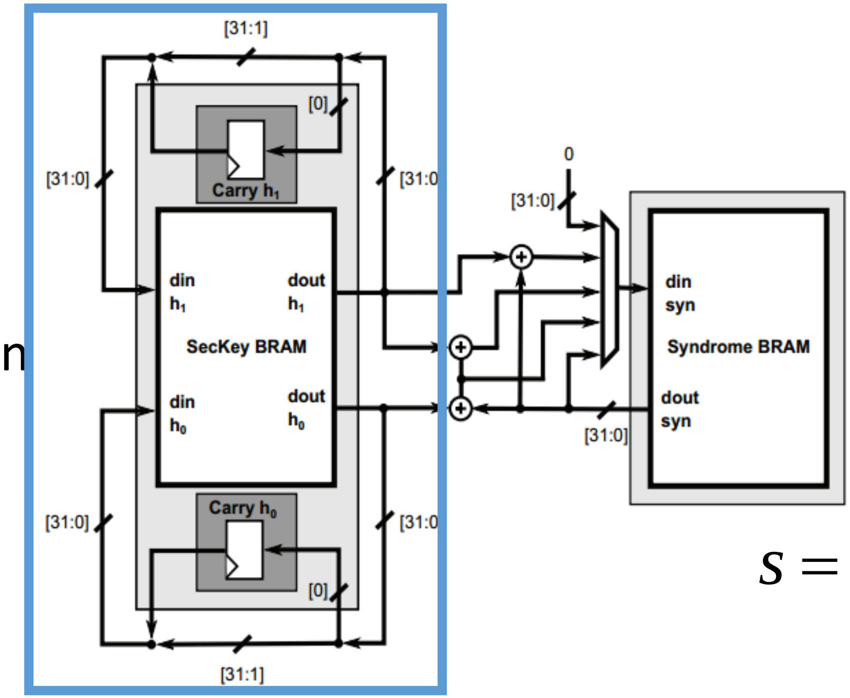
- Finds HW of key information via SPA
  - Final key recovery requires significant guessing

- **DPA not possible**, as key not classically mixed into state.

[3] Heyse, Moradi, Paar: *Practical Power Analysis Attacks on Software Implementations of McEliece* PQCrypto 2010

Efficient FPGA Implementation [MG14]

Key rotation



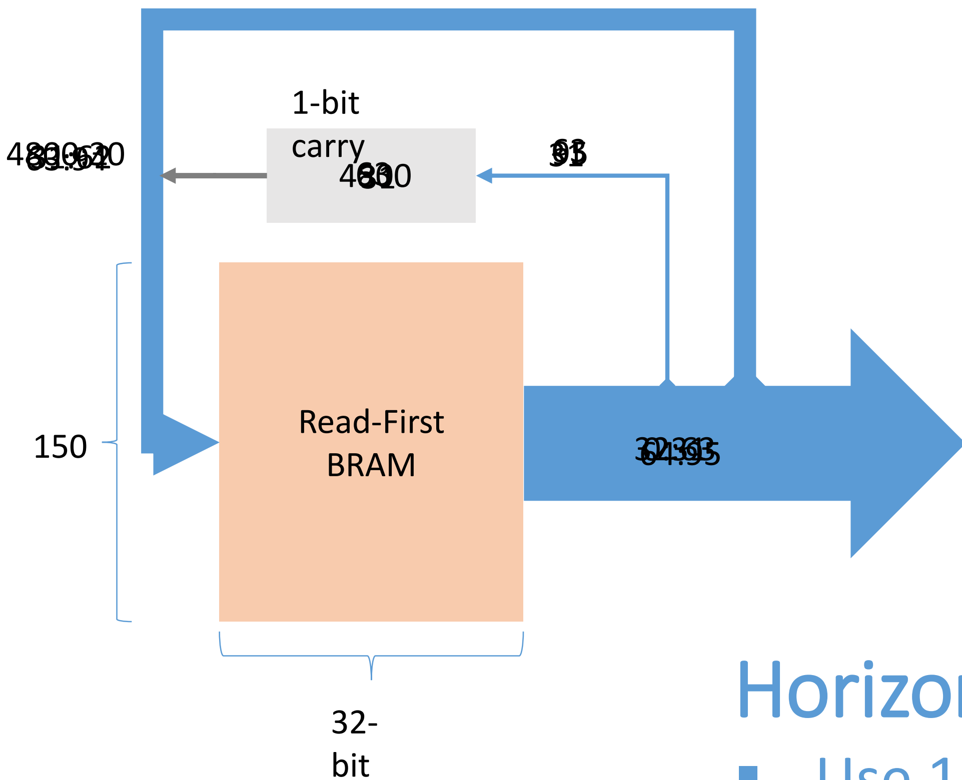
Syndrome computation

$$s = H \cdot x^T$$

$$s <= s + (h_0 >>> i) \cdot x_i + (h_1 >>> i) \cdot x_{4801+i}$$

[MG14] von Maurich, I.; Güneysu, T., "Lightweight code-based cryptography: QC-MDPC McEliece encryption on reconfigurable devices," *DATE 2014*

Key Rotation (KR) of 4801-bit $h_0[0:4800]$



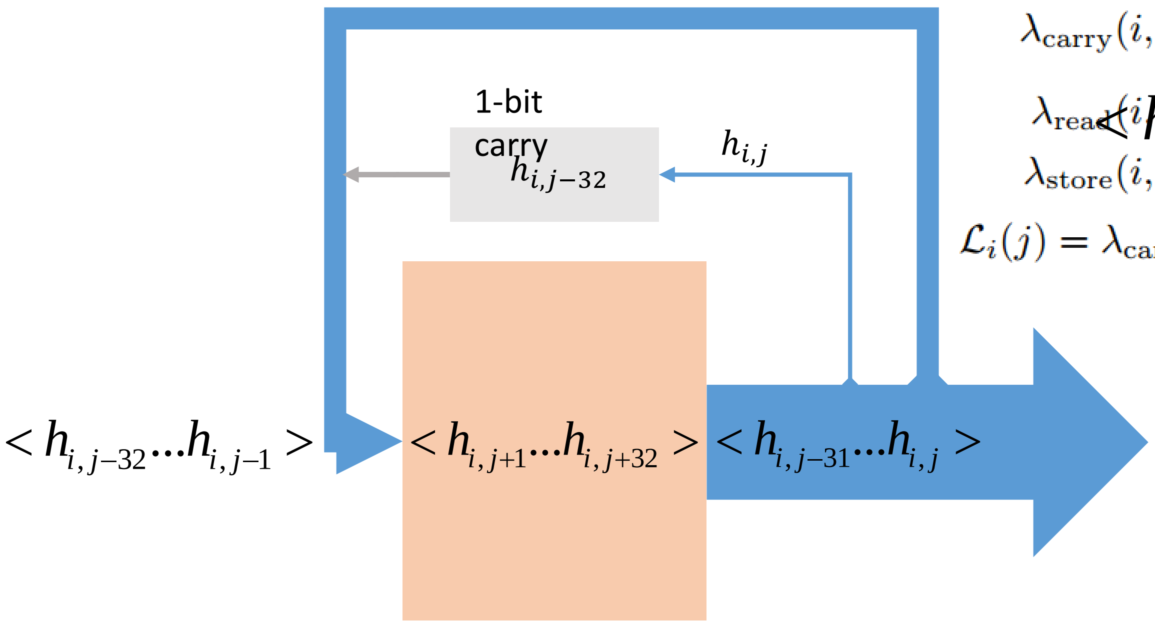
- 4801 = 150×32 + 1 bits
- 150 clock cycles per rotation
- 150 bits overwrite register
- 4801 rotations during KR;
- 4801×150 times overwriting;
- Each bit has 150 chances overwriting the carry reg during one decryption

Horizontal Attack:

- Use 150 leakages from one trace!

Leakage Model

For any key bit $h_{i,j}, i \in \{0,1\}, j \in [0,4800]$ the leakage when it overwrites carry register:



$$\lambda_{\text{carry}}(i,j) = w_1 \cdot \text{wt}(h_{i,j-32} \oplus h_{i,j})$$

$$\lambda_{\text{read}}(i,j) = w_2 \cdot \text{wt}(\langle h_{i,j+1} \ldots h_{i,j+32} \rangle)$$

$< h_{i,j-32} \ldots h_{i,j} \ldots h_{i,j+32} >$

$$\lambda_{\text{store}}(i,j) = w_3 \cdot \text{wt}(\langle h_{i,j-32} \ldots h_{i,j-1} \rangle),$$

$$\mathcal{L}_i(j) = \lambda_{\text{carry}}(i,j) + \lambda_{\text{read}}(i,j) + \lambda_{\text{store}}(i,j) + \mathcal{N}$$

Leakage Exploitation

Experiment Setup

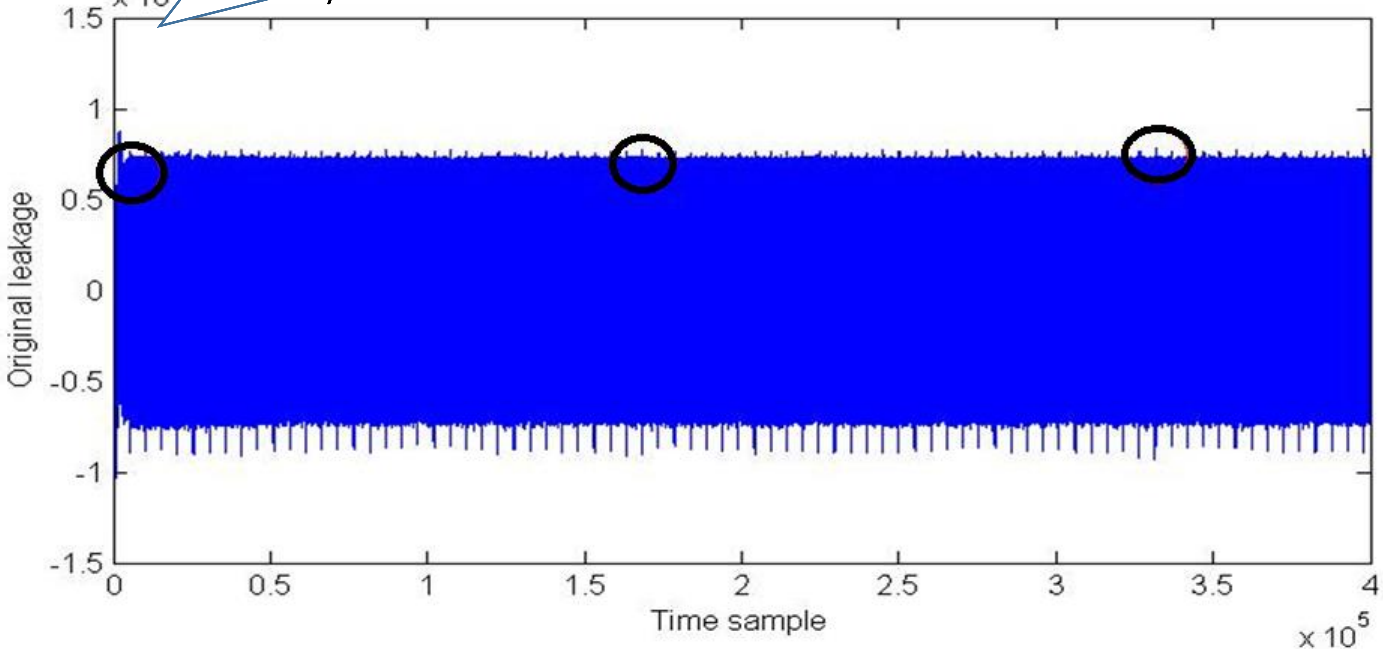
- SASEBO-GII SCA evaluation board

-- Clocked at 3MHz

- Tektronix DPO 5104 oscilloscope

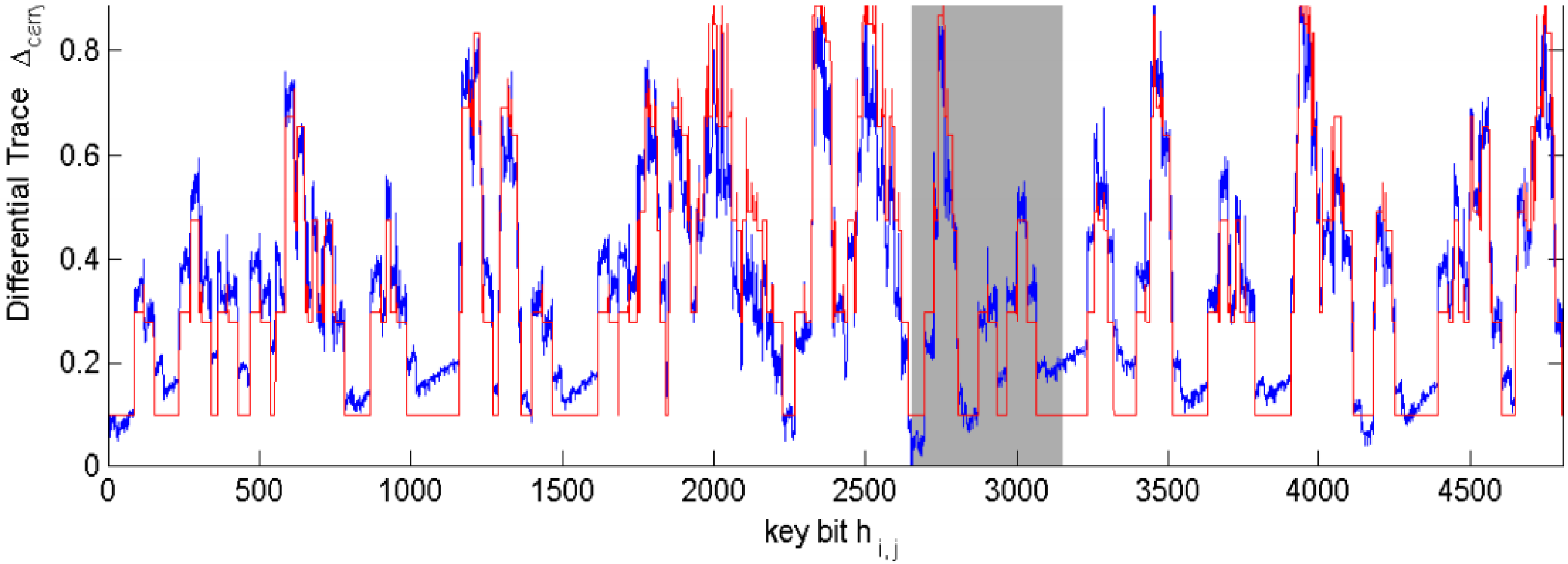
-- Sampling rate: 100MS/s

bit 31 overwrites carry



Differential Trace

$$
\begin{aligned}
\Delta_{\text{carry}}(j) &= \frac{1}{150}\sum_{l=1}^{150}\left(\mathcal{L}_0(j,l)+\mathcal{L}_1(j,l)\right)\\
&= \text{avg}\left(\lambda_{\text{carry}}(0,j)+\lambda_{\text{read}}(0,j)+\lambda_{\text{store}}(0,j)+\lambda_{\text{carry}}(1,j)+\lambda_{\text{read}}(1,j)+\lambda_{\text{store}}(1,j)\right)
\end{aligned}
$$



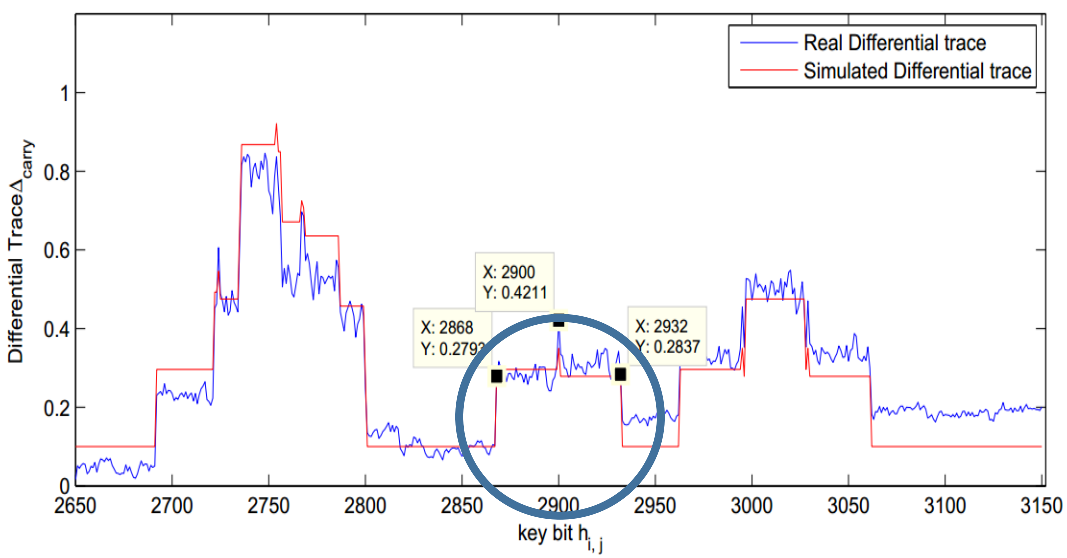
Key Bits Recovery

Shape Definition

- Find a clear characteristic shape caused by a set bit in the differential trace
- Define threshold based on this shape

Shape Detection

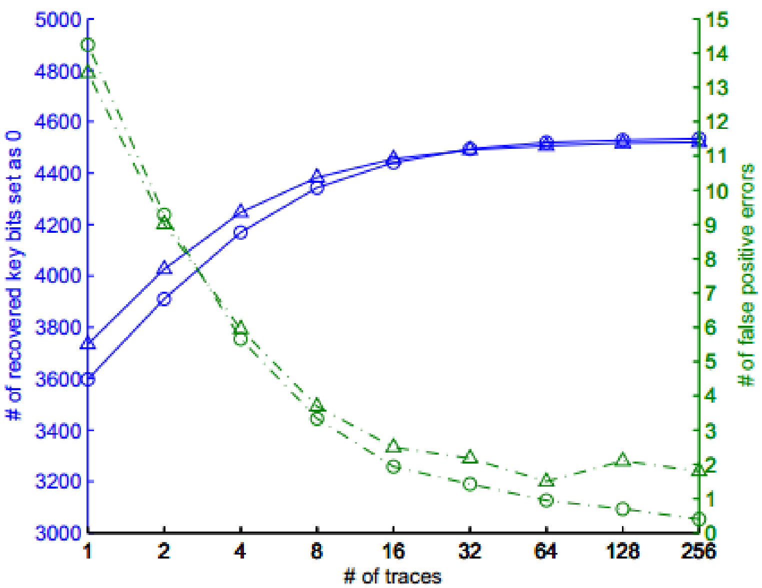
- Browse the differential trace to find more characteristic shapes
- Recover bit 0 and bit 1



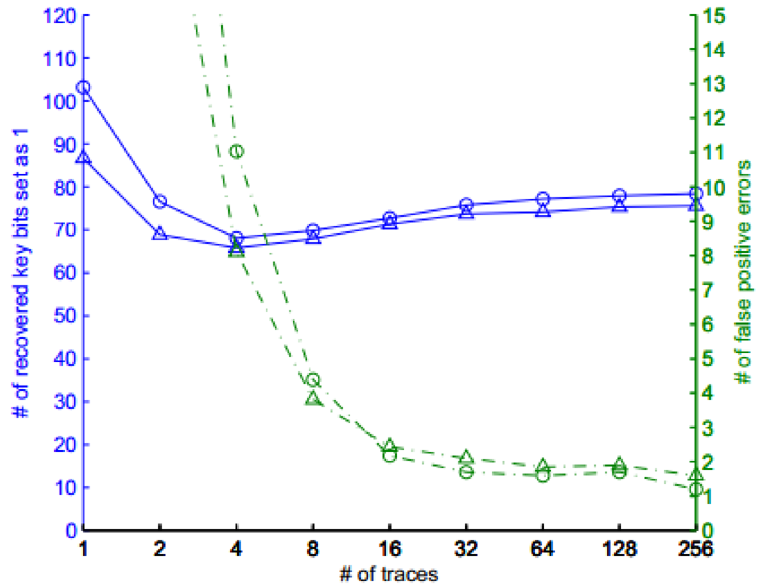
$$< h_{i,j-32} \ldots\ldots h_{i,j} \ldots\ldots h_{i,j+32} >$$

0	...0...	0	0	0	...0...	0

DPA results for $(h_1 + h_0)$



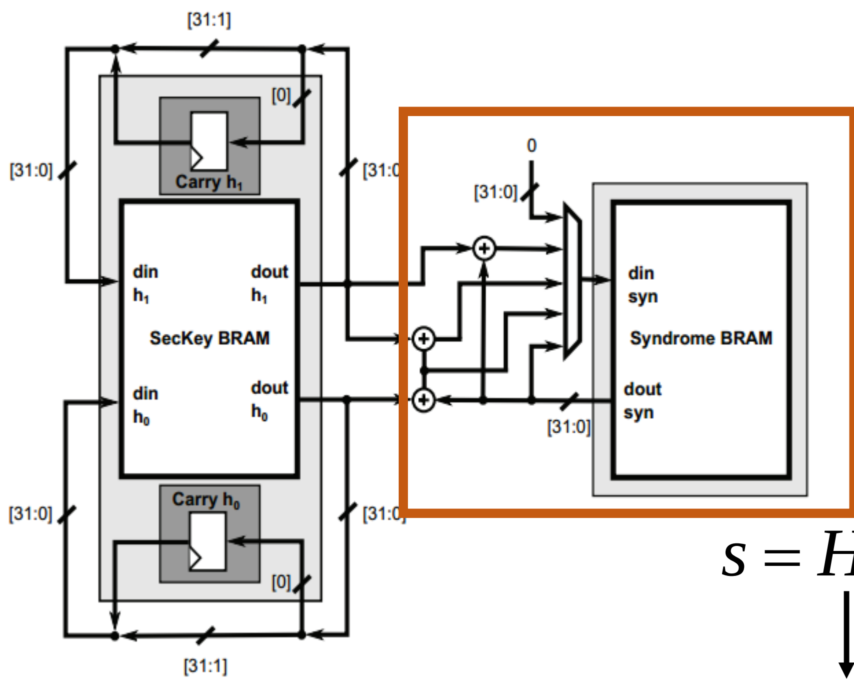
Recovered key bits of 0 vs. false positives



Recovered key bits of 1 vs. false positives

Vertical Attack on Syndrome Computation

Key rotation



Syndrome computation

$$s = H \cdot x^T$$

$$s <= s + (h_0 >>> i) \cdot x_i + (h_1 >>> i) \cdot x_{4801+i}$$

Idea: set single bit in x_i and see h_0 written in s

→4801 different leakages for h_0

Vertical Attack on Syndrome Leakage

- Leakage model: Hamming weight of H written to empty syndrome:

$$\lambda_{j,\text{syn}} = w_0 \cdot \text{wt}\left(\langle h_{i,j-l} \ldots h_{i,j} \ldots h_{i,j-l+31} \rangle\right)$$

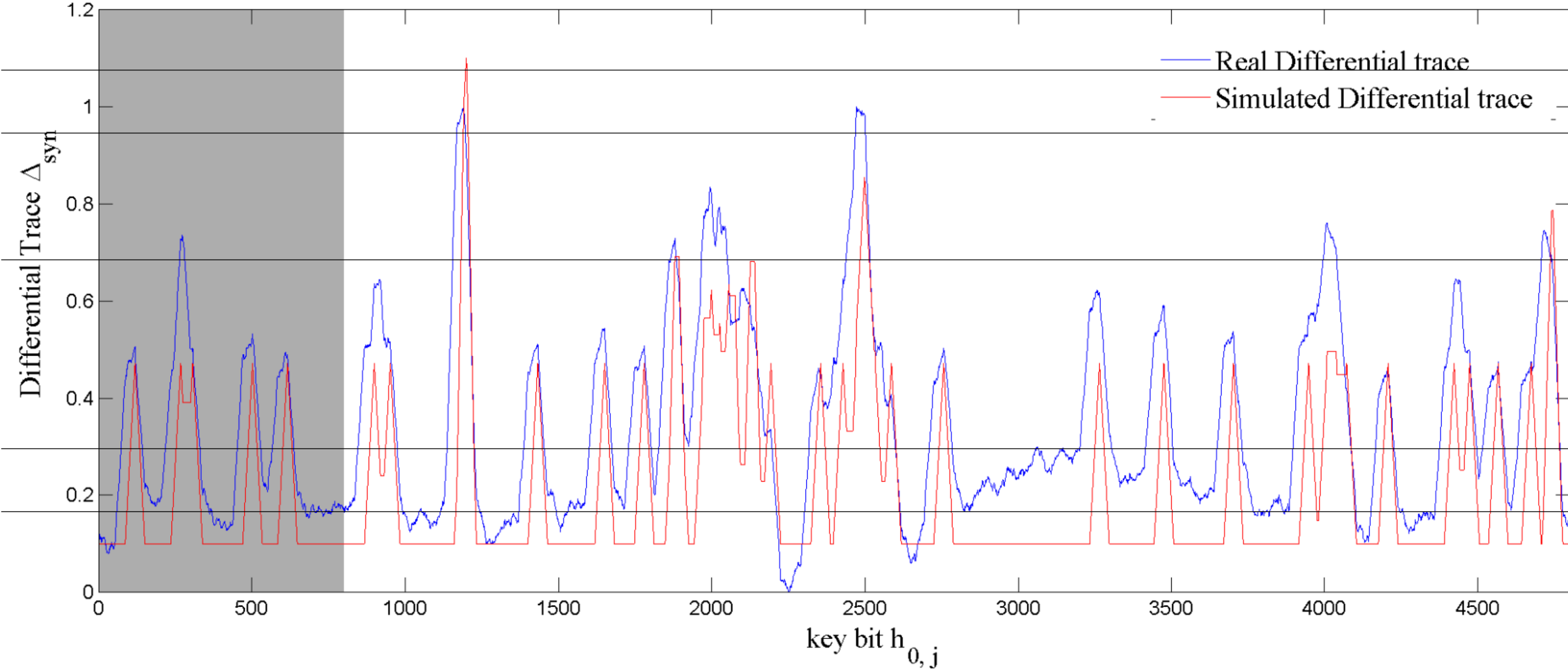
- Differential Trace:

$$\Delta_{\text{syn}}(j) = \sum_{l=0}^{4800} \left(\mathcal{L}_{j,\text{syn}}(l) - \mathcal{L}_{j,\text{const}}(l)\right)$$

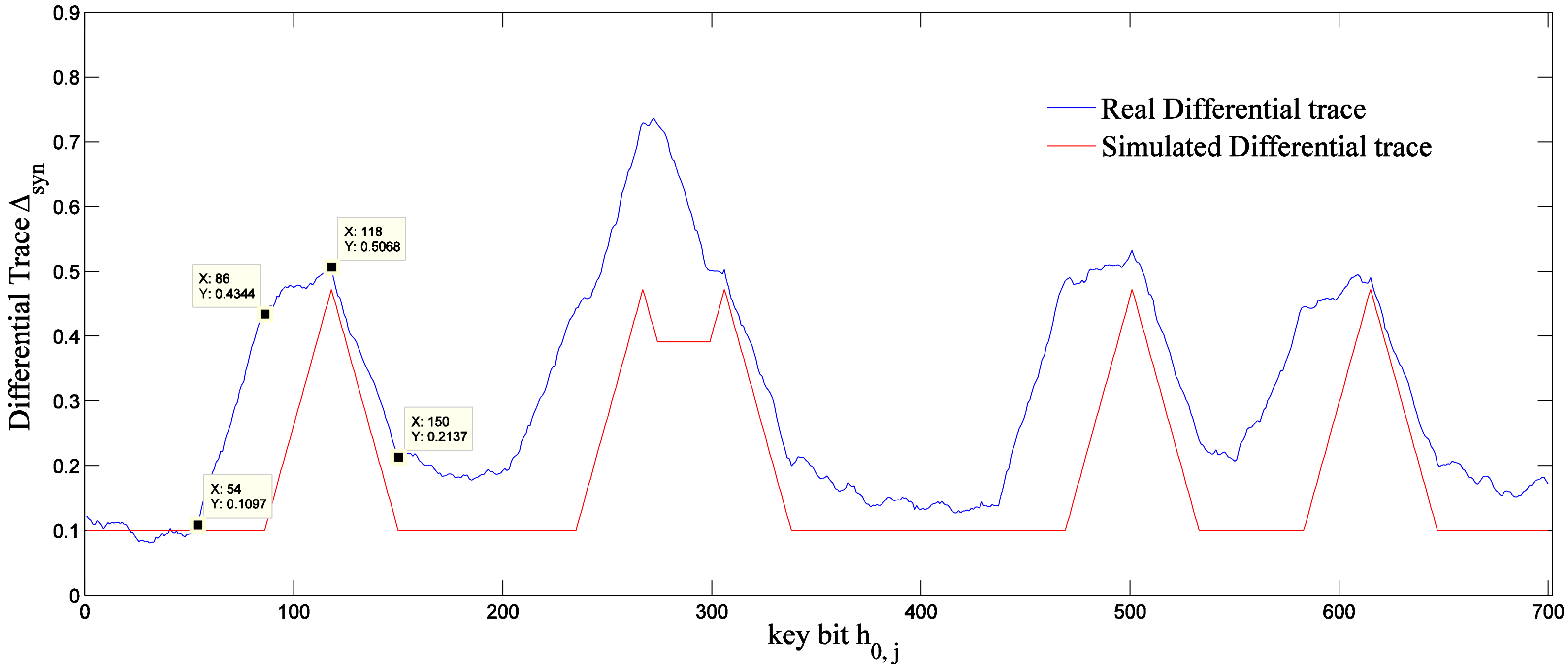
  - Subtract base behavior (leakage w/o syndrome update)

➢ Sparse 1's leave clear mark in trace

Vertical Attack: Leakage



Vertical Attack: Leakage



- Each 1 leaks in 32 neighboring bits
- Low HW key makes attack feasible

Full Key Recovery

Relationship between h_0 and h_1

Known public key: $Q = (H_1^{-1} \cdot H_0)^T$

$$h_0 = h_1 \cdot Q^T$$

$$h_0 \oplus h_1 = h_1 \cdot Q^T \oplus h_1 = h_1 \cdot (Q^T \oplus I_{4801})$$

DPA recovers :

$h_0 + h_1$	[0 0 1 0 * 0 0 1 0 0 …… * 0 0 1 0 0 1]
$h_0 \oplus h_1$	[0 0 * 0 * 0 0 * 0 0 …… * 0 0 * 0 0 *]
h_1	[0 0 * 0 * 0 0 * 0 0 …… * 0 0 * 0 0 *]

Solving the equation

$$h_0 \oplus h_1 = h_1 \cdot (Q^T \oplus I_{4801})$$

$$\begin{bmatrix} 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} * & \dots & * \end{bmatrix} \times \begin{bmatrix} 1 & \dots\dots & 0 \\ \dots\dots & \dots\dots & \dots\dots \\ 1 & \dots\dots & 0 \end{bmatrix}$$

N > 4801-N

=> N > 2400

DPA recovers 4400 0s with some errors

Select N=2401 from 4400 without error.

The probability is between $\dfrac{\binom{4395}{2401}}{\binom{4400}{2401}} \approx 0.02$ $\dfrac{\binom{4398}{2401}}{\binom{4400}{2401}} \approx 0.21$

Summary

- Post-Quantum Cryptography does **not** solve implementation issues of cryptography
- QC-MDPC code reduces the key size but makes DPA feasible
    - Vertical attack more generic
    - Horizontal attack more efficient
- Full key recovery using secret key's algebraic property

[CEMS15] Chen, Eisenbarth, von Maurich, Steinwandt*: Differential Power Analysis of a McEliece Cryptosystem* ACNS 2015
[CEMS16] Chen, Eisenbarth, von Maurich, Steinwandt*: Horizontal and Vertical Side Channel Analysis of a McEliece Cryptosystem* IEEE TIFS 2016

Masking McEliece

Masked syndrome computation

parity-check matrix H has quasi-cyclic structure

⇩

use uniformly random masks $m_0, \ldots, m_{n_0-1}$ to mask $h_0, \ldots, h_{n_0-1}$

⇩

quasi-cyclic shifting yields mask matrix M

⇩

split H into two shares $H = H_m \oplus M$

⇩

masked syndrome $s_m = H_m x^T$ and syndrome mask $m_s = Mx^T$
can be computed independently

⇩

one mask suffices!

Masked error-correction decoder

Algorithm 1 Masked Error Correction Decoder

Input: H_m, M_1, M_2, s_m, m_{s_1}, m_{s_2}, x, $B = b_0, ..., b_{max-1}$, max
Output: Error free codeword x or DecodingFailure

for $i = 0$ to max-1 **do**
  for every ciphertext bit x_j **do**
    $\#_{\mathrm{upc}} = \mathsf{SecHW}(\mathsf{SecAND}(s_m, m_{s_1}, m_{s_2}, H_{m,j}, M_{1,j}, M_{2,j}))$
    $d = (\#_{\mathrm{upc}} > b_i)$, $d \in \{0, 1\}$
    $x = x \oplus (d \cdot 1_j)$ ▷ Flip the jth bit of x
    $s_m = s_m \oplus (d \cdot H_{m,j} \oplus \bar{d} \cdot M_{2,j})$ ▷ Update syndrome
    $m_{s_1} = m_{s_1} \oplus M_{1,j}$ ▷ Update masks
    $m_{s_2} = m_{s_2} \oplus M_{2,j} \oplus (\bar{d} \cdot M_{1,j})$
  end for
  if $\mathsf{SecHW}(s_m, m_{s_1}, m_{s_2}) == 0$ **then** ▷ Check for remaining errors
    return x
  end if ▷ For constant run time, this if-statement can be moved after the for-loop
end for
return DecodingFailure

SecAND: bitwise AND of syndrome and row of H

- Adopt Threshold Implementation (TI) for bit-wise AND of H and s [NRR06]
  $\rightarrow$ requires three shares
  - expand syndrome representation $s_m \oplus m_{s_1} \oplus m_{s_2}$
  - expand key representation $H_{m,j} \oplus M_{1,j} \oplus M_{2,j}$

  $\rightarrow$ Additional random vectors r_1, r_2
  - to ensure uniformity

- Shares of the result (bitwise AND):
  - $(s_m \wedge H_{m,j}) \oplus (s_m \wedge M_{1,j}) \oplus (H_{m,j} \wedge m_{s_1}) \oplus r_1$
  - $(m_{s_1} \wedge M_{1,j}) \oplus (m_{s_1} \wedge M_{2,j}) \oplus (M_{1,j} \wedge m_{s_2}) \oplus r_2$
  - $(m_{s_2} \wedge M_{2,j}) \oplus (m_{s_2} \wedge H_{m,j}) \oplus (M_{2,j} \wedge s_m) \oplus r_1 \oplus r_2$

[NRR06] Nikova, Rechberger, Rjimen *Threshold Implementations against sidechannel attacks and glitches* ISC 2006

SecHW: Secure Hamming weight computation

- Unprotected implementation: obtain Hamming weight wt as accumulation of look-ups with pre-computed table
- Here: secure conversion from Boolean to arithmetic masking to facilitate secure accumulation [CGV14]
  Independent sums for each bit position

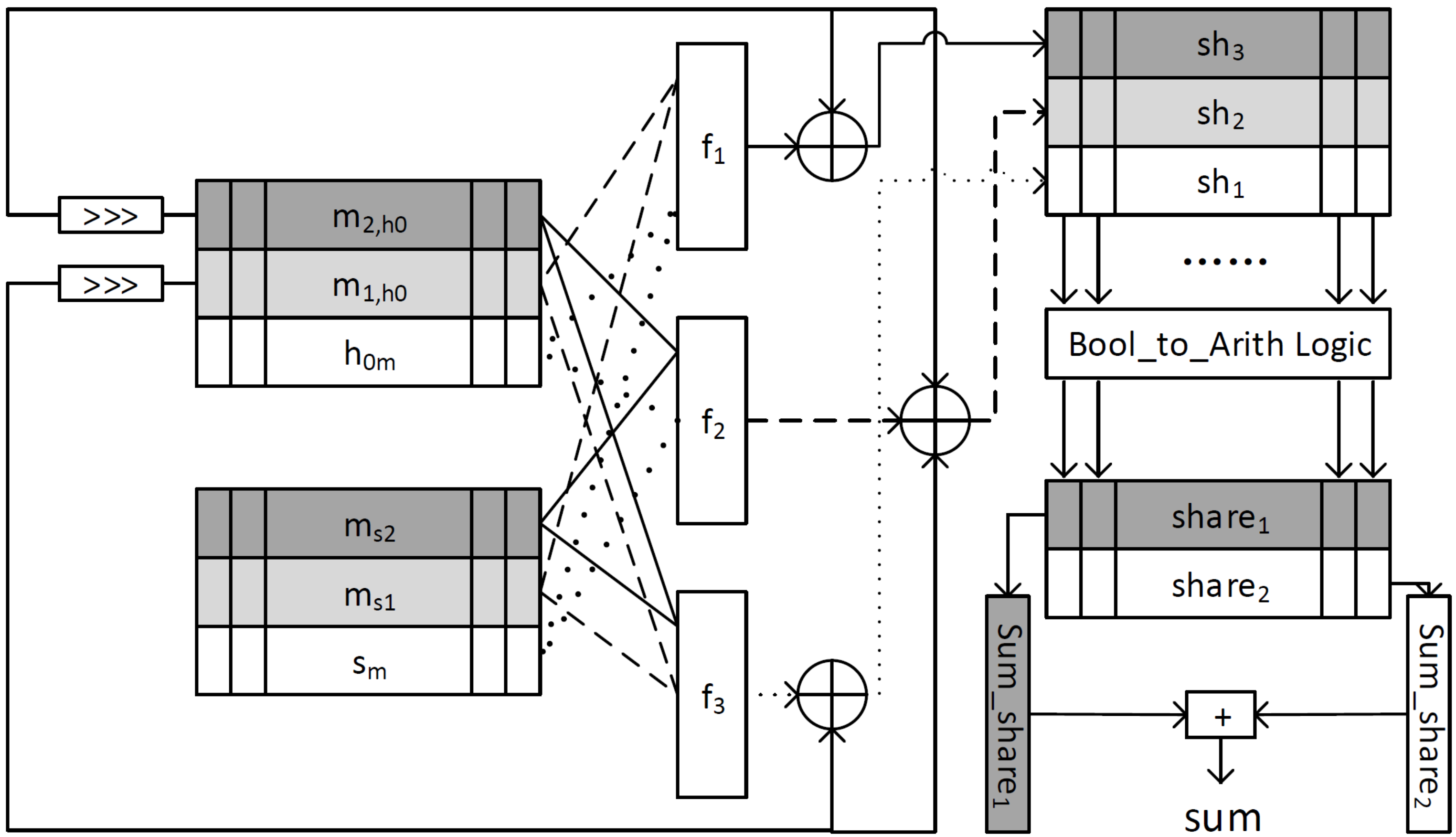
$$\mathrm{wt}(sh) = (sh_{1,\,1} \oplus sh_{2,\,1} \oplus sh_{3,\,1}) + \ldots + (sh_{1,\,|sh|} \oplus sh_{2,\,|sh|} \oplus sh_{3,\,|sh|})$$

$$\Downarrow$$

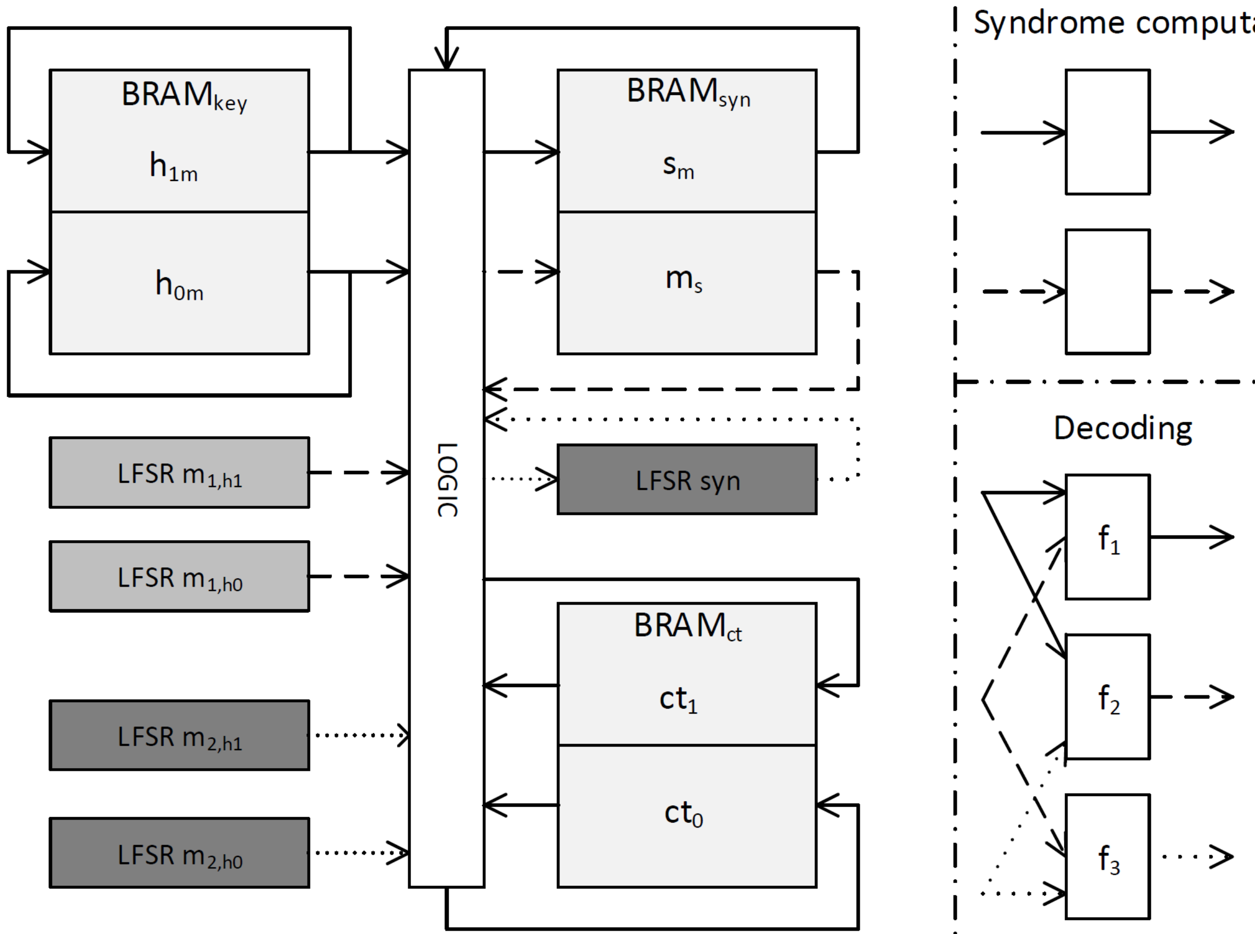
$$\mathrm{wt}(sh) = \quad A_{1,1} + A_{2,1}, \quad + \ldots + \quad A_{1,\,|sh|} + A_{2,\,|sh|}$$
$$= (A_{1,1} + \ldots + A_{1,\,|sh|}) + (A_{2,1} + \ldots + A_{2,\,|sh|})$$

[CGV14] Coron Großschädl Vadnala *Secure Conversion between Boolean and Arithmetic Masking of Any Order* CHES 2014

Overview of Decoder



Overview of masked implementation



Implementation results

VHDL design, synthesized for Virtex-5 XC5VLX50 FPGA,

	FFs	LUTs	Slices	BRAMs	Freq.
Unprotected	412	568	148	3	318
Masked	3045	4672	1549	3	73
Overhead	7.4x	8.2x	10.5x	1x	4.3x

Overhead (4x) not out of line (cf. Moradi et al.'s AES implementation – EC 2011)

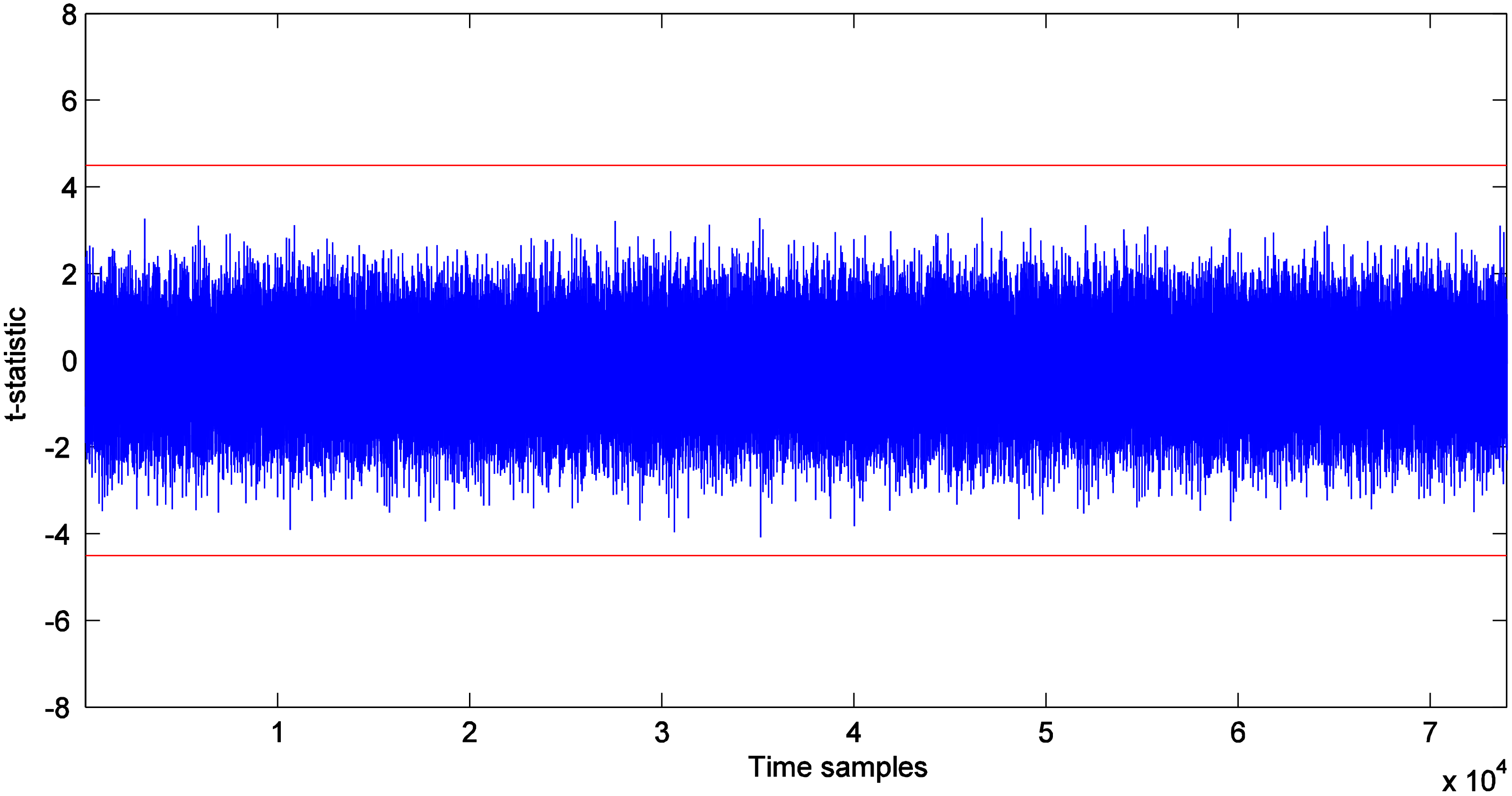
Leakage Analysis

Leakage analysis

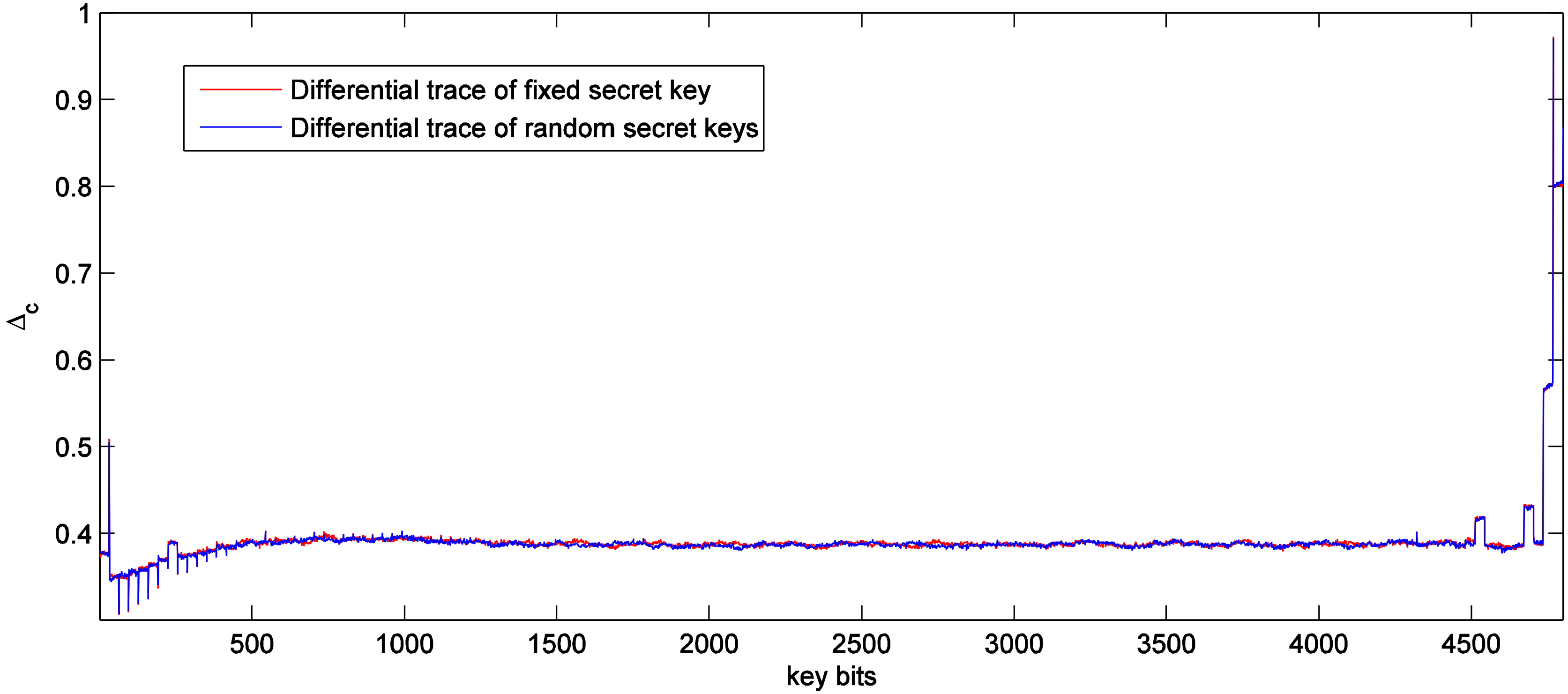
- implementation on Xilinx Virtex-5 XC5VLX50 FPGA of SASEBO-GII board, Tektronix DSO 5104 oscilloscope
- board clocked at 3MHz, sampling rate 100M samples per second, Tektronix DSO 5104 oscilloscope
- T-Test based leakage Detection (TVLA)
- Fixed vs. Random Key!
  - 5,000 repetitions of a fixed key
  - 5,000 random keys.

- T-test validates indistinguishability between key sets
- Attack from ACNS 2015 fails

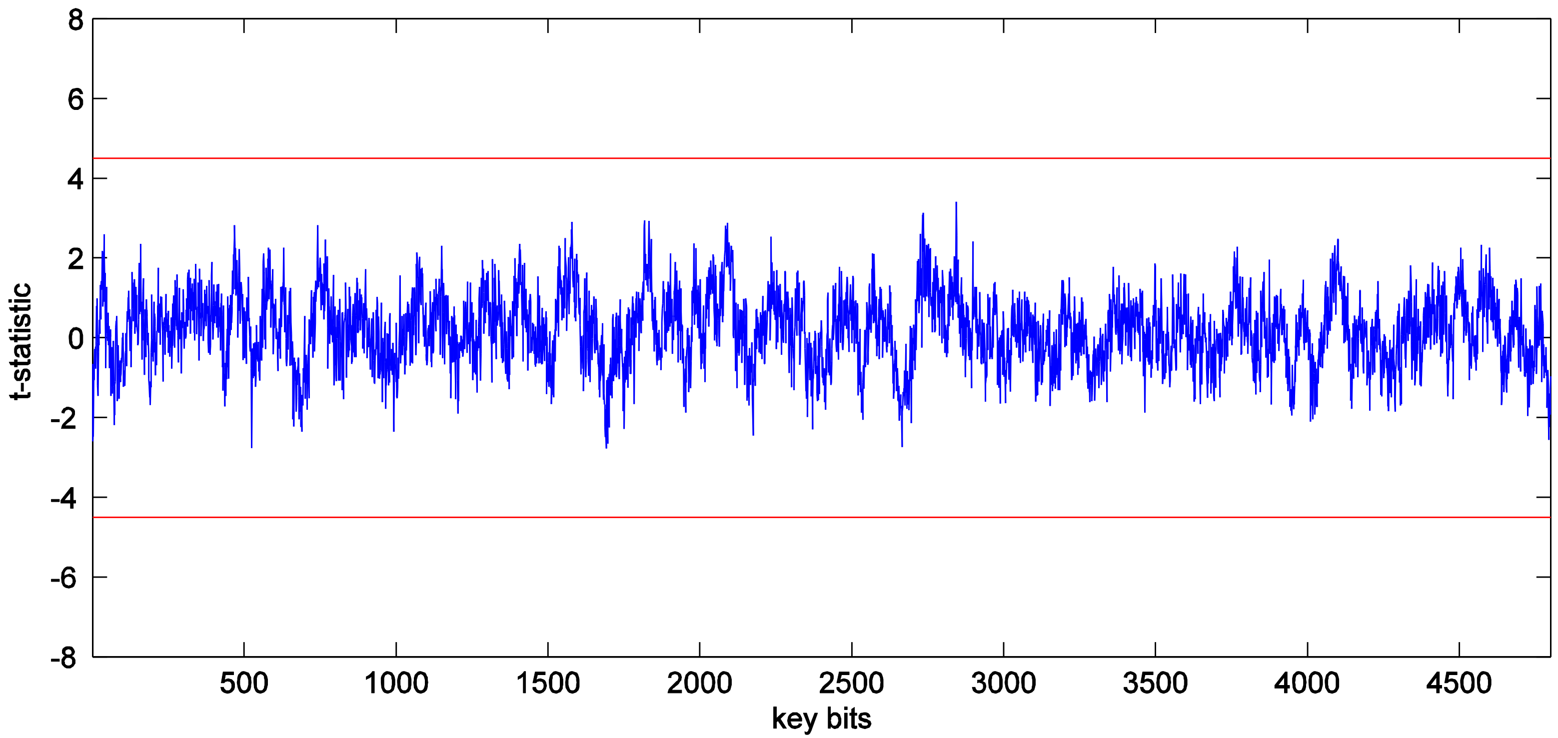
T-test with original traces



Comparison of differential traces



T-test of differential traces



Conclusion

- 1st masked McEliece implementation
- area overhead, incl. on-the-fly mask generation, about 4×
- reduction in clock frequency
- leakage analysis supports effectiveness

- Masking the ciphertext?
- Enforce constant number of iterations for decoder?

[CEMS15] Chen, Eisenbarth, von Maurich, Steinwandt
Masking Large Keys in Hardware: A Masked Implementation of McEliece SAC 2015

Thank you!

More information:

users.wpi.edu/~teisenbarth

teisenbarth@wpi.edu

v.wpi.edu