Protecting against Statistical Ineffective Fault Attacks

Joan Daemen, Christoph Dobraunig, Maria Eichlseder, Hannes Gross, Florian Mendel and Robert Primas

CHES 2020

Motivation

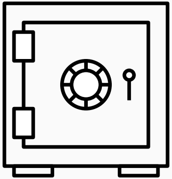
Using crypto in the wild requires:

- Mathematically secure cryptographic schemes

Motivation

Using crypto in the wild requires:

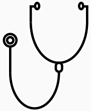
- Mathematically secure cryptographic schemes
- Additional defenses mechanisms against implementation attacks:



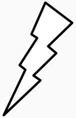
Motivation

Using crypto in the wild requires:

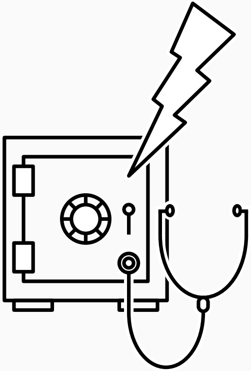
- Mathematically secure cryptographic schemes
- Additional defenses mechanisms against implementation attacks:



Power Analysis



Fault Attacks



Motivation

- Statistical Ineffective Fault Attacks (SIFA) were first presented at CHES2018:
  - Work against block ciphers, AEAD, etc. . .
  - Circumvent redundancy/infection countermeasures
  - Only one fault injection per cipher execution

Motivation

- Statistical Ineffective Fault Attacks (SIFA) were first presented at CHES2018:
  - Work against block ciphers, AEAD, etc...
  - Circumvent redundancy/infection countermeasures
  - Only one fault injection per cipher execution
- In a follow-up at ASIACRYPT2018 it was shown that:
  - SIFA can additionally circumvent (higher-order) masking/TI

Motivation

- Statistical Ineffective Fault Attacks (SIFA) were first presented at CHES2018:
    - Work against block ciphers, AEAD, etc. . .
    - Circumvent redundancy/infection countermeasures
    - Only one fault injection per cipher execution
- In a follow-up at ASIACRYPT2018 it was shown that:
    - SIFA can additionally circumvent (higher-order) masking/TI
- Proposed countermeasures at the time:
    - Error correction
    - Hiding
    - Self destruction

Motivation cont.

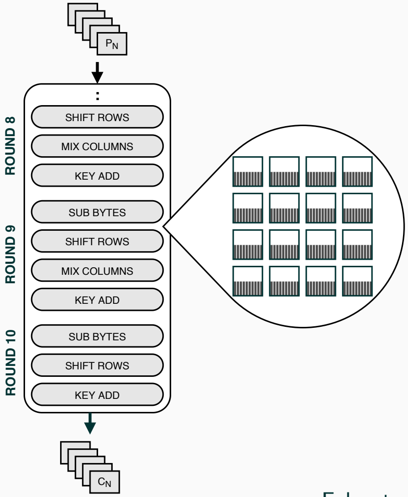
- Many proposed SIFA countermeasures so far utilize error correction:
  - Rather expensive (masking!)
  - How much error correction is necessary?
  - What about DFA?

Motivation cont.

- Many proposed SIFA countermeasures so far utilize error correction:
  - Rather expensive (masking!)
  - How much error correction is necessary?
  - What about DFA?
- We propose efficient SIFA countermeasure strategies:
  - "Careful" combination of redundancy with masking
  - Low overhead for lightweight schemes
  - Moderate overhead for "bulky" schemes like AES

Statistical Fault Attacks on AES-128

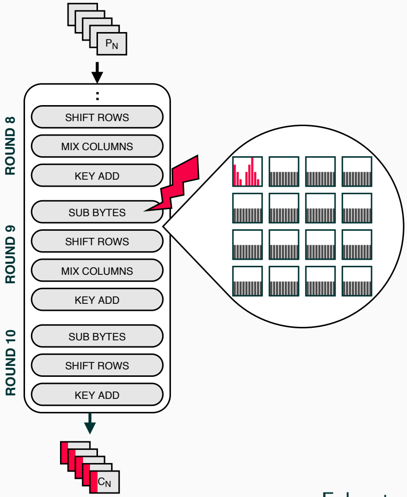
- AES is a PRP:
  - Distribution of ciphertext bytes is uniform
  - (Also after only 9 rounds)



Fuhr et al. [Fuh+13]

Statistical Fault Attacks on AES-128

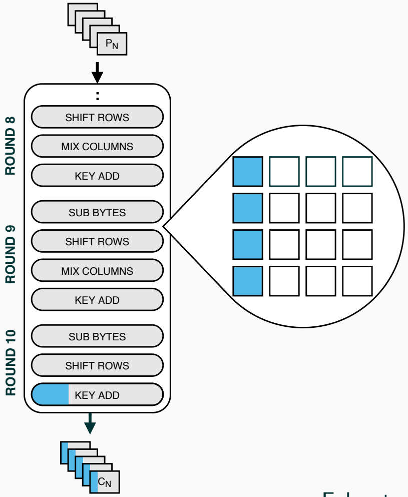
- Assume fault that disturbs distribution of one state byte in round 9
  - Stuck-at, bitflip, random, etc.
  - Attacker does not need to know the caused bias
  - 4 ciphertext bytes are affected



Fuhr et al. [Fuh+13]

Statistical Fault Attacks on AES-128

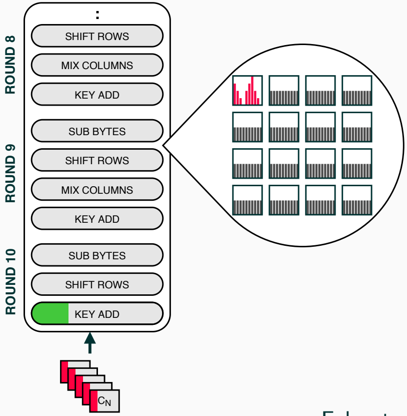
- 4 state bytes in round 9 can be calculated from:
    - 4 ciphertext bytes
    - 4 key bytes



Fuhr et al. [Fuh+13]

Statistical Fault Attacks on AES-128

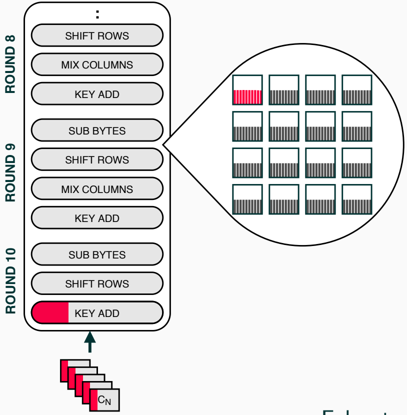
- 4 state bytes in round 9 can be calculated from:
  - 4 ciphertext bytes
  - 4 key bytes (correct)



Fuhr et al. [Fuh+13]

Statistical Fault Attacks on AES-128

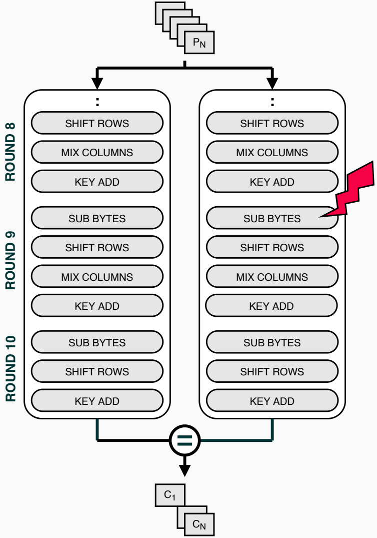
- 4 state bytes in round 9 can be calculated from:
  - 4 ciphertext bytes
  - 4 key bytes (incorrect)



Fuhr et al. [Fuh+13]

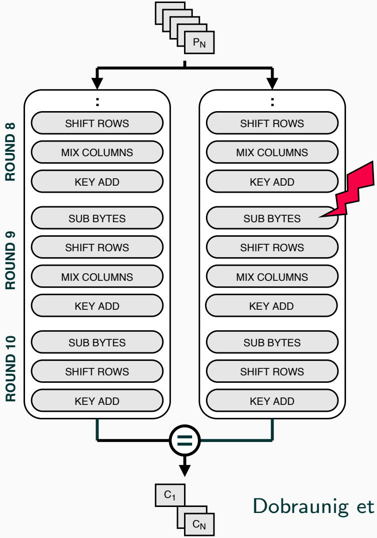
Statistical Ineffective Fault Attacks on AES-128

- Redundant computation fixes the problem!



Statistical Ineffective Fault Attacks on AES-128

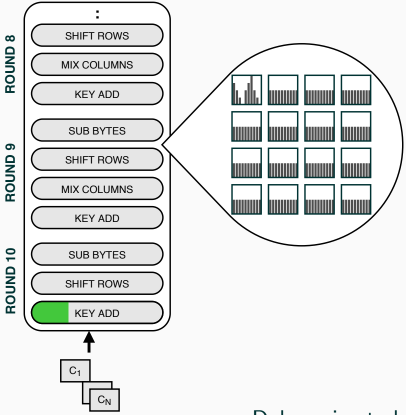
- Redundant computation fixes the problem!
- Except it doesn't



Dobraunig et al. [Dob+18b]

Statistical Ineffective Fault Attacks on AES-128

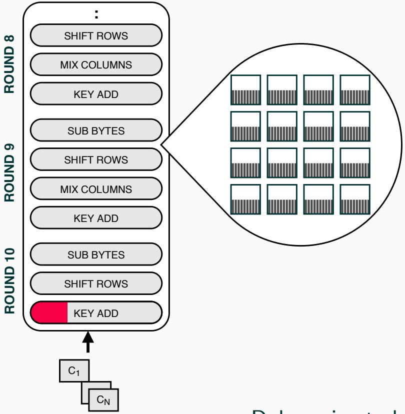
- For simplicity, assume stuck-at zero fault (others work as well)
- "Effective" faults are filtered out
- Correct ciphertexts still show bias in round 9
- Exploitation works same as before



Dobraunig et al. [Dob+18b]

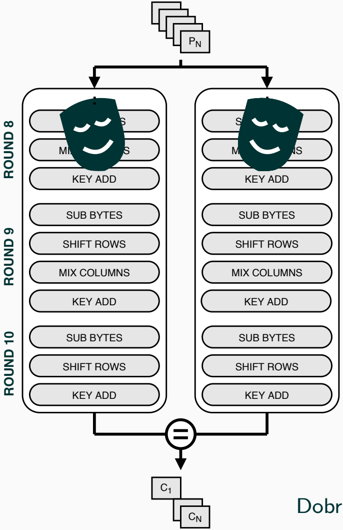
Statistical Ineffective Fault Attacks on AES-128

- For simplicity, assume stuck-at zero fault (others work as well)
- “Effective” faults are filtered out
- Correct ciphertexts still show bias in round 9
- Exploitation works same as before



Dobraunig et al. [Dob+18b]

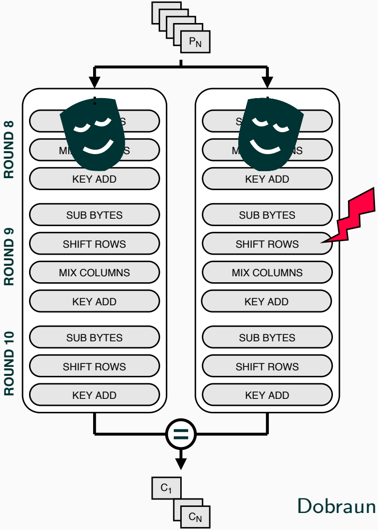
- Masking fixes the problem!



Dobraunig et al. [Dob+18b]

Statistical Ineffective Fault Attacks on AES-128

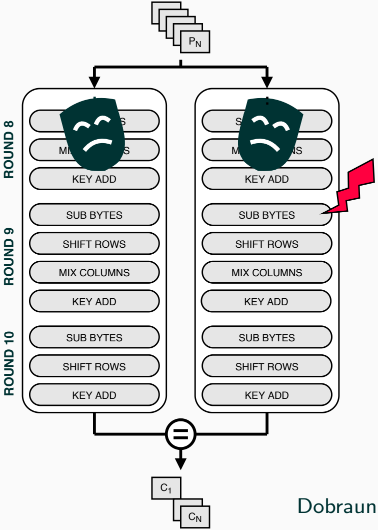
- Masking fixes the problem!



Dobraunig et al. [Dob+18b]

Statistical Ineffective Fault Attacks on AES-128

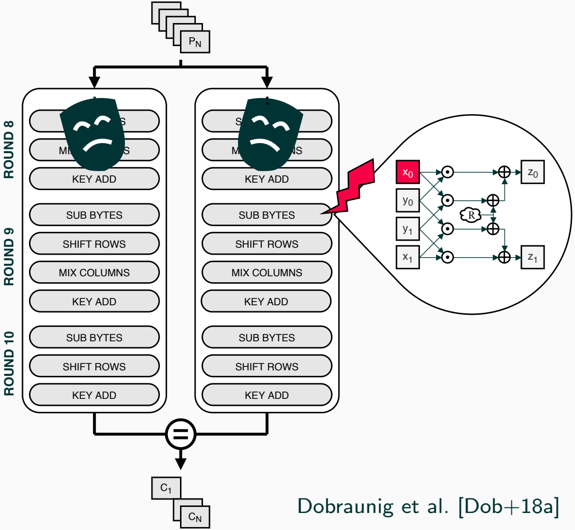
- Masking fixes the problem!
- Except it doesn't



Dobraunig et al. [Dob+18a]

Statistical Ineffective Fault Attacks on AES-128

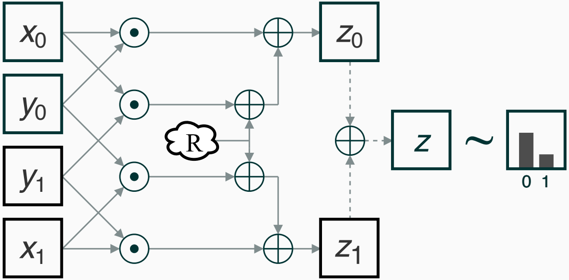
- Masking fixes the problem!
- Except it doesn't



Dobraunig et al. [Dob+18a]

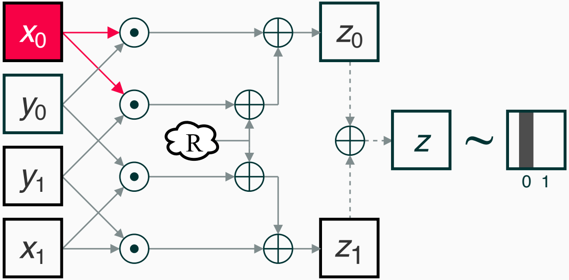
Statistical Ineffective Fault Attacks on AES-128

- Masked AND-gate
- Naturally, when x and y are uniform then z has bias towards 0



Dobraunig et al. [Dob+18a]

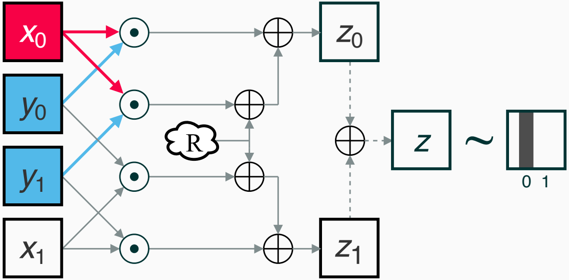
- Assume a fault causes difference in x_0 (to redundant computation)



Dobraunig et al. [Dob+18a]

Statistical Ineffective Fault Attacks on AES-128

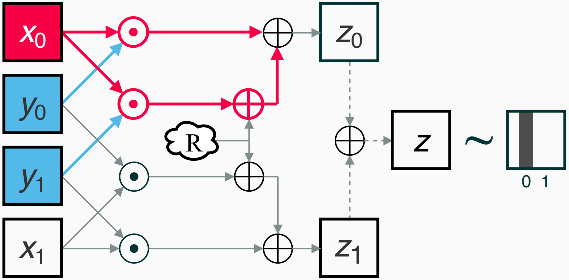
- Assume a fault causes difference in x_0 (to redundant computation)
- Difference cancels if either:
  - y_0, y_1 are both 0



Dobraunig et al. [Dob+18a]

Statistical Ineffective Fault Attacks on AES-128

- Assume a fault causes difference in x_0 (to redundant computation)
- Difference cancels if either:
  - y_0, y_1 are both 0
  - y_0, y_1 are both 1

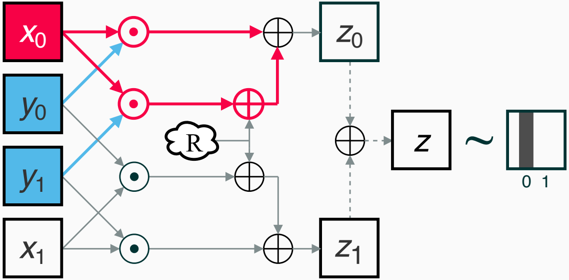


Dobraunig et al. [Dob+18a]

Statistical Ineffective Fault Attacks on AES-128

- Assume a fault causes difference in x_0 (to redundant computation)
- Difference cancels if either:
  - y_0, y_1 are both 0
  - y_0, y_1 are both 1
- Fault is ineffective iff native value y is zero

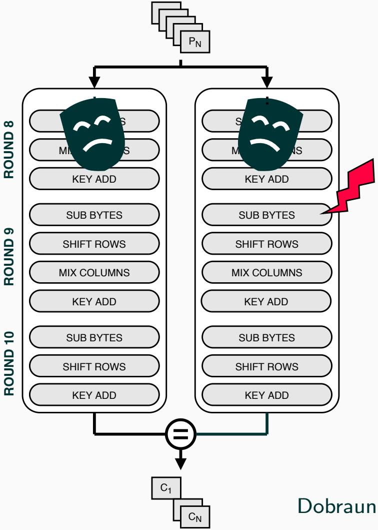
  $\Rightarrow$ "Dangerous fault"



Dobraunig et al. [Dob+18a]

Statistical Ineffective Fault Attacks on AES-128

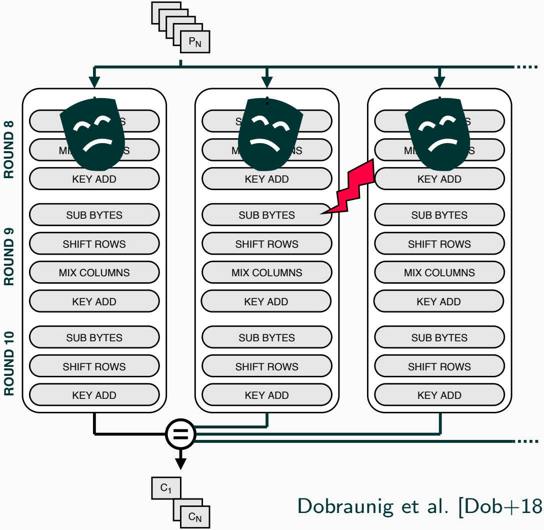
- SIFA can circumvent both masking and redundant computation



Dobraunig et al. [Dob+18a]

Statistical Ineffective Fault Attacks on AES-128

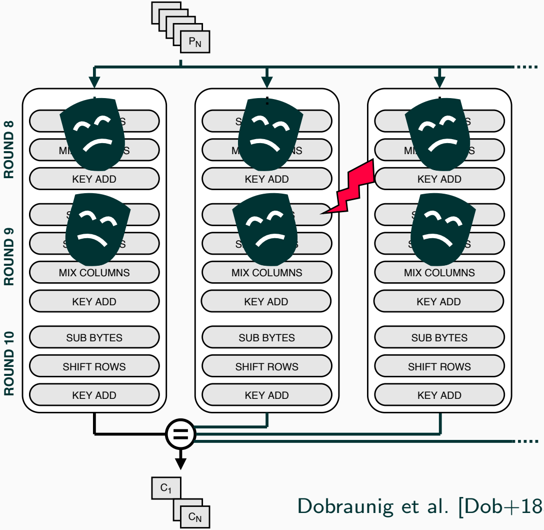
- SIFA can circumvent both masking and redundant computation
- More redundancy doesn't help



Dobraunig et al. [Dob+18a]

Statistical Ineffective Fault Attacks on AES-128

- SIFA can circumvent both masking and redundant computation
- More redundancy doesn't help
- Higher-order masking doesn't help

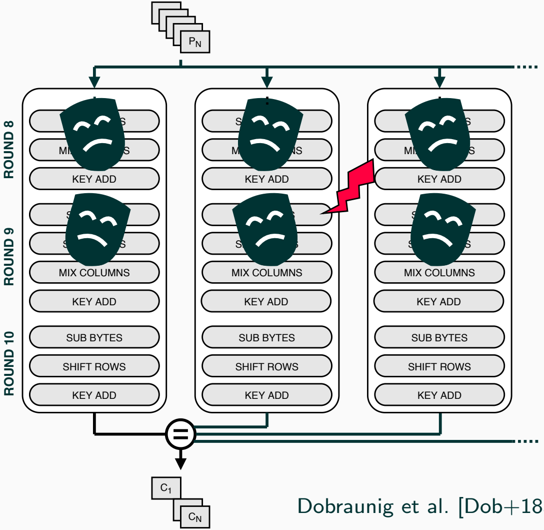


Dobraunig et al. [Dob+18a]

Statistical Ineffective Fault Attacks on AES-128

- SIFA can circumvent both masking and redundant computation
- More redundancy doesn't help
- Higher-order masking doesn't help

⇒ We now show how to counteract SIFA using masking + redundancy . . .



Dobraunig et al. [Dob+18a]

Notations

- We express a cipher as a circuit:
  - Input: Array of variables
  - Output: Array of variables

Notations

- We express a cipher as a circuit:
  - Input: Array of variables
  - Output: Array of variables
- Circuits can be split into sub-circuits:
  - Input: Cipher's input or other sub-circuit's output

Notations

- We express a cipher as a circuit:
  - Input: Array of variables
  - Output: Array of variables
- Circuits can be split into sub-circuits:
  - Input: Cipher's input or other sub-circuit's output
- Splitting is done recursively until we have basic circuits:
  - Only consist of simple operations such as addition/multiplication

The High-level Strategy

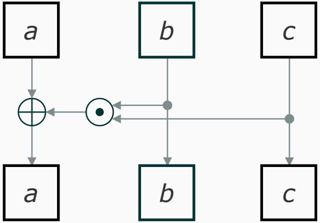
- Build cipher circuit such that "dangerous" faults can always be detected . . .
  - at the S-box output
  - at the cipher output

The High-level Strategy

- Build cipher circuit such that "dangerous" faults can always be detected . . .
  - at the S-box output
  - at the cipher output
- Build masked + redundant cipher circuit where each basic circuit . . .
  - only operates on incomplete set of shares
  - is a permutation (optional)

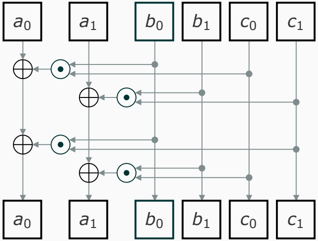
The High-level Strategy

- Build cipher circuit such that "dangerous" faults can always be detected ...
  - at the S-box output
  - at the cipher output
- Build masked + redundant cipher circuit where each basic circuit ...
  - only operates on incomplete set of shares
  - is a permutation (optional)
- Permutation can either be:
  - A linear function
  - A variant of the Toffoli gate (simplest invertible non-linear function)

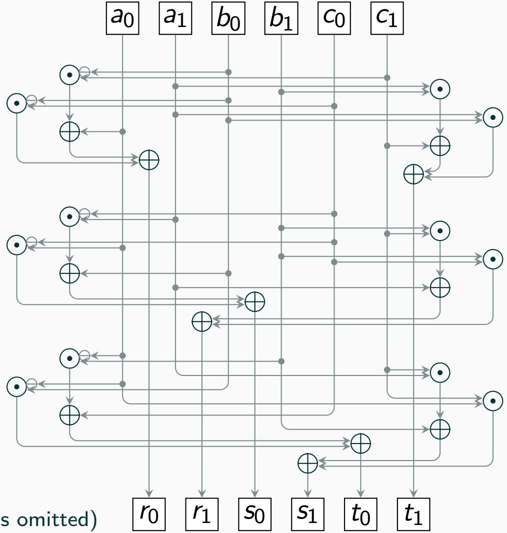


The High-level Strategy

- Build cipher circuit such that "dangerous" faults can always be detected . . .
  - at the S-box output
  - at the cipher output
- Build masked + redundant cipher circuit where each basic circuit . . .
  - only operates on incomplete set of shares
  - is a permutation (optional)
- Permutation can either be:
  - A linear function
  - A variant of the Toffoli gate (simplest invertible non-linear function)



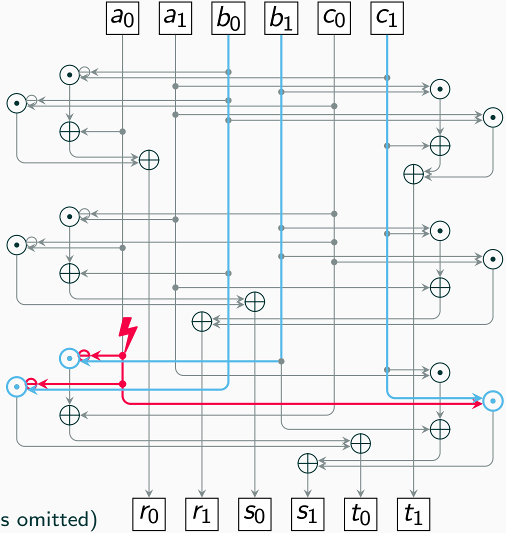
Case study: SIFA on Masked Chi-3 S-box



(Refreshing of shares omitted)

Case study: SIFA on Masked Chi-3 S-box

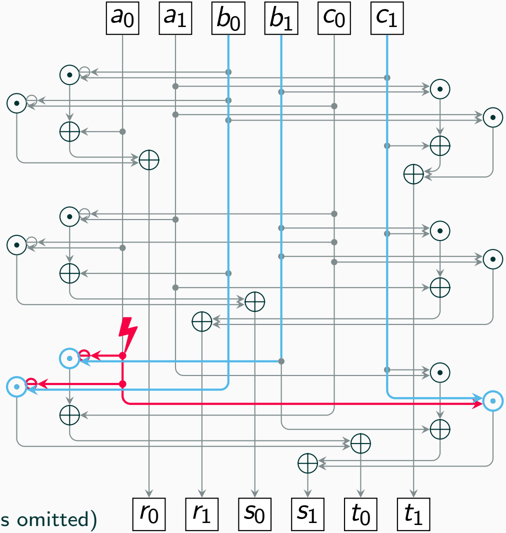
- Same problem as before. . .



(Refreshing of shares omitted)

Case study: SIFA on Masked Chi-3 S-box

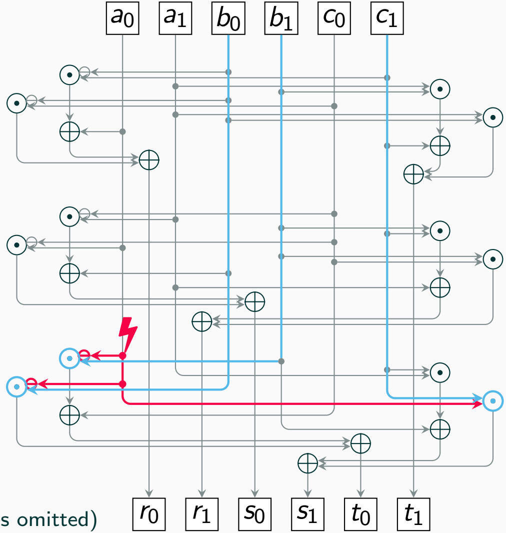
- Same problem as before...
- Difference cancels depending on b_0, b_1 and c_1



(Refreshing of shares omitted)

Case study: SIFA on Masked Chi-3 S-box

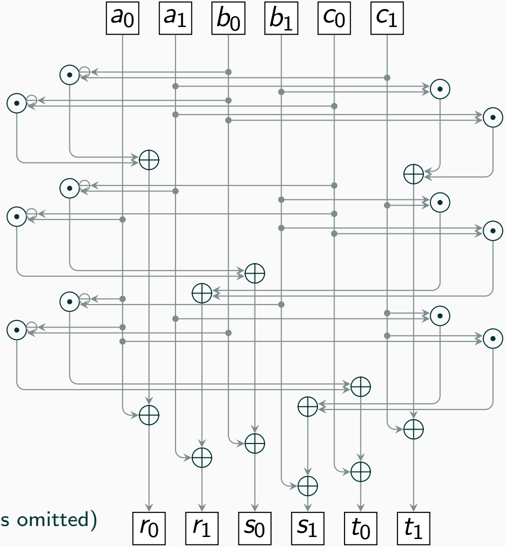
- Same problem as before...
- Difference cancels depending on b_0, b_1 and c_1
- If computation correct despite fault:
  - $b = 0$
  - Bias at S-box output



(Refreshing of shares omitted)

Case study: SIFA Protected Chi-3 S-box

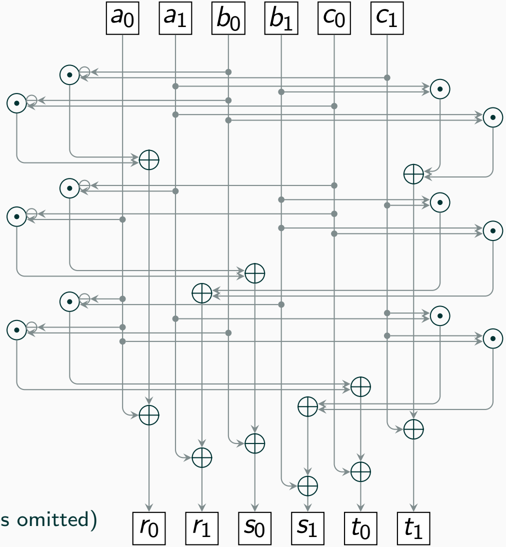
- Basic circuits are incomplete (but not permutations)



(Refreshing of shares omitted)

Case study: SIFA Protected Chi-3 S-box

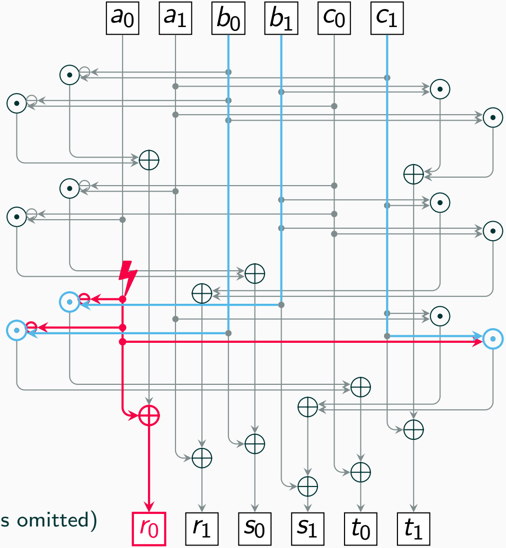
- Basic circuits are incomplete (but not permutations)
- “Dangerous” faults are always visible on the S-box output



(Refreshing of shares omitted)

Case study: SIFA Protected Chi-3 S-box

- Basic circuits are incomplete (but not permutations)
- “Dangerous” faults are always visible on the S-box output



(Refreshing of shares omitted)

Additional Remarks

- This approach can be implemented efficiently
  - No noticeable performance difference to ordinary masked Chi-3
  - Can also be implemented without fresh randomness (3×repeated application of Toffoli-gate)

Additional Remarks

- This approach can be implemented efficiently
    - No noticeable performance difference to ordinary masked Chi-3
    - Can also be implemented without fresh randomness (3×repeated application of Toffoli-gate)
- In the paper we:
    - Prove applicability for all 3-bit and many 4-bit S-boxes
    - Show applicability for Chi-5-ish S-boxes

Additional Remarks

- This approach can be implemented efficiently
  - No noticeable performance difference to ordinary masked Chi-3
  - Can also be implemented without fresh randomness (3×repeated application of Toffoli-gate)
- In the paper we:
  - Prove applicability for all 3-bit and many 4-bit S-boxes
  - Show applicability for Chi-5-ish S-boxes
- What about larger S-boxes like in AES?

Additional Remarks

- This approach can be implemented efficiently
  - No noticeable performance difference to ordinary masked Chi-3
  - Can also be implemented without fresh randomness (3×repeated application of Toffoli-gate)
- In the paper we:
  - Prove applicability for all 3-bit and many 4-bit S-boxes
  - Show applicability for Chi-5-ish S-boxes
- What about larger S-boxes like in AES?
  - Here we can use the Toffoli gate for bigger fields

AES S-box using Incomplete Permutation Basic Circuits

- Based on Canright's description [Can05]

AES S-box using Incomplete Permutation Basic Circuits

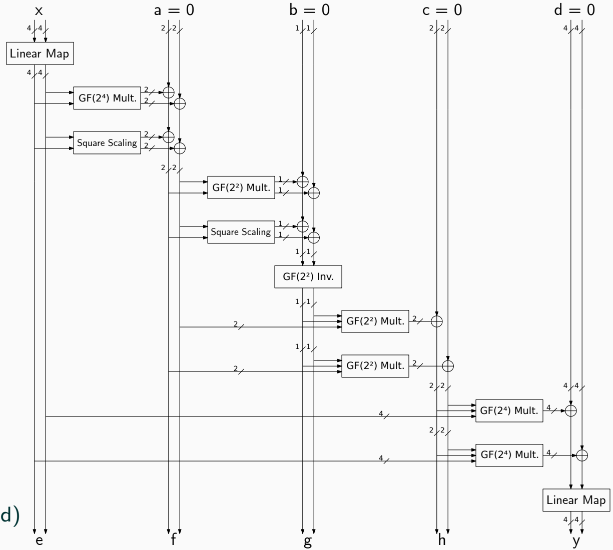
- Based on Canright's description [Can05]
- Convert idea of Sugawara from 3-share to 2-share masking [Sug19]

AES S-box using Incomplete Permutation Basic Circuits

- Based on Canright's description [Can05]
- Convert idea of Sugawara from 3-share to 2-share masking [Sug19]
- Replace all $F(2^n)$ multiplications by Toffoli gates operating in $F(2^n)$, using additional inputs that are set to zero

AES S-box using Incomplete Permutation Basic Circuits

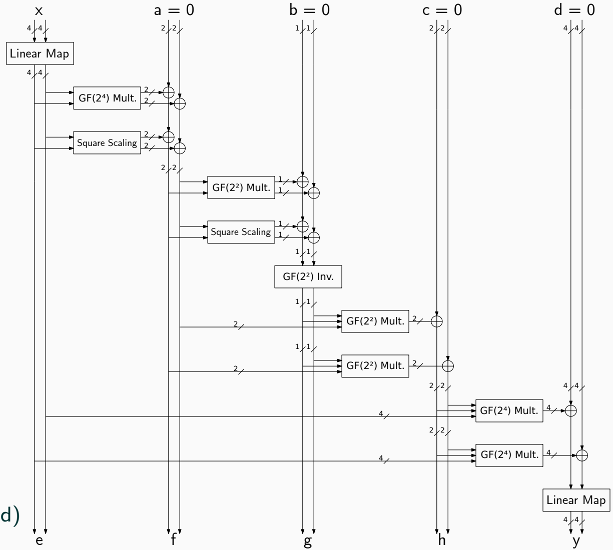
- Inputs:
  - x (8-bits)
  - a, b, c, d (18-bits)



(Masking omitted)

AES S-box using Incomplete Permutation Basic Circuits

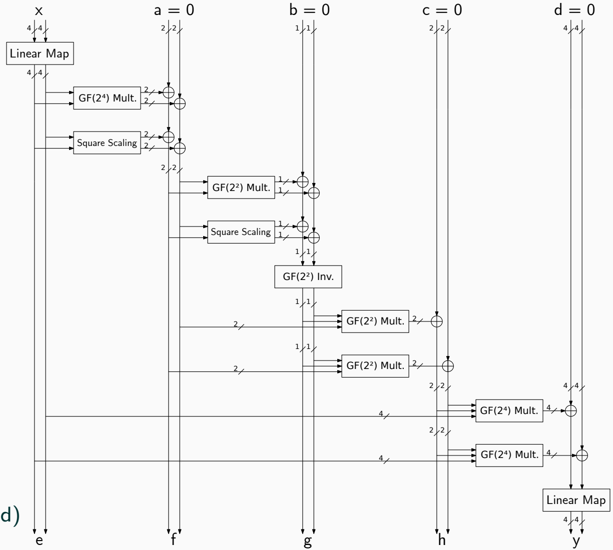
- Inputs:
  - x (8-bits)
  - a, b, c, d (18-bits)
- Outputs:
  - y (8-bits)
  - e, f, g, h (18-bits)



(Masking omitted)

AES S-box using Incomplete Permutation Basic Circuits

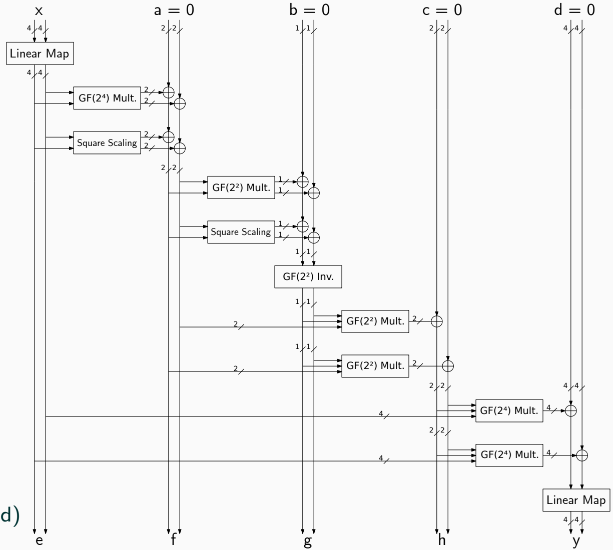
- When masked:
  - x_0, x_1 (16-bits)
  - y_0, y_1 (16-bits)
  - a_0, b_0, c_0, d_0 (18-bits, random)
  - e_0, f_0, g_0, h_0 (18-bits, reusable)



(Masking omitted)

AES S-box using Incomplete Permutation Basic Circuits

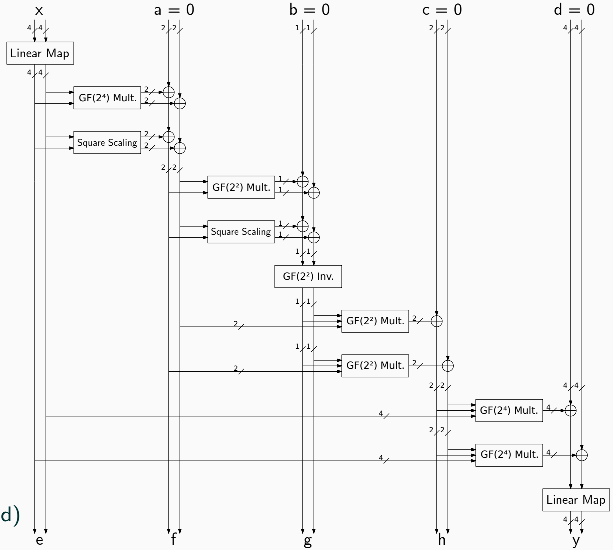
- When masked:
  - x_0, x_1 (16-bits)
  - y_0, y_1 (16-bits)
  - a_0, b_0, c_0, d_0 (18-bits, random)
  - e_0, f_0, g_0, h_0 (18-bits, reusable)
- No need for additional randomness within masked S-box



(Masking omitted)

AES S-box using Incomplete Permutation Basic Circuits

- When masked:
  - x_0, x_1 (16-bits)
  - y_0, y_1 (16-bits)
  - a_0, b_0, c_0, d_0 (18-bits, random)
  - e_0, f_0, g_0, h_0 (18-bits, reusable)
- No need for additional randomness within masked S-box
- Redundancy checks needed after each S-box



(Masking omitted)

Final Remarks

In the paper we:

- Give a complete description of single-fault SIFA resistant (masked) AES S-box

Final Remarks

In the paper we:

- Give a complete description of single-fault SIFA resistant (masked) AES S-box
- Discuss additional implementation aspects for SW/HW

Final Remarks

In the paper we:

- Give a complete description of single-fault SIFA resistant (masked) AES S-box
- Discuss additional implementation aspects for SW/HW
- Present an alternative countermeasure strategy
  - Based on fine-grained redundancy checks
  - Can protect against multi-fault SIFA (but then not so efficient)

Final Remarks

In the paper we:

- Give a complete description of single-fault SIFA resistant (masked) AES S-box
- Discuss additional implementation aspects for SW/HW
- Present an alternative countermeasure strategy
    - Based on fine-grained redundancy checks
    - Can protect against multi-fault SIFA (but then not so efficient)

Side-note: SIFA protection also possible on mode-level (NIST LWC):

- DryGASCON, ISAP

Thank you!

Protecting against Statistical Ineffective Fault Attacks

Joan Daemen, Christoph Dobraunig, Maria Eichlseder, Hannes Gross, Florian Mendel and Robert Primas

CHES 2020

References

[Can05] D. Canright. A Very Compact S-Box for AES. In: CHES. Vol. 3659. Lecture Notes in Computer Science. Springer, 2005, pp. 441–455.

[Dob+18a] C. Dobraunig, M. Eichlseder, H. Groß, S. Mangard, F. Mendel, and R. Primas. Statistical Ineffective Fault Attacks on Masked AES with Fault Countermeasures. In: ASIACRYPT (2). Vol. 11273. Lecture Notes in Computer Science. Springer, 2018, pp. 315–342.

[Dob+18b] C. Dobraunig, M. Eichlseder, T. Korak, S. Mangard, F. Mendel, and R. Primas. SIFA: Exploiting Ineffective Fault Inductions on Symmetric Cryptography. In: IACR Trans. Cryptogr. Hardw. Embed. Syst. 2018.3 (2018), pp. 547–572.

[Fuh+13] T. Fuhr, É. Jaulmes, V. Lomné, and A. Thillard. Fault Attacks on AES with Faulty Ciphertexts Only. In: FDTC. IEEE Computer Society, 2013, pp. 108–118.

[Sug19] T. Sugawara. 3-Share Threshold Implementation of AES S-box without Fresh Randomness. In: IACR Trans. Cryptogr. Hardw. Embed. Syst. 2019.1 (2019), pp. 123–145.