pace
Physical Analysis and Cryptographic Engineering

FDTC 2018
Fault Diagnosis and Tolerance in Cryptography

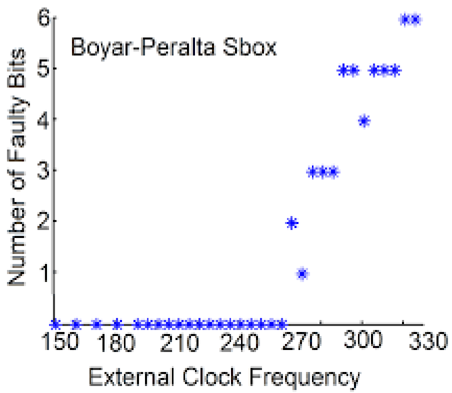
Breaking Redundancy-Based Countermeasures with Random Faults and Power Side Channel

Sayandeep Saha, Dirmanto Jap, Jakub Breier, Shivam Bhasin, Debdeep Mukhopadhyay, and Pallab Dasgupta

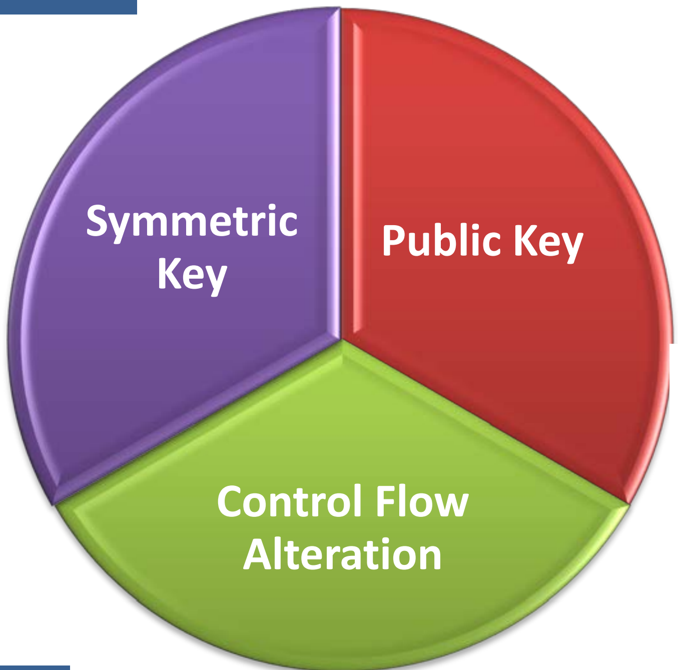
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Introduction

Localized Random Faults

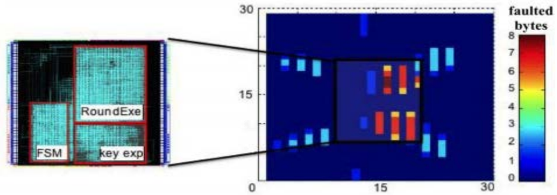


Biased Faults



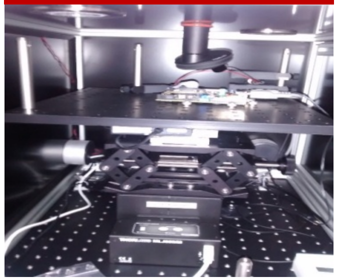
Instruction Skip/Modify

```
1 LDD  R24, Y+i   // load subkey
2 LD   R25, X     // load state
3 EOR  R24, R25   // ExclusiveOR
4 STD  Z+i, R24   // store result
```



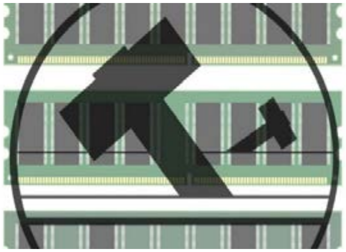
Cartographic view of internal registers

Laser-FI



EM-FI

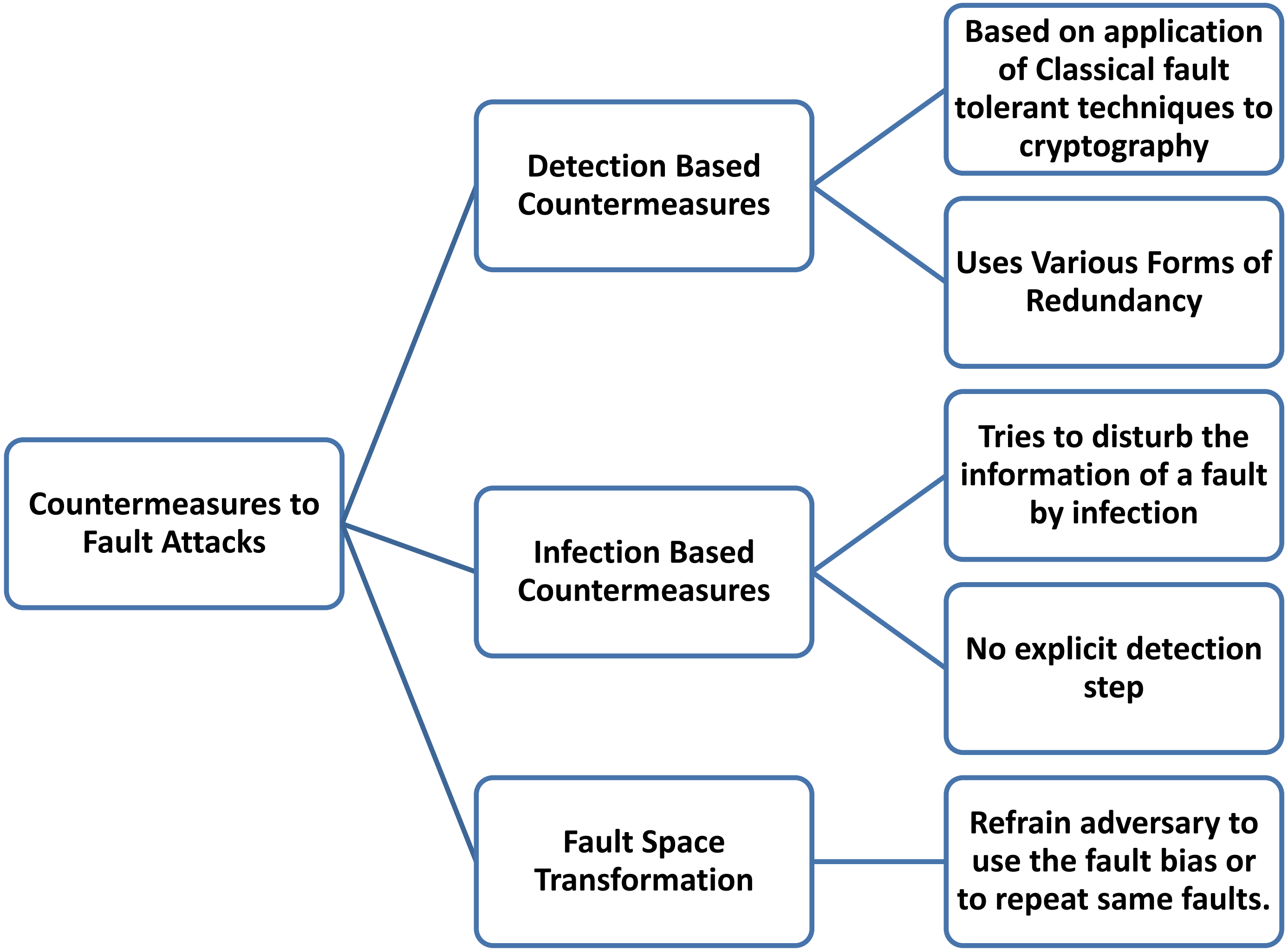
Row-Hammer



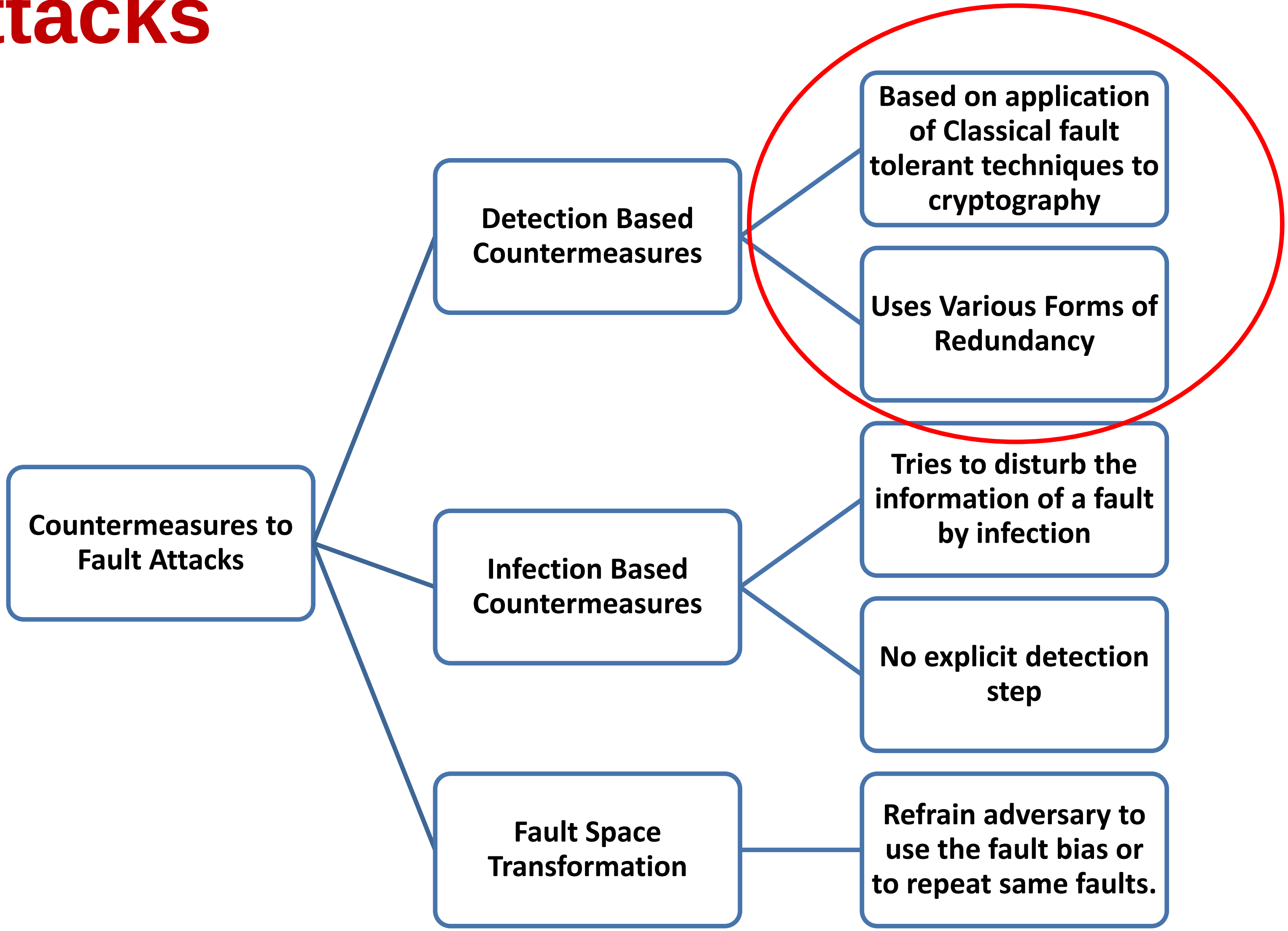
Voltage-Glitch

Clock-Glitch

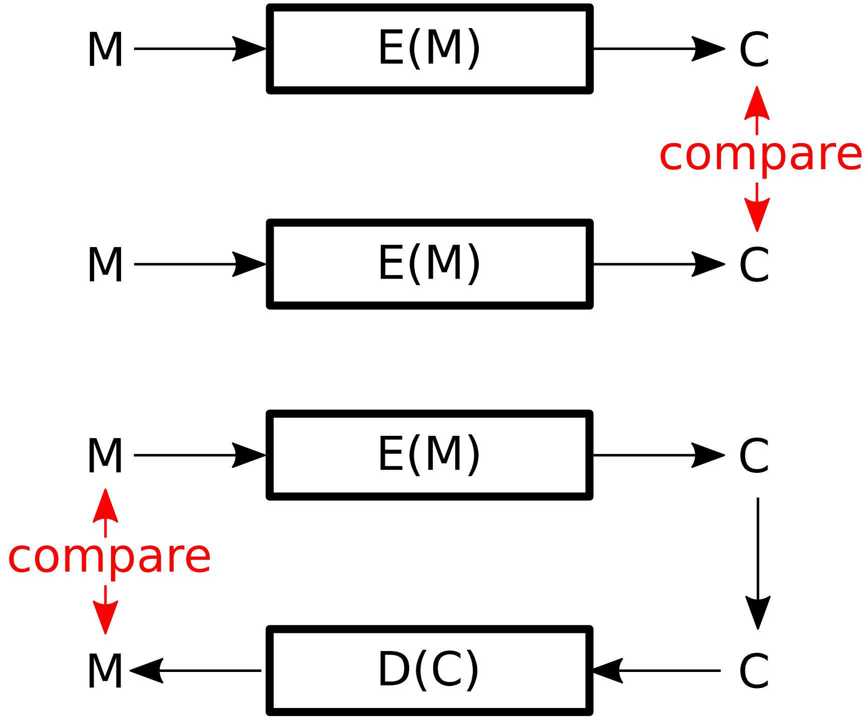
Countermeasures Against Fault Attacks



Countermeasures Against Fault Attacks

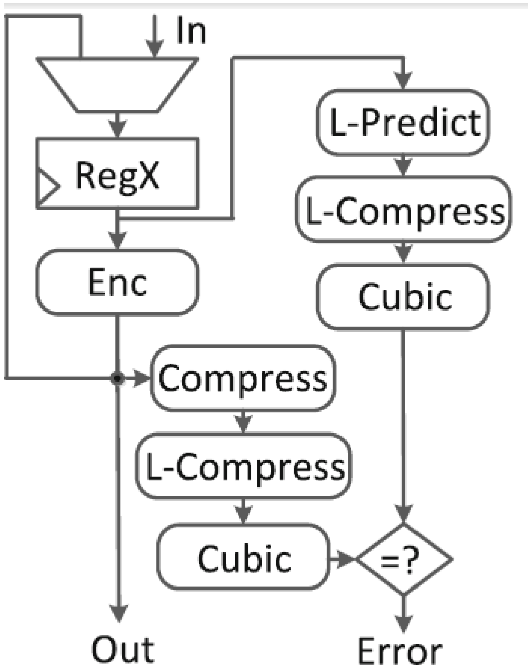


Redundancy Based Countermeasures

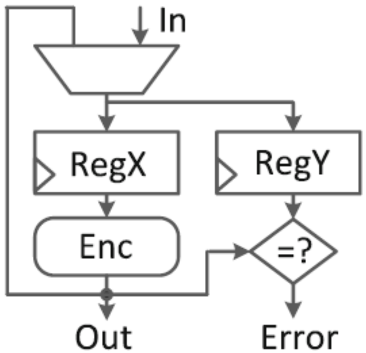


- Follows from classical fault tolerance.
- Simple redundancy executes the encryption twice and then compares the result.
- Another method is to execute the encryption, take the ciphertext, decrypt it, and compare the message.

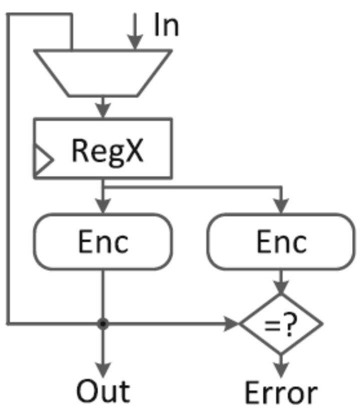
Redundancy Based Countermeasures



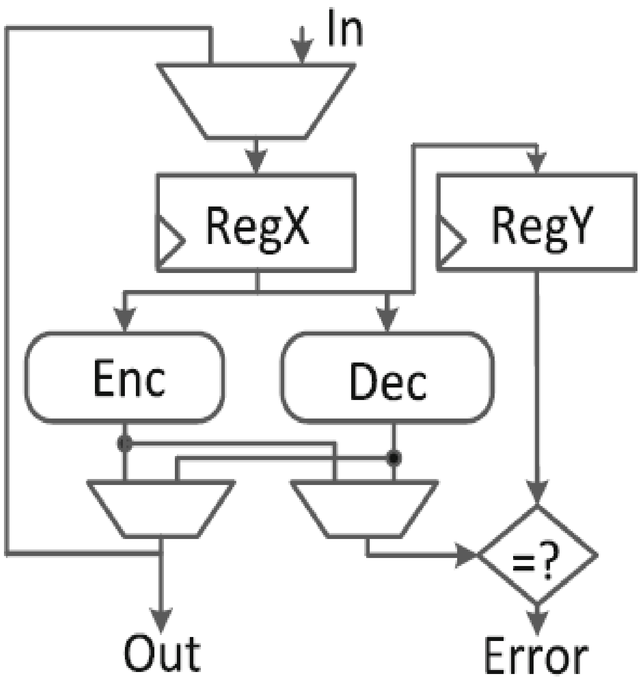
Information Redundancy – Robust Codes



Time Redundancy



Hardware Redundancy



Hybrid Redundancy - REPO

Source : Guo et. al. , Security analysis of concurrent error detection against differential fault analysis – Journal of Cryptographic Engineering, 2014

Simple is Best…

- Simplest form of Redundancy :
  - Execute the encryption twice and then compares the ciphertexts.
- Applications in safety and reliability
- Easy to implement
- Reasonably high fault coverage.
- Relatively low overhead.
- Used widely in industries.

Attacks on Redundancy

2014
- Guo et. al. (JCEN)
- Practically bypass concurrent error detection with biased faults.

2015
- Patranabis et. al. (COSADE)
- Biased faults to bypass time-redundancy.

2016
- Selmake et. al. (FDTC)
- Biased faults to bypass hardware-redundancy.

2017
- Breier et. al. (JCEN)
- Practically bypass information redundancy.

2017
- Wiersma et. al. (FDTC)
- Attack on commercial processors having ASIL-D security.

Attacks on Redundancy

2014
- Guo et. al. (JCEN)
- Practically bypass concurrent error detection with biased faults.

2015
- Patranabis et. al. (COSADE)
- Biased faults to bypass time-redundancy.

2016
- Selmake et. al. (FDTC)
- Biased faults to bypass hardware-redundancy.

2017
- Breier et. al. (JCEN)
- Practically bypass information redundancy.

2017
- Wiersma et. al. (FDTC)
- Attack on commercial processors having ASIL-D security.

Our Contributions

Previous Work	We Propose
➢ Bypass the countermeasures	➢ Use the countermeasure to leak
➢ Use Biased Faults	➢ Use Random Faults
➢ Corrupt all computation branches	➢ Corrupt single branch

Our Contributions

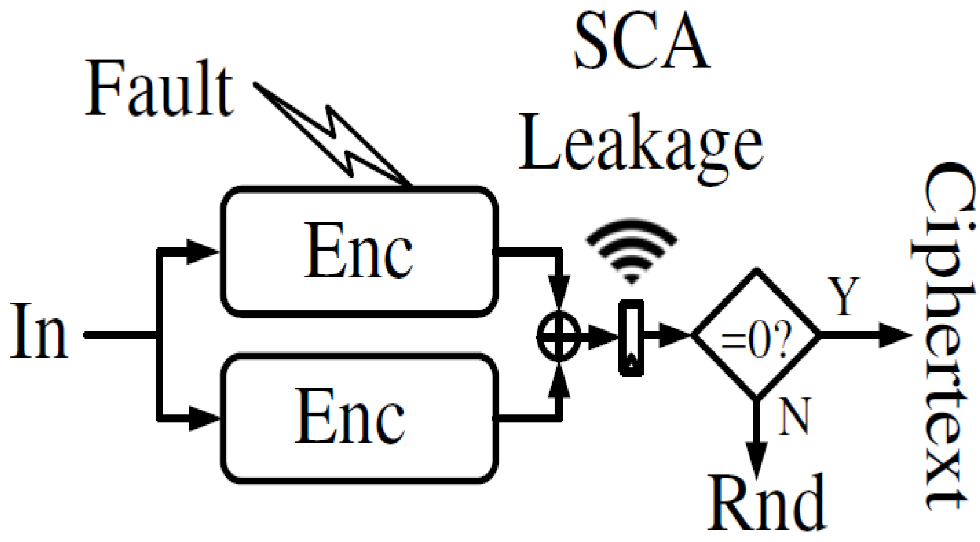
Previous Work	We Propose
➢ Bypass the countermeasures	➢ Use the countermeasure to leak
➢ Us	
➢ Corrupt all computation branches	➢ Corrupt single branch

Even Countermeasures Leak

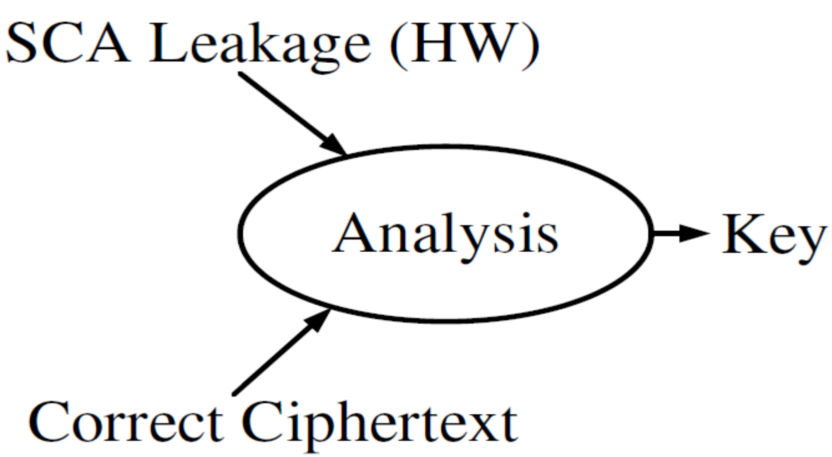
SCA Assisted DFA

- Proposed in FDTC 2017 by Patranabis et. al.
  - One Plus One is More than Two: A Practical Combination of Power and Fault Analysis Attacks on PRESENT and PRESENT-Like Block Ciphers.
- Uses side-channel to expose certain properties of bit permutation on fault injection
- Attacks on unprotected implementation of bit-permutation based ciphers
- Here we use side-channel to capture the leakage from countermeasures

SCA Assisted DFA: The Context of Countermeasures



- Assumptions:
  - Two or more redundant cipher computation and equality check of ciphertexts.
  - Side-channel measurement from the comparison operation
  - Random faults corrupting one single computation branch.



The Main Idea

- What we have and what we don't :
  - Correct ciphertexts: C → Known
  - Faulty Ciphertexts: C^* → Unknown
  - $HW(\delta) = HW(C \oplus C^*)$ → Known (bytewise)

With:

$HW(\delta) = w$

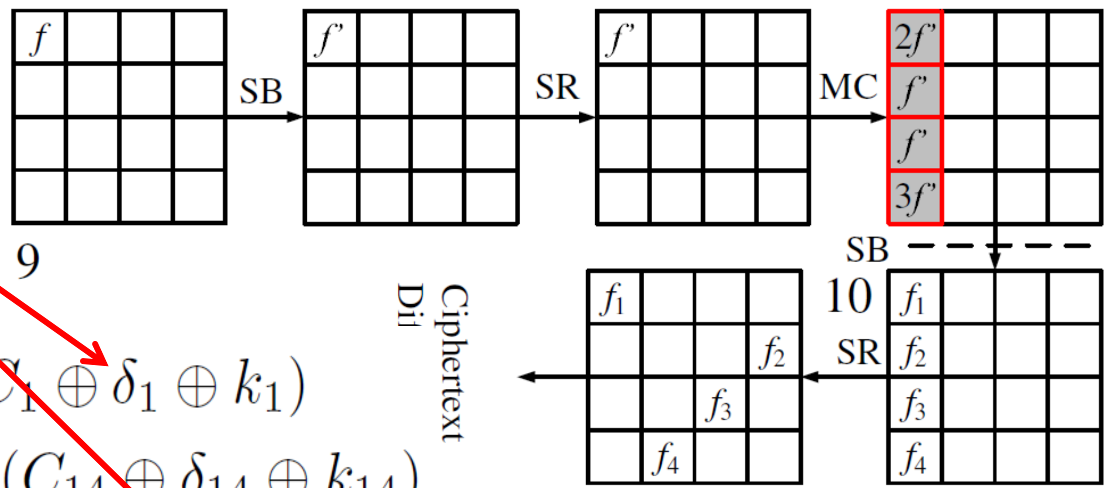
$\binom{8}{w}$ choices for C^*

Without:

2^8 choices for C^*

Can be low for certain choices of w

Case Study I: AES



$W(\delta_i)$

$$2f_1 = S^{-1}(C_1 \oplus k_1) \oplus S^{-1}(C_1 \oplus \delta_1 \oplus k_1)$$
$$f_1 = S^{-1}(C_{14} \oplus k_{14}) \oplus S^{-1}(C_{14} \oplus \delta_{14} \oplus k_{14})$$
$$f_1 = S^{-1}(C_{11} \oplus k_{11}) \oplus S^{-1}(C_{11} \oplus \delta_{11} \oplus k_{11})$$
$$3f_1 = S^{-1}(C_9 \oplus k_9) \oplus S^{-1}(C_9 \oplus \delta_9 \oplus k_9)$$

$$|\mathcal{R}| = 2^8 \times \prod_{i \in I} \binom{8}{W(\delta_i)}$$
$$|\mathcal{F}| = 1$$

- For each choice of $W(\delta_i)$ we have $2^8 \times \binom{8}{W(\delta_i)}$ choices for (k_i, δ_i)

Case Study I: AES

$$|\mathcal{R}| = 2^8 \times \prod_{i \in I} \binom{8}{W(\delta_i)}$$

$$|\mathcal{F}| = 1$$

For practical attack:

$$|\mathcal{R}| \leq 2^{32}$$

$$|\mathcal{F}| \leq 2^{32}$$

Worst Case

$$\binom{8}{W(\delta_i)} = \binom{8}{4} = 70$$

$$|\mathcal{R}| = 2^{32}$$

Best Case

$$\binom{8}{W(\delta_i)} = \binom{8}{8} = 1$$

$$|\mathcal{R}| = 2^8$$

Case Study I: AES

Wait and see…

$$S = \{1,2,3,4,5,6,7,8\}$$ → All possible HW values

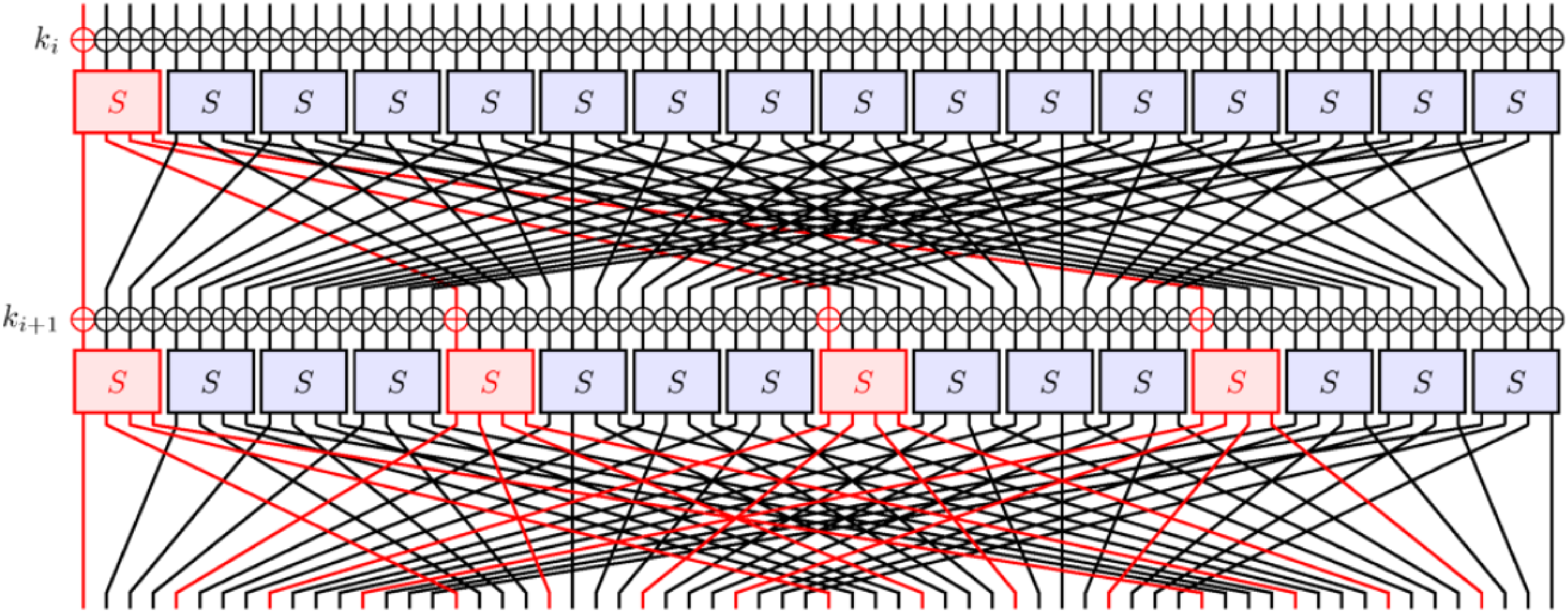
Let's just consider → $S - \{3,4,5\}$

Worst Case: $|\mathcal{F}| = \frac{2^{32}}{(28)^4} \approx \frac{2^{32}}{(2^5)^4} \approx 2^{12} \quad |\mathcal{R}| = 2^{18}$

Fairly Reasonable

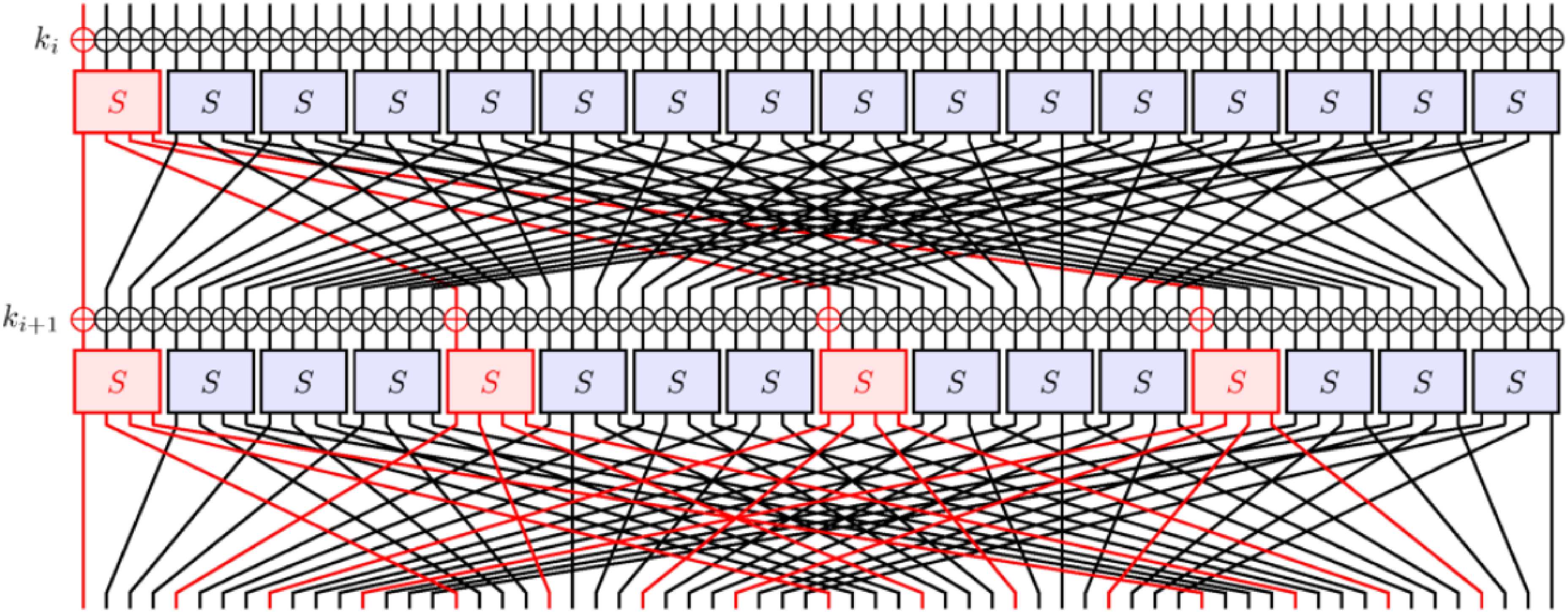
On average, the 128-bit AES key can be recovered with 2^{25} injections.

Case Study II: PRESENT



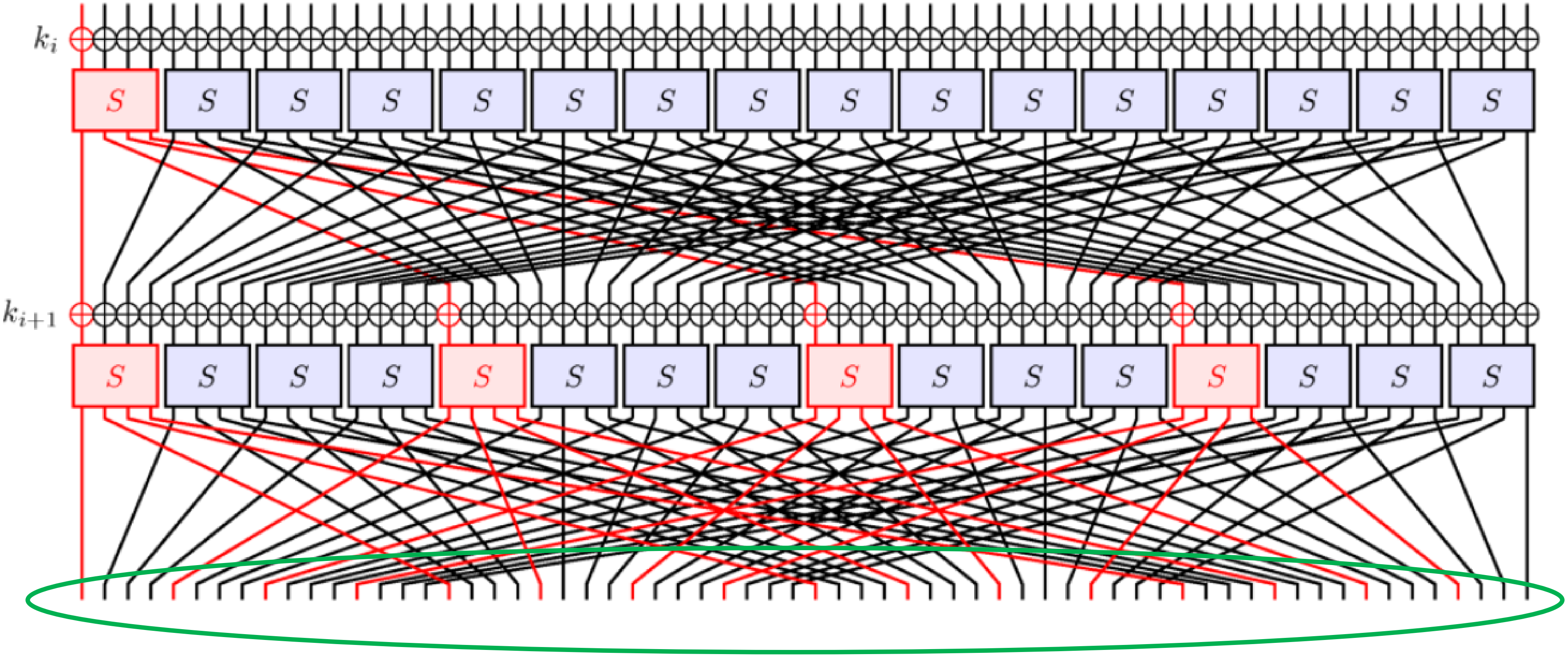
$$f_i = S^{-1}(C_i \oplus K_i) \oplus S^{-1}(C_i \oplus \delta_i \oplus K_i), \text{for } i \in [0, 15]$$

Case Study II: PRESENT



- We want nibble-wise Hamming weights
- We get byte-wise Hamming weights
- How to get nibble-wise values for byte-wise values???

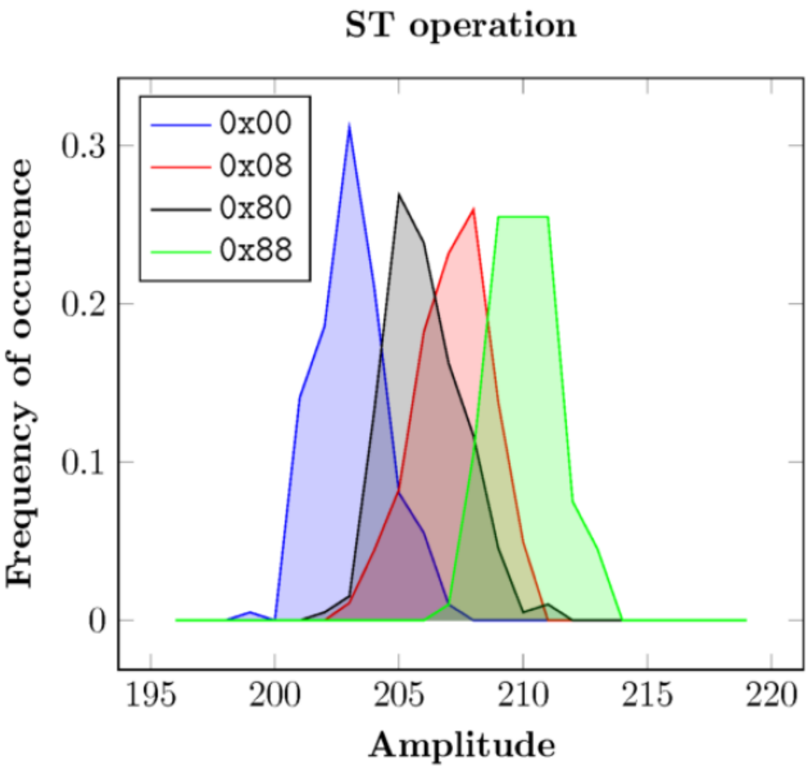
Case Study II: PRESENT



- No two consecutive nibbles in a byte are active simultaneously
- Only 3 byte-wise Hamming weights can be observed: 0, 1, 2

Case Study II: PRESENT

Templates: General approach for extracting nibble-wise Hamming weights

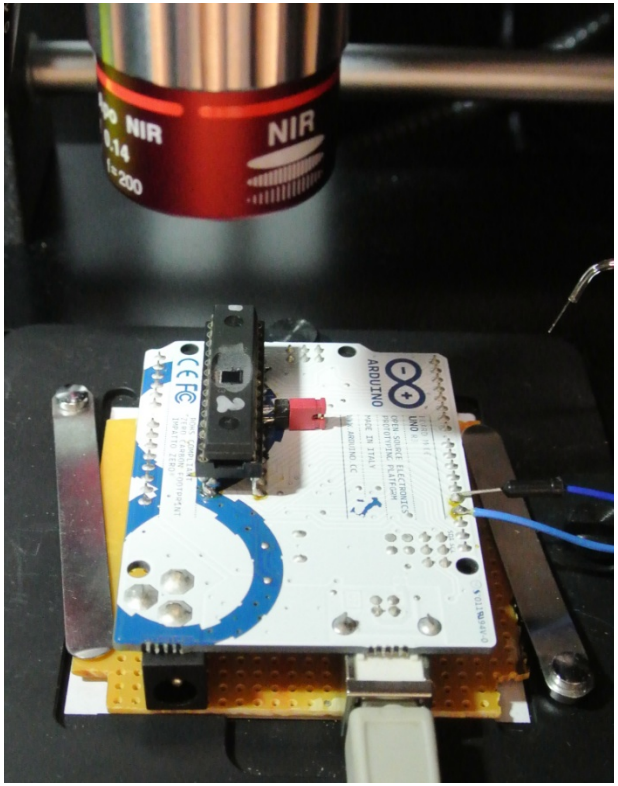


- 4 possible byte values: 00, 08, 80, 88
- All are clearly distinguishable from templates.
- Each nibble Hamming weight and nibble value has one-to-one correspondence.
  - We can uniquely extract the ciphertexts.

With 4 fault injections, the last round key can be determined uniquely

Practical Validation

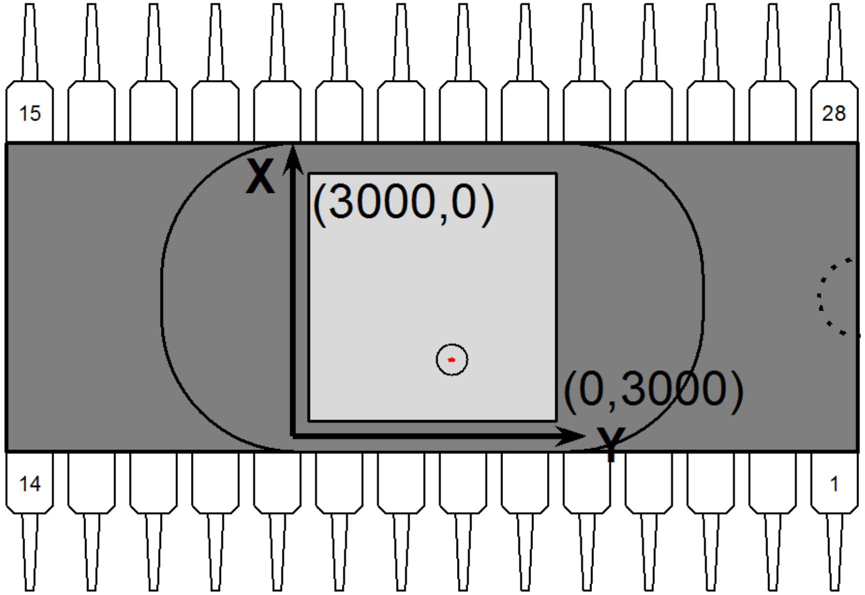
Laser fault injection on an ATmega328P 8-bit microcontroller



- Near-infrared diode pulse laser
- Maximum output power of 20 W
- For the experiments, 20x magnifying objective lens was used
- As a DUT, ATmega328P was used – an 8-bit microcontroller running at 16 MHz
- Chip was depackaged from the backside to be accessible by the laser

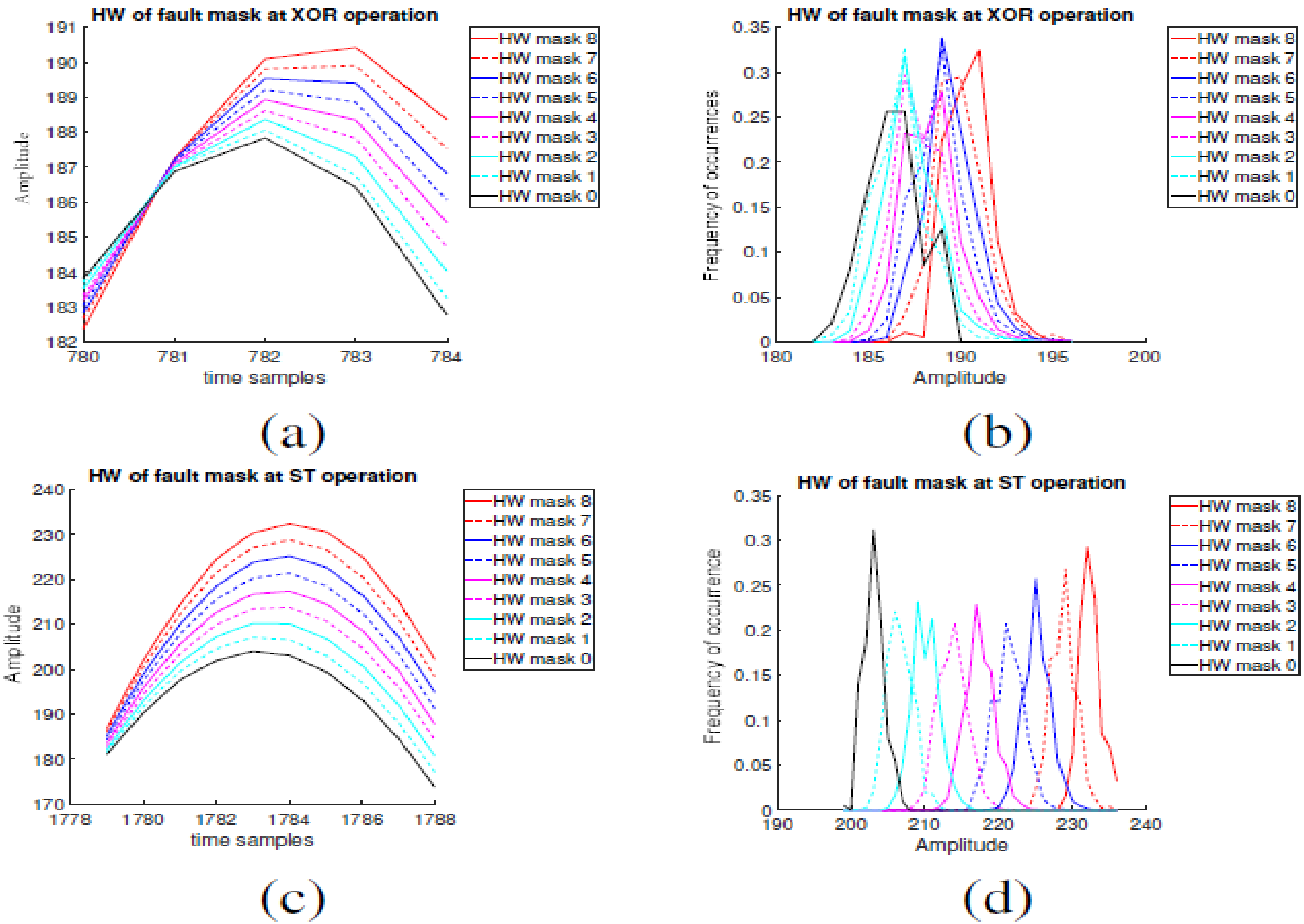
Practical Validation

Laser fault injection on an ATmega328P 8-bit microcontroller

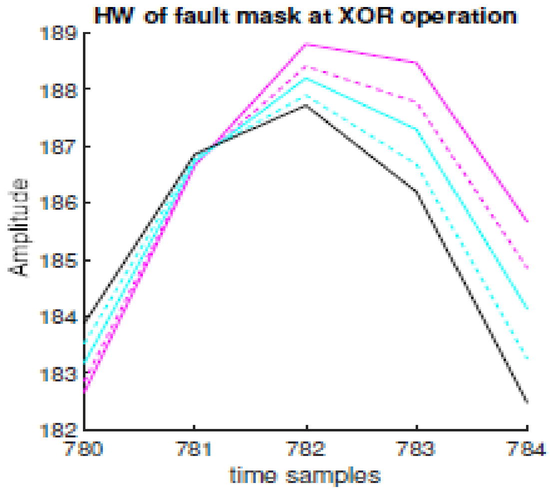


- Total area vulnerable to experiments was <1% of the entire chip area
- Reproducibility of faults was near to 100% with the same laser settings

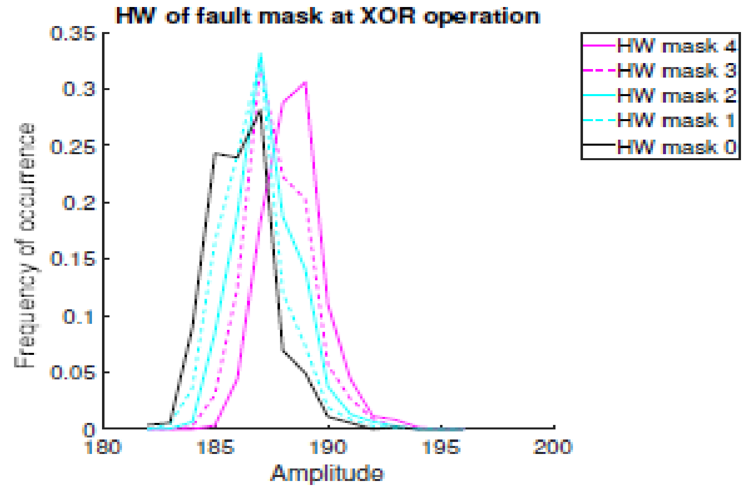
Practical Validation



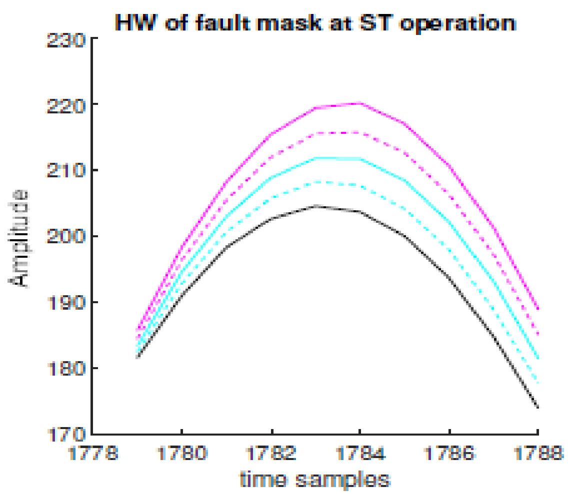
Practical Validation



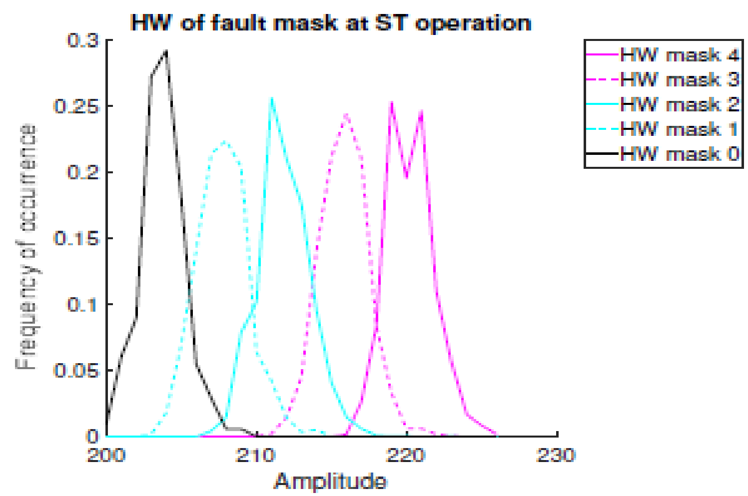
(a)



(b)



(c)



(d)

Practical Validation

Cipher	Code Size (bytes)	T_{ENC}	N_{EXP}	$(\|\mathcal{E}\|, \|\mathcal{F}\|, \|\mathcal{R}\|)$
AES-128	7570	0.326	$2^{26.98}$	$(2^{43}, 2^{25}, 1)$
PRESENT-80	7110	4.01	$2^{23.36}$	$(2^{4}, 4, 1)$

2^{25} injections can be performed within a day !!!

Summary

- Redundancy based countermeasures are simple and practical.
- Usage: very simple.
- Caution!!!
  - They leak unless properly constructed.

- Potential Solutions:
  - Mask the comparison block ⇒ Resource overhead
  - Redundancy at each round ⇒ May not be secured

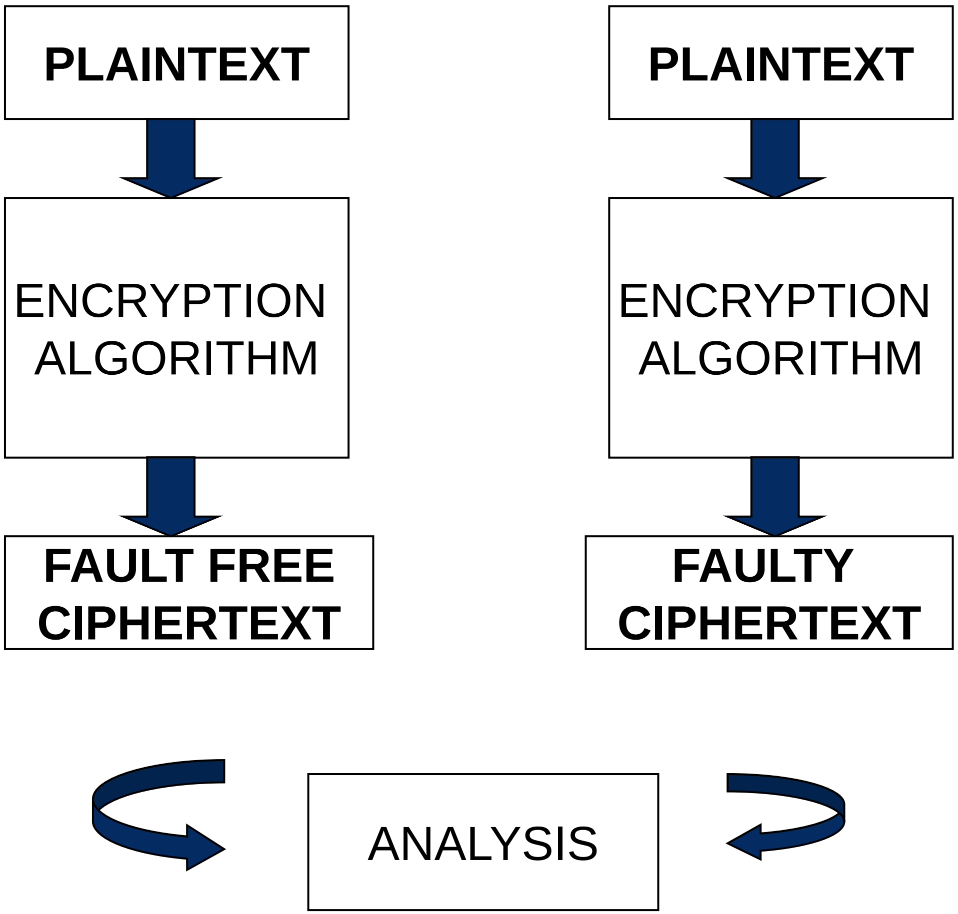
- Future works:
  - Extension for more general form of redundancies.
  - Low-cost but leakage-free countermeasure construction.

Thank you

Questions?

Introduction

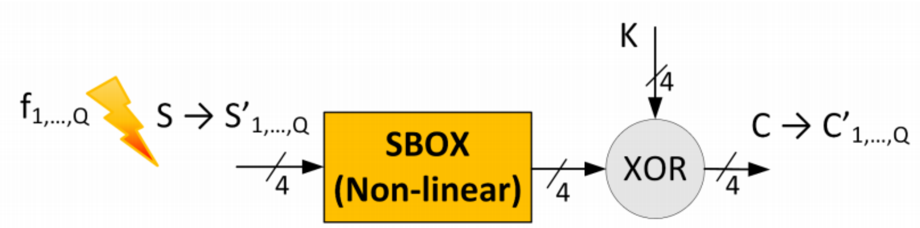
Differential Fault Analysis (DFA)



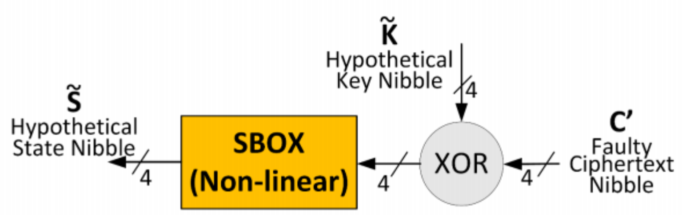
- Most widely explored
- Low fault complexity
- Complex analysis
- Fault Locations
  - Datapath
  - Key-schedule
- Fault models
  - Bit based
  - Nibble based
  - Byte based
  - Multiple byte based

Introduction

- Step 1: Biased Fault Injection
  - Apply Q different fault intensities ($f_{1,\ldots,Q}$)
  - Induce biased faults ($S'_{1,\ldots,Q}$)
  - Collect faulty ciphertexts ($C'_{1,\ldots,Q}$)



Step 2: Hypothesis Test with Biased Faults

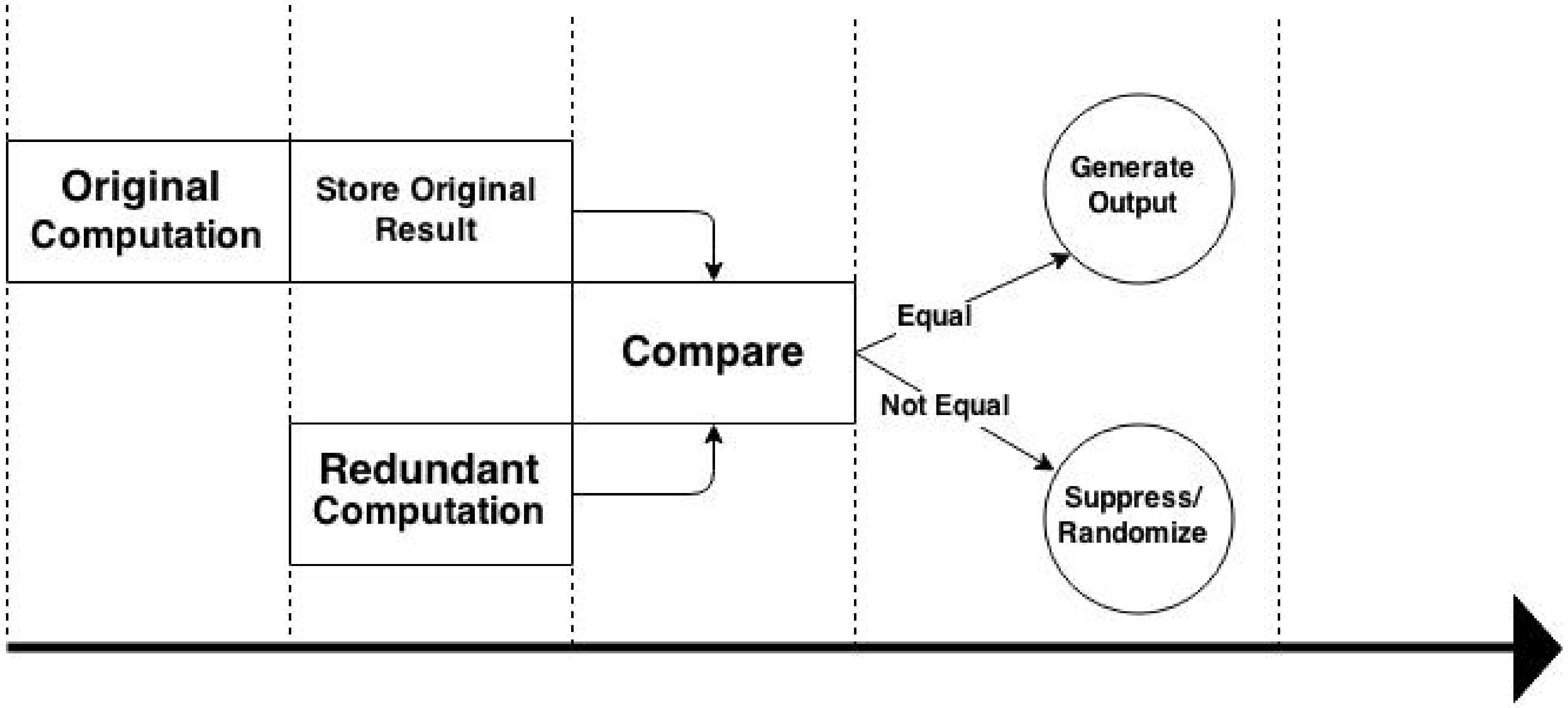


Given: C' and a KNOWN fault bias f
Find: Most likely key nibble $\tilde{K}$

For all $\tilde{K}$, find $\tilde{S} = SBOX^{-1}(C' \oplus \tilde{K})$
Accumulate $\rho_{\tilde{K}} = \sum HD(\tilde{S})$
Select K = argmin ρ

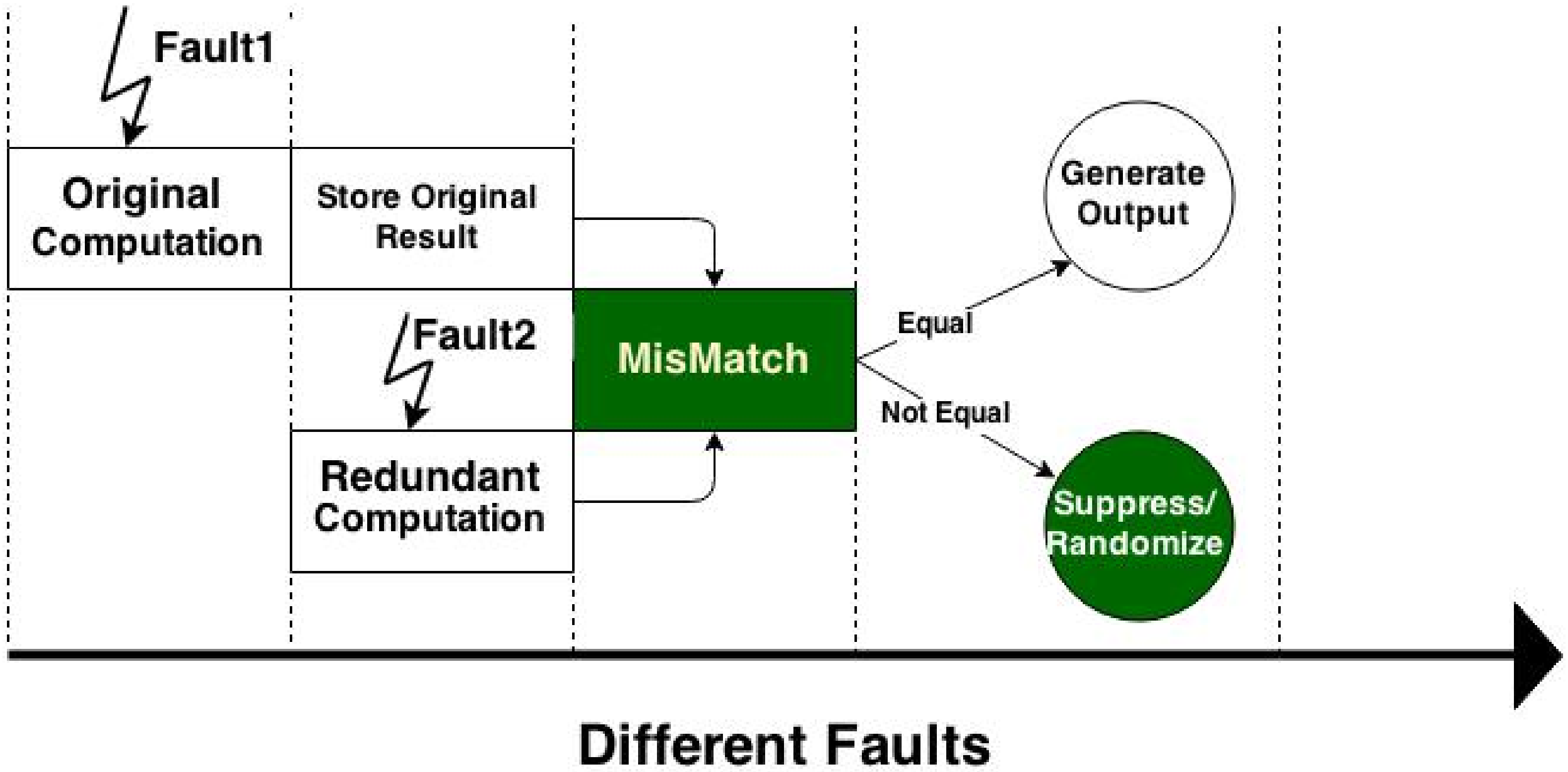
- **Biased Faults**: Distribution of the faulty values are non-uniform.
- Bias is exploited for key extraction by means of Hypothesis Testing.
- Utilizes device properties to the highest extent.
- Requires only faulty ciphertexts – But many of them.

Attacks on Redundancy



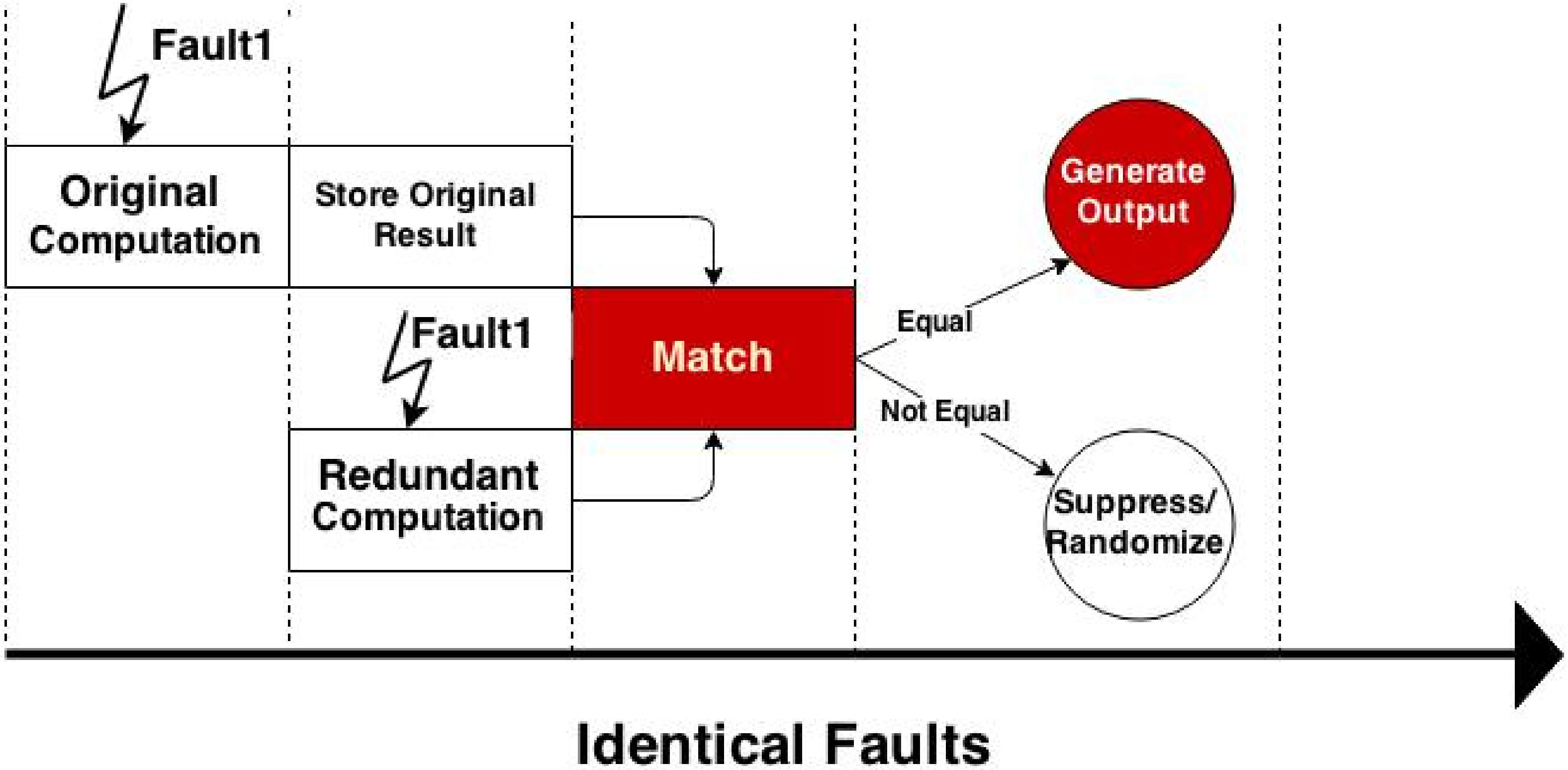
S.Patranabis, A.Chakraborty, P.H.Nguyen and D.Mukhopadhyay. A Biased Fault Attack on the Time Redundancy Countermeasure for AES. In *Proceedings of Constructive Side Channel Analysis and Secure Design 2015 (COSADE 2015*), Berlin, Germany, April 2015

Attacks on Redundancy



S.Patranabis, A.Chakraborty, P.H.Nguyen and D.Mukhopadhyay. A Biased Fault Attack on the Time Redundancy Countermeasure for AES. In *Proceedings of Constructive Side Channel Analysis and Secure Design 2015 (COSADE 2015*), Berlin, Germany, April 2015

Attacks on Redundancy



S.Patranabis, A.Chakraborty, P.H.Nguyen and D.Mukhopadhyay. A Biased Fault Attack on the Time Redundancy Countermeasure for AES. In *Proceedings of Constructive Side Channel Analysis and Secure Design 2015 (COSADE 2015*), Berlin, Germany, April 2015