FDTC 2020
Fault Diagnosis and Tolerance in Cryptography

RUHR UNIVERSITÄT BOCHUM

RUB

MAX PLANCK INSTITUTE FOR SECURITY AND PRIVACY

SPFA – SFA ON MULTIPLE PERSISTENT FAULTS

Authors: Susanne Engels[1,2] , Falk Schellenberg[2] , Christof Paar[2]

[1] Ruhr University Bochum, Germany

[2] Max Planck Institute for Security and Privacy, Bochum, Germany

PERSISTENT FAULTS AND PFA

DURATION OF A FAULT

Transient	Persistent	Permanent
Limited period of time	Lasts until reset	permanent
Examples:	Examples:	Examples:
• Laser Beams • Temperature • Voltage	• SRAM	• Defects • Failure • Destruction

PERSISTENT FAULT ANALYSIS (PFA): TIMELINE

- **First introduced by Zhang et al. at CHES 2018**

Persistent Fault Analysis on Block Ciphers

Fan Zhang[1,2], Xiaoxuan Lou[1,2], Xinjie Zhao[4], Shivam Bhasin[5], Wei He[6], Ruyi Ding[1,7], Samiya Qureshi[1] and Kui Ren[3]

- **Practically evaluated on CAESAR finalists by Gruber et al. FDTC 2019**

Persistent Fault Analysis of OCB, DEOXYS and COLM

Michael Gruber, Matthias Probst, Michael Tempelmeier

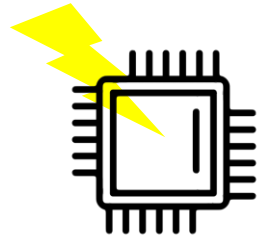
- **Enhanced version by Zhang et al. at CHES 2020**

Persistent Fault Attack in Practice

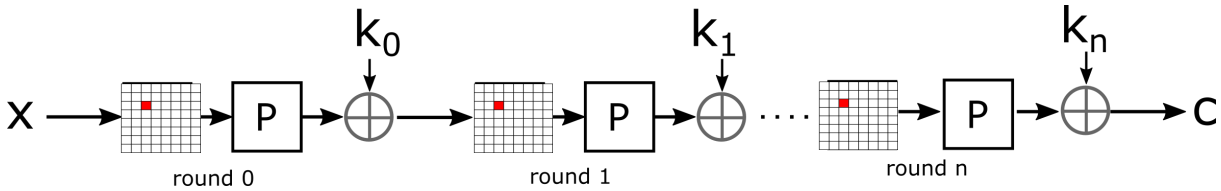
Fan Zhang[1,2,4], Yiran Zhang[1,3,4], Huilong Jiang[5], Xiang Zhu[5], Shivam Bhasin[6], Xinjie Zhao[7], Zhe Liu[2,8(✉)], Dawu Gu[9] and Kui Ren[1,4]

PFA: FAULT MODEL

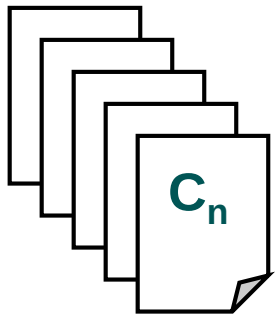
1. The adversary can inject faults prior to the encryption.



2. Injected faults are persistent, i.e., last for several computations.

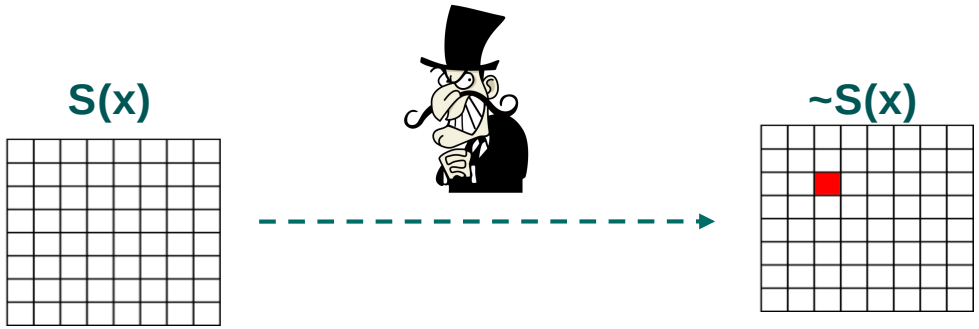


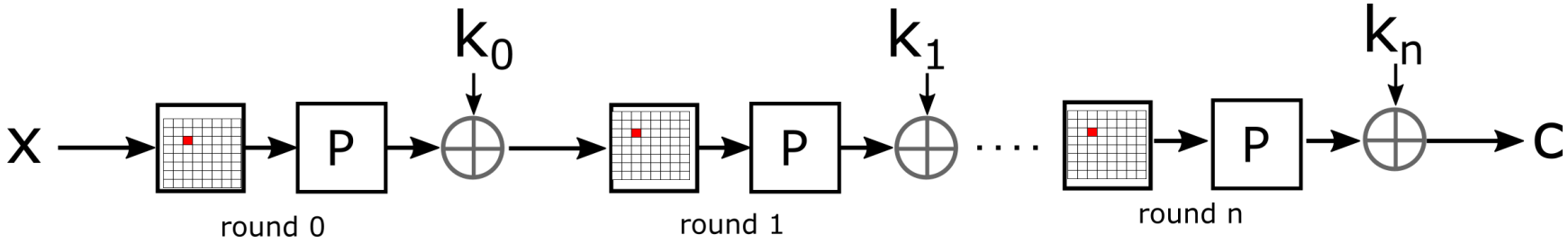
3. Multiple faulty ciphertexts can be collected.



PFA: WORKING PRINCIPLE

Step 1: Preparation



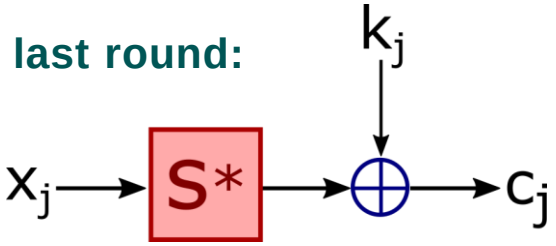


Step 2: Collect faulty ciphertexts

Step 3: Analysis

PFA: THE ANALYSIS PART

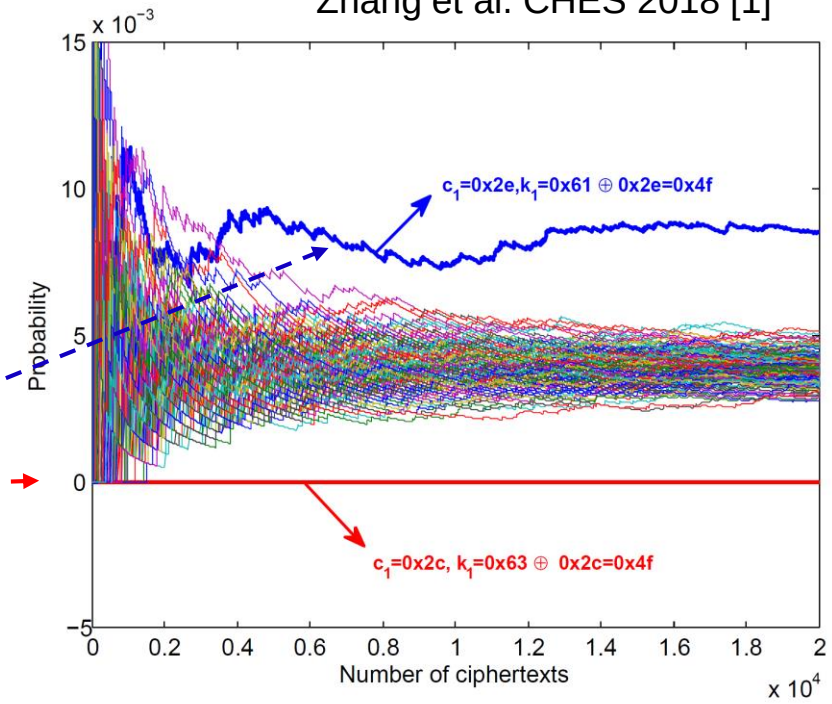
- **Targets last round:**



- **Three attack strategies:**

1. $k_j = v + t_{min}$
2. $k_j \neq v + t$
3. $k_j = v^* + t_{max}$

Zhang et al. CHES 2018 [1]



(a) Extract k_1 using the distribution of c_1

CLASSICAL PFA: LIMITS

- **Adversary needs to know or bruteforce the value and/or position of the fault**

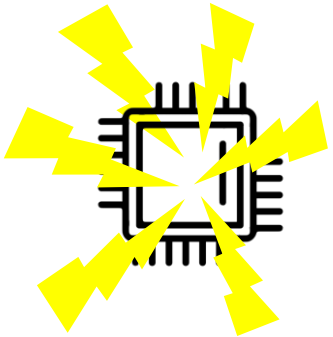
$$k_j = v + t_{min} \qquad k_j \neq v + t \qquad k_j = v^* + t_{max}$$

- **Impromptu key recovery only possible for single byte faults**

✓ **For single faults, improved variant PFA using MLE (Zhang et al., CHES 2020)**

➢ **Open research questions:**

What about multiple unknown faults?

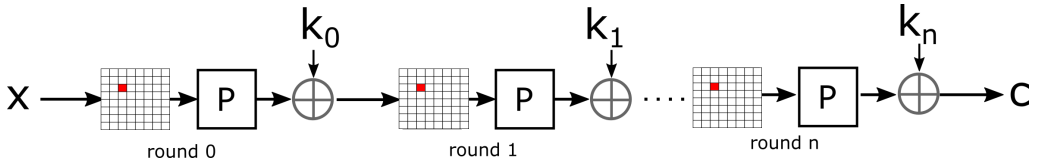


STATISTICAL PERSISTENT FAULT ANALYSIS (SPFA)

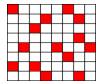
SPFA: COMBINING PFA WITH CLASSICAL SFA

PFA

Persistent faults



SFA



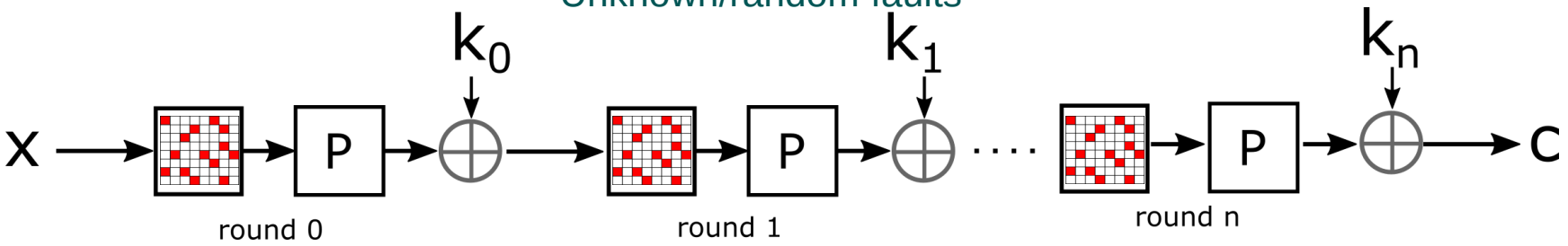
Multiple faults

$$d(\hat{K}) = \sum_{\delta=0}^{2^s-1} \left(\frac{\#\{i \mid f^{-1}(\hat{K}, \tilde{C}_i) = \delta\}}{N} - \frac{1}{2^s} \right)^2$$

Unknown faults

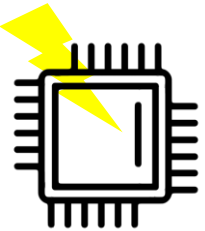
SPFA

Multiple persistent faults
Unknown/random faults

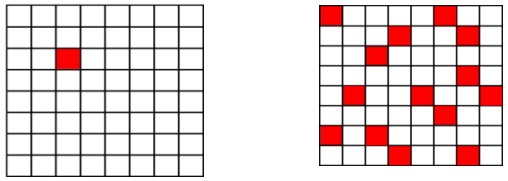


SPFA: RELAXED FAULT MODEL

1. The adversary can inject faults prior to the encryption.

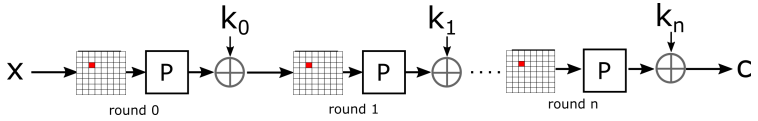


2. Faults can range from affecting a single byte to larger structures.

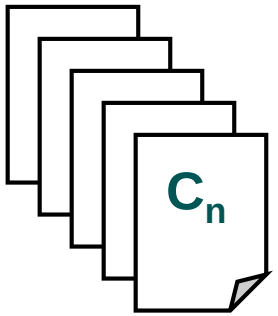


3. The faults cause a non-uniform distribution.

4. Injected faults are persistent, i.e., affect all following computations.

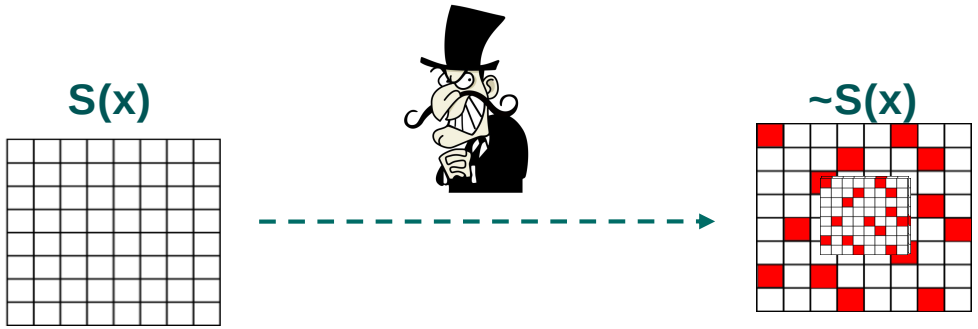


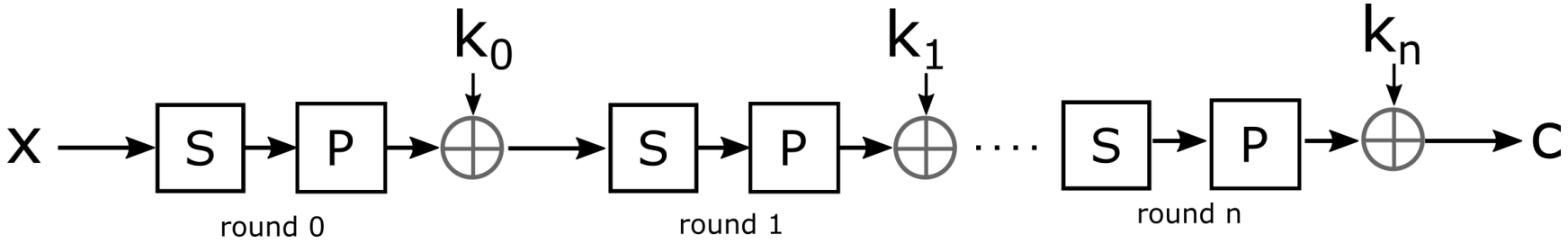
5. Multiple faulty ciphertexts can be collected.



SPFA: WORKING PRINCIPLE

Step 1: Preparation

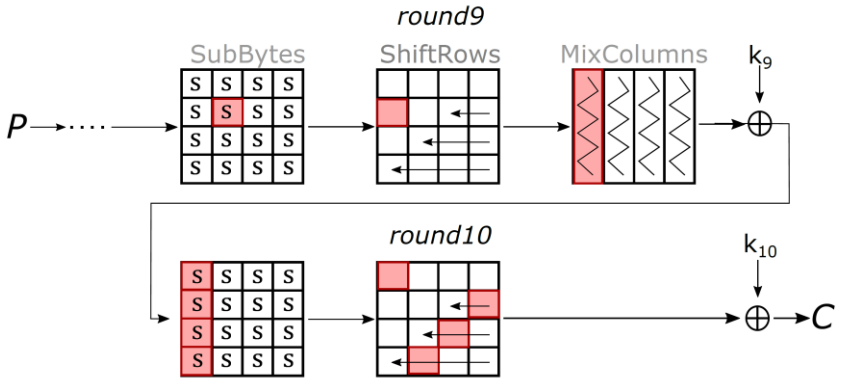




Step 2: Collect faulty ciphertexts

Step 3: Analysis

SPFA: ANALYSIS



- **Exploits fault in penultimate round**
- **The key to our analysis is to detect the bias caused by the faults**
- **Two scenarios:**
  1. Wrong key guess: approximately uniform distribution
  2. Correct key guess: faulty intermediate value will cause bias in the distribution
- **We use the Squared Euclidean Imbalance to rank key hypotheses**

$$SEI(\hat{k}) = \sum_{\delta=0}^{2^s-1} \left(\frac{\#\{i \mid \tilde{S}_r^{(\hat{k},\tilde{c}_i)}[t] = \delta\}}{N} - \frac{1}{2^s} \right)^2$$

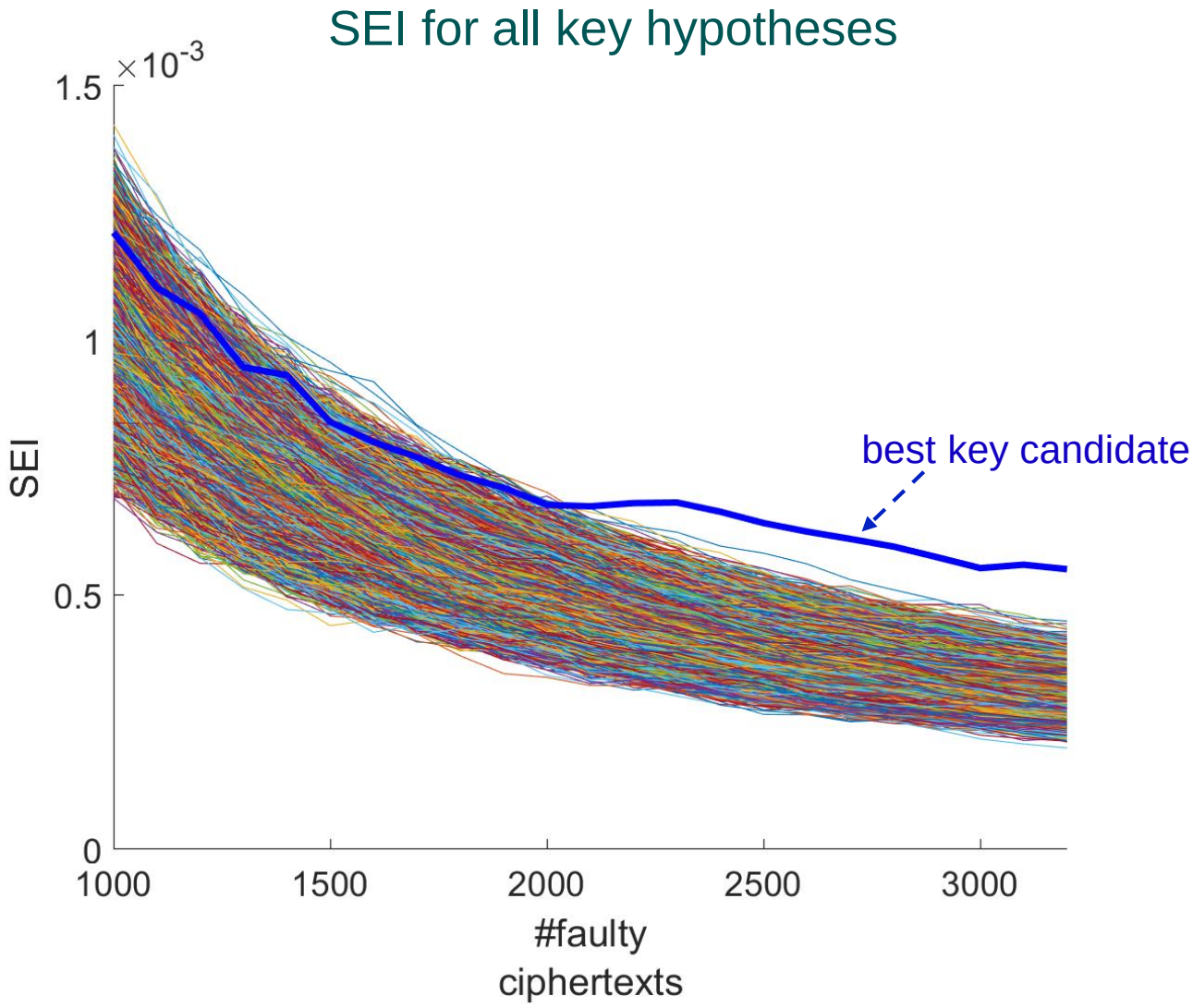
SPFA: RESULTS

- **What does a successful SPFA attack look like?**

```
for (u32 k0 = 0; k0 < 256; k0++)
{
    for (u32 k1 = 0; k1 < 256; k1++)
    {
        for (u32 k2 = 0; k2 < 256; k2++)
        {
            for (u32 k3 = 0; k3 < 256; k3++)
            {
```

```
0.0008270833
0.0008937500
0.0007354167
0.0008701389
0.0009604166
0.0008229167
0.0007854167
0.0008534722
0.0007354167
0.0008159722
0.0008118056
0.0007881944
0.0009479167
0.0008631944
0.0008243056
```

SPFA: RESULTS



EXPERIMENTS

PRACTICAL EVALUATION: SETUP

- **C++ implementation of block ciphers LED and AES**
- **IntelCore i7-6700 with 8 threads**
- **Serialized implementation → single Sbox is stored as a static table**

- ***Preparation*: manipulate (parts of) the Sbox**
- ***Collection*: collect faulty ciphertexts and store (along with encryption key)**
- ***Analysis*:**
  1. Calculate SEI for each key hypothesis: $2^8 * 2^8 * 2^8 * 2^8 * 2^8 = 2^{48}$
  2. Find the maximum SEI → MATLAB

```
for (u32 k0 = 0; k0 < 256; k0++)
{
    for (u32 k1 = 0; k1 < 256; k1++)
    {
        for (u32 k2 = 0; k2 < 256; k2++)
        {
            for (u32 k3 = 0; k3 < 256; k3++)
            {

                #pragma omp parallel for reduction (+: sum)
                for (u32 im = 0; im < INTERMEDIATES; im++)
                {
```

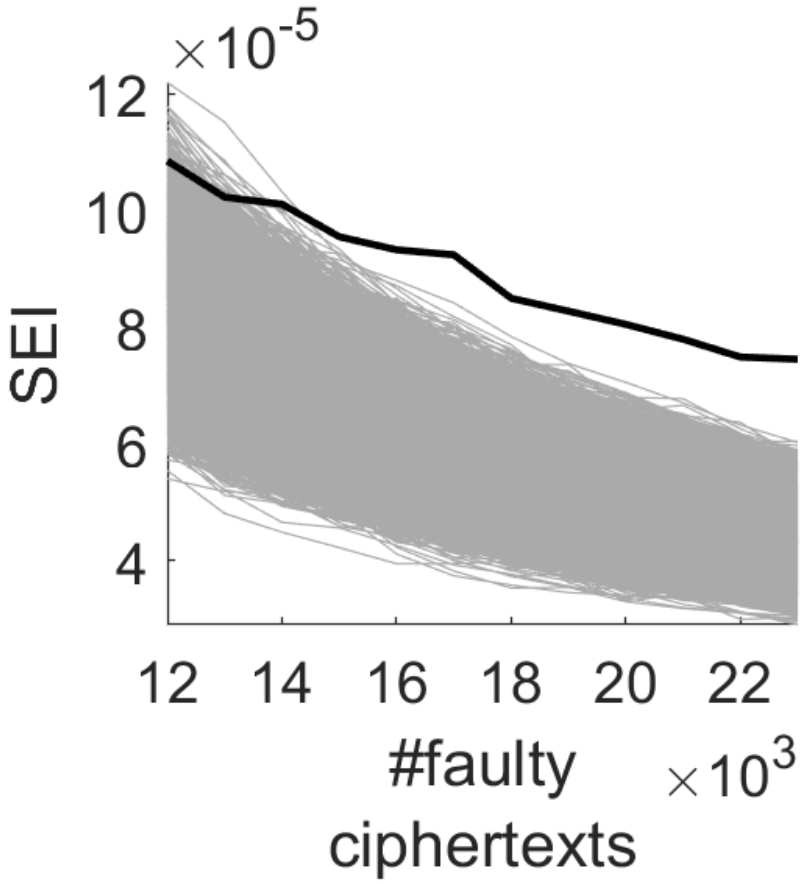
EXPERIMENTAL RESULTS FOR AES

- **Question 1: How many ciphertexts are needed for a succesful analysis?**

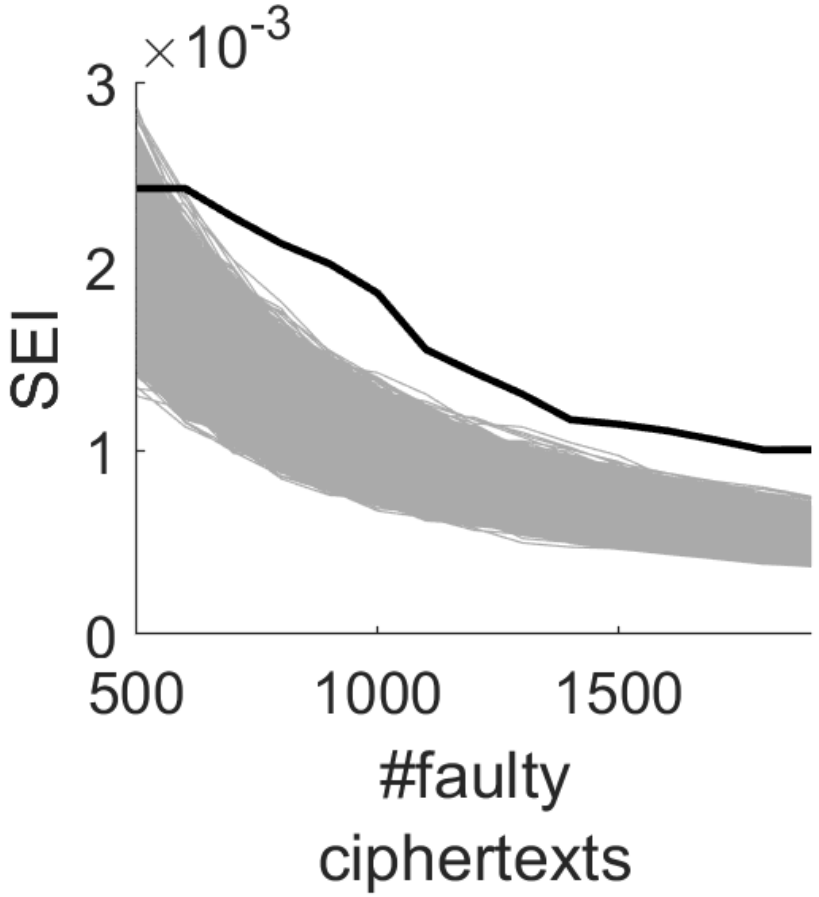
- **Question 2: How „faulty“ can the Sbox be?**

EXPERIMENTAL RESULTS FOR AES

single byte fault

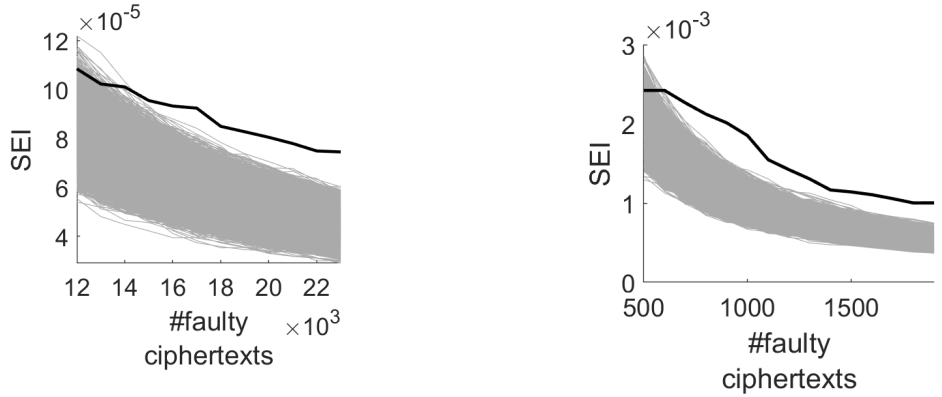


32 faulty bytes



EXPERIMENTAL RESULTS FOR AES

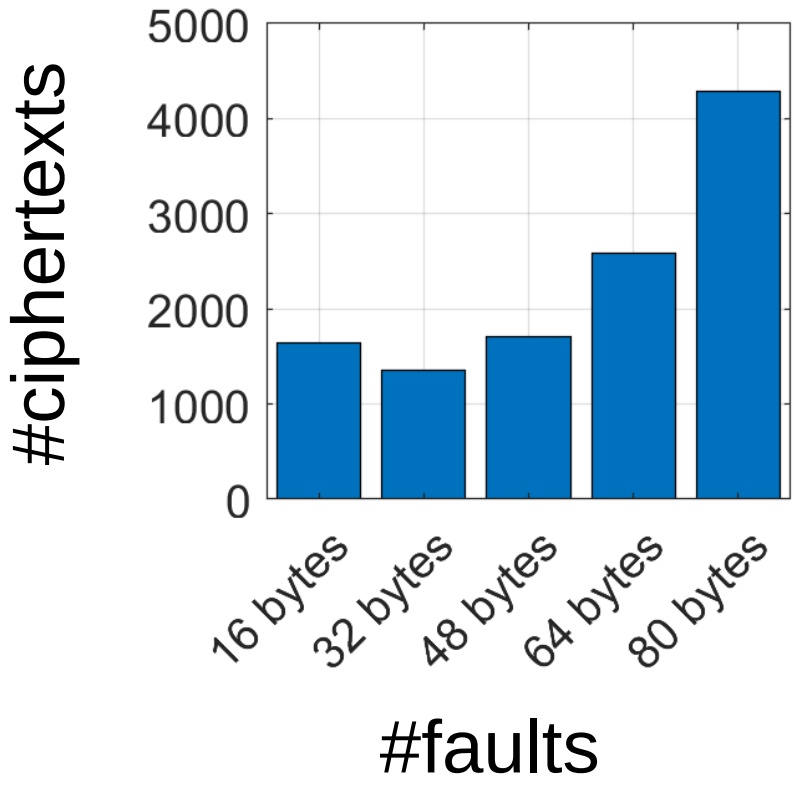
- **Question 1: How many ciphertexts are needed for a succesful analysis?**



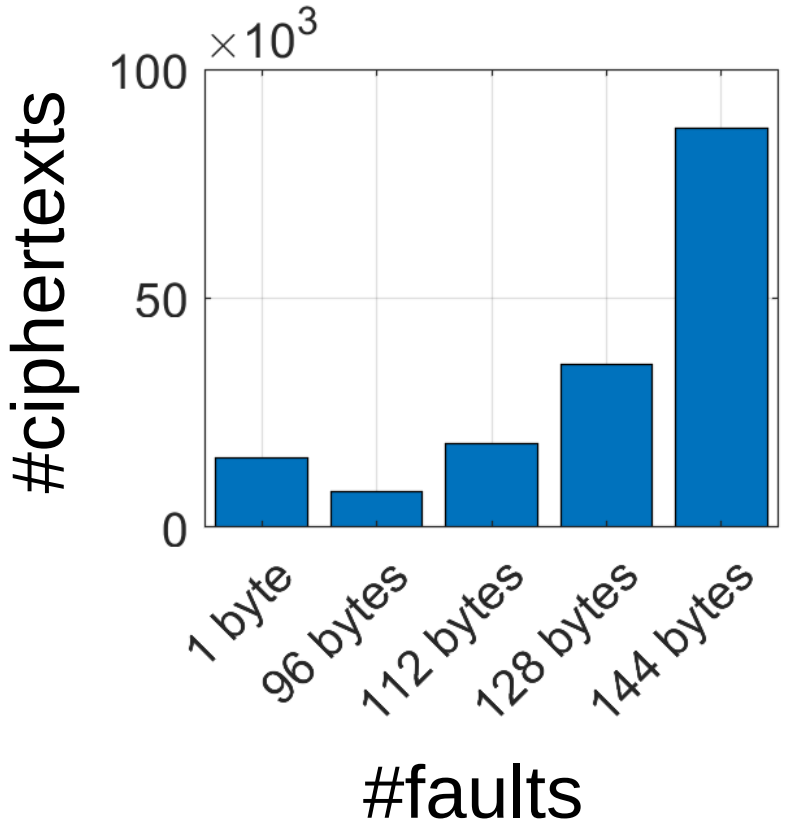
- **Question 2: How „faulty“ can the Sbox be?**

EXPERIMENTAL RESULTS FOR AES

single byte fault

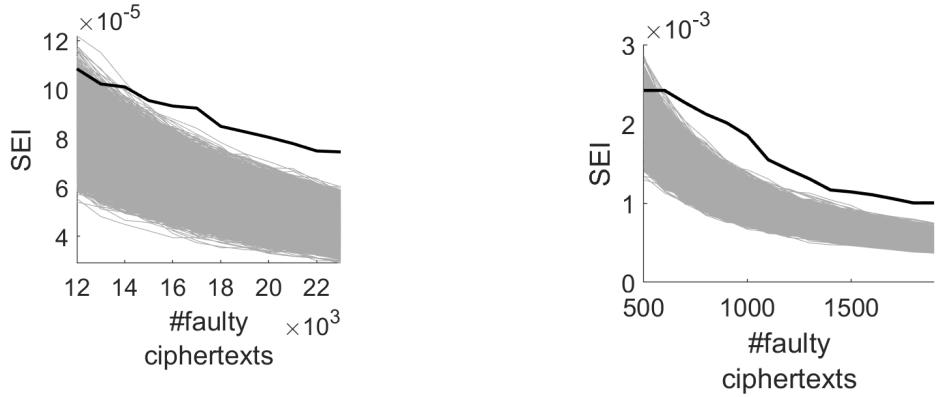


32 faulty bytes

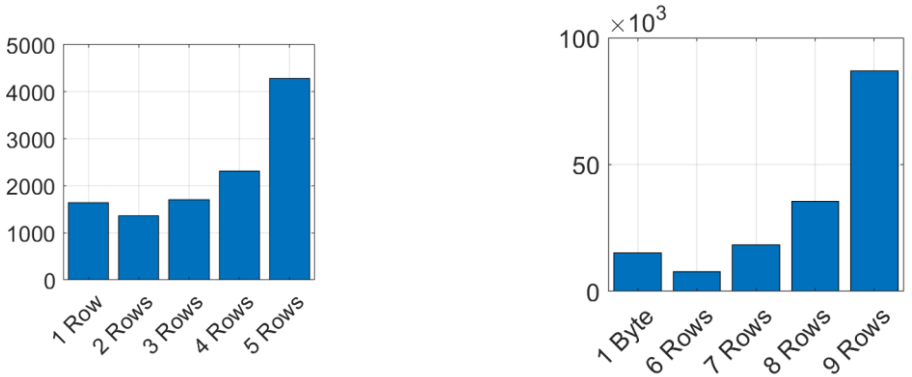


EXPERIMENTAL RESULTS FOR AES

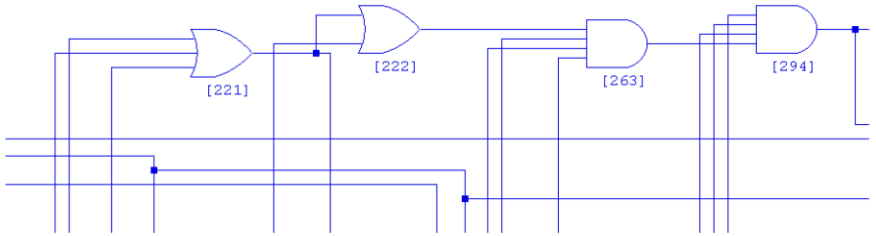
- **Question 1: How many ciphertexts are needed for a succesful analysis?**



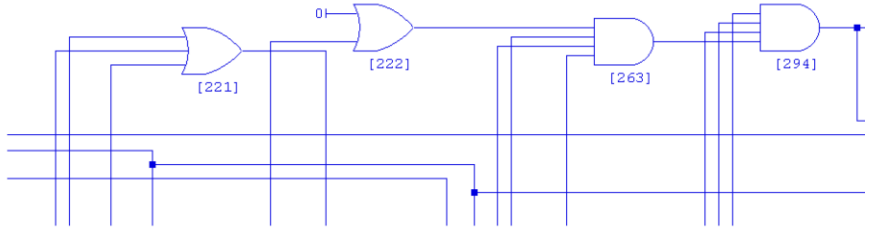
- **Question 2: How „faulty“ can the Sbox be?**



FAULTY SBOX THROUGH RE-WIRING



(a) Original wiring at gate 222.

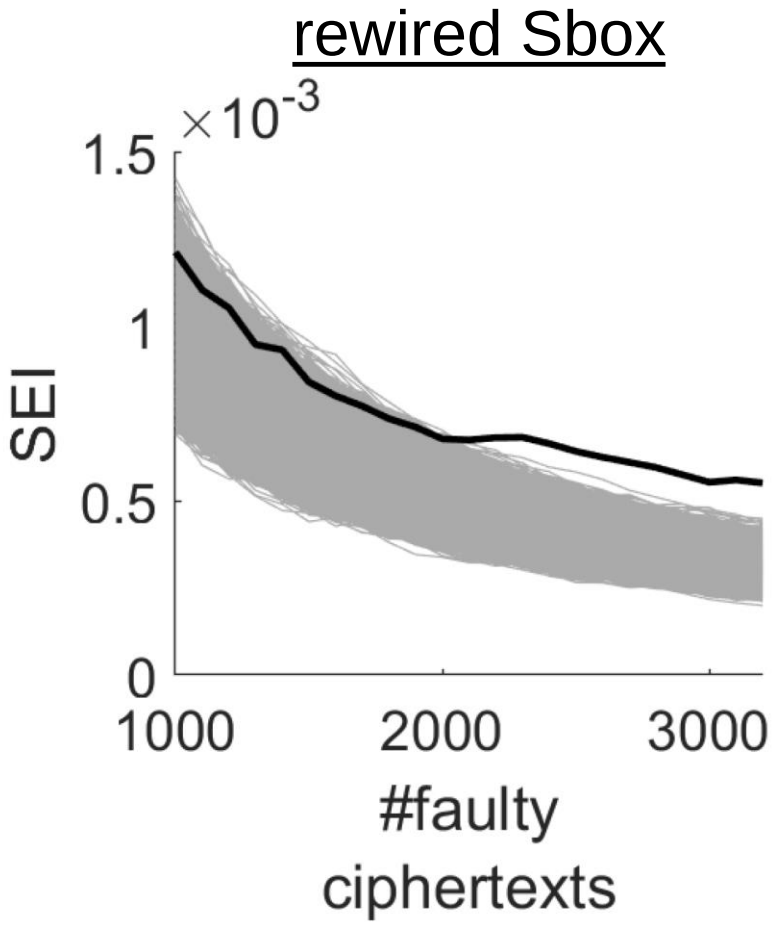


(b) First input of gate 222 fixed to 0.

resulting Sbox

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	B7	26	36	3F	F7	CC	34	A5	E5	3D	65	FC	25	15
3	04	C7	27	3F	1C	96	05	1E	07	76	84	66	EF	27	B6	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	0	ED	20	FC	B1	5D	6A	CB	BE	39	4A	4C	58	CF
6	D4	EF	8E	FF	EF	4D	17	85	45	35	86	7F	4C	3C	9F	8C
7	9D	AF	04	8F	FE	9D	04	F5	BC	B5	9E	2D	1C	FF	F7	D6
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E4	06	3E	0E	65	06	24	5C	E6	C7	AC	66	E5	95	E4	75
B	E7	CC	37	6D	8D	D5	4E	5D	6C	56	F4	6E	65	7E	AE	0C
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	8E
E	AD	AC	AC	ED	6D	5D	8E	94	D7	1E	87	ED	CE	55	2C	DF
F	8C	AD	8D	0D	BF	E6	A6	4C	65	9D	2D	0F	D4	54	BF	16

FAULTY SBOX THROUGH RE-WIRING



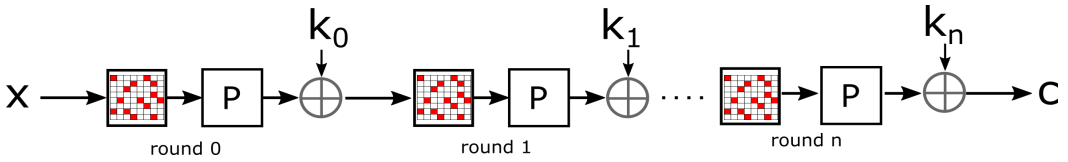
COMPARISON WITH CLASSICAL PFA

#faults	#ciphertexts			complexity	
	w/o MLE	MLE	Our work	Zhang et.al.	Our work
1	2273	1641	15650	0	2^{50}
2	Ca. 2000	n/a	7775	2^{16}	2^{50}
8	Ca. 2000	n/a	2008	2^{50}	2^{50}
16	Ca. 2000	n/a	1643	2^{64}	2^{50}

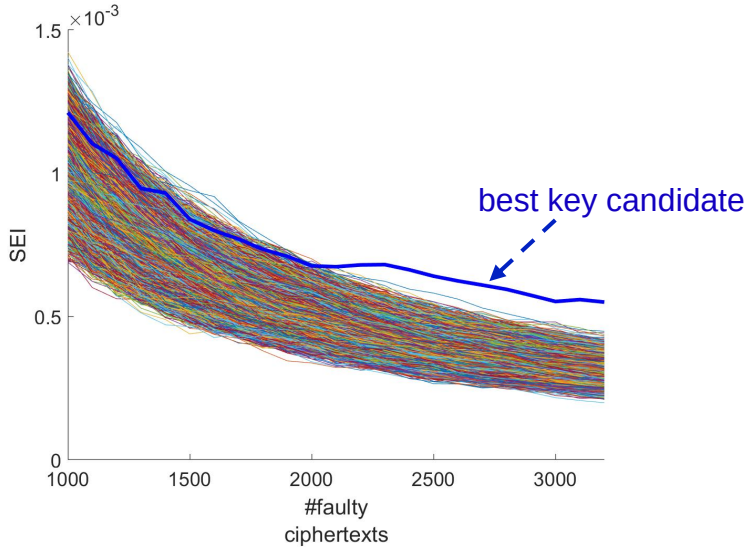
CONCLUSION

SUMMARY

- **We presented SPFA, a combination of SFA and PFA**
  - ✓ **SPFA allows for a more relaxed fault model wrt to PFA**
  - ✓ **SPFA allows multiple random faults, e.g., faults on gate level or bitstream manipulations**



- **We verified our attack on the block ciphers LED and AES**



- **Future work:**
  - ➢ SPFA and ineffective faults?
  - ➢ SPFA against other Sbox designs

REFERENCES

[1] Fan Zhang, Xiaoxuan Lou, Xinjie Zhao, Shivam Bhasin, Wei He, Ruyi Ding, Samiya Qureshi, Kui Ren: Persistent Fault Analysis on Block Ciphers. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2018(3): 150-172 (2018)

[2] Fan Zhang, Yiran Zhang, Huilong Jiang, Xiang Zhu, Shivam Bhasin, Xinjie Zhao, Zhe Liu, Dawu Gu, Kui Ren: Persistent Fault Attack in Practice. IACR Trans. CHES 2020(2): 172-195 (2020)

[3] Michael Gruber, Matthias Probst, Michael Tempelmeier: Persistent Fault Analysis of OCB, DEOXYS and COLM. FDTC 2019: 17-24

[4] Thomas Fuhr, Éliane Jaulmes, Victor Lomné, Adrian Thillard: Fault Attacks on AES with Faulty Ciphertexts Only. FDTC 2013: 108-118

[5] Christoph Dobraunig, Maria Eichlseder, Thomas Korak, Victor Lomné, Florian Mendel: Statistical Fault Attacks on Nonce-Based Authenticated Encryption Schemes. ASIACRYPT (1) 2016: 369-395

FDTC 2020
Fault Diagnosis and Tolerance in Cryptography

RUHR UNIVERSITÄT BOCHUM

MAX PLANCK INSTITUTE FOR SECURITY AND PRIVACY

THANK YOU FOR YOUR ATTENTION!

SPFA – SFA on Multiple Persistent Faults

Authors: Susanne Engels[1,2] , Falk Schellenberg[2] , Christof Paar[2]

[1] Ruhr University Bochum, Germany

[2] Max Planck Institute for Security and Privacy, Bochum, Germany