PRACTICAL IMPROVEMENTS TO STATISTICAL INEFFECTIVE FAULT ATTACKS

riscure

driving your security forward

Barış Ege[1]
Bob Swinkels[1]
Dilara Toprakhisar[2]
Praveen Kumar Vadnala[1]

[1]Riscure B.V., Delft, The Netherlands
lastname@riscure.com

[2]COSIC, KU Leuven, Leuven, Belgium,
dilara.toprakhisar@esat.kuleuven.be

April 9, 2024

OUTLOOK

We present methods for Statistical Ineffective Fault Attacks that...

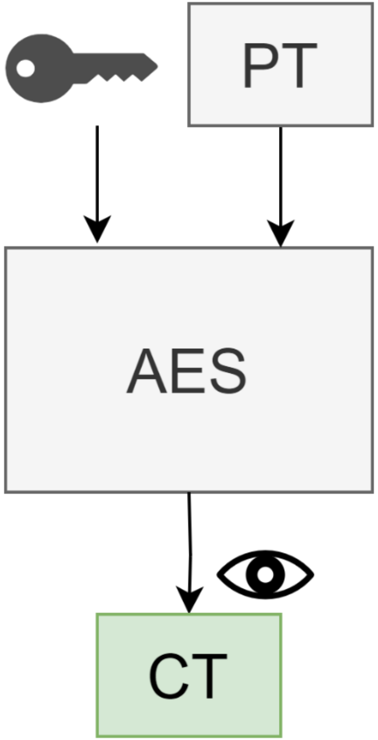
- Improve the effectiveness of SIFA on AES in the presence of jitter
  - Defy clock randomization countermeasures
- Facilitate white-box analysis on AES
  - Chosen plaintext attack significantly reduces the brute force space
  - Apply analysis on 4 columns simultaneously

FAULT ATTACKS

Workings

- With access to a device:
  - Set plaintexts
  - Observe ciphertexts
  - Cause faulty outputs at specific locations
  - Observe faulty outputs

- What can we do with this?
  - Perform DFA [1]

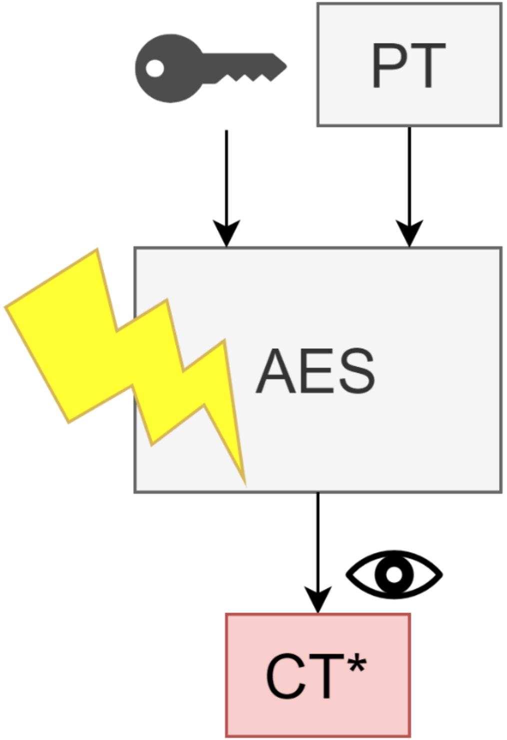


FAULT ATTACKS

Workings

- With access to a device:
  - Set plaintexts
  - Observe ciphertexts
  - Cause faulty outputs at specific locations
  - Observe faulty outputs

- What can we do with this?
  - Perform DFA [1]

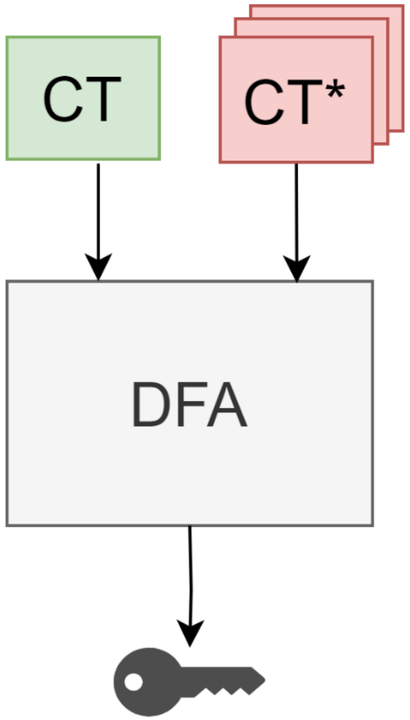


FAULT ATTACKS

Workings

- With access to a device:
  - Set plaintexts
  - Observe ciphertexts
  - Cause faulty outputs at specific locations
  - Observe faulty outputs

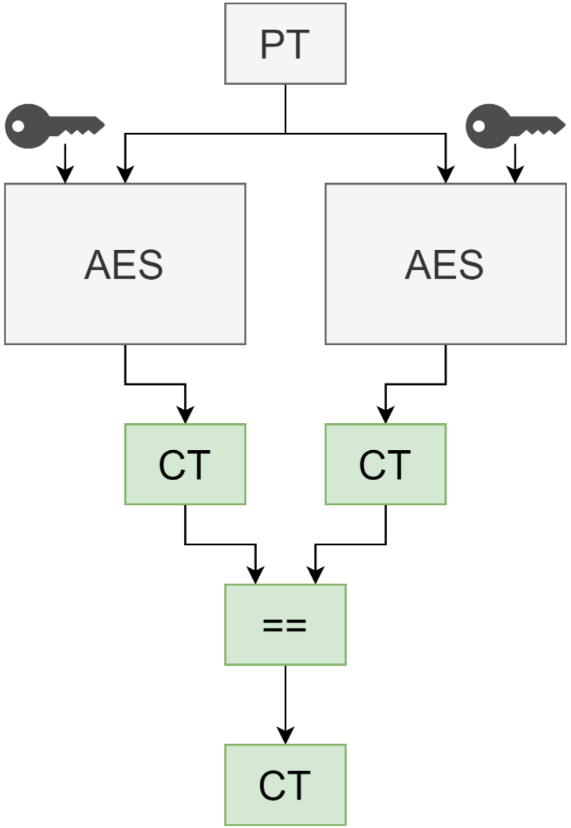
- What can we do with this?
  - Perform DFA [1]



FAULT ATTACKS

Countermeasures

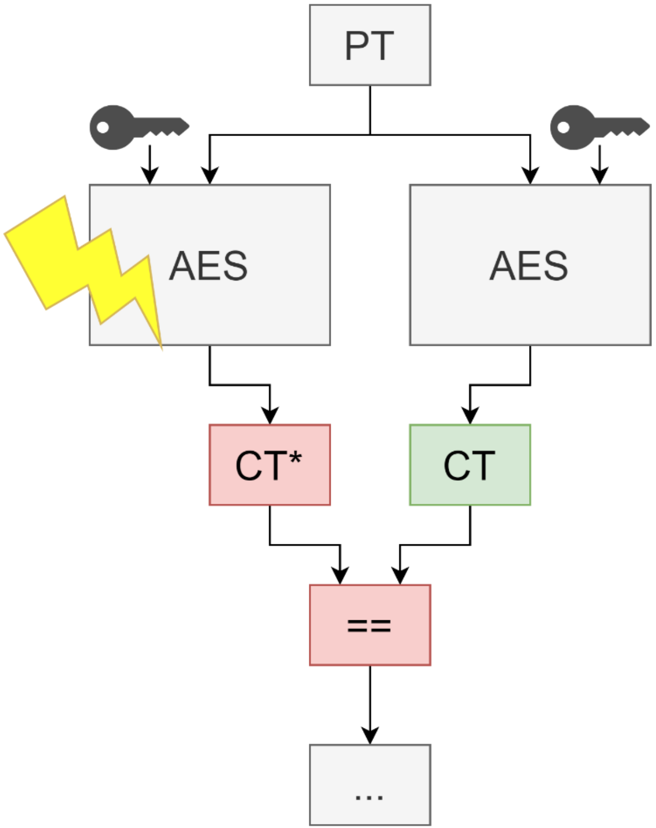
- Redundancy Countermeasure
  - Fault detected == no ciphertext
- Infection
  - Faults are amplified therefore ciphertext is not related to the key anymore
  - Key recovery using DFA not possible



FAULT ATTACKS

Countermeasures

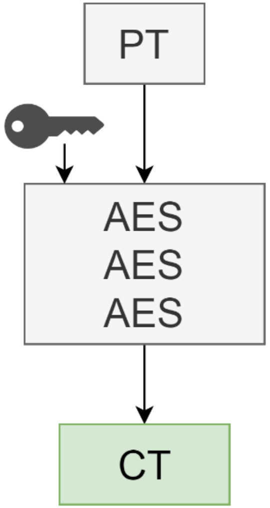
- Redundancy Countermeasure
  - Fault detected == no ciphertext
- Infection
  - Faults are amplified therefore ciphertext is not related to the key anymore
  - Key recovery using DFA not possible



FAULT ATTACKS

Countermeasures

- Redundancy Countermeasure
  - Fault detected == no ciphertext
- Infection
  - Faults are amplified therefore ciphertext is not related to the key anymore
  - Key recovery using DFA not possible

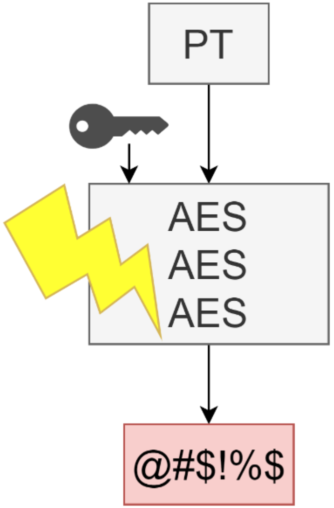


FAULT ATTACKS

Countermeasures

- Redundancy Countermeasure
  - Fault detected == no ciphertext

- Infection
  - Faults are amplified therefore ciphertext is not related to the key anymore
  - Key recovery using DFA not possible



FAULT ATTACKS

Attacking in the Presence of Countermeasures

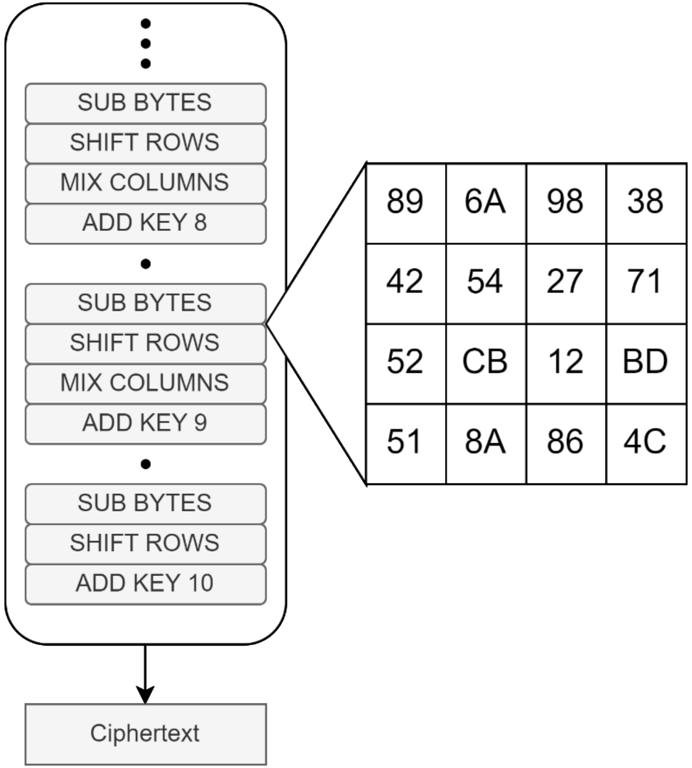
- Ineffective Fault Attacks (IFA) by Clavier et al. [2]
  - Exploits only correct ciphertexts
  - Requires precise faults with known effect
- Statistical Ineffective Fault Attacks (SIFA) by Dobraunig et al. [3]
  - Combines IFA with Statistical Fault Analysis (SFA) by Fuhr et al. [4]
  - Exploits only correct ciphertexts
  - Any fault, even if its effect is unknown
  - Analysis takes long because of 2^{32} brute force space

SIFA ON AES

Acquisition phase

For multiple encryptions on AES…

- Intermediate bytes are random uniformly distributed
- Fault between last two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Works the same for ineffective faults
  - The target still outputs the expected cipher text after the fault is injected
  - Attacker gets "access to a subset of the ciphertexts"

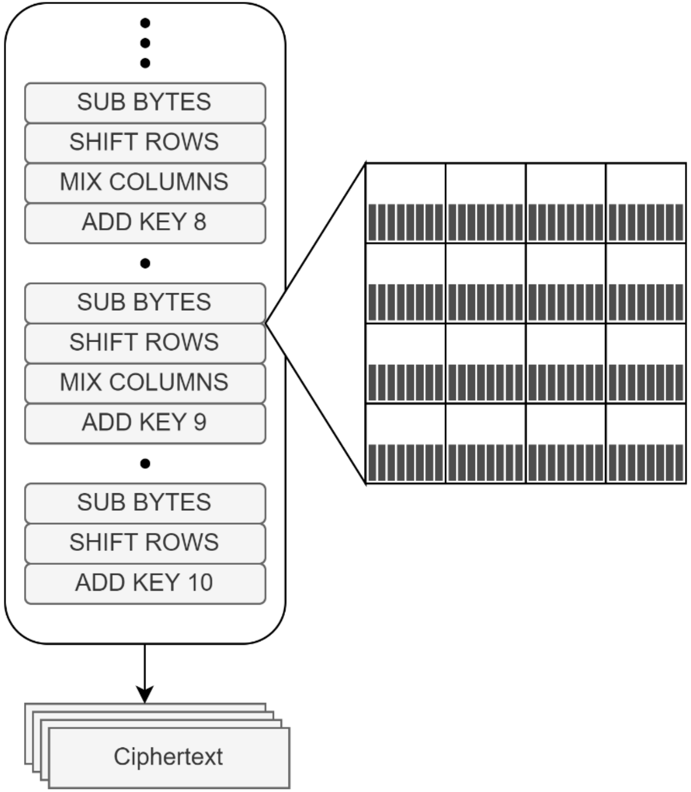


SIFA ON AES

Acquisition phase

For multiple encryptions on AES...

- Intermediate bytes are random uniformly distributed
- Fault between last two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Works the same for ineffective faults
  - The target still outputs the expected cipher text after the fault is injected
  - Attacker gets "access to a subset of the ciphertexts"

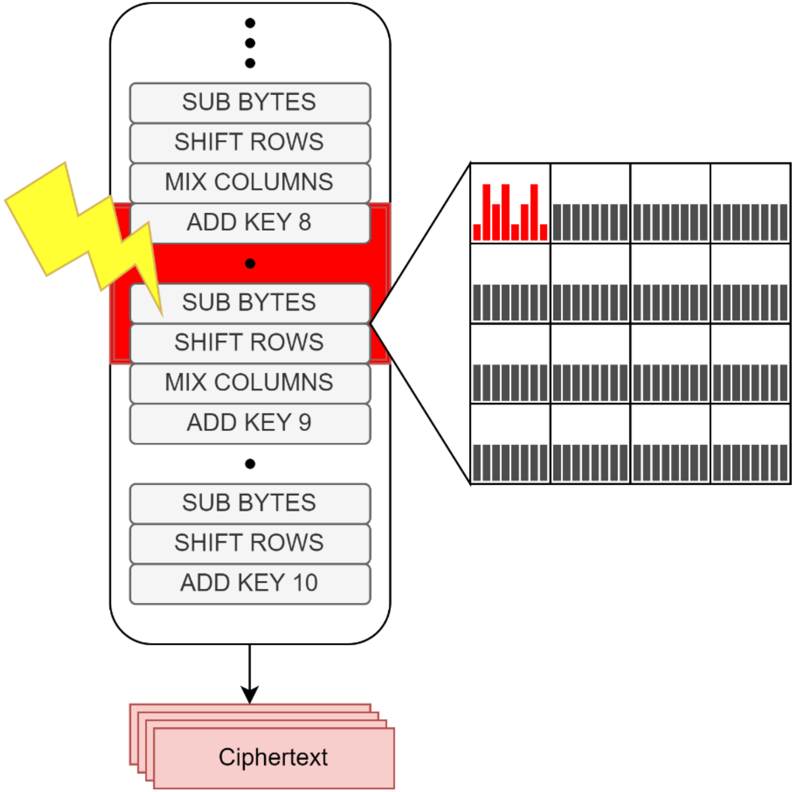


SIFA ON AES

Acquisition phase

For multiple encryptions on AES…

- Intermediate bytes are random uniformly distributed
- Fault between last two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Works the same for ineffective faults
  - The target still outputs the expected cipher text after the fault is injected
  - Attacker gets "access to a subset of the ciphertexts"

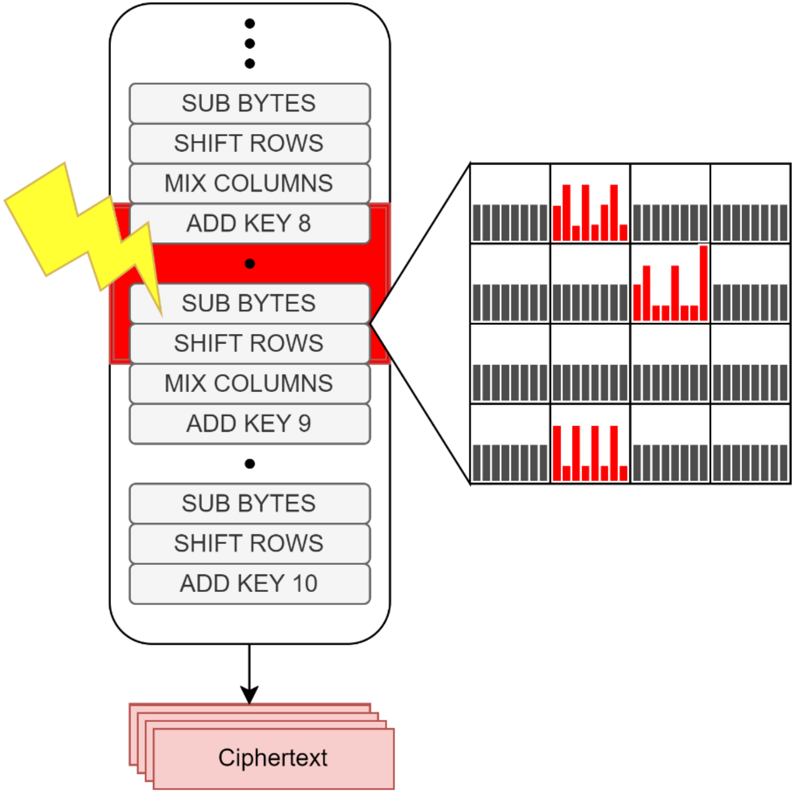


SIFA ON AES

Acquisition phase

For multiple encryptions on AES...

- Intermediate bytes are random uniformly distributed
- Fault between last two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Works the same for ineffective faults
  - The target still outputs the expected cipher text after the fault is injected
  - Attacker gets "access to a subset of the ciphertexts"



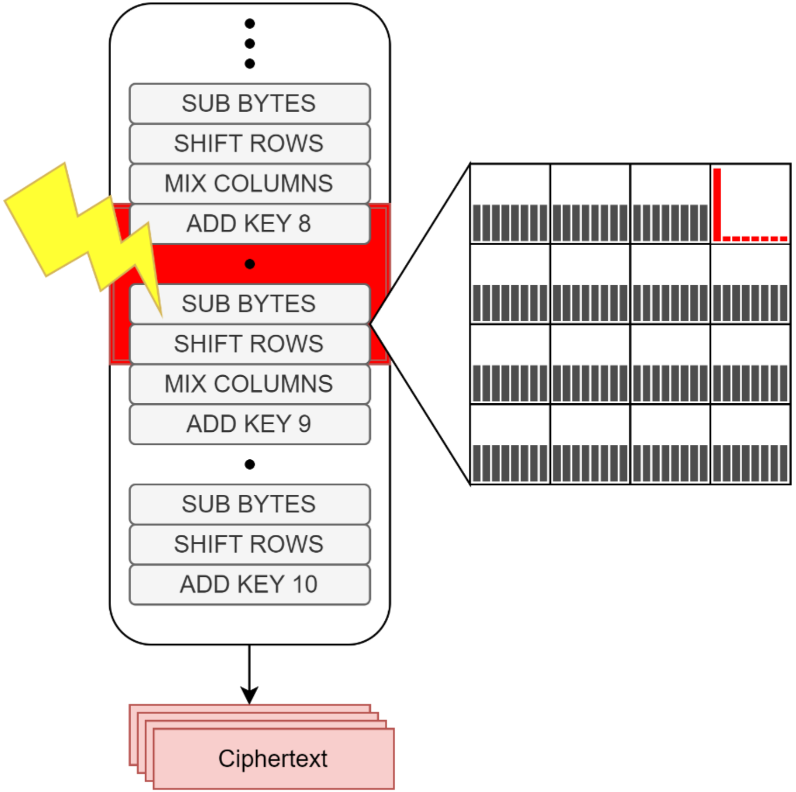
Public

SIFA ON AES

Acquisition phase

For multiple encryptions on AES...

- Intermediate bytes are random uniformly distributed
- Fault between last two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Works the same for ineffective faults
  - The target still outputs the expected cipher text after the fault is injected
  - Attacker gets "access to a subset of the ciphertexts"

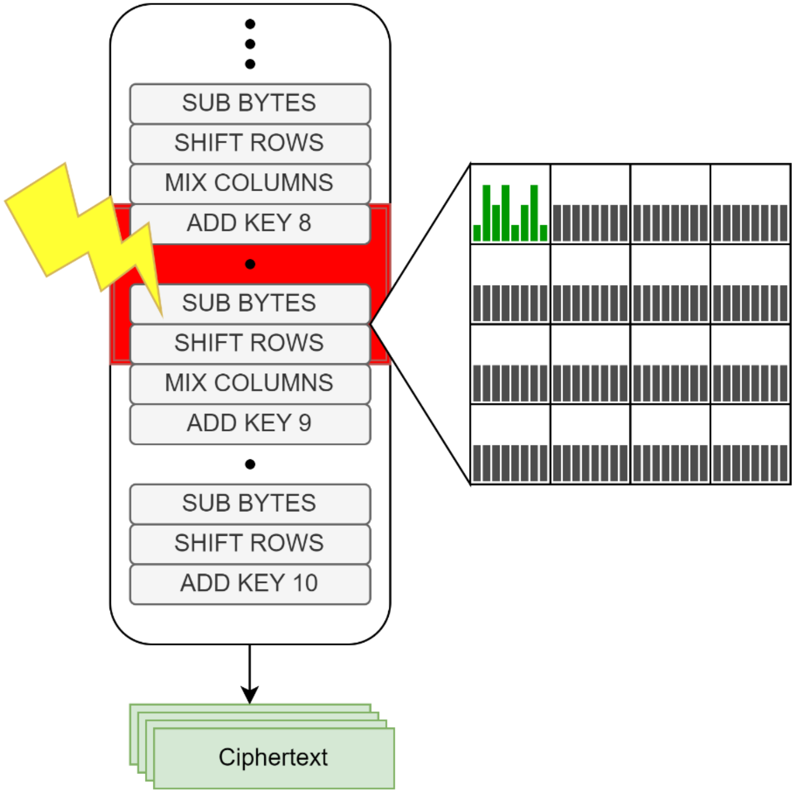


SIFA ON AES

Acquisition phase

For multiple encryptions on AES…

- Intermediate bytes are random uniformly distributed
- Fault between last two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Works the same for ineffective faults
  - The target still outputs the expected cipher text after the fault is injected
  - Attacker gets "access to a subset of the ciphertexts"



SIFA ON AES

Analysis phase

- Collect set of correct ciphertexts $\mathcal{C}_1 \ldots \mathcal{C}_n$ from faulted encryptions
- Guess 32-bit sub key $\mathcal{K}_{10}$ and calculate state $\mathcal{S}_i$ in round 9 ($\mathcal{K}_9$ is not needed):

$$\mathcal{S}_i = \mathrm{MC}^{-1} \circ \mathrm{SB}^{-1} \circ \mathrm{SR}^{-1} \circ (\mathcal{C}_i \oplus \mathcal{K}_{10})$$

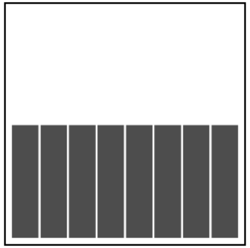
- Wrong key candidate: $\mathcal{S}_1 \ldots \mathcal{S}_n$ is uniformly distributed
- Correct key candidate: $\mathcal{S}_1 \ldots \mathcal{S}_n$ is non-uniformly distributed
- Measure uniformity using a statistical test and rank all 2^{32} possible sub keys
- The four key bytes of the highest ranking subkey are likely correct

SIFA ON AES

Analysis phase

- Collect set of correct ciphertexts $\mathcal{C}_1 \dots \mathcal{C}_n$ from faulted encryptions
- Guess 32-bit sub key $\mathcal{K}_{10}$ and calculate state $\mathcal{S}_i$ in round 9 ($\mathcal{K}_9$ is not needed):

$$\mathcal{S}_i = \mathrm{MC}^{-1} \circ \mathrm{SB}^{-1} \circ \mathrm{SR}^{-1} \circ (\mathcal{C}_i \oplus \mathcal{K}_{10})$$



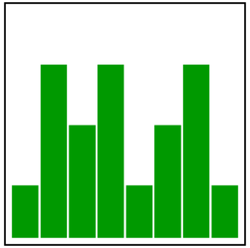
- Wrong key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is uniformly distributed
- Correct key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is non-uniformly distributed
- Measure uniformity using a statistical test and rank all 2^{32} possible sub keys
- The four key bytes of the highest ranking subkey are likely correct

SIFA ON AES

Analysis phase

- Collect set of correct ciphertexts $\mathcal{C}_1 \dots \mathcal{C}_n$ from faulted encryptions
- Guess 32-bit sub key $\mathcal{K}_{10}$ and calculate state $\mathcal{S}_i$ in round 9 ($\mathcal{K}_9$ is not needed):

$$\mathcal{S}_i = \mathrm{MC}^{-1} \circ \mathrm{SB}^{-1} \circ \mathrm{SR}^{-1} \circ (\mathcal{C}_i \oplus \mathcal{K}_{10})$$



- Wrong key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is uniformly distributed
- Correct key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is non-uniformly distributed
- Measure uniformity using a statistical test and rank all 2^{32} possible sub keys
- The four key bytes of the highest ranking subkey are likely correct

SIFA ON AES

Analysis phase

- Collect set of correct ciphertexts $\mathcal{C}_1 \dots \mathcal{C}_n$ from faulted encryptions
- Guess 32-bit sub key $\mathcal{K}_{10}$ and calculate state $\mathcal{S}_i$ in round 9 ($\mathcal{K}_9$ is not needed):

$$\mathcal{S}_i = \mathrm{MC}^{-1} \circ \mathrm{SB}^{-1} \circ \mathrm{SR}^{-1} \circ (\mathcal{C}_i \oplus \mathcal{K}_{10})$$

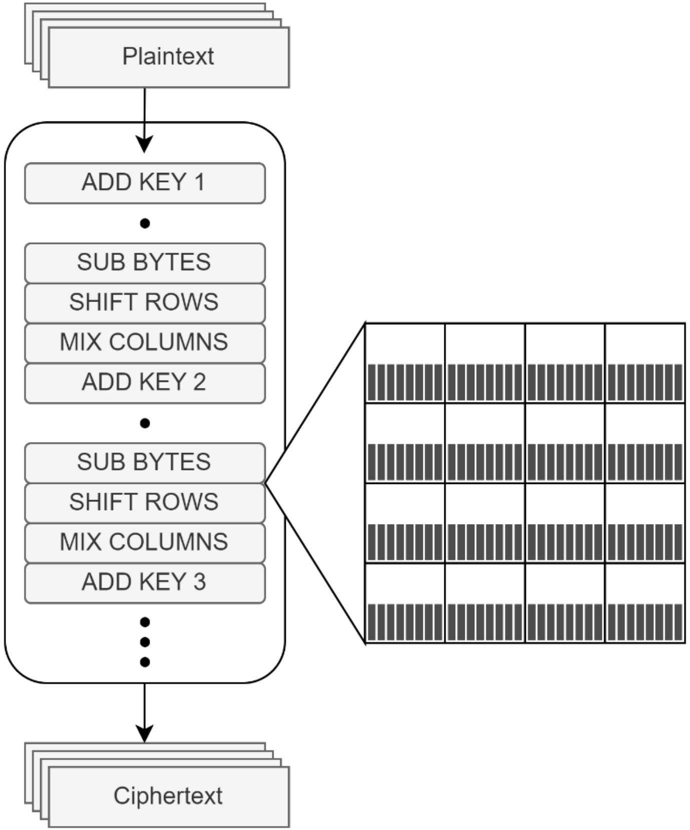
- Wrong key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is uniformly distributed
- Correct key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is non-uniformly distributed
- Measure uniformity using a statistical test and rank all 2^{32} possible sub keys
- The four key bytes of the highest ranking subkey are likely correct

CONTRIBUTION 1: SIFA FROM INPUT SIDE

Acquisition phase

For multiple encryptions of, uniformly distributed, random plaintexts with AES...

- Inject faults between the first two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Collect the subset of *plaintexts* from ineffective faults

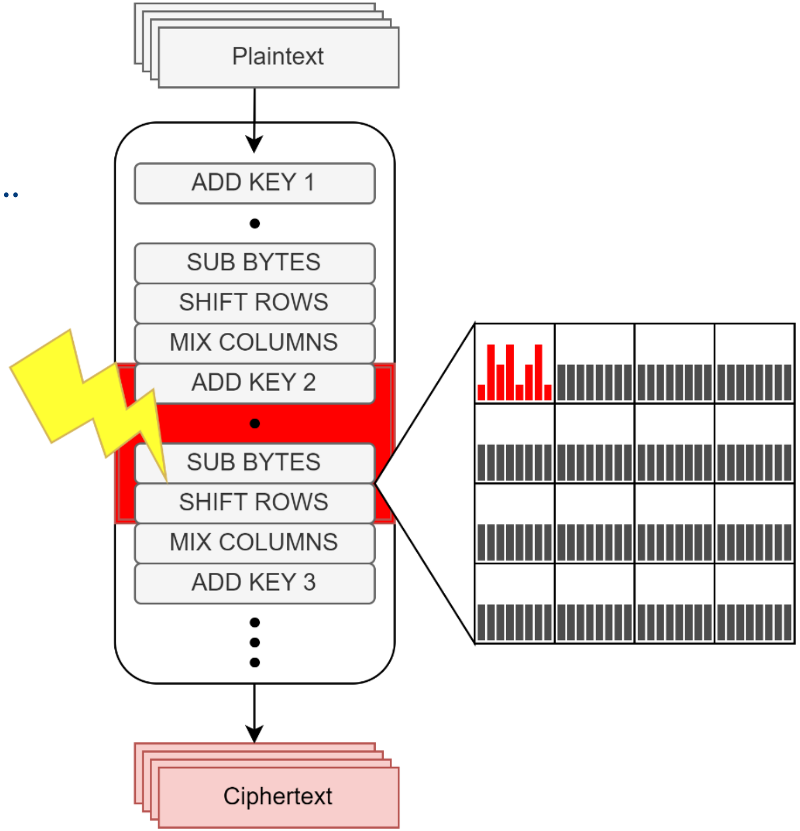


CONTRIBUTION 1: SIFA FROM INPUT SIDE

Acquisition phase

For multiple encryptions of, uniformly distributed, random plaintexts with AES…

- Inject faults between the first two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Collect the subset of *plaintexts* from ineffective faults

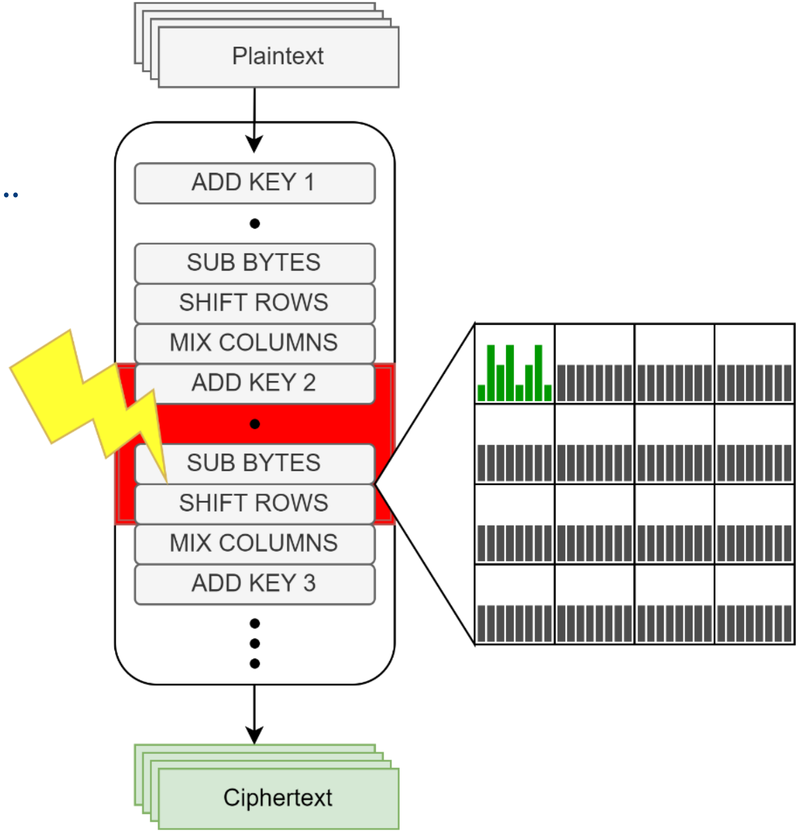


CONTRIBUTION 1: SIFA FROM INPUT SIDE

Acquisition phase

For multiple encryptions of, uniformly distributed, random plaintexts with AES...

- Inject faults between the first two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Collect the subset of *plaintexts* from ineffective faults



CONTRIBUTION 1: SIFA FROM INPUT SIDE

Analysis phase

- Collect set of plaintexts $\mathcal{P}_1 \dots \mathcal{P}_n$ from faulted encryptions corresponding to ineffective faults
- Guess 32-bit sub key $\mathcal{K}_1$ and calculate state $\mathcal{S}_i$ in round 2 ($\mathcal{K}_2$ is not needed):

$$\mathcal{S}_i = (\mathcal{P}_i \oplus \mathcal{K}_1) \circ \mathrm{SB} \circ \mathrm{SR} \circ \mathrm{MC}$$

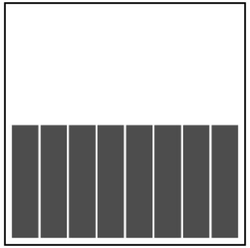
- Wrong key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is uniformly distributed
- Correct key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is non-uniformly distributed
- Measure uniformity using a statistical test and rank all 2^{32} possible sub keys
- The four key bytes of the highest ranking subkey are likely correct

CONTRIBUTION 1: SIFA FROM INPUT SIDE

Analysis phase

- Collect set of plaintexts $\mathcal{P}_1 \dots \mathcal{P}_n$ from faulted encryptions corresponding to ineffective faults
- Guess 32-bit sub key $\mathcal{K}_1$ and calculate state $\mathcal{S}_i$ in round 2 ($\mathcal{K}_2$ is not needed):

$$\mathcal{S}_i = (\mathcal{P}_i \oplus \mathcal{K}_1) \circ \mathrm{SB} \circ \mathrm{SR} \circ \mathrm{MC}$$



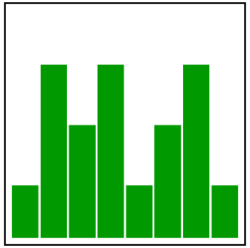
- Wrong key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is uniformly distributed
- Correct key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is non-uniformly distributed
- Measure uniformity using a statistical test and rank all 2^{32} possible sub keys
- The four key bytes of the highest ranking subkey are likely correct

CONTRIBUTION 1: SIFA FROM INPUT SIDE

Analysis phase

- Collect set of plaintexts $\mathcal{P}_1 \ldots \mathcal{P}_n$ from faulted encryptions corresponding to ineffective faults
- Guess 32-bit sub key $\mathcal{K}_1$ and calculate state $\mathcal{S}_i$ in round 2 ($\mathcal{K}_2$ is not needed):

$$\mathcal{S}_i = (\mathcal{P}_i \oplus \mathcal{K}_1) \circ \mathrm{SB} \circ \mathrm{SR} \circ \mathrm{MC}$$



- Wrong key candidate: $\mathcal{S}_1 \ldots \mathcal{S}_n$ is uniformly distributed
- Correct key candidate: $\mathcal{S}_1 \ldots \mathcal{S}_n$ is non-uniformly distributed
- Measure uniformity using a statistical test and rank all 2^{32} possible sub keys
- The four key bytes of the highest ranking subkey are likely correct

CONTRIBUTION 1: SIFA FROM INPUT SIDE

Analysis phase

- Collect set of plaintexts $\mathcal{P}_1 \dots \mathcal{P}_n$ from faulted encryptions corresponding to ineffective faults
- Guess 32-bit sub key $\mathcal{K}_1$ and calculate state $\mathcal{S}_i$ in round 2 ($\mathcal{K}_2$ is not needed):

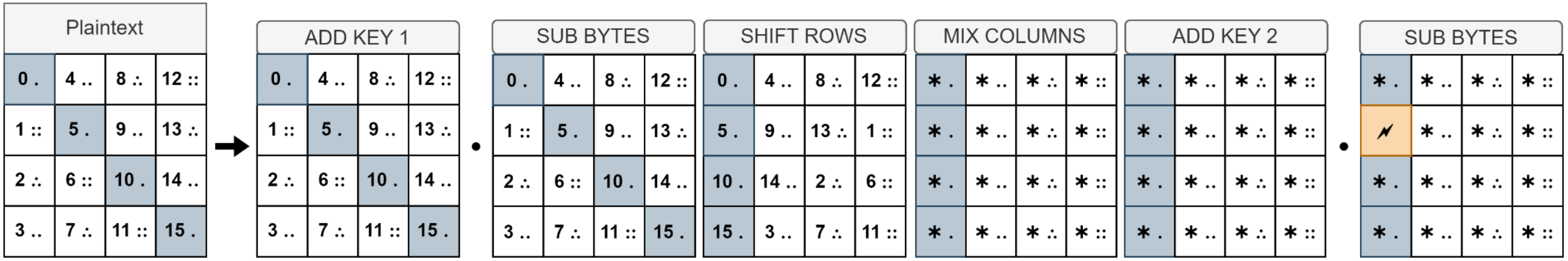
$$\mathcal{S}_i = (\mathcal{P}_i \oplus \mathcal{K}_1) \circ \mathrm{SB} \circ \mathrm{SR} \circ \mathrm{MC}$$

- Wrong key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is uniformly distributed
- Correct key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is non-uniformly distributed
- Measure uniformity using a statistical test and rank all 2^{32} possible sub keys
- The four key bytes of the highest ranking subkey are likely correct

Public

CONTRIBUTION 1: SIFA FROM INPUT SIDE

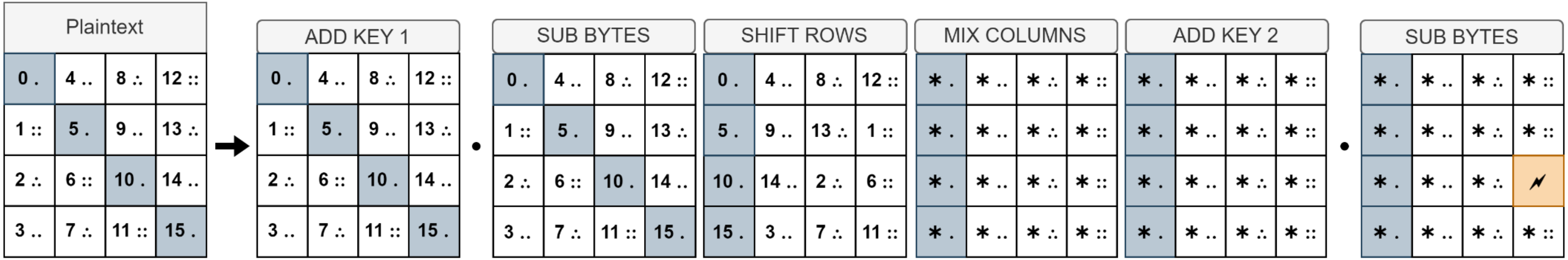
Propagation



- Each intermediate column corresponds to 4 input bytes
- No need repeat the analysis 4 times
- Can use Intel AES-NI for simultaneous calculation off all columns

CONTRIBUTION 1: SIFA FROM INPUT SIDE

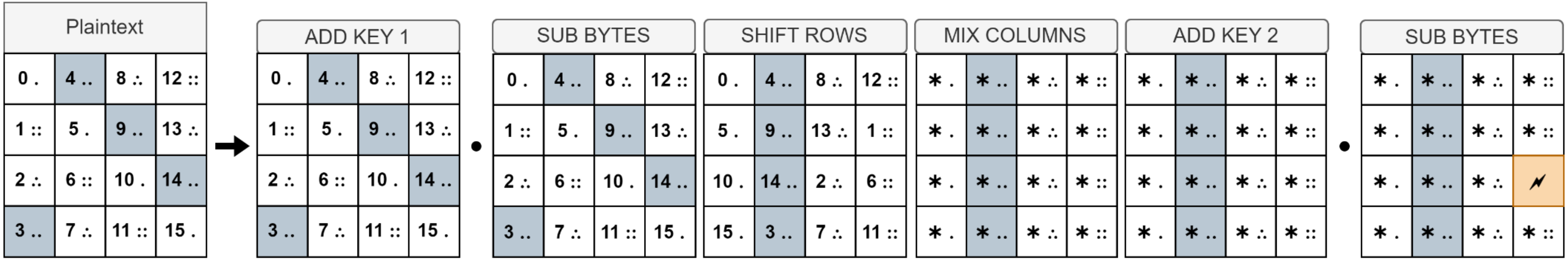
Propagation



- Each intermediate column corresponds to 4 input bytes
- No need repeat the analysis 4 times
- Can use Intel AES-NI for simultaneous calculation off all columns

CONTRIBUTION 1: SIFA FROM INPUT SIDE

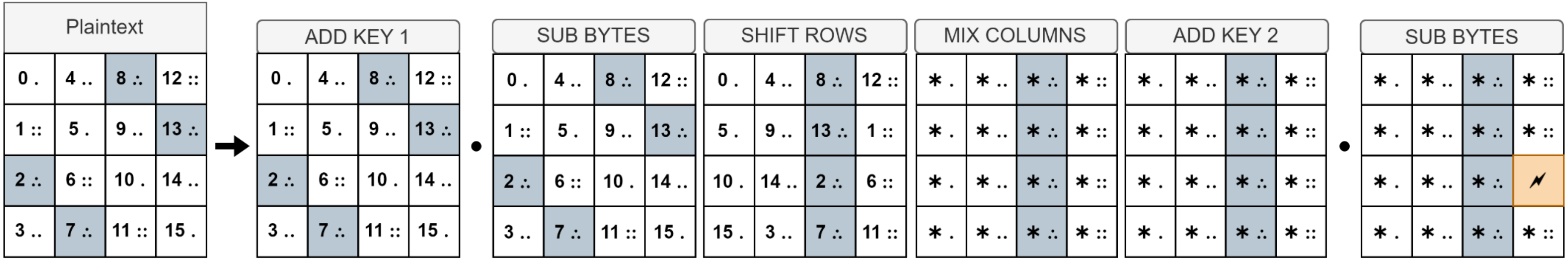
Propagation



- Each intermediate column corresponds to 4 input bytes
- No need repeat the analysis 4 times
- Can use Intel AES-NI for simultaneous calculation off all columns

CONTRIBUTION 1: SIFA FROM INPUT SIDE

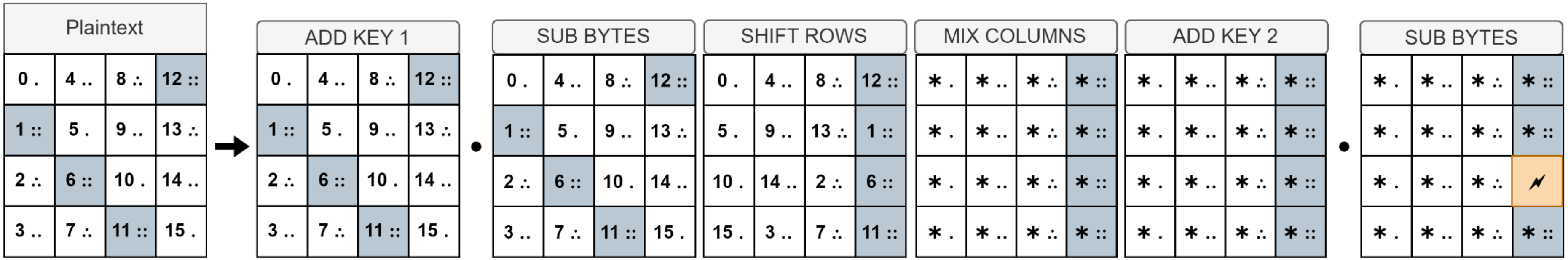
Propagation



- Each intermediate column corresponds to 4 input bytes
- No need repeat the analysis 4 times
- Can use Intel AES-NI for simultaneous calculation off all columns

CONTRIBUTION 1: SIFA FROM INPUT SIDE

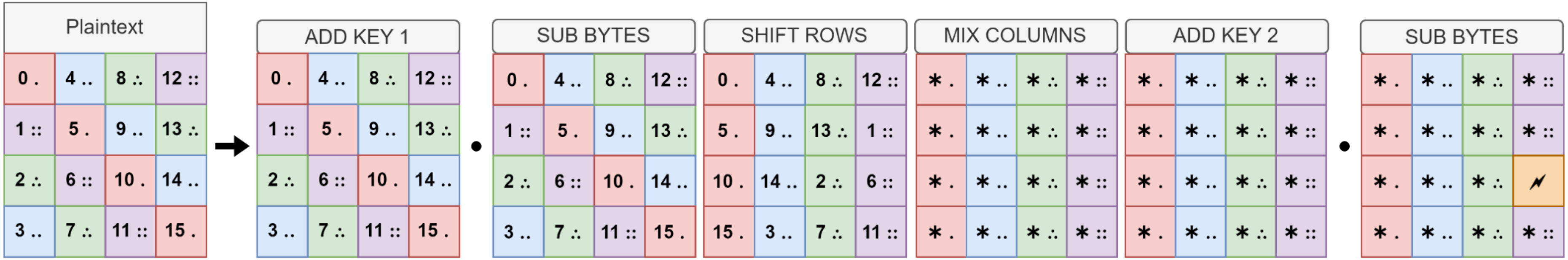
Propagation



- Each intermediate column corresponds to 4 input bytes
- No need repeat the analysis 4 times
- Can use Intel AES-NI for simultaneous calculation off all columns

CONTRIBUTION 1: SIFA FROM INPUT SIDE

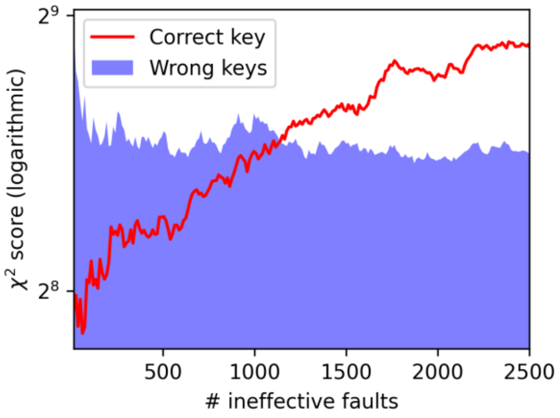
Propagation

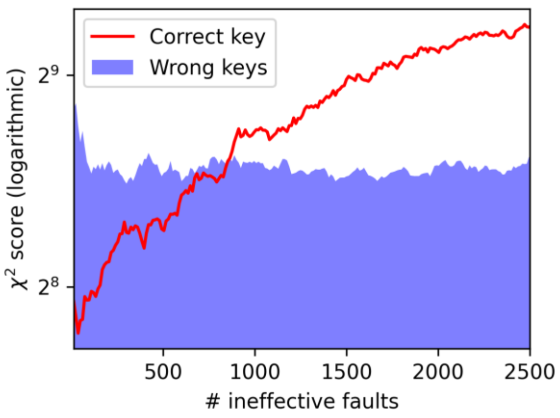


- Each intermediate column corresponds to 4 input bytes
- No need repeat the analysis 4 times
- Can use Intel AES-NI for simultaneous calculation off all columns

CONTRIBUTION 1: SIFA FROM INPUT SIDE

Practical results





- Voltage glitch on STM32F407IG M4
- 8-bit "textbook" software AES (Section 4.1 of [5])
- After ≈ 1150 ineffective faults

- Voltage glitch on STM32F407IG M4
- 32-bit t-table software AES implementation (Section 4.2 of [5])
- After ≈ 865 ineffective faults

CONTRIBUTION 1: SIFA FROM INPUT SIDE

Pros and Cons

- Known inputs, randomly distributed/ attacker-controlled inputs
- Attack needs to be repeated 3 times (+ 32-bit bruteforce) to retrieve the full key
- AES execution time can be non-constant
  - Can be modeled as an Irwin-Hall distribution
    - $n =$ number of rounds
    - Mean: $\mu = \frac{n}{2}$
    - Variance: $\sigma^2 = \frac{n}{12}$

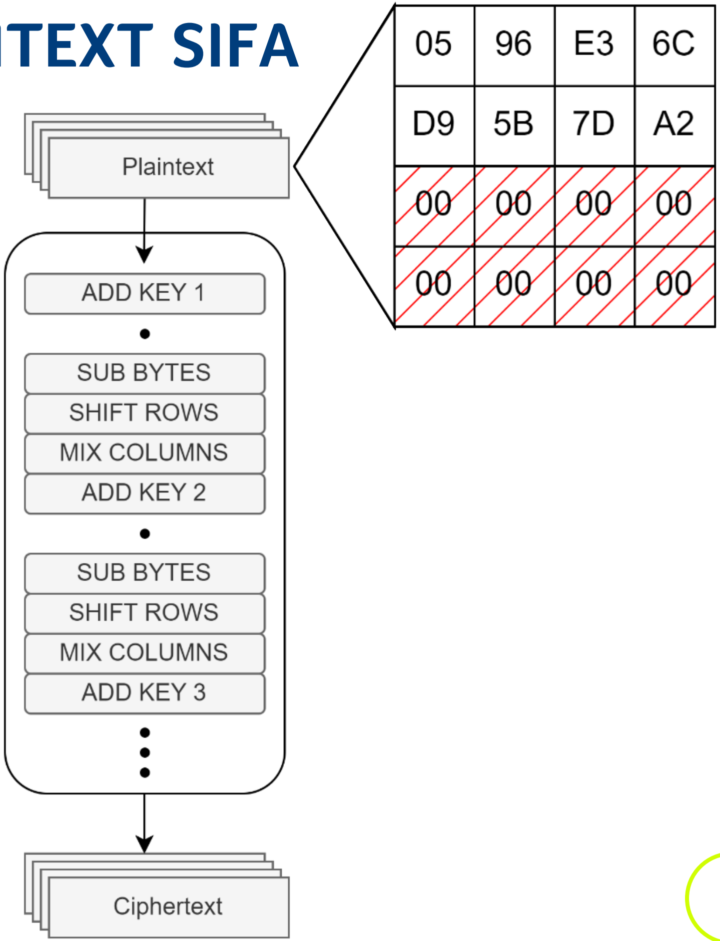
- Attacking in an earlier round → smaller error & more consistent fault model
- Great for blackbox analysis:
  Performs better than regular SIFA in the presence of (clock) jitter

CONTRIBUTION 2: CHOSEN PLAINTEXT SIFA

Acquisition phase

For multiple encryptions of, uniformly distributed, random plaintexts with AES…

- Special plaintexts are crafted where two of the four rows are set to a fixed value (e.g. zero)
- Inject faults between the *first* two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Collect the subset of *plaintexts* from ineffective faults

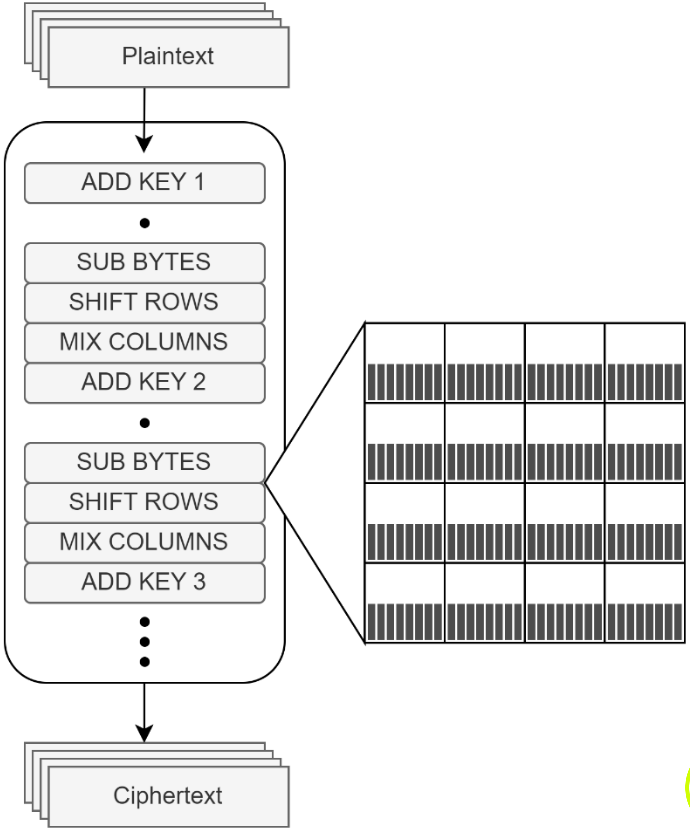


CONTRIBUTION 2: CHOSEN PLAINTEXT SIFA

Acquisition phase

For multiple encryptions of, uniformly distributed, random plaintexts with AES...

- Special plaintexts are crafted where two of the four rows are set to a fixed value (e.g. zero)
- Inject faults between the *first* two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Collect the subset of *plaintexts* from ineffective faults

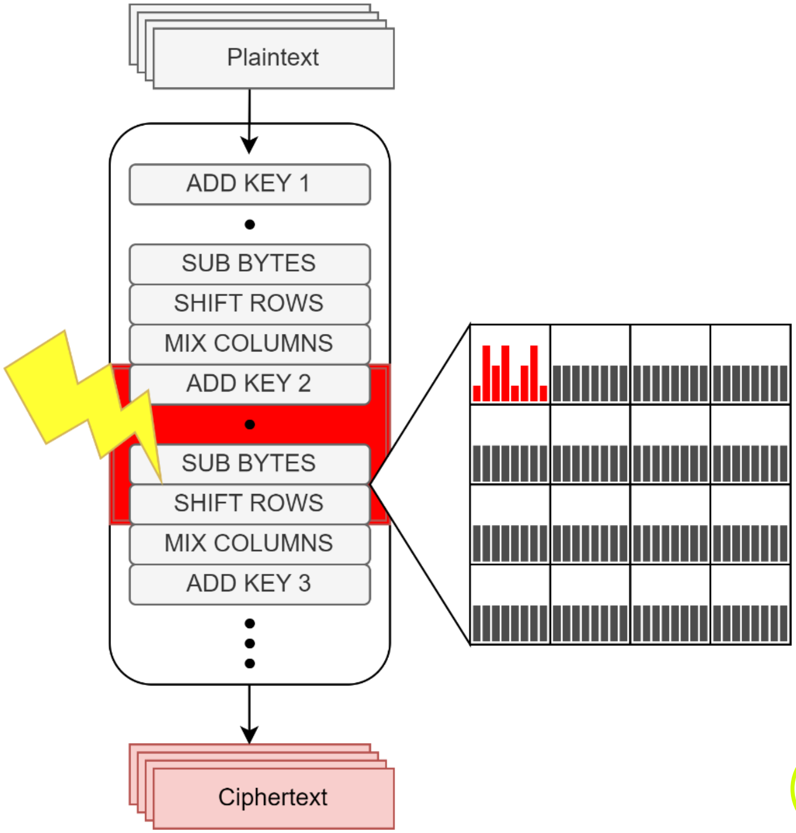


CONTRIBUTION 2: CHOSEN PLAINTEXT SIFA

Acquisition phase

For multiple encryptions of, uniformly distributed, random plaintexts with AES…

- Special plaintexts are crafted where two of the four rows are set to a fixed value (e.g. zero)
- Inject faults between the *first* two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Collect the subset of *plaintexts* from ineffective faults

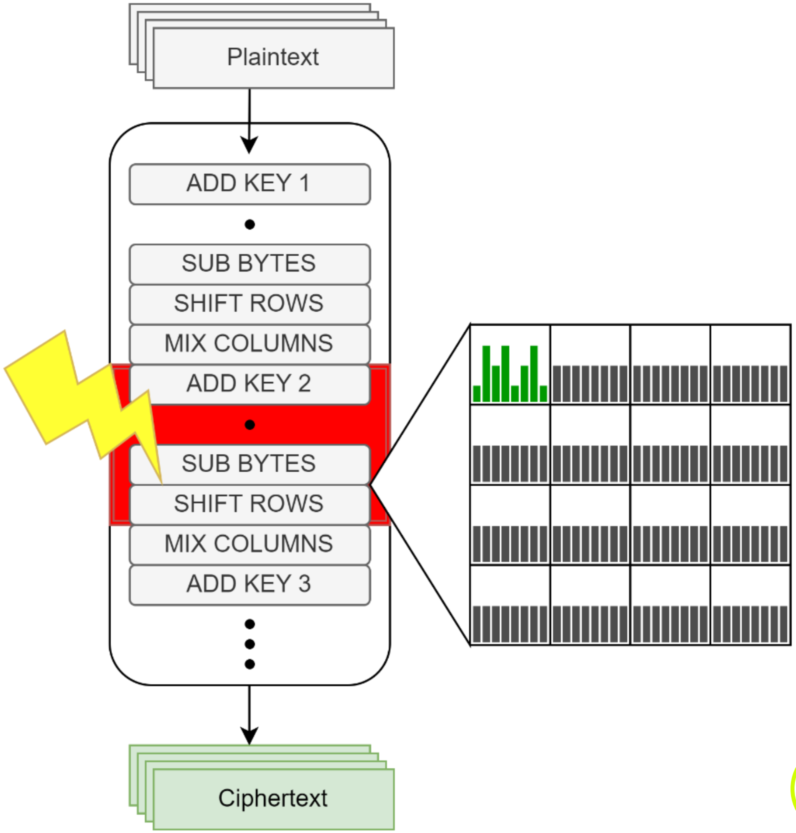


CONTRIBUTION 2: CHOSEN PLAINTEXT SIFA

Acquisition phase

For multiple encryptions of, uniformly distributed, random plaintexts with AES...

- Special plaintexts are crafted where two of the four rows are set to a fixed value (e.g. zero)
- Inject faults between the *first* two MixColumns operations
- Bias distribution of one or more intermediate bytes
- Collect the subset of *plaintexts* from ineffective faults



CONTRIBUTION 2: CHOSEN PLAINTEXT SIFA

Analysis phase

- Collect set of plaintexts $\mathcal{P}_1 \ldots \mathcal{P}_n$ from faulted encryptions corresponding to ineffective faults
- Guess 32-bit sub key $\mathcal{K}_1$ where the same two respective bytes are set to a fixed value as for the plaintext and calculate state $\mathcal{S}_i$ in round 2 ($\mathcal{K}_2$ is not needed):

$$\mathcal{S}_i = (\mathcal{P}_i \oplus \mathcal{K}_1) \circ \mathrm{SB} \circ \mathrm{SR} \circ \mathrm{MC}$$

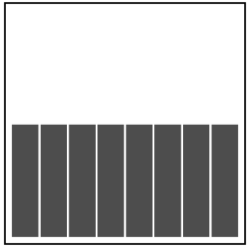
- Wrong key candidate: $\mathcal{S}_1 \ldots \mathcal{S}_n$ is uniformly distributed
- Correct key candidate: $\mathcal{S}_1 \ldots \mathcal{S}_n$ is non-uniformly distributed
- Measure uniformity using a statistical test and rank all 2^{16} possible sub keys
- The two non-fixed key bytes of the highest ranking subkey are likely correct
- Repeat the attack but with the opposite two rows set to zero to recover the other two key bytes

CONTRIBUTION 2: CHOSEN PLAINTEXT SIFA

Analysis phase

- Collect set of plaintexts $\mathcal{P}_1 \dots \mathcal{P}_n$ from faulted encryptions corresponding to ineffective faults
- Guess 32-bit sub key $\mathcal{K}_1$ where the same two respective bytes are set to a fixed value as for the plaintext and calculate state $\mathcal{S}_i$ in round 2 ($\mathcal{K}_2$ is not needed):

$$\mathcal{S}_i = (\mathcal{P}_i \oplus \mathcal{K}_1) \circ \mathrm{SB} \circ \mathrm{SR} \circ \mathrm{MC}$$



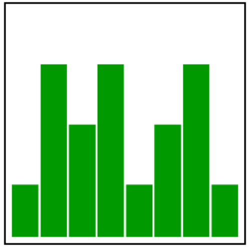
- Wrong key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is uniformly distributed
- Correct key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is non-uniformly distributed
- Measure uniformity using a statistical test and rank all 2^{16} possible sub keys
- The two non-fixed key bytes of the highest ranking subkey are likely correct
- Repeat the attack but with the opposite two rows set to zero to recover the other two key bytes

CONTRIBUTION 2: CHOSEN PLAINTEXT SIFA

Analysis phase

- Collect set of plaintexts $\mathcal{P}_1 \ldots \mathcal{P}_n$ from faulted encryptions corresponding to ineffective faults
- Guess 32-bit sub key $\mathcal{K}_1$ where the same two respective bytes are set to a fixed value as for the plaintext and calculate state $\mathcal{S}_i$ in round 2 ($\mathcal{K}_2$ is not needed):

$$\mathcal{S}_i = (\mathcal{P}_i \oplus \mathcal{K}_1) \circ \mathrm{SB} \circ \mathrm{SR} \circ \mathrm{MC}$$



- Wrong key candidate: $\mathcal{S}_1 \ldots \mathcal{S}_n$ is uniformly distributed
- Correct key candidate: $\mathcal{S}_1 \ldots \mathcal{S}_n$ is non-uniformly distributed
- Measure uniformity using a statistical test and rank all 2^{16} possible sub keys
- The two non-fixed key bytes of the highest ranking subkey are likely correct
- Repeat the attack but with the opposite two rows set to zero to recover the other two key bytes

CONTRIBUTION 2: CHOSEN PLAINTEXT SIFA

Analysis phase

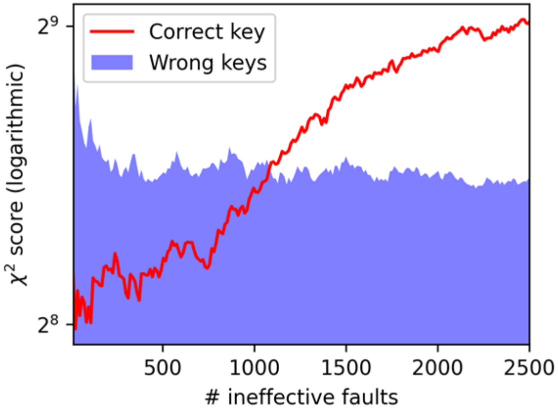
- Collect set of plaintexts $\mathcal{P}_1 \dots \mathcal{P}_n$ from faulted encryptions corresponding to ineffective faults
- Guess 32-bit sub key $\mathcal{K}_1$ where the same two respective bytes are set to a fixed value as for the plaintext and calculate state $\mathcal{S}_i$ in round 2 ($\mathcal{K}_2$ is not needed):

$$\mathcal{S}_i = (\mathcal{P}_i \oplus \mathcal{K}_1) \circ \mathrm{SB} \circ \mathrm{SR} \circ \mathrm{MC}$$

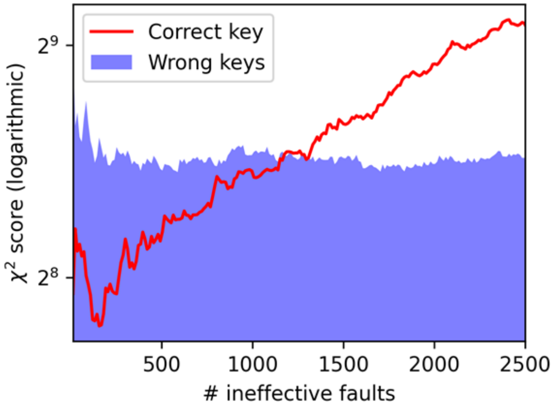
- Wrong key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is uniformly distributed
- Correct key candidate: $\mathcal{S}_1 \dots \mathcal{S}_n$ is non-uniformly distributed
- Measure uniformity using a statistical test and rank all 2^{16} possible sub keys
- The two non-fixed key bytes of the highest ranking subkey are likely correct
- Repeat the attack but with the opposite two rows set to zero to recover the other two key bytes

CONTRIBUTION 2: CHOSEN PLAINTEXT SIFA

Practical results



- Voltage glitch on STM32F407IG M4
- 8-bit "textbook" software AES (Section 4.1 of [5])
- After ≈ 1085 ineffective faults



- Voltage glitch on STM32F407IG M4
- 32-bit t-table software AES implementation (Section 4.2 of [5])
- After ≈ 1310 ineffective faults

CONTRIBUTION 2: CHOSEN PLAINTEXT SIFA

Pros and Cons

- Attacker requires input control
- Brute force 16-bits at a time (instead of 32-bits)
- Attack needs to be repeated 6 times (+ 32-bit bruteforce) to retrieve the full key
- Same benefits and equal leakage to SIFA form input side
- Great for white-box analysis:
  Reduces the brute force complexity (analysis time) by a factor of 32768

SUMMARY

SIFA from the input side...

- Perform better than regular SIFA in the presence of clock jitter
- Known inputs (randomly distributed)/attacker-controlled inputs
- Allow for analysis on all 4 columns simultaneously → blackbox

Chosen Plaintext SIFA...

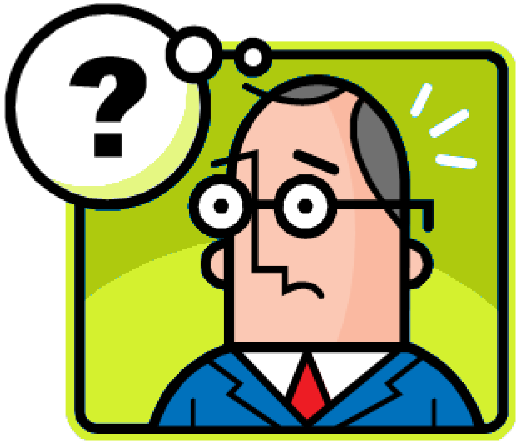
- Has the same benefits as SIFA from the input side
- Attacker controlled inputs
- Reduces the brute force complexity (analysis time) by a factor of 32768 → whitebox

QUESTIONS OR REMARKS?

Bob Swinkels

Security Analyst at Riscure

swinkels@riscure.com



Riscure B.V.
Frontier Building, Delftechpark 49
2628 XJ Delft
The Netherlands
Phone: +31 15 251 40 90
www.riscure.com

Riscure North America
550 Kearny St., Suite 330
San Francisco, CA 94108 USA
Phone: +1 650 646 99 79
inforequest@riscure.com

Riscure China
Room 2030-31, No. 989, Changle Road,
Shanghai 200031
China
Phone: +86 21 5117 5435
inforcn@riscure.com

riscure

driving your security forward

SEI & CHI-SQUARED STATISTIC

$$\mathrm{SEI} = \sum_{x \in \mathcal{X}} \left(\widehat{p_k}(x) - \theta(x) \right)^2$$

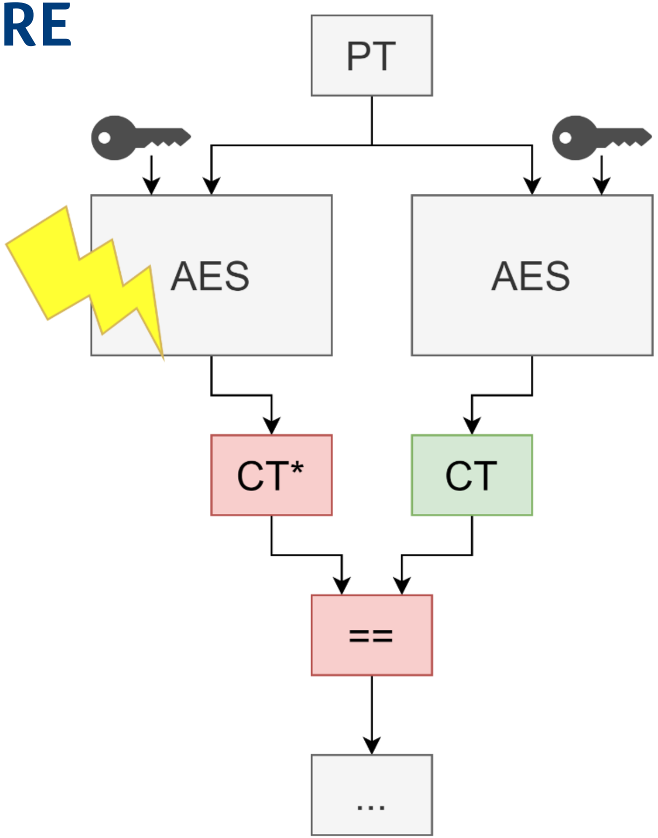
$$\chi^2(\hat{p}, \theta) = N \sum_{x \in \mathcal{X}} \frac{\left(\widehat{p_k}(x) - \theta(x) \right)^2}{\theta(x)}$$

GLITCH PARAMETERS

	Input side SIFA		Chosen Input SIFA	
Parameters	Textbook	T-Table	Textbook	T-Table
Normal voltage	3.3 V	3.3 V	3.3 V	3.3 V
Glitch voltage	1.0 V	1.0 V	1.0 V	1.0 V
Glitch length	123 ns	123 ns	123 ns	123 ns
Glitch delay	32500 ns	5550 ns	32500 ns	5550 ns

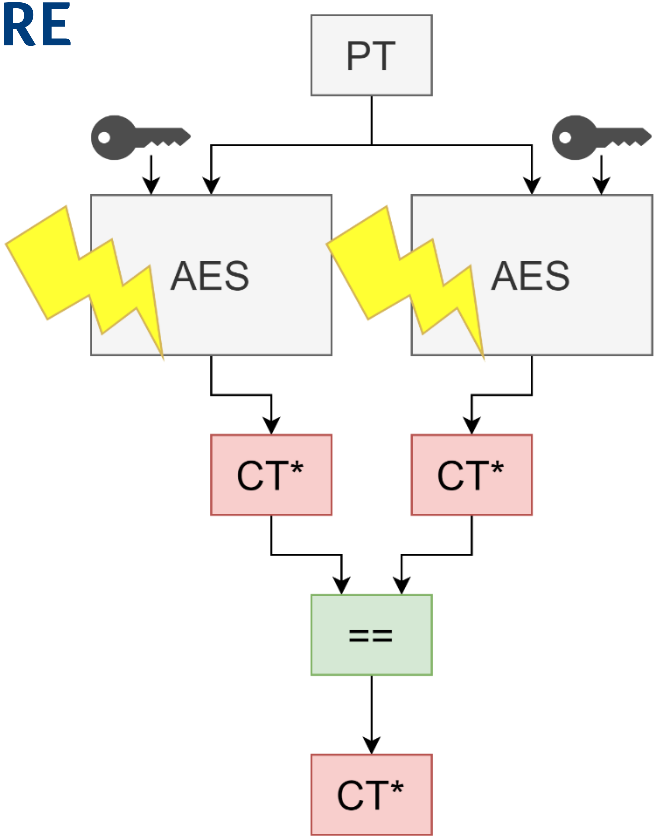
REDUNDANCY COUNTERMEASURE

- Fault detected == no ciphertext
- 2 identical faults needed for DFA



REDUNDANCY COUNTERMEASURE

- Fault detected == no ciphertext
- 2 identical faults needed for DFA



REFERENCES

[1] P. Dusart, G. Letourneux, and O. Vivolo, "Differential fault analysis on A.E.S," IACR Cryptol. ePrint Arch., vol. 2003, p. 10, 2003. [Online]. Available: http://eprint.iacr.org/2003/010

[2] C. Clavier, "Secret external encodings do not prevent transient fault analysis," in Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings, ser. Lecture Notes in Computer Science, P. Paillier and I. Verbauwhede, Eds., vol. 4727. Springer, 2007, pp. 181–194. [Online]. Available: https://doi.org/10.1007/978-3-540-74735-2 13

[3] C. Dobraunig, M. Eichlseder, T. Korak, S. Mangard, F. Mendel, and R. Primas, "SIFA: exploiting ineffective fault inductions on symmetric cryptography," IACR Trans. Cryptogr. Hardw. Embed. Syst., vol. 2018, no. 3, pp. 547–572, 2018. [Online]. Available: https://doi.org/10.13154/tches.v2018.i3.547-572

[4] T. Fuhr, É. Jaulmes, V. Lomné, and A. Thillard, "Fault attacks on AES with faulty ciphertexts only," in 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Los Alamitos, CA, USA, August 20, 2013, W. Fischer and J. Schmidt, Eds. IEEE Computer Society, 2013, pp. 108–118. [Online]. Available: https://doi.org/10.1109/FDTC.2013.18

[5] J. Daemen and V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard, ser. Information Security and Cryptography. Springer, 2002. [Online]. Available: https://doi.org/10.1007/978-3-662-04722-4