Cryptography and Embedded System Security
CRAESS_I

Xiaolu Hou

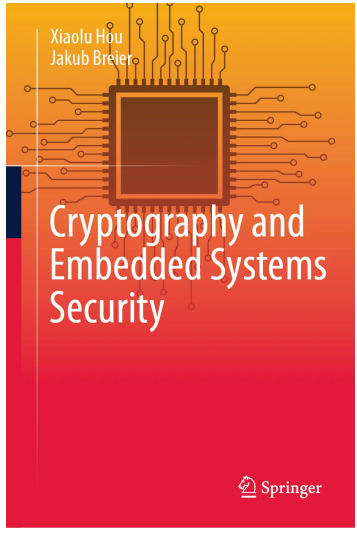
FIIT, STU
xiaolu.hou @ stuba.sk

Course Outline

- Abstract algebra and number theory
- Introduction to cryptography
- Symmetric block ciphers and their implementations
- RSA, RSA signatures, and their implementations
- Probability theory and introduction to SCA
- SPA and non-profiled DPA
- Profiled DPA
- SCA countermeasures
- FA on RSA and countermeasures
- FA on symmetric block ciphers
- FA countermeasures for symmetric block cipher
- Practical aspects of physical attacks
  - Invited speaker: Dr. Jakub Breier, Senior security manager, TTControl GmbH

Recommended reading

- Textbook
  - Sections 5.1.1, 5.1.2



Lecture Outline

- Differential Fault Analysis
- Statistical Fault Analysis
- Other Fault Attacks

Fault attacks on symmetric block ciphers

- The specifications of round functions and key schedules are public (Kerckhoffs' principle)
- The master key, hence also the round keys, are secret.
- We also assume that throughout the attack, the same master key is used and the goal of the attacker is normally to recover certain round key(s).
- The methodologies that we will discuss can be applied to any unprotected implementations of symmetric block cipher proposed up to now
- Fault attacks normally aim to recover the last/first round key(s), then use the inverse key schedule to find the master key
  - DES: any round key $\rightarrow 48$ bits of master key
  - AES: any round key $\rightarrow$ master key
  - PRESENT: any round key $\rightarrow 64$ bits of master key, brute force the rest or recover another round key

Remark

- We will see two attacks on AES
  - Differential fault analysis attack
  - Statistical fault analysis attack
- There are many more fault attack methods

Fault mask

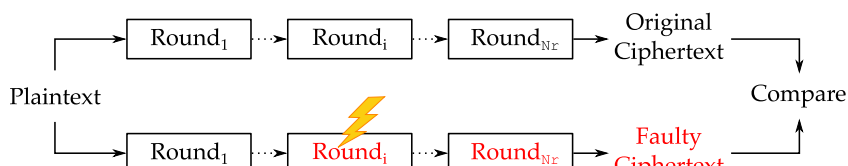
- If the fault injected in an intermediate value x results in a faulty value x'
- We refer to $\varepsilon := x \oplus x'$ as the *fault mask*, which represents the change in the faulted value.

FA on symmetric block ciphers

- Differential Fault Analysis
- Statistical Fault Analysis
- Other Fault Attacks

Attack methodology

- Differential Fault Analysis (DFA) was first introduced by Biham et al.[1] in 1997.
- It has been studied by numerous researchers in different settings and is one of the most popular fault attack analysis methods for symmetric block ciphers.
- DFA considers a fault injection into the intermediate state of the cipher, normally in the last few rounds.
- Then the difference between correct and faulty ciphertexts is analyzed to recover the round key(s).



[1] Biham, E., & Shamir, A. (1997, August). Differential fault analysis of secret key cryptosystems. In Annual international cryptology conference (pp. 513-525). Springer, Berlin, Heidelberg.

Difference distribution table

Definition

For an Sbox SB: $\mathbb{F}_2^{\omega_1} \to \mathbb{F}_2^{\omega_2}$, the *(extended) difference distribution table (DDT)* of SB is a $2-$dimensional table T of size $(2^{\omega_1} - 1) \times 2^{\omega_2}$ such that for any $0 < \delta < 2^{\omega_1}$ and $0 \leq \Delta < 2^{\omega_2}$, the entry of T at the Δth row and δth column is given by

$$T[\Delta, \delta] = \{\, \boldsymbol{a} \;\mid\; \boldsymbol{a} \in \mathbb{F}_2^{\omega_1}, \mathsf{SB}(\boldsymbol{a} \oplus \delta) \oplus \mathsf{SB}(\boldsymbol{a}) = \Delta \,\}.$$

We refer to δ as the *input difference*, and Δ as the *output difference*.

Example (DDT of PRESENT Sbox)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

If input is 9, input difference $\delta = 3$, what is the output difference?

$$\mathsf{SB}_{\mathsf{PRESENT}}(9 \oplus 3) \oplus \mathsf{SB}_{\mathsf{PRESENT}}(9) = ?$$

Difference distribution table

Definition

$$T[\Delta, \delta] = \{\, \boldsymbol{a} \mid \boldsymbol{a} \in \mathbb{F}_2^{\omega_1}, \mathsf{SB}(\boldsymbol{a} \oplus \delta) \oplus \mathsf{SB}(\boldsymbol{a}) = \Delta \,\}.$$

We refer to δ as the *input difference*, and Δ as the *output difference*.

Example (DDT of PRESENT Sbox)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

If input is 9, input difference $\delta = 3$, the output difference is given by

$$\mathsf{SB}_{\mathsf{PRESENT}}(9 \oplus 3) \oplus \mathsf{SB}_{\mathsf{PRESENT}}(9) = \mathsf{SB}_{\mathsf{PRESENT}}(\mathtt{A}) \oplus 1110 = 1111 \oplus 1110 = 0001 = 1.$$

Thus 9 is in $T[1, 3]$. Similarly, 7 is in $T[?, 3]$

Difference distribution table

Definition

$$T[\Delta, \delta] = \{\, \boldsymbol{a} \;\mid\; \boldsymbol{a} \in \mathbb{F}_2^{\omega_1}, \mathsf{SB}(\boldsymbol{a} \oplus \delta) \oplus \mathsf{SB}(\boldsymbol{a}) = \Delta \,\}.$$

We refer to δ as the *input difference*, and Δ as the *output difference*.

Example (DDT of PRESENT Sbox)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

If input is 9, input difference $\delta = 3$, the output difference is given by

$$\mathsf{SB}_{\mathsf{PRESENT}}(9 \oplus 3) \oplus \mathsf{SB}_{\mathsf{PRESENT}}(9) = \mathsf{SB}_{\mathsf{PRESENT}}(\mathtt{A}) \oplus 1110 = 1111 \oplus 1110 = 0001 = 1.$$

Thus 9 is in $T[1, 3]$. Similarly, 7 is in $T[4, 3]$

$$\mathsf{SB}_{\mathsf{PRESENT}}(7 \oplus 3) \oplus \mathsf{SB}_{\mathsf{PRESENT}}(7) = \mathsf{SB}_{\mathsf{PRESENT}}(4) \oplus 1101 = 1001 \oplus 1101 = 0100 = 4.$$

DDT of PRESENT Sbox

Δ \ δ	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1			9A		36		078F				5E		1C		24BD
2						8E	34		09	5F		1D	67AB	2C	
3	CDEF	46	12					3B		0A			58	79	
4			47		8D				35AC		0B		2F		169E
5		CDEF		0145						2389		67AB			
6		9B	CDEF	37		06	25		18					4A	
7	67AB		03	8C				5D				2E	49	1F	
8						17	AD		6F	4E	2389	0C		5B	
9	0145			9D	BE			2A			7C	3F		68	
A		02	56	BF	9C					7D	1A	48	3E		
B			8B		27	35AC		169E			4F		0D		
C		8a		26	0145	9F	BC		7E					3D	
D	2389	57			AF			4C		1B	6D			0E	
E		13		AE					24BD	6C		59			078F
F						24BD	169E	078F							35AC

Table: Difference distribution table for PRESENT Sbox, where the row corresponding to output difference $\Delta = 0$ is omitted since it is empty

DDT – Example

Example

Find the DDT for the following Sbox

x	0	1	2	3	4	5	6	7
SB(x)	4	7	0	5	2	6	3	1

Δ \ δ	1	2	3	4	5	6	7
1		?	?	15	27		
2	67	13			05	24	
3	01		47	26		35	
4	45	02		37			16
5	23		56		14		07
6				04	36	17	25
7		57	12			06	34

DDT – Example

Example

x	0	1	2	3	4	5	6	7
$\mathrm{SB}(x)$	4	7	0	5	2	6	3	1

Δ \ δ	1	2	3	4	5	6	7
1		46	03	15	27		
2	67	13			05	24	
3	01		47	26		35	
4	45	02		37			16
5	23		56		14		07
6				04	36	17	25
7		57	12			06	34

$$\mathrm{SB}(0) \oplus \mathrm{SB}(0 \oplus 3) = 4 \oplus \mathrm{SB}(3) = 4 \oplus 5 = 1$$

How DFA works on a simple example

- Let us consider the `AND` operation that takes inputs $a, b \in \mathbb{F}_2$ and outputs

$$c = a \ \& \ b.$$

- All possible values of a, b, c are given by

a	b	$c = a \ \& \ b$
0	0	0
0	1	0
1	0	0
1	1	1

- Suppose the output c can be observed by the attacker and a, b are unknown.
- The attacker injects a fault in b by flipping it.
- By observing the output c, how does the attacker recover value of a?

How DFA works on a simple example

- All possible values of a, b, c are given by

a	b	$c = a \ \& \ b$
0	0	0
0	1	0
1	0	0
1	1	1

- Suppose the output c can be observed by the attacker and a, b are unknown.
- The attacker injects a fault in b by flipping it.
- If the output c stays the same, then $a = 0$; otherwise $a = 1$.

How DFA works on PRESENT Sbox

- SB: PRESENT Sbox
- Let $\boldsymbol{a} \in \mathbb{F}_2^4,\ \boldsymbol{b} \in \mathbb{F}_2^4$ be fixed secret values
- Define

$$\begin{aligned} f : \mathbb{F}_2^4 &\rightarrow \mathbb{F}_2^4 \\ \boldsymbol{x} &\mapsto \mathrm{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}. \end{aligned}$$

- We will show how to recover the values of $\boldsymbol{a}$ and $\boldsymbol{b}$ with DFA
- Attack assumption
  - Fault location: input of f
  - Fault model: bit flip
  - Fault mask: $\varepsilon \in \mathbb{F}_2^4$ s.t. $\boldsymbol{x}' = \boldsymbol{x} \oplus \varepsilon$
  - Attacker knowledge: Sbox design, inputs and outputs of f, fault mask
  - Attacker goal: recover values of $\boldsymbol{a}$ and $\boldsymbol{b}$
  - Attacker can repeat the computation with the same input (not chosen by attacker)

How DFA works on PRESENT Sbox

$$\begin{aligned} f : \mathbb{F}_2^4 &\rightarrow \mathbb{F}_2^4 \\ \boldsymbol{x} &\mapsto \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}. \end{aligned}$$

- Attack assumption
  - Fault location: input of f
  - Fault model: bit flip
  - Fault mask: $\varepsilon \in \mathbb{F}_2^4$ s.t. $\boldsymbol{x}' = \boldsymbol{x} \oplus \varepsilon$
  - Attacker knowledge: Sbox design, inputs and outputs of f, fault mask
  - Attacker goal: recover values of $\boldsymbol{a}$ and $\boldsymbol{b}$
  - Attacker can repeat the computation with the same input (not chosen by attacker)
- Attack steps:
  - Compute DDT of the Sbox, T
  - Inject fault
  - Reduce guesses for $\boldsymbol{a}$ with knowledge of fault mask, input and outputs
  - Reduce guesses for $\boldsymbol{b}$ with guesses of $\boldsymbol{a}$, knowledge of the correct input and output

How DFA works on PRESENT Sbox

$$\begin{aligned} f : \mathbb{F}_2^4 &\rightarrow \mathbb{F}_2^4 \\ \boldsymbol{x} &\mapsto \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}. \end{aligned}$$

Attack steps:

- Compute DDT of the Sbox, T
- Inject fault
- Reduce guesses for $\boldsymbol{a}$ with knowledge of fault mask, input and outputs
  - Let Δ denote the difference between the correct and faulty output, then

    $$\Delta = (\mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}) \oplus (\mathsf{SB}(\boldsymbol{x}' \oplus \boldsymbol{a}) \oplus \boldsymbol{b}) = \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \mathsf{SB}(\boldsymbol{x}' \oplus \boldsymbol{a}) = \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a} \oplus \varepsilon)$$

  - Thus the value $\boldsymbol{x} \oplus \boldsymbol{a}$ is in the entry corresponding to input difference $\delta = ?$ and output difference ? of T
- Reduce guesses for $\boldsymbol{b}$ with guesses of $\boldsymbol{a}$, knowledge of the correct input and output

How DFA works on PRESENT Sbox

$$\begin{aligned} f : \mathbb{F}_2^4 &\rightarrow \mathbb{F}_2^4 \\ \boldsymbol{x} &\mapsto \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}. \end{aligned}$$

Attack steps:

- Compute DDT of the Sbox, T
- Inject fault
- Reduce guesses for $\boldsymbol{a}$ with knowledge of fault mask, input and outputs
  - Let Δ denote the difference between the correct and faulty output, then

    $$\Delta = (\mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}) \oplus (\mathsf{SB}(\boldsymbol{x}' \oplus \boldsymbol{a}) \oplus \boldsymbol{b}) = \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \mathsf{SB}(\boldsymbol{x}' \oplus \boldsymbol{a}) = \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a} \oplus \varepsilon)$$

  - Thus the value $\boldsymbol{x} \oplus \boldsymbol{a}$ is in the entry of T corresponding to input difference $\delta = \varepsilon$ and output difference Δ
- Reduce guesses for $\boldsymbol{b}$ with guesses of $\boldsymbol{a}$, knowledge of the correct input and output

How DFA works on PRESENT Sbox – Example

$$\begin{aligned} f : \mathbb{F}_2^4 &\rightarrow \mathbb{F}_2^4 \\ \boldsymbol{x} &\mapsto \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}. \end{aligned}$$

Reduce guesses for $\boldsymbol{a}$ with knowledge of fault mask, input and outputs

- Let Δ denote the difference between the correct and faulty output, then

  $$\Delta = (\mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}) \oplus (\mathsf{SB}(\boldsymbol{x}' \oplus \boldsymbol{a}) \oplus \boldsymbol{b}) = \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \mathsf{SB}(\boldsymbol{x}' \oplus \boldsymbol{a}) = \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a} \oplus \varepsilon)$$

- Thus the value $\boldsymbol{x} \oplus \boldsymbol{a}$ is in the entry of T corresponding to input difference $\delta = \varepsilon$ and output difference Δ

Example

- Suppose the attacker fixes the input to be $\boldsymbol{x} = 0$ and they know that the correct output of f is 0
- When the attacker injects fault in $\boldsymbol{x}$ with fault mask $\varepsilon_1 = 3$, they get a faulty output 1, which gives $\Delta_1 = ?$

How DFA works on PRESENT Sbox – Example

Δ \ δ	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1			9A		36		078F				5E		1C		24BD
2						8E	34		09	5f		1D	67AB	2C	
3	CDEF	46	12					3B		0A			58	79	
4			47		8D				35AC		0B		2F		169E
5		CDEF		0145						2389		67AB			
6		9B	CDEF	37		06	25		18					4A	
7	67AB		03	8C				5D				2E	49	1F	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	

Example

- Input: $\boldsymbol{x} = 0$; correct output: 0
- fault mask: $\varepsilon_1 = 3$; faulty output: 1, which gives $\Delta_1 = 0 \oplus 1 = 1$.
- $\boldsymbol{x} \oplus \boldsymbol{a}$ is in the entry corresponding to input difference $\delta = 3$ and output difference 1 of T
- The possible values for $\boldsymbol{x} \oplus \boldsymbol{a}$ are given by?

How DFA works on PRESENT Sbox – Example

Reduce guesses for $\boldsymbol{a}$ with knowledge of fault mask, input and outputs

- Let Δ denote the difference between the correct and faulty output, then

$$\Delta = (\mathrm{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}) \oplus (\mathrm{SB}(\boldsymbol{x}' \oplus \boldsymbol{a}) \oplus \boldsymbol{b}) = \mathrm{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \mathrm{SB}(\boldsymbol{x}' \oplus \boldsymbol{a}) = \mathrm{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \mathrm{SB}(\boldsymbol{x} \oplus \boldsymbol{a} \oplus \varepsilon)$$

- Thus the value $\boldsymbol{x} \oplus \boldsymbol{a}$ is in the entry of T corresponding to input difference $\delta = \varepsilon$ and output difference Δ

Example

- Input: $\boldsymbol{x} = 0$; correct output: 0
- fault mask: $\varepsilon_1 = 3$; faulty output: $1 \rightarrow$ the possible values for $\boldsymbol{x} \oplus \boldsymbol{a}$ are 9 and A
- When the attacker injects another fault with fault mask $\varepsilon_2 = 2$, they get a faulty output 6. $\Delta_2 = ?$. $\boldsymbol{x} \oplus \boldsymbol{a}$ is in the entry of T corresponding to input difference $\delta = ?$ and output difference ?

How DFA works on PRESENT Sbox – Example

Δ \ δ	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1			9A		36		078F				5E		1C		24BD
2						8E	34		09	5f		1D	67AB	2C	
3	CDEF	46	12					3B		0A			58	79	
4			47		8D				35AC		0B		2F		169E
5		CDEF		0145						2389		67AB			
6		9B	CDEF	37		06	25		18					4A	
7	67AB		03	8C				5D				2E	49	1F	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Example

- Input: $\boldsymbol{x} = 0$; correct output: 0
- fault mask: $\varepsilon_1 = 3$; faulty output: 1 $\rightarrow$ the possible values for $\boldsymbol{x} \oplus \boldsymbol{a}$ are 9 and A
- fault mask $\varepsilon_2 = 2$; faulty output: 6 $\rightarrow \Delta_2 = 6$
- $\boldsymbol{x} \oplus \boldsymbol{a}$ is in the entry of T corresponding to input difference $\delta = 2$ and output difference 6
- Possible values of $\boldsymbol{x} \oplus \boldsymbol{a}$ are ?

How DFA works on PRESENT Sbox – Example

$$\begin{aligned} f : \mathbb{F}_2^4 &\rightarrow \mathbb{F}_2^4 \\ \boldsymbol{x} &\mapsto \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}. \end{aligned}$$

Reduce guesses for $\boldsymbol{b}$ with guesses of $\boldsymbol{a}$, knowledge of the correct input and output

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Example

- Input: $\boldsymbol{x} = 0$; correct output: 0
- fault mask: $\varepsilon_1 = 3$; faulty output: $1 \rightarrow$ the possible values for $\boldsymbol{x} \oplus \boldsymbol{a}$ are 9 and A
- fault mask $\varepsilon_2 = 2$; faulty output: $6 \rightarrow$ possible values for $\boldsymbol{x} \oplus \boldsymbol{a}$ are 9 and B
- Thus $\boldsymbol{a} = ?$, $\boldsymbol{b} = ?$

How DFA works on PRESENT Sbox – Example

$$\begin{aligned} f : \mathbb{F}_2^4 &\rightarrow \mathbb{F}_2^4 \\ \boldsymbol{x} &\mapsto \mathsf{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}. \end{aligned}$$

Reduce guesses for $\boldsymbol{b}$ with guesses of $\boldsymbol{a}$, knowledge of the correct input and output

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Example

- Input: $\boldsymbol{x} = 0$; correct output: 0
- fault mask: $\varepsilon_1 = 3$; faulty output: $1 \rightarrow$ the possible values for $\boldsymbol{x} \oplus \boldsymbol{a}$ are 9 and A
- fault mask $\varepsilon_2 = 2$; faulty output: $6 \rightarrow$ possible values for $\boldsymbol{x} \oplus \boldsymbol{a}$ are 9 and B
- $\boldsymbol{a} = 9$, $\boldsymbol{b} = \mathtt{E}$

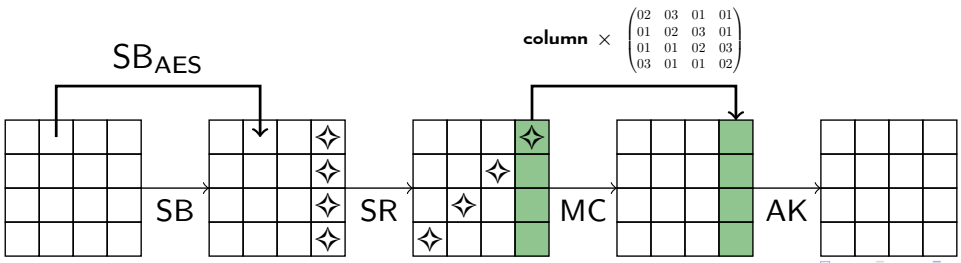
How many faults are needed

Δ \ ε	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1			9a		36		078f				5e		1c		24bd
2						8e	34		09	5f		1d	67ab	2c	
3	cdef	46	12					3b		0a			58	79	
4			47		8d				35ac		0b		2f		169e
5		cdef		0145						2389		67ab			
6		9b	cdef	37		06	25		18					4a	
7	67ab		03	8c				5d				2e	49	1f	
8						17	ad		6f	4e	2389	0c		5b	
9	0145			9d	be			2a			7c	3f		68	
a		02	56	bf	9c					7d	1a	48	3e		
b			8b		27	35ac		169e			4f		0d		
c		8a		26	0145	9f	bc		7e					3d	
d	2389	57			af			4c		1b	6d			0e	
e		13		ae					24bd	6c		59			078f
f						24bd	169e	078f							35ac

- Chosen fault mask: 2 (e.g. 3 and 5)
- Random fault mask: at most 4

AES encryption

- An initial AddRoundKey
- Round function for $\texttt{Nr}-1$ rounds: SubBytes, ShiftRows, MixColumns, AddRoundKey
- Last round, round $\texttt{Nr}$: SubBytes, ShiftRows, AddRoundKey
- AddRoundKey is bitwise $\texttt{XOR}$ with the round key
- SubBytes is the application of $8-$bit Sboxes.
- ShiftRows permutes the bytes
- MixColumns is a function on $32-$bit values (four bytes).



Fault propagation in AES

- Recall that AES cipher state can be represented as a four-by-four matrix of bytes:

$$\begin{pmatrix} s_{00} & s_{01} & s_{02} & s_{03} \\ s_{10} & s_{11} & s_{12} & s_{13} \\ s_{20} & s_{21} & s_{22} & s_{23} \\ s_{30} & s_{31} & s_{32} & s_{33} \end{pmatrix}.$$

- Let us represent those bytes by squares for the purpose of visual illustration.
- Suppose a fault is injected at the beginning of one round (except for the last round) in byte s_{00}.
- Then the fault propagation in this round can be represented by

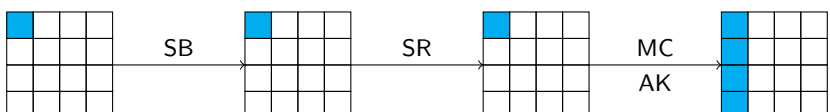


Figure: Blue squares correspond to bytes that can be affected by the fault.

Fault propagation in AES

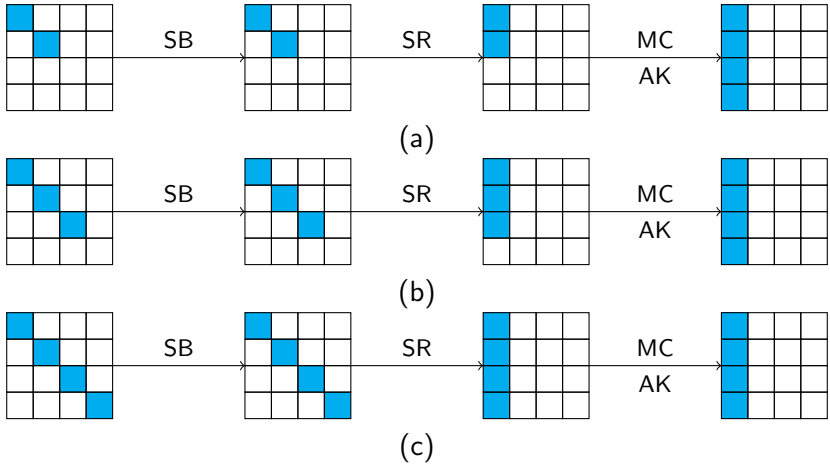


Figure: Visual illustration of how the fault propagates when a fault is injected at the beginning of one AES round in bytes (a) s_{00}, s_{11}, (b) s_{00}, s_{11}, s_{22}, and (c) $s_{00}, s_{11}, s_{22}, s_{33}$. Blue squares correspond to bytes that can be affected by the fault.

Fault at end of round 7

- Let us refer to the bytes $s_{00}, s_{11}, s_{22}, s_{33}$ as a *diagonal* of AES state
- We consider a fault attack where a random byte fault is injected in the diagonal of the AES state at the end of round 7.
- By the above discussion, we know that at the end of round 8, the whole first column might be affected by the fault.
- Similarly, we can study the fault propagation in round 9.
- Recall that MixColumns multiplies one column by the following matrix

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}.$$

Fault propagation in round 9

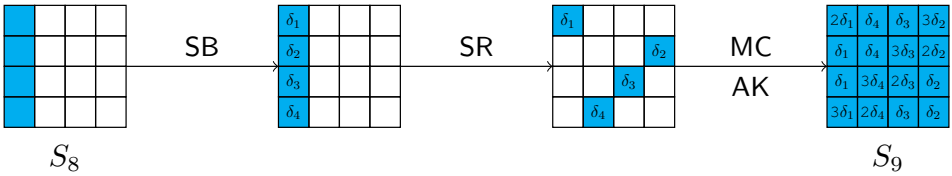


Figure: Visual illustration of fault propagation in the 9th round of AES when the fault was injected in the diagonal $s_{00}, s_{11}, s_{22}, s_{33}$ of the AES cipher state at the end of round 7. δ_i $(i = 1, 2, 3, 4)$ denote the differences between the four correct and faulty bytes in the first column of the cipher state after SubBytes in round 9.

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}.$$

Notations

- S_9: the cipher state at the end of round nine
- c: the correct ciphertext
- K_{10}: the last round key

$$S_9 = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix}, \; c = \begin{pmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{pmatrix}, \; K_{10} = \begin{pmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{pmatrix}.$$

- c': faulty ciphertext

$$c' = \begin{pmatrix} c'_{00} & c'_{01} & c'_{02} & c'_{03} \\ c'_{10} & c'_{11} & c'_{12} & c'_{13} \\ c'_{20} & c'_{21} & c'_{22} & c'_{23} \\ c'_{30} & c'_{31} & c'_{32} & c'_{33} \end{pmatrix}.$$

- Round 10: SubBytes, ShiftRows, and AddRoundKey.

$$c_{00} = \mathsf{SB}_{\mathsf{AES}}(a_{00}) \oplus k_{00}, \quad c_{13} = \mathsf{SB}_{\mathsf{AES}}(a_{10}) \oplus k_{13},$$
$$c_{22} = \mathsf{SB}_{\mathsf{AES}}(a_{20}) \oplus k_{22}, \quad c_{31} = \mathsf{SB}_{\mathsf{AES}}(a_{30}) \oplus k_{31}.$$

- Then

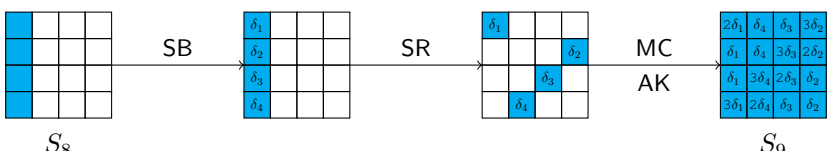
$$\begin{aligned} a_{00} &= \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{00} \oplus k_{00}) \\ a_{10} &= \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{13} \oplus k_{13}) \\ a_{20} &= \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{22} \oplus k_{22}) \\ a_{30} &= \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{31} \oplus k_{31}). \end{aligned}$$

- Similarly

$$\begin{aligned} a'_{00} &= \mathsf{SB}^{-1}_{\mathsf{AES}}(c'_{00} \oplus k_{00}) \\ a'_{10} &= \mathsf{SB}^{-1}_{\mathsf{AES}}(c'_{13} \oplus k_{13}) \\ a'_{20} &= \mathsf{SB}^{-1}_{\mathsf{AES}}(c'_{22} \oplus k_{22}) \\ a'_{30} &= \mathsf{SB}^{-1}_{\mathsf{AES}}(c'_{31} \oplus k_{31}). \end{aligned}$$

- Let $\delta = \delta_1$ and by observing the first column of S_9, we have

$$\begin{aligned}
2\delta &= a_{00} \oplus a'_{00} = \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{00} \oplus k_{00}) \oplus \mathsf{SB}^{-1}_{\mathsf{AES}}(c'_{00} \oplus k_{00}) \\
\delta &= a_{10} \oplus a'_{10} = \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{13} \oplus k_{13}) \oplus \mathsf{SB}^{-1}_{\mathsf{AES}}(c'_{13} \oplus k_{13}) \\
\delta &= a_{20} \oplus a'_{20} = \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{22} \oplus k_{22}) \oplus \mathsf{SB}^{-1}_{\mathsf{AES}}(c'_{22} \oplus k_{22}) \\
3\delta &= a_{30} \oplus a'_{30} = \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{31} \oplus k_{31}) \oplus \mathsf{SB}^{-1}_{\mathsf{AES}}(c'_{31} \oplus k_{31}).
\end{aligned}$$



Key recovery

$$
\begin{aligned}
2\delta &= a_{00} \oplus a'_{00} = \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{00} \oplus k_{00}) \oplus \mathsf{SB}^{-1}_{\mathsf{AES}}(c'_{00} \oplus k_{00}) \\
\delta &= a_{10} \oplus a'_{10} = \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{13} \oplus k_{13}) \oplus \mathsf{SB}^{-1}_{\mathsf{AES}}(c'_{13} \oplus k_{13}) \\
\delta &= a_{20} \oplus a'_{20} = \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{22} \oplus k_{22}) \oplus \mathsf{SB}^{-1}_{\mathsf{AES}}(c'_{22} \oplus k_{22}) \\
3\delta &= a_{30} \oplus a'_{30} = \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{31} \oplus k_{31}) \oplus \mathsf{SB}^{-1}_{\mathsf{AES}}(c'_{31} \oplus k_{31}).
\end{aligned}
$$

- For each value of δ, the possible values for $(k_{00}, k_{13}, k_{22}, k_{31})$ are restricted
- $a_{00} = \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{00} \oplus k_{00})$ can be considered as an AES Sbox input that corresponds to input difference 2δ and output difference $c_{00} \oplus c'_{00}$
- $a_{10} = \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{13} \oplus k_{13})$ is an AES Sbox input that gives output difference $c_{13} \oplus c'_{13}$ when the input difference is δ
- On average[1], the key hypotheses for $(k_{00}, k_{13}, k_{22}, k_{31})$ can be reduced to 2^8

[1] Saha, D., Mukhopadhyay, D., & RoyChowdhury, D. (2009). A diagonal fault attack on the advanced encryption standard. Cryptology ePrint Archive.

Diagonal DFA

- In this attack, we assume the attacker has the knowledge of
  - The fault location: diagonal of cipher state at the end of round 7
  - Fault model: random byte
  - Output of AES: correct and faulty ciphertext
- Since the attack is on the diagonal of the cipher state, it is also called the *diagonal DFA*.

Other diagonals

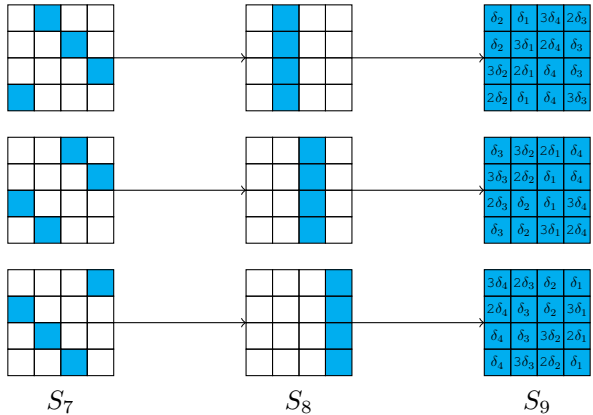


Figure: Fault propagation for random byte fault injected in the "diagonals" of the cipher state at the end of round 7. S_i denotes the cipher state at the end of the ith round

FA on symmetric block ciphers

- Differential Fault Analysis
- Statistical Fault Analysis
- Other Fault Attacks

Random Experiments

- Probability theory studies the mathematical theory behind random experiments.
- A random experiment is an experiment whose output cannot be predicted with certainty in advance.
- However, if the experiment is repeated many times, we can see "regularity" in the average output.
- For example, if we roll a die, we cannot predict the output of one roll.
- But if we roll it many times, we would expect to see the number 1 in $1/6$ of the outcomes assuming the die is fair.

Sample Space and Events

- For a given random experiment, we define *sample space*, denoted by Ω, to be the set of all possible outcomes.
- A subset A of Ω is called an *event*.
- If the outcome of the experiment is contained in A, then we say that A has *occurred*.
- The empty set $\emptyset$ denotes the event that consists of no outcomes.
- $\emptyset$ is also called the *impossible event*.

Example

- When the random experiment is rolling a die, the sample space $\Omega = \{ 1, 2, 3, 4, 5, 6 \}$. $A = \{ 1, 2, 3 \} \subseteq \Omega$ is an event.
- When the random experiment is rolling two dice, $\Omega = \{ (i, j) \mid 1 \leq i, j \leq 6 \}$. One possible event is $A = \{ (1, 2), (1, 1) \}$.

Sample space and its power set

- Ω: sample space
- $\mathcal{A}$: power set of Ω, 2^{Ω}

Example

Let us consider the random experiment of tossing a coin, the sample space $\Omega = \{ H, T \}$. $\mathcal{A} = 2^{\Omega} = \{ \emptyset, \Omega, \{ H \}, \{ T \} \}$.

Probability

Definition

A *probability measure* defined on $(\Omega, \mathcal{A})$ is a function $P : \mathcal{A} \to [0, 1]$ such that

- $P(\Omega) = 1$, $P(\emptyset) = 0$.
- For any $A_i \in \mathcal{A}$ that are pairwise disjoint, i.e. $A_{i_1} \cap A_{i_2} = \emptyset$ for $i_1 \neq i_2$, *countable additivity*

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i).$$

$P(A)$ is called the *probability* of A.

Example

Tossing a coin, $\Omega = \{ H, T \}$. $\mathcal{A} = 2^{\Omega} = \{ \emptyset, \Omega, \{ H \}, \{ T \} \}$. P defined as follows is a probability measure on $(\Omega, \mathcal{A})$:

$$P(\emptyset) = 0, \quad P(\Omega) = 1, \quad P(\{ H \}) = \frac{1}{2}, \quad P(\{ T \}) = \frac{1}{2}.$$

Partition of Ω

Sample space Ω, power set $\mathcal{A} = 2^{\Omega}$.

Definition

A set of events $\{ E_1, E_2, \ldots \mid E_i \in \mathcal{A} \}$, is called a *partition of* Ω if they are pairwise disjoint, $P(E_i) > 0$ for all i, and $\cup_i E_i = \Omega$.

Example

Let $\Omega = \{ 1, 2, 3, 4, 5, 6 \}$, $\mathcal{A} = 2^{\Omega}$, and P be the uniform probability measure on $(\Omega, \mathcal{A})$. Let $E_1 = \{ 1, 2, 3 \}$, $E_2 = \{ 4, 5 \}$, $E_3 = \{ 6 \}$. Then, $\{ E_1, E_2, E_3 \}$ is a finite partition of Ω. We can also calculate that

$$P(E_1) = \frac{1}{2}, \quad P(E_2) = \frac{1}{3}, \quad P(E_3) = \frac{1}{6}.$$

Lemma

Lemma

Let $\{ E_1, E_2, \dots \mid E_i \in \mathcal{A} \}$ *be a finite or countable partition of* Ω*. Then, for any* $A \in \mathcal{A}$*, we have*

$$P(A) = \sum_i P(A|E_i)P(E_i).$$

Example

$\Omega = \{ 1,2,3,4,5,6 \}$, $E_1 = \{ 1,2,3 \}$, $E_2 = \{ 4,5 \}$, $E_3 = \{ 6 \}$, $A = \{ 2,4 \}$.

$$P(A) = 1/3, \quad A \cap E_1 = \{ 2 \}, \quad A \cap E_2 = \{ 4 \}, \quad A \cap E_3 = \emptyset.$$

$$P(A|E_1) = \frac{1/6}{1/2} = \frac{1}{3}, \quad P(A|E_2) = \frac{1/6}{1/3} = \frac{1}{2}, \quad P(A|E_3) = 0.$$

$$\sum_{i=1}^{3} P(A|E_i)P(E_i) = \frac{1}{3} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{3} = \frac{1}{3} = P(A).$$

Fault distribution table

- Consider fault models that change an intermediate value x to x'.
- Model these two values as random variables X and X'.
- Based on the fault properties, we can draw a table with probabilities for the value x to be changed to x', i.e. $P(X' = x' | X = x)$ – *fault distribution table*

Example

- Let us consider the case when x is just one bit.
- stuck-at-0: changes x to 0 with probability 1.
- bit flip fault: changes x to $x \oplus 1$ with probability 1.
- random fault: changes x to $x \oplus 1$ with probability 0.5.

x \ x'

x	0	1
0	1	0
1	1	0

stuck-at-0

x	0	1
0	?	?
1	?	?

bit flip

x	0	1
0	?	?
1	?	?

random

Fault distribution table

- Consider fault models that change an intermediate value x to x'.
- Model these two values as random variables X and X'.
- Based on the fault properties, we can draw a table with probabilities for the value x to be changed to x', i.e. $P(X' = x' | X = x)$ – *fault distribution table*

Example

- Let us consider the case when x is just one bit.
- stuck-at-0: changes x to 0 with probability 1.
- bit flip fault: changes x to $x \oplus 1$ with probability 1.
- random fault: changes x to $x \oplus 1$ with probability 0.5.

stuck-at-0

x \ x'	0	1
0	1	0
1	1	0

bit flip

x \ x'	0	1
0	0	1
1	1	0

random

x \ x'	0	1
0	0.5	0.5
1	0.5	0.5

Assumption

- Statistical Fault Analysis (SFA)[1] assumes no knowledge of plaintext or correct ciphertext for the attacker.
- Only knowledge of faulty ciphertext and a non-uniform fault model is required.
- We say that the fault model is *non-uniform* if

$$P(X' = x' | X = x) \neq \frac{1}{2^b}$$

for some x and x', where b is the maximum bit length of x.

[1] Fuhr, T., Jaulmes, É., Lomné, V., & Thillard, A. (2013, August). Fault attacks on AES with faulty ciphertexts only. In 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography (pp. 108-118). IEEE.

Non-uniform fault model – Example

Example

- Let us consider the case when x is just one bit.
- In this case, the bit length of x is 1, and a fault model is non-uniform if

$$P(X' = x' | X = x) \neq 0.5$$

for some x and x'.

- Which fault models are non-uniform?

stuck-at-0

x \ x'	0	1
0	1	0
1	1	0

bit flip

x \ x'	0	1
0	0	1
1	1	0

random

x \ x'	0	1
0	0.5	0.5
1	0.5	0.5

Non-uniform fault model – Example

Example

- Let us consider the case when x is just one bit.
- In this case, the bit length of x is 1, and a fault model is non-uniform if

$$P(X' = x' | X = x) \neq 0.5$$

  for some x and x'.
- stuck-at-0 and bit flip fault models are non-uniform.

stuck-at-0

x \ x'	0	1
0	1	0
1	1	0

bit flip

x \ x'	0	1
0	0	1
1	1	0

random

x \ x'	0	1
0	0.5	0.5
1	0.5	0.5

Fault injection in AES round 9

- S_9: the cipher state at the end of round nine
- c: the correct ciphertext
- K_{10}: the last round key

$$S_9 = \begin{pmatrix} s_{00} & s_{01} & s_{02} & s_{03} \\ s_{10} & s_{11} & s_{12} & s_{13} \\ s_{20} & s_{21} & s_{22} & s_{23} \\ s_{30} & s_{31} & s_{32} & s_{33} \end{pmatrix}, \; c = \begin{pmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{pmatrix}, \; K_{10} = \begin{pmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{pmatrix}.$$

- A fault in s_{00} with a non-uniform fault model.
- S_{00}: random variable corresponding to s_{00}
- S'_{00}: random variable corresponding to faulty value of S_{00}, s'_{00}
- Attacker knowledge: the fault location and the fault distribution table, i.e. the probabilities

$$P(S'_{00} = s'_{00} | S_{00} = s_{00})$$

- Goal of the attacker: recover k_{00}.

SFA – Example

Example

Let us consider a stuck-at-0 fault model, then

$$P(S'_{00} = s'_{00} | S_{00} = s_{00}) = \begin{cases} 1 & s'_{00} = 00 \\ 0 & \text{Otherwise} \end{cases},$$

for all $s_{00} \in \mathbb{F}_2^8$.
In this case, one faulty ciphertext is enough to recover k_{00}. Since the attacker knows that the faulty value s'_{00} is always 00, they can recover k_{00} by computing

$$k_{00} = c'_{00} \oplus \mathsf{SB}_{\mathsf{AES}}(00) = c'_{00} \oplus 63.$$

Fault injection in AES round 9

We assume S_{00} follows a uniform distribution, i.e.

$$P(S_{00} = s_{00}) = \frac{1}{256} \quad \forall s_{00} \in \mathbb{F}_2^8.$$

Then by the lemma

Lemma

Let $\{ E_1, E_2, \dots \mid E_i \in \mathcal{A} \}$ *be a finite or countable partition of* Ω*. Then, for any* $A \in \mathcal{A}$*, we have*

$$P(A) = \sum_i P(A|E_i)P(E_i).$$

$$\begin{aligned} P(S'_{00} = s'_{00}) &= \sum_{s_{00}=0}^{255} P(S'_{00} = s'_{00}|S_{00} = s_{00})P(S_{00} = s_{00}) \\ &= \frac{1}{256} \sum_{s_{00}=0}^{255} P(S'_{00} = s'_{00}|S_{00} = s_{00}). \end{aligned}$$

Probability of a faulty value

We assume S_{00} follows a uniform distribution, i.e.

$$P(S_{00} = s_{00}) = \frac{1}{256} \quad \forall s_{00} \in \mathbb{F}_2^8.$$

Then

$$\begin{aligned} P(S'_{00} = s'_{00}) &= \sum_{s_{00}=0}^{255} P(S'_{00} = s'_{00} | S_{00} = s_{00}) P(S_{00} = s_{00}) \\ &= \frac{1}{256} \sum_{s_{00}=0}^{255} P(S'_{00} = s'_{00} | S_{00} = s_{00}). \end{aligned}$$

Recall – Fault injection in AES round 9

- S_9: the cipher state at the end of round nine
- c: the correct ciphertext
- K_{10}: the last round key

$$S_9 = \begin{pmatrix} s_{00} & s_{01} & s_{02} & s_{03} \\ s_{10} & s_{11} & s_{12} & s_{13} \\ s_{20} & s_{21} & s_{22} & s_{23} \\ s_{30} & s_{31} & s_{32} & s_{33} \end{pmatrix}, \; c = \begin{pmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{pmatrix}, \; K_{10} = \begin{pmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{pmatrix}.$$

$$c_{00} = \mathsf{SB}_{\mathsf{AES}}(s_{00} \oplus k_{00}) \Longrightarrow s_{00} = \mathsf{SB}^{-1}_{\mathsf{AES}}(c_{00} \oplus k_{00}).$$

Attack steps

- Injects fault in s_{00}
- Collects a set of m faulty ciphertexts $\left\{ c'^1, c'^2, \ldots, c'^m \right\}$.
- Let $\hat{k}_{00}$ denote a key hypothesis for k_{00}.
- Then for each c'^i, we can compute a hypothetical value for s'_{00}, denoted $\hat{s}^i_{00}$, as follows:

$$\hat{s}^i_{00} = \mathsf{SB}^{-1}_{\mathsf{AES}}(c'^i{}_{00} \oplus \hat{k}_{00}).$$

- Last round: SubBytes, ShiftRows, AddRoundKey

Attack steps

$$P(S'_{00} = s'_{00}) = \sum_{s_{00}=0}^{255} P(S'_{00} = s'_{00} | S_{00} = s_{00}) P(S_{00} = s_{00})$$
$$= \frac{1}{256} \sum_{s_{00}=0}^{255} P(S'_{00} = s'_{00} | S_{00} = s_{00}).$$

- The probability that the faulty value of s_{00} in the ith encryption actually equals to $\hat{s}^i_{00}$ can be found by the knowledge of the fault distribution table using the above formula:

$$P(S'_{00} = \hat{s}^i_{00}) = \frac{1}{256} \sum_{s_{00}=0}^{255} P(S'_{00} = \hat{s}^i_{00} | S_{00} = s_{00}).$$

Attack steps

- Define $\ell(\hat{k}_{00})$ to be the probability that the faulty value of s_{00} in the ith encryption equals to the hypothetical value $\hat{s}_{00}^i$ for all i, i.e.

$$\ell(\hat{k}_{00}) := \prod_{i=1}^{m} P(S'_{00} = \hat{s}_{00}^i). \tag{1}$$

- Then the correct key can be found using the *maximum likelihood* approach, namely

$$k_{00} = \arg\max_{\hat{k}_{00}} \ell(\hat{k}_{00}).$$

Attack results

- It was shown[1] that with high probability, the correct key byte can be recovered with only a few faults.
- The same method can recover other bytes of K_{10}.
- Each byte can be recovered in parallel, hence the number of faults to recover the full round key depends on the number of bytes that can be faulted with one fault injection.

[1] Fuhr, T., Jaulmes, É., Lomné, V., & Thillard, A. (2013, August). Fault attacks on AES with faulty ciphertexts only. In 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography (pp. 108-118). IEEE.

FA on symmetric block ciphers

- Differential Fault Analysis
- Statistical Fault Analysis
- Other Fault Attacks

Ineffective Fault Analysis

- Clavier, C. (2007, September). Secret external encodings do not prevent transient fault analysis. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 181-194). Springer, Berlin, Heidelberg.
- Faults that do not change the intermediate values are exploited.
- Those faults are called *ineffective faults*.
- Normally a particular fault model is assumed, e.g. a stuck-at-0 fault model.

Statistical Ineffective Fault Attack (SIFA)

- Dobraunig, C., Eichlseder, M., Korak, T., Mangard, S., Mendel, F., & Primas, R. (2018). SIFA: exploiting ineffective fault inductions on symmetric cryptography. IACR Transactions on Cryptographic Hardware and Embedded Systems, 547-572.
- A non-uniform fault model is assumed and the attack exploits ineffective faults.
- The dependency between the fault induction being ineffective and the data that is processed is exploited.
- Different from SFA, SIFA does not require each fault to be successful, but the attack requires repeated plaintext, and knowledge of the correct ciphertext (or whether each ciphertext is correct or not).
- The fault injection is the same as for SFA
- In the original paper the authors provide a detailed theoretical analysis of the number of ciphertexts needed and extensive experimental results.

Persistent Fault Analysis (PFA)

- Zhang, Fan, Xiaoxuan Lou, Xinjie Zhao, Shivam Bhasin, Wei He, Ruyi Ding, Samiya Qureshi, and Kui Ren. Persistent fault analysis on block ciphers. IACR Transactions on Cryptographic Hardware and Embedded Systems (2018): 150-172.
- Fault in the memory, Sbox lookup table
- Knowledge of ciphertext only

Algebraic Fault Analysis (AFA)

- Courtois, N. T., Jackson, K., & Ware, D. (2010). Fault-algebraic attacks on inner rounds of DES. In E-Smart'10 Proceedings: The Future of Digital Security Technologies. Strategies Telecom and Multimedia.
- Similar to DFA, exploits differences between correct and faulty ciphertext
- DFA – manual analysis
- AFA – expresses cryptographic algorithm in the form of algebraic equations and utilizes SAT solver[1] to recover the key.

[1] A SAT solver solves Boolean satisfiability problems. It takes a Boolean logic formula and checks if there is a solution satisfying the formula.

Collision Fault Analysis

- Blömer, J., & Seifert, J. P. (2003, January). Fault based cryptanalysis of the advanced encryption standard (AES). In International Conference on Financial Cryptography (pp. 162-181). Springer, Berlin, Heidelberg.
- Injects fault in the earlier rounds of a block cipher implementation.
- Then the attacker records the faulty ciphertext and finds plaintext that produces the same ciphertext, but without fault.
- Further analysis using those plaintexts can recover the round key.
- If the fault only changes one bit or one byte of the intermediate value, the attacker can try different plaintexts that only differ at one bit or one byte.

Fault Sensitivity Analysis

- Li, Y., Sakiyama, K., Gomisawa, S., Fukunaga, T., Takahashi, J., & Ohta, K. (2010, August). Fault sensitivity analysis. In International workshop on cryptographic hardware and embedded systems (pp. 320-334). Springer, Berlin, Heidelberg.
- Exploits the sensitivity of a device to faults
- The attack analyzes when a faulty output begins to exhibit some detectable characteristics and utilizes the information to recover the secret key.
- No knowledge of faulty ciphertext is required for the attack.