Fault Attacks on AES with Faulty Ciphertexts Only

Thomas FUHR, Eliane JAULMES, Victor LOMNE,
Adrian THILLARD

ANSSI (French Network and Information Security Agency)

FDTC 2013, Tuesday, August 20th
Santa Barbara, CA

- Embedded Systems integrating Cryptography are susceptible to Physical Attacks
- In this work we consider the security of Block Ciphers (particularly AES) vs Fault Attacks

Differential Fault Attacks

$$
\begin{aligned}
M_1 &\rightarrow (C_1, \widetilde{C_1}) \\
M_2 &\rightarrow (C_2, \widetilde{C_2}) \\
M_3 &\rightarrow (C_3, \widetilde{C_3}) \\
M_4 &\rightarrow (C_4, \widetilde{C_4}) \\
M_5 &\rightarrow (C_5, \widetilde{C_5}) \\
&\ldots
\end{aligned}
$$

$\longrightarrow$  $\longrightarrow$ 

Correct/faulty ciphertexts | Statistical treatment | Secret key

Countermeasure: Guilley et al. FDTC10

- Idea: Unable to encrypt twice the same message
  $\Longrightarrow$ no attack!
- Modify protocol:
  - input (M)
  - randomly draw r
  - output $C = (Enc(M \oplus r), r)$
- r renewed at each encryption, preventing differential attacks

Non differential fault attacks

$$M_1 \to \widetilde{C_1}$$
$$M_2 \to \widetilde{C_2}$$
$$M_3 \to \widetilde{C_3}$$
$$M_4 \to \widetilde{C_4}$$
$$M_5 \to \widetilde{C_5}$$
$$\ldots$$

Faulty ciphertexts   Statistical treatment   Secret key

Fault models

- If the fault is uniform $\implies$ no attack
- We consider non uniform faults, i.e. faults s.t. the distribution of the faulty value $\widetilde{X}$ is non uniform
- We study different degrees of knowledge/control of the attacker on the distribution of $\widetilde{X}$:
  - Perfect knowledge
  - Partial knowledge (eg `AND` with unknown value)
  - No knowledge (except for the non-uniform property)

Sketch of attack on AES: 9-th round

- The fault is injected just after the penultimate `AddRoundKey` operation on a single byte of the state

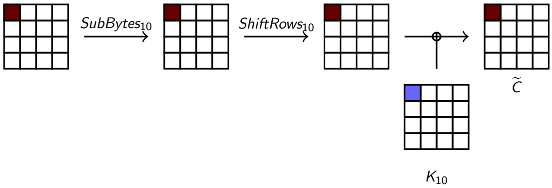
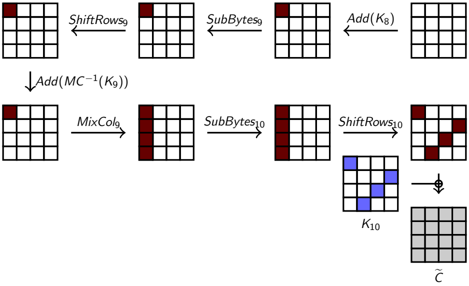


Figure : Brown bytes are modified due to the fault. A guess is made on the blue byte.

- For each $\widetilde{C}_i$, make an hypothesis $\hat{k}$ on the secret and compute the corresponding intermediate value $X_{i,\hat{k}}$
- Distribution of $X_{i,\hat{k}}$ matches our model when $\hat{k}$ is correct.

Sketch of attack on AES: 8-th round

- The fault is injected just after the antepenultimate `AddRoundKey` operation on a single byte of the state



- For each $\widetilde{C}_i$, make an hypothesis $\hat{k}$ on the secret and compute the corresponding intermediate value $X_{i,\hat{k}}$
- Distribution of $X_{i,\hat{k}}$ matches our model when $\hat{k}$ is correct. Otherwise uniform

Find the perfect match?

Distinguishers: depending on the knowledge of the fault:

- Maximum likelihood:

$$\prod_{i=1}^{n} P(\widetilde{X} = X_{i,\hat{k}})$$

- Min/Max mean HW:

$$\frac{1}{n}\sum_{i=1}^{n} HW(X_{i,\hat{k}})$$

- Squared Euclidian Imbalance (SEI):

$$\sum_{\delta=0}^{255} \left(\frac{\#\{i | X_{i,\hat{k}} = \delta\}}{n} - \frac{1}{256} \right)^2$$

Results

To study how well these distinguishers perform, we simulated several fault effects:

- (a) $\widetilde{X} = X \texttt{ AND } 0$ with probability 1
- (b) $\widetilde{X} = X \texttt{ AND } 0$ with probability $\frac{1}{2}$, $\widetilde{X} = X \texttt{ AND } e$ otherwise
- (c) $\widetilde{X} = X \texttt{ AND } e$ with e uniform

Results

	Max. likelihood	Min. mean HW	SEI
$\widetilde{X} = X \texttt{ AND } 0$	1	1	N/A
$\widetilde{X} = X \texttt{ AND } \{0, e\}$	10	14	N/A
$\widetilde{X} = X \texttt{ AND } e$	14	18	N/A

Figure : 9-th round: Number of required faults to retrieve (one byte of) K_{10} with a 99% probability. Note that the SEI is useless in this context.

- On the 8-th round: the SEI allows to retrieve (4 bytes!) of K_{10} using 6, 14 and 80 faulty ciphertexts respectively

Can we get much higher ?

- What if both last rounds are duplicated as a countermeasure ?

- Too many bytes of K_{10} to guess!

- Possible to attack if we strengthen fault models

7-th round

- The fault is injected between the 7-th `MixColumns` and the 8-th `ShiftRows`
- A diagonal of the state is stucked-at an unknown value
- Relevant fault model on 32-bit architecture

Sketch of attack

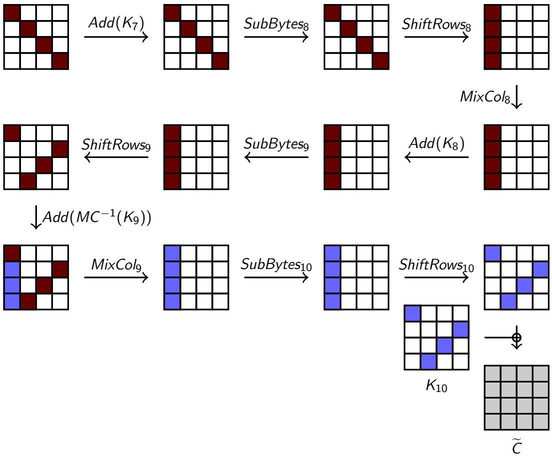


Figure : Brown bytes are constant due to the fault. Blue bytes can be computed from the ciphertext and a guess on K_{10}.

Sketch of attack

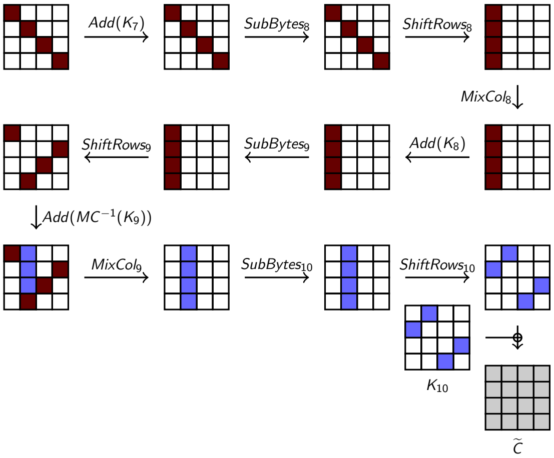


Figure : Brown bytes are constant due to the fault. Blue bytes can be computed from the ciphertext and a guess on K_{10}.

Sketch of attack

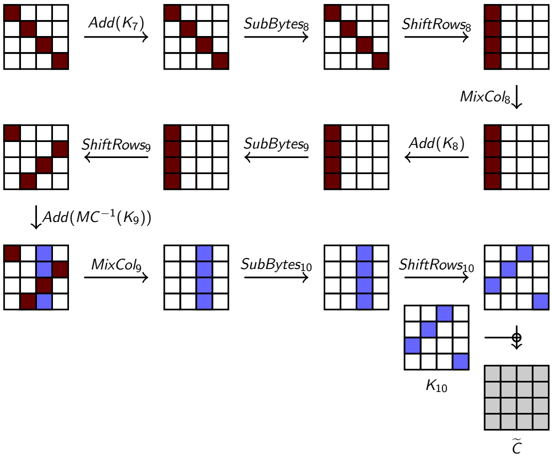


Figure : Brown bytes are constant due to the fault. Blue bytes can be computed from the ciphertext and a guess on K_{10}.

Sketch of attack

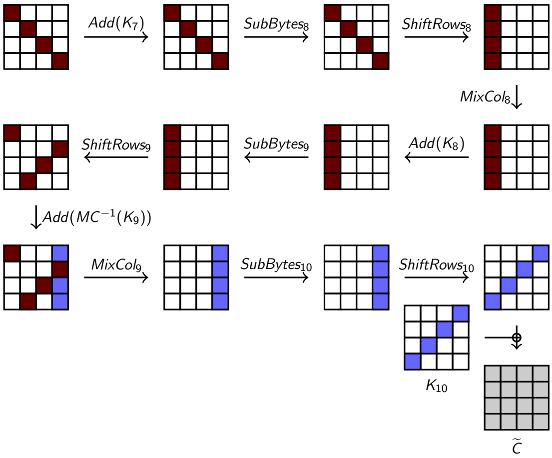


Figure : Brown bytes are constant due to the fault. Blue bytes can be computed from the ciphertext and a guess on K_{10}.

Results of attack

- Considering ℓ faults the number of possible hypotheses for 4 bytes of key is $2^{32-8(\ell-1)}$
- 5 faults leave only the correct key
- Computation phase costs $4\ell 2^{32}$
- Complexity : $2^{128-32(\ell-1)} + 4\ell 2^{32}$
- In our paper, we show that it is possible to retrieve the correct key even with failed injections

6-th round

- The fault is injected between the 6-th `MixColumns` and the 7-th `ShiftRows`
- Three diagonals of the state are stucked-at an unknown value

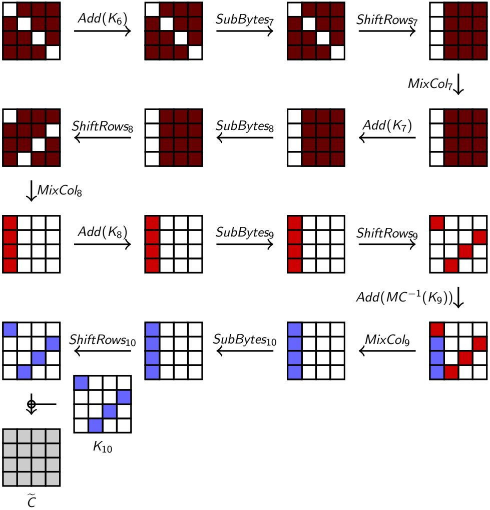


Figure : Brown bytes are constant due to the fault. Light red bytes have a reduced number of possible values. Blue bytes can be computed from the ciphertext and a guess on K_{10}.

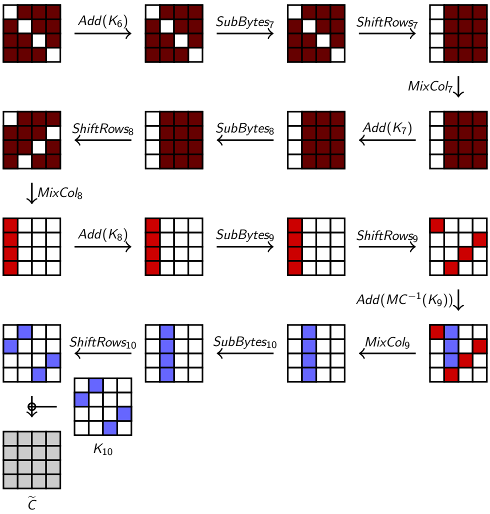


Figure : Brown bytes are constant due to the fault. Light red bytes have a reduced number of possible values. Blue bytes can be computed from the ciphertext and a guess on K_{10}.

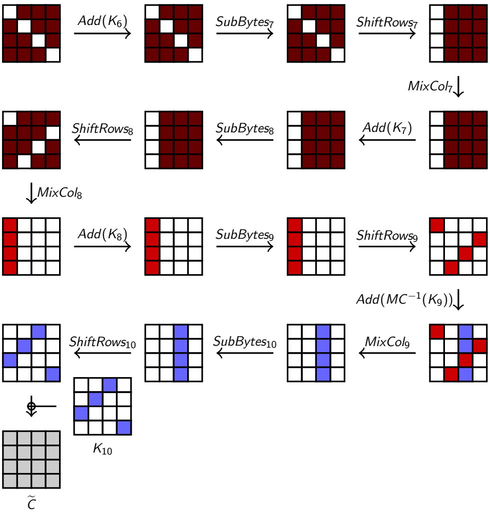


Figure : Brown bytes are constant due to the fault. Light red bytes have a reduced number of possible values. Blue bytes can be computed from the ciphertext and a guess on K_{10}.

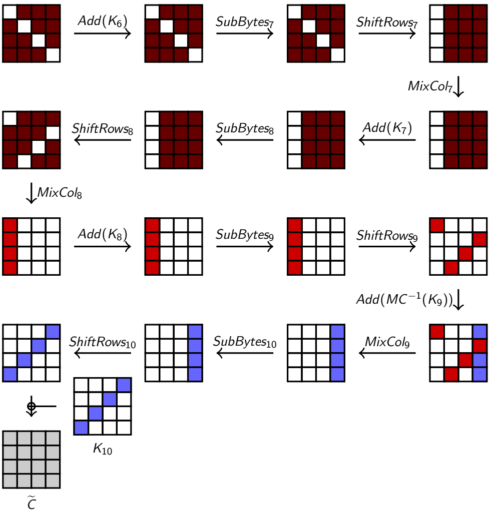


Figure : Brown bytes are constant due to the fault. Light red bytes have a reduced number of possible values. Blue bytes can be computed from the ciphertext and a guess on K_{10}.

Attack on the 6-th round

- For each $\widetilde{C}$ we have the values of light red bytes for each key hypothesis on each diagonal $\implies 4 \times 2^{32}$
- Guess which ciphertexts collide $\implies$ look for a collision for each diagonal and find the possible keys

Results

- τ collisions amongst ℓ faults
- Complexity: $\binom{\ell}{\tau} 2^{128-32(\ell-1)} + 4\ell 2^{32}$
- High number of faults: with 245 faults, success prob $= 54\%$ and around 2^{32} candidates left
- In our paper, we show that it is possible to retrieve the correct key even with failed injections

Conclusion/perspective

- Conclusion
  - Attack without correct ciphertexts/messages based on the non-uniformity of injected faults
  - Applicable on the last 4 rounds of AES
- Perspectives
  - Weaken fault models?
  - Improve complexity/decrease number of faults?

$\ddot{\smile}$ Thank you for your attention!