Reverse Engineering of a Secret AES-like Cipher by Ineffective Fault Analysis

Antoine Wurcker  
antoine.wurcker@xlim.fr

Christophe Clavier  
christophe.clavier@unilim.fr

Université de Limoges

FDTC 2013

20-08-2013
Outline

1. Introduction
   - Advanced Encryption Standard
   - Ineffective Fault Analysis

2. Scope of the Attack
   - Modifications on AES
   - Constraints on Attacker

3. Attack Steps

4. Conclusion
   - Global Results
   - Future Works
Outline

1 Introduction
   - Advanced Encryption Standard
   - Ineffective Fault Analysis

2 Scope of the Attack
   - Modifications on AES
   - Constraints on Attacker

3 Attack Steps

4 Conclusion
   - Global Results
   - Future Works
AES Datapath

\[ M \]

\[ \rightarrow \]

\[ \text{AddRoundKey}(K_0) \]

\[ \rightarrow S_0 \]

\[ r = 0 \]

\[ S_{r-1} \]

\[ \downarrow \text{SubBytes} \]

\[ \downarrow \text{ShiftRows} \]

\[ \downarrow \text{MixColumns} \]

\[ \rightarrow \text{AddRoundKey}(K_r) \]

\[ \rightarrow S_r \]

\[ r = 1, \ldots, 9 \]

\[ S_9 \]

\[ \downarrow \text{SubBytes} \]

\[ \downarrow \text{ShiftRows} \]

\[ \rightarrow \text{AddRoundKey}(K_{10}) \]

\[ \rightarrow C \]

\[ r = 10 \]

**Figure:** The AES encryption path.
AES KeySchedule

Figure: The AES key schedule.
Ineffective Fault Analysis

Fault Model: Stuck at 0 a precise byte.

Fault effect:
- Ciphertext not modified $\Rightarrow$ the value was already 0.
- Ciphertext modified $\Rightarrow$ the value was not 0.

Remark:
- IFA by-pass dual-execution countermeasure.
**Figure**: Example of no-occurrence of IFA.
Figure: Example of occurrence of IFA.
Notations

$m_i$: Byte number $i$ of the input plaintext $M$.

$c_i$: Byte number $i$ of the output ciphertext $C$.

$K_r$: 128-bit Key of round number $r$.

$k_{r,i}$: Byte number $i$ of the round key $K_r$.

$S()$: Function SubBytes.

$S^{-1}(0)$: Preimage of 0 value by S-Box table

$\mu_i = k_{0,i} \oplus S^{-1}(0)$

$X_r = \{x_{r,0}, \ldots, x_{r,15}\}$: Input state of SubBytes step of round $r$

$Y_r = \{y_{r,0}, \ldots, y_{r,15}\}$: Input state of ShiftRows step of round $r$

$Z_r = \{z_{r,0}, \ldots, z_{r,15}\}$: Input state of MixColumns step of round $r$

$T_r = \{t_{r,0}, \ldots, t_{r,15}\}$: Input state of AddRoundKey step of round $r$
Outline

1. Introduction
   - Advanced Encryption Standard
   - Ineffective Fault Analysis

2. Scope of the Attack
   - Modifications on AES
   - Constraints on Attacker

3. Attack Steps

4. Conclusion
   - Global Results
   - Future Works
The modifications allowed have to respect the constraints from the NIST document describing the AES:

1. The SBOX operation is a permutation table.  
   $\Rightarrow$ $256!$ possible SBOX ($\approx 2^{1684}$).

2. The ShiftRows operation keeps shifting rows.  
   $\Rightarrow$ $2^8$ possible ShiftRows.

3. The MixColumns matrix stays circulant with four parameters ($\neq 0$).  
   $\Rightarrow$ $255^4$ possible MixColumns ($\approx 2^{32}$).

4. The RotWord operation keeps shifting word.  
   $\Rightarrow$ $2^2$ possible RotWord.

5. The $Rcon$ vectors keeps the form $[\rho^{r-1}, 0, 0, 0]$.  
   $\Rightarrow$ $2^8$ possible sets of $Rcon$ vectors.
Modifications on AES

Figure: ShiftRows parameters.

Figure: MixColumns matrix.

Figure: RotWord parameter.

Figure: Rcon[r] parameter.
Constraints on Attacker

We placed main constraints on an attacker:

1. The SBOX table is unknown.
2. The MixColumns coefficients are unknown.
3. The ShiftRows coefficients are unknown.
4. The fault can only be applied on SBOX output.
5. The key $K$ is unknown.

The Key-Schedule operation is also constrained:

1. RotWord coefficient is unknown.
2. Rcon parameter is unknown.
3. Unavailable to fault injection (e.g. pre-computation).
Outline

1. Introduction
   - Advanced Encryption Standard
   - Ineffective Fault Analysis

2. Scope of the Attack
   - Modifications on AES
   - Constraints on Attacker

3. Attack Steps

4. Conclusion
   - Global Results
   - Future Works
Retrieving $K_0$ up to a Constant Byte

We obtain $\mu_i = k_0$, by exhausting $m_i$ while faulting the output of the $i$th S-Box of the first round. Eventually an IFA occurs and we obtain the equation:

$$S(m_i \oplus k_0, i) = 0$$

$m_i \oplus k_0, i = S_{-1}(0)$

$m_i = k_0, i \oplus S_{-1}(0)$

We retrieve every $\mu_i$ values by applying this method on each position.

$\Rightarrow$ The set of candidates for $K_0$ is reduced from $2^{128}$ to $2^8$. 
Retrieving $K_0$ up to a Constant Byte

We obtain $\mu_i = k_{0,i} \oplus S^{-1}(0)$ by exhausting $m_i$ while faulting the output of $i^{th}$ S-Box of first round.

Eventually an IFA occurs and we obtain the equation:

$$S(m_i \oplus k_{0,i}) = 0$$

$$m_i \oplus k_{0,i} = S^{-1}(0)$$

$$m_i = k_{0,i} \oplus S^{-1}(0)$$

$$m_i = \mu_i$$

We retrieve every $\mu_i$ values by applying this method on each position.

$\Rightarrow$ The set of candidates for $K_0$ is reduced from $2^{128}$ to $2^8$. 
Lemma: “Choosing” S-Box Input

Lemma: “Choosing” S-Box Input
Lemma: "Choosing" S-Box Input

Lemma

The knowledge of $\mu_i$ values allows us to choose any value $x_{1,i}$ up to the constant value $S^{-1}(0)$.

Proof.

Playing value $m_i = v \oplus \mu_i$ implies that:

\[ x_{1,i} = m_i \oplus k_{0,i} \]
\[ x_{1,i} = v \oplus \mu_i \oplus k_{0,i} \]
\[ x_{1,i} = v \oplus S^{-1}(0) \oplus k_{0,i} \oplus k_{0,i} \]
\[ x_{1,i} = v \oplus S^{-1}(0) \]

Remark: if $v = 0$ it implies $x_{1,i} = S^{-1}(0) \Rightarrow y_{1,i} = 0$
Reversing ShiftRows Operation

First step: Playing random messages until an IFA occurs.

Second step: Playing previous message with only one byte modified each time.

On each row 1 position will break the IFA when 3 will not.

We play the second step until we get the 4 values that break IFA, revealing the 4 ShiftRows parameters.

⇒ The ShiftRows operation is reversed.
Reversing ShiftRows Operation

Fault position: first S-Box of second round.

First step: Playing random messages until an IFA occurs.

Second step: Playing previous message with only one byte modified each time.

On each row 1 position will break the IFA when 3 will not.

We play the second step until we get the 4 values that break IFA, revealing the 4 ShiftRows parameters.

⇒ The ShiftRows operation is reversed.
Reversing ShiftRows Operation

\[
\begin{align*}
\bigoplus K_0 & \quad \Rightarrow \quad SB \\
\Rightarrow K_0 & \quad \Rightarrow \quad SB \\
\end{align*}
\]

\[
\begin{align*}
\bigoplus K_1 & \quad \Rightarrow \quad MC \\
\end{align*}
\]

Figure: Position of IFA
Reversing ShiftRows Operation

Figure: Proof: shift parameter of second row is not 0
Reversing ShiftRows Operation

Figure: Proof: shift parameter of second row is 1
Reversing ShiftRows Operation

Figure: Proof: shift parameter of second row is not 2
Reversing ShiftRows Operation

Figure: Proof: shift parameter of second row is not 3
Lemma: Retrieving $mk_{i,j}$ Values
Lemma: Retrieving $mk_{i,j}$ Values

**Definition**

$mk_{i,j}$ are particular values that verifies: $\alpha_j \ast S(mk_{i,j}) = k_{1,i} \oplus S^{-1}(0)$

**Lemma**

*The knowledge of $\mu_i$ values and ShiftRows parameters allows us to calculate any value $mk_{i,j}$ up to $S^{-1}(0)$.*

**Proof.**

We can play a full 0 state as input of first round MixColumns, except the position $t = \lfloor i/4 \rfloor + 4 \ast j$. This induces, with chosen $\nu$:

$$x_{2,i} = \alpha_j \ast S(\nu \oplus S^{-1}(0)) \oplus k_{1,i}$$

When $\nu$ provokes an IFA on $y_{2,i}$: $\nu = mk_{i,j} \oplus S^{-1}(0)$
Goal: find multiplicative order of $\beta_{i,j} = \alpha_i \alpha_j$.

Remark: We place ourselves in cases where at least one of the 6 orders of values $\beta_{i,j}$ is equal to 255. It's concerning 95.28% of cases.

Example: recovery of order of $\beta_{1,2}$.

Equation given by an IFA on first S-Box of second round:

$$\begin{align*}
x_{2,0} &= \alpha_0 \ast z_{1,0} \oplus \alpha_1 \ast z_{1,1} \oplus \alpha_2 \ast z_{1,2} \oplus \alpha_3 \ast z_{1,3} \oplus k_{1,0} \\
x_{2,0} &= S^{-1}(0)
\end{align*}$$
Reducing MixColumns by Retreiving Cycles Orders (1/3)

Goal: find multiplicative order of $\beta_{i,j} = \frac{\alpha_i}{\alpha_j}$.

Remark

We place ourselves in case where at least one of the 6 orders of values $\beta_{i,j}$ is equals to 255. It’s concerning to 95.28% of cases.

Example: recovery of order of $\beta_{1,2}$.

Equation given by an IFA on first S-Box of second round:

\[
\begin{align*}
\begin{cases}
x_{2,0} &= \alpha_0 \ast z_{1,0} \oplus \alpha_1 \ast z_{1,1} \oplus \alpha_2 \ast z_{1,2} \oplus \alpha_3 \ast z_{1,3} \oplus k_{1,0} \\
x_{2,0} &= S^{-1}(0)
\end{cases}
\Rightarrow \alpha_0 \ast z_{1,0} \oplus \alpha_1 \ast z_{1,1} \oplus \alpha_2 \ast z_{1,2} \oplus \alpha_3 \ast z_{1,3} \oplus k_{1,0} &= S^{-1}(0)
\end{align*}
\]
Reducing MixColumns by Retrieving Cycles Orders (2/3)

Knowledge of $mk_{0,0}$ allows to play a plaintext byte value inducing:

$$z_{1,0} = S(mk_{0,0}) \Rightarrow \alpha_0 \ast z_{1,0} = k_{1,0} \oplus S^{-1}(0)$$

That clean $K_1$ and $S^{-1}(0)$ from previous equation:

$$\alpha_0 \ast z_{1,0} \oplus \alpha_1 \ast z_{1,1} \oplus \alpha_2 \ast z_{1,2} \oplus \alpha_3 \ast z_{1,3} \oplus k_{1,0} = S^{-1}(0)$$

$$k_{1,0} \oplus S^{-1}(0) \oplus \alpha_1 \ast z_{1,1} \oplus \alpha_2 \ast z_{1,2} \oplus \alpha_3 \ast z_{1,3} \oplus k_{1,0} = S^{-1}(0)$$

$$\alpha_1 \ast z_{1,1} \oplus \alpha_2 \ast z_{1,2} \oplus \alpha_3 \ast z_{1,3} = 0$$

Knowledge of $\mu_i$ values allows to play a plaintext byte value inducing:

$$z_{1,3} = S(S^{-1}(0)) = 0 \Rightarrow \alpha_3 \ast z_{1,3} = 0$$

That clean $\alpha_3$ from previous equation:

$$\alpha_1 \ast z_{1,1} \oplus \alpha_2 \ast z_{1,2} \oplus \alpha_3 \ast z_{1,3} = 0$$

$$\alpha_1 \ast z_{1,1} \oplus \alpha_2 \ast z_{1,2} = 0$$
Reducing MixColumns by Retrieving Cycles Orders

We use a random value $\theta_{1,2}^{(0)}$:

$$z_{1,1} = \tau_{1,2}^{(0)} = S(\theta_{1,2}^{(0)} \oplus k_{0,0})$$

We exhaust $z_{1,2}$ until an IFA occurs revealing the value $\theta_{1,2}^{(1)}$ such as:

$$z_{1,2} = \tau_{1,2}^{(1)} = S(\theta_{1,2}^{(1)} \oplus k_{0,0})$$

We then reveal the sequence of $\theta_{1,2}^{(k)}$ that verifies:

$$\alpha_1 \ast \tau_{1,2}^{(k)} \oplus \alpha_2 \ast \tau_{1,2}^{(k+1)} = 0$$

$$\tau_{1,2}^{(k+1)} = \beta_{1,2} \ast \tau_{1,2}^{(k)}$$

$$\Rightarrow \tau_{1,2}^{(k)} = (\beta_{1,2})^k \ast \tau_{1,2}^{(0)}$$

Eventually $\tau_{1,2}^{(n)} = \tau_{1,2}^{(0)}$ revealing that $(\beta_{1,2})^n = 1$. $n_{1,2} = n$, order of $\beta_{1,2}$.
Exploiting Data from Orders Retrieval

For each candidate for \( \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \) we are now able to test order of every \( \beta_{i,j} \) and drop the solutions that do not verify found orders \( n_{i,j} \).

We imposed that at least one order is equals to 255, it induces that during orders recovery we produced a sequence of 255 values \( \{\theta_{i,j}^{(0)}, \ldots, \theta_{i,j}^{(255)}\} \). That particular sequence will be set as reference for further steps and noted \( \{\theta^{(0)}, \ldots, \theta^{(255)}\} \). The concerned \( \beta_{i,j} \) will also be noted \( \beta \).

Then we know that:

\[
\tau^{(i)} = S(\theta^{(i)} \oplus k_{0,0})
\]

\[
\tau^{(i)} = \beta \cdot \tau^{(i-1)}
\]
Lemma: Relation $K_1 - K_0$
Lemma: Relation $K_1-K_0$

This reduction of MixColumns candidates will use particular properties brought by KeySchedule scheme:

**Lemma**

For $i \in \{0, 4, 1, 5, 2, 6, 3, 7\}$, we have $k_{1,i} \oplus k_{1,i+8} = \mu_{i+4} \oplus \mu_{i+8}$.

**Proof.**

\[
\begin{align*}
  k_{1,i+4} &= k_{1,i} \oplus k_{0,i+4} \\
  k_{1,i+8} &= k_{1,i+4} \oplus k_{0,i+8}
\end{align*}
\]

$\implies k_{1,i} \oplus k_{1,i+8} = k_{0,i+4} \oplus k_{0,i+8} = \mu_{i+4} \oplus \mu_{i+8}$
Reducing MixColumns Using $K_1$ Relations (1/2)

We will force the $K_0 - K_1$ relation to appear in IFA equations. As in previous step we use $m_{K_0,0}$ knowledge to clean $K_1$ and $S^{-1}(0)$:

$$\alpha_{0} \ast z_{1,0} \oplus \alpha_{1} \ast z_{1,1} \oplus \alpha_{2} \ast z_{1,2} \oplus \alpha_{3} \ast z_{1,3} \oplus k_{1,0} = S^{-1}(0)$$

Then we use knowledge of $m_{K_1,1,i}$ and $m_{K_2,2,i+8}$ to have $z_{1,1} = S(m_{K_1,1,i})$ and $z_{1,2} = S(m_{K_2,2,i+8})$:

$$\alpha_{1} \ast z_{1,1} \oplus \alpha_{2} \ast z_{1,2} \oplus \alpha_{3} \ast z_{1,3} = 0$$

$$k_{1,i} \oplus S^{-1}(0) \oplus k_{1,i+8} \oplus S^{-1}(0) \oplus \alpha_{3} \ast z_{1,3} = 0$$

$$\mu_{i+4} \oplus \mu_{i+8} \oplus \alpha_{3} \ast z_{1,3} = 0$$

Then we exhaust value for $z_{1,3}$ until we got an IFA.
Reducing MixColumns Using $K_1$ Relations

We will force the $K_0-K_1$ relation to appear in IFA equations. As in previous step we use $mk_{0,0}$ knowledge to clean $K_1$ and $S^{-1}(0)$:

$$\alpha_0 \ast z_{1,0} \oplus \alpha_1 \ast z_{1,1} \oplus \alpha_2 \ast z_{1,2} \oplus \alpha_3 \ast z_{1,3} \oplus k_{1,0} = S^{-1}(0)$$

$$k_{1,0} \oplus S^{-1}(0) \oplus \alpha_1 \ast z_{1,1} \oplus \alpha_2 \ast z_{1,2} \oplus \alpha_3 \ast z_{1,3} \oplus k_{1,0} = S^{-1}(0)$$

$$\alpha_1 \ast z_{1,1} \oplus \alpha_2 \ast z_{1,2} \oplus \alpha_3 \ast z_{1,3} = 0$$

Then we use knowledge of $mk_{1,i}$ and $mk_{2,i+8}$ to have $z_{1,1} = S(mk_{1,i})$ and $z_{1,2} = S(mk_{2,i+8})$:

$$\alpha_1 \ast z_{1,1} \oplus \alpha_2 \ast z_{1,2} \oplus \alpha_3 \ast z_{1,3} = 0$$

$$k_{1,i} \oplus S^{-1}(0) \oplus k_{1,i+8} \oplus S^{-1}(0) \oplus \alpha_3 \ast z_{1,3} = 0$$

$$k_{1,i} \oplus k_{1,i+8} \oplus \alpha_3 \ast z_{1,3} = 0$$

$$\mu_{i+4} \oplus \mu_{i+8} \oplus \alpha_3 \ast z_{1,3} = 0$$

Then we exhaust value for $z_{1,3}$ until we got an IFA.
Reducing MixColumns Using $K_1$ Relations

We recognise the message byte inducing the colliding $z_{1,3}$ as a $\theta(p)$ value, then we know that $z_{1,3} = \tau(p)$:

\[
\mu_i+4 \oplus \mu_i+8 \oplus \alpha_3 \ast \tau(p) = 0
\]
\[
\mu_i+4 \oplus \mu_i+8 \oplus \alpha_3 \ast \beta^p \ast \tau(0) = 0
\]
\[
\tau(0) = \frac{\mu_i+4 \oplus \mu_i+8}{\alpha_3 \ast \beta^p}
\]

That type of relations constraint MixColumns parameters.

**Lemma**

*Two equations of previous step allows to reduce the set of candidates for MixColumns parameters to 255 elements.*
Retrieving MixColumns and RotWord parameters

In this step we use two types of equations combined:

\[
\begin{align*}
    k_1,0 &= k_0,0 \oplus S(k_0,12 + \eta) \oplus \rho_0 \\
    k_1,0 &= \alpha_j \ast S(mk_0, j \oplus k_0,0) \oplus S^{-1}(0) \\
    S(k_0,12 + \eta) &= k_0,0 \oplus S^{-1}(0) \oplus 1 \oplus \alpha_j \ast S(mk_0, j \oplus k_0,0) \\
    S(k_0,12 + \eta) &= \mu_0 \oplus \alpha_j \ast \tau(q_1) \\
    \end{align*}
\]

For each MixColumns parameter candidate we are able to calculate \(S(k_0,12 + \eta)\) and recognise it as a known \(\tau(q_2)\) value:

\[
\begin{align*}
    \Rightarrow S(k_0,12 + \eta) &= \tau(q_2) = S(\theta(q_2) \oplus k_0,0) \\
    \Rightarrow k_0,12 + \eta &= \theta(q_2) \oplus k_0,0 \\
    \Rightarrow \theta(q_2) &= \mu_0 \oplus \mu_12 + \eta
\end{align*}
\]

Then we get only 4 valid solutions, a second equation let only 1.
Retrieving MixColumns and RotWord parameters

In this step we use two types of equations combined:

\[
\begin{align*}
\left\{
 k_{1,0} &= k_{0,0} \oplus S(k_{0,12+\eta}) \oplus \rho^0 \\
 k_{1,0} &= \alpha_j \ast S(\eta_{mk_{0,j} \oplus k_{0,0}}) \oplus S^{-1}(0)
\end{align*}
\]

\(\Rightarrow S(k_{0,12+\eta}) = k_{0,0} \oplus S^{-1}(0) \oplus 1 \oplus \alpha_j \ast S(\eta_{mk_{0,j} \oplus k_{0,0}})\)

\(\Rightarrow S(k_{0,12+\eta}) = \mu_0 \oplus 1 \oplus \alpha_j \ast \tau(q_1)\)

For each MixColumns parameter candidate we are able to calculate \(S(k_{0,12+\eta})\) and recognise it as a known \(\tau(q_2)\) value:

\(\Rightarrow S(k_{0,12+\eta}) = \tau(q_2) = S(\theta(q_2) \oplus k_{0,0})\)

\(\Rightarrow k_{0,12+\eta} = \theta(q_2) \oplus k_{0,0}\)

\(\Rightarrow \theta(q_2) = \mu_0 \oplus \mu_{12+\eta}\)

Then we got only 4 valid solutions, a second equation let only 1.
We are now able to calculate $k_1, 4$, due to equations from KeySchedule:

$$k_1, 0 = k_0, 0 \oplus \tau(q_2) \oplus 1 \oplus k_0, 4 \Rightarrow k_1, 4 = \mu_0 \oplus S^{-1}(0) \oplus \tau(q_2) \oplus 1 \oplus \mu_4 \oplus S^{-1}(0) \Rightarrow k_1, 4 = \tau(q_2) \oplus 1 \oplus \mu_0 \oplus \mu_4$$

We then use $k_1, 4$ to derive $S^{-1}(0)$ from a $mk_i,j$ equation:

$$k_1, 4 = \alpha_j^* S(\theta(q_3) \oplus k_0, 0) \oplus S^{-1}(0)$$
$$S^{-1}(0) = \alpha_j^* \tau(q_3) \oplus k_1, 4$$

Remark

We are now able to infer the values of: S-Box, $K_0$ and $K_1$. 

Antoine Wurcker  (Université de Limoges)
Retrieving $S^{-1}(0)$

We are now able to calculate $k_{1,4}$, due to equations from KeySchedule:

\[
\begin{align*}
    k_{1,0} &= k_{0,0} \oplus \tau(q_2) \oplus 1 \\
    k_{1,4} &= k_{1,0} \oplus k_{0,4} \\
\end{align*}
\]

\[\Rightarrow k_{1,4} = k_{0,0} \oplus \tau(q_2) \oplus 1 \oplus k_{0,4} \]

\[\Rightarrow k_{1,4} = \mu_0 \oplus S^{-1}(0) \oplus \tau(q_2) \oplus 1 \oplus \mu_4 \oplus S^{-1}(0) \]

\[\Rightarrow k_{1,4} = \tau(q_2) \oplus 1 \oplus \mu_0 \oplus \mu_4 \]

We then use $k_{1,4}$ to derive $S^{-1}(0)$ from a $mk_{i,j}$ equation:

\[k_{1,4} = \alpha_j \ast S(mk_{4,j} \oplus k_{0,0}) \oplus S^{-1}(0) \]

\[S^{-1}(0) = \alpha_j \ast S(\theta(q_3) \oplus k_{0,0}) \oplus k_{1,4} \]

\[S^{-1}(0) = \alpha_j \ast \tau(q_3) \oplus k_{1,4} \]

**Remark**

We are now able to infer the values of: S-Box, $K_0$ and $K_1$. 
Retrieving Rcon parameter

We know all AES parameters except $\rho$, that allows to control $T_2$ state. We exhaust $t_2, 0$ values until an IFA occurs on first S-Box of third round: $y_3, 0 = 0$

$$S(t_2, 0 \oplus k_2, 0) = 0$$

We learn $k_2, 0$ and then we can simply calculate $\rho$:

$$k_2, 0 = k_1, 0 \oplus S(k_1, 12 + \eta) \oplus \rho$$

$$\rho = k_1, 0 \oplus S(k_1, 12 + \eta) \oplus k_2, 0$$
Retrieving Rcon parameter

We know all AES parameters except \( \rho \), that allows to control \( T_2 \) state. We exhaust \( t_{2,0} \) values until an IFA occurs on first S-Box of third round:

\[
\begin{align*}
y_{3,0} &= 0 \\
S(x_{3,0}) &= 0 \\
S(t_{2,0} \oplus k_{2,0}) &= 0 \\
k_{2,0} &= t_{2,0} \oplus S^{-1}(0)
\end{align*}
\]

We learn \( k_{2,0} \) and then we can simply calculate \( \rho \):

\[
\begin{align*}
k_{2,0} &= k_{1,0} \oplus S(k_{1,12+\eta}) \oplus \rho \\
\rho &= k_{1,0} \oplus S(k_{1,12+\eta}) \oplus k_{2,0}
\end{align*}
\]
### Simulations Results

<table>
<thead>
<tr>
<th>Step</th>
<th># of faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieving $\mu^i$ values</td>
<td>2055.96</td>
</tr>
<tr>
<td>Retrieving ShiftRows</td>
<td>138.50</td>
</tr>
<tr>
<td>Retrieving $\beta_{i,j}$ orders</td>
<td>22339.80</td>
</tr>
<tr>
<td>Retrieving cross-orders relations</td>
<td>0</td>
</tr>
<tr>
<td>Retrieving $K_1$ relations</td>
<td>915.77</td>
</tr>
<tr>
<td>Retrieving MixColumns and RotWord</td>
<td>64.30</td>
</tr>
<tr>
<td>Retrieving $S^{-1}(0)$</td>
<td>0</td>
</tr>
<tr>
<td>Retrieving $Rcon$</td>
<td>127.5</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>25641.83</strong></td>
</tr>
</tbody>
</table>

Figure: Experimental results on an unprotected implementation.
## Simulations Results

<table>
<thead>
<tr>
<th>Step</th>
<th># of faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieving $\mu_i$ values</td>
<td>2055.96</td>
</tr>
<tr>
<td>Retrieving ShiftRows</td>
<td>138.50</td>
</tr>
<tr>
<td>Retrieving $\beta_{i,j}$ orders</td>
<td>22339.80</td>
</tr>
<tr>
<td>Retrieving cross-orders relations</td>
<td>0</td>
</tr>
<tr>
<td>Retrieving $K_1$ relations</td>
<td>915.77</td>
</tr>
<tr>
<td>Retrieving MixColumns and RotWord</td>
<td>64.30</td>
</tr>
<tr>
<td>Retrieving $S^{-1}(0)$</td>
<td>0</td>
</tr>
<tr>
<td>Retrieving Rcon</td>
<td>127.5</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>25641.83</strong></td>
</tr>
</tbody>
</table>

**Figure:** Experimental results on an unprotected implementation.
Outline

1. Introduction
   - Advanced Encryption Standard
   - Ineffective Fault Analysis

2. Scope of the Attack
   - Modifications on AES
   - Constraints on Attacker

3. Attack Steps

4. Conclusion
   - Global Results
   - Future Works
Global Results

We by-pass the dual-execution countermeasure.

In 95.28% of cases we retrieve the whole algorithm specifications in an average of $\sim 25k$ required fault number.

With reasonable over-costs, we are able to extend our attack to two harder configurations:

1. Full entropy MixColumns matrix: MixColumns matrix is no more circulant and is composed of 16 independent parameters. This new attack is valid in 99.99% of cases (instead of 95.28%).

2. Extended Rcon parameters: Rcon is no more dependant from an unique value $\rho$ but each round have it’s own independent value.
Future Works

- Search tricks in order to reduce fault number.
- Extend attack to 5% remaining cases.
- Adapt attack when fault is done on exclusive-or (⊕) operations instead of table lookup.
- Study adaptability of this attack in presence of different type of counter-measures.
- Study how the knowledge of the key facilitates the attack (a decryption function available on the device give ability to find the key).
Thank you for your attention.
Any Question?
Proof: Only 255 MixColumns Candidates Remains

Proof.

\[
\begin{align*}
\tau^{(0)} &= \frac{\mu_{1,i+4} \oplus \mu_{1,i+8}}{\alpha_3 \beta_{P_1}} \\
\tau^{(0)} &= \frac{\mu_{1,i+8} \oplus \mu_{1,i+12}}{\alpha_3 \beta_{P_2}}
\end{align*}
\]

\[\Rightarrow \beta_{P_1} - \beta_{P_2} = \frac{\mu_{1,i+4} \oplus \mu_{1,i+8}}{\mu_{1,i+8} \oplus \mu_{1,i+12}} \]

\[\Rightarrow \frac{\alpha_{i^*}^{P_1} - \alpha_{j^*}^{P_2}}{\alpha_{j^*}^{P_1} - \alpha_{j^*}^{P_2}} = \frac{\mu_{1,i+4} \oplus \mu_{1,i+8}}{\mu_{1,i+8} \oplus \mu_{1,i+12}} \]

\[\Rightarrow \alpha_{i^*}^{P_1} - \alpha_{j^*}^{P_2} = \frac{\mu_{1,i+4} \oplus \mu_{1,i+8}}{\mu_{1,i+8} \oplus \mu_{1,i+12}} * \alpha_{j^*}^{P_1} - \alpha_{j^*}^{P_2}
\]

It remains 255 valid pairs \((\alpha_{i^*}, \alpha_{j^*})\). Already acquired relations extend this property to other MixColumns parameters.

Remark

For each of 255 candidates for MixColumns parameters we are able to calculate \(\tau^{(0)}\) and \(\beta\), then the whole sequence \((\tau^{(k)})_{k=0,\ldots,254}\).
Simulation’s Oracle

We ran simulations using an oracle taking as input:

- the parameters of the modified AES
- the round and S-Box position that is considered as faulted
- the message we decide to play

it gives back a boolean value indicating if the fault was ineffective or not.