IIS Tutorial - Advanced Fault Attack

Robert Primas, IIS Tutor @ IAIK

25th November 2015

Content

Organisatorial Information

- Tutorial slides are already available on the wiki
- No specification needed for advanced Fault Attack
- Data for DPA and advanced Fault attack will be released soon

Notation

- Δ denotes the XOR difference between two values

$$\Delta_{A,B} = A \oplus B$$
$$\Delta_{5,7} = 2$$
$$\Delta_{6,6} = 0$$

Notation

- AES State in round [Round] after [Operation]
  (128-bit = 4 x 4 x 8 bytes)

$$\begin{pmatrix} A_0 & A_4 & A_8 & A_{12} \\ A_1 & A_5 & A_9 & A_{13} \\ A_2 & A_6 & A_{10} & A_{14} \\ A_3 & A_7 & A_{11} & A_{15} \end{pmatrix}$$

Valid State: $S_{[Round],[Operation]}$

Faulty State: $F_{[Round],[Operation]}$

Notation

- Differential AES State in round [Round] after [Operation]

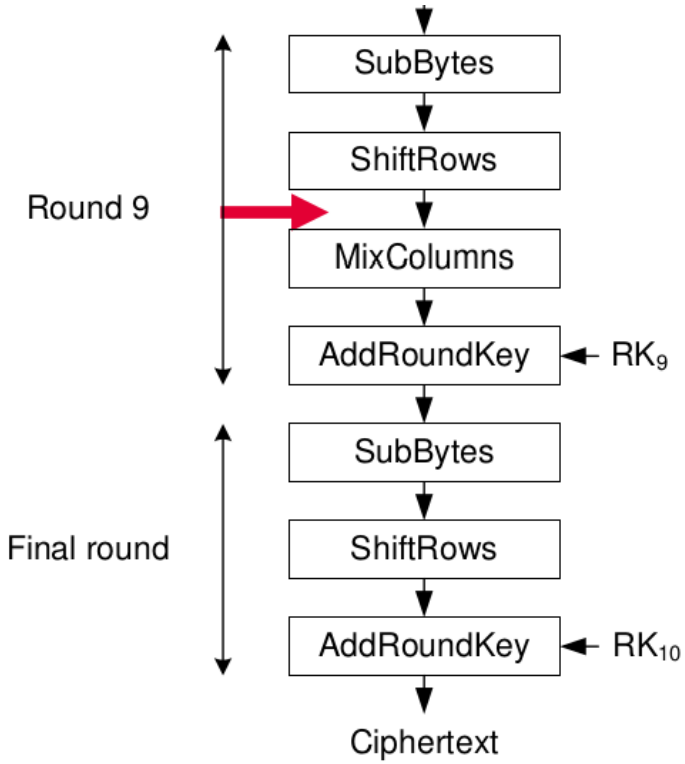
$$\begin{pmatrix} \Delta_0 & \Delta_4 & \Delta_8 & \Delta_{12} \\ \Delta_1 & \Delta_5 & \Delta_9 & \Delta_{13} \\ \Delta_2 & \Delta_6 & \Delta_{10} & \Delta_{14} \\ \Delta_3 & \Delta_7 & \Delta_{11} & \Delta_{15} \end{pmatrix}$$

$$\Delta_{[Round],[Operation]}$$

$$= S_{[Round],[Operation]} \oplus F_{[Round],[Operation]}$$

Attack Overview

- Induce random (non-zero) error in one byte before MixColums in Round 9
- Collect valid/faulty ciphertext pairs
- Recover key bytes...



Fault Propagation

- Induce random (non-zero) error
- Error is in one byte of the state
- 255 possibilities ($2^8 - 1$)
- Before MixColums and after ShiftRows in Round 9
- Exact position of error not known
- Here the error is induced in the first byte

$$\begin{pmatrix} \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$\Delta_{9,ShiftRows}$

Fault Propagation

- MixColumns spreads 1 byte difference over the whole column
- Now 4 bytes contain differences
- The 4 differences are not equal
- Still only 255 possibilities because MixColumns is linear

$$\begin{pmatrix} \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 \end{pmatrix}$$

$\Delta_{9,MixColumns}$

Fault Propagation

- AddRoundKey is just XOR with constant value
- No effect on differences

$$\begin{pmatrix} \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 \end{pmatrix}$$

$\Delta_{9,AddRoundKey}$

Fault Propagation

- SubBytes changes the values of the differences
- No additional differences are introduced

$$\begin{pmatrix} \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 \end{pmatrix}$$

$\Delta_{10,SubBytes}$

Fault Propagation

- ShiftRows exchanges positions of bytes in each row of the state

$$\begin{pmatrix} \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta \\ 0 & 0 & \Delta & 0 \\ 0 & \Delta & 0 & 0 \end{pmatrix}$$

$\Delta_{10,\mathit{ShiftRows}}$

Fault Propagation

- MixColumns is skipped (round 10)
- AddRoundKey has no effect on differences
- This is the observable output difference

$$\begin{pmatrix} \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta \\ 0 & 0 & \Delta & 0 \\ 0 & \Delta & 0 & 0 \end{pmatrix}$$

$\Delta_{10,AddRoundKey}$

Example

1. $$S_{9,ShiftRows} = \begin{pmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{pmatrix}$$
2. Induce fault “1E” in first byte
3. $$F_{9,ShiftRows} = \begin{pmatrix} 99 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{pmatrix}$$

Example

4.

$$S_{10,AddRoundKey} = \begin{pmatrix} 39 & 02 & DC & 19 \\ 25 & DC & 11 & 6A \\ 84 & 09 & 85 & 0B \\ 1D & FB & 97 & 32 \end{pmatrix}$$

$$F_{10,AddRoundKey} = \begin{pmatrix} DE & 02 & DC & 19 \\ 25 & DC & 11 & 3B \\ 84 & 09 & C2 & 0B \\ 1D & 62 & 97 & 32 \end{pmatrix}$$

5. Observable differences are “E7”, “99”, “47“ and “51”

Fault Propagation Cont.

- So we know that...

$$\begin{pmatrix} \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \Longrightarrow \begin{pmatrix} \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta \\ 0 & 0 & \Delta & 0 \\ 0 & \Delta & 0 & 0 \end{pmatrix}$$

- But also...

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \Longrightarrow \begin{pmatrix} \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta \\ 0 & 0 & \Delta & 0 \\ 0 & \Delta & 0 & 0 \end{pmatrix}$$

- We need to consider all four cases
- Be aware of what happens if the fault is induced in column 2-4 (not needed for this assignment)

Recover Key

- Generate 1020 (4 x 255) possible output differentials for MixColumns in Round 9
- Guess 4 key bytes and calculate the output difference for MixColumns in Round 9
- Make list of all key combinations that result in one of the 1020 possible differentials
- Repeat attack with different valid/faulty ciphertext pairs and narrow down key combinations

Recover Key

- Guessing 4 key bytes is quite some work (2^{32} operations)
- Calculate SubBytes^{-1} for one key byte
- Check if differentials are actually possible
- Repeat for other key bytes

Remarks

- Basis for attack from G. Piret and J. Quisquater [PQ03]
- Also works if fault is instead induced somewhere between MixColumns in round 8 and MixColumns in round 9 (not needed for this assignment)
- Already quite practical attack
- Can be extended to an even more powerful attack easily

Submissions

For Final Delivery

- Implement simple Fault Attack
- Implement advanced Fault Attack
  - You only need to find 4 bytes of the key (error in first column)
  - The other 12 bytes of the last round key are provided by us
  - We know that simple brute force would work as well...
- Implement DPA

Submissions

- Programming language is Matlab/Octave or Sage
- Add Readme with instructions if necessary
- We need to be able to run your code
- Write summary for each solved challenge
  (2-4 Pages in total)

Timeline

- $<$ 24th December - Give progress update to tutor
- 21th January - Final deliverable
- $\sim$ 28th January - Final interviews