Combined Fault and Side-Channel Attacks on the AES Key Schedule

François DASSANCE
Inside Secure

Alexandre VENELLI
Inside Secure

FDTC 2012
09/09/2012

DRIVING TRUST

Outline

Combined attack

- Combines a fault attack with a leakage analysis
- Main goal: attack implementations resistant against fault and leakage analysis
- New implementations + new countermeasures often necessary

Example of combined attack

Algorithm 1 Binary SPA-FA resistant exponentiation

Input: $x \in \mathbb{G}$ and $d = (d_{k-1}, \ldots, d_0)_2 \in \mathbb{N}$
Output: x^d

```
A ← x
R[0] ← x
R[1] ← 1
for i = 0 to k − 1 do
    R[d_i] ← R[d_i].A
    A ← A^2
end for
R[0] ← R[0].R[1]
if (R[0] ≠ A) then
    error
end if
return R[1]
```

Example of combined attack

Algorithm 1 Binary SPA-FA resistant exponentiation

Input: $x \in \mathbb{G}$ and $d = (d_{k-1}, \ldots, d_0)_2 \in \mathbb{N}$

Output: x^d

```
A ← x
R[0] ← x
R[1] ← 1
for i = 0 to k − 1 do
    R[d_i] ← R[d_i].A
    A ← A^2
end for
R[0] ← R[0].R[1]
if (R[0] ≠ A) then
    error
end if
return R[1]
```

Skip instruction

Outline

1. Combined attack
2. Related work on combined attacks
    1. Asymmetric cryptosystems
    2. Symmetric cryptosystems
    3. Roche et al.'s attack on AES
3. Combined attacks on AES key schedule
    1. Recursive structure of the key schedule
    2. RCON
    3. Affine transformation
4. Complexity of our attacks
5. Countermeasures
6. Conclusion

Asymmetric cryptosystems

- Fault Analysis + Simple Side-Channel Analysis

- Attack on atomic left-to-right exponentiation
  - Amiel, Villegas, Feix, Marcel - 2007

- Resistant algorithms for RSA and ECC
  - Schmidt, Tunstall, Avanzi, Kizhvatov, Kasper, Oswald - 2010

- Attack on scalar multiplication
  - Fan, Gierlichs, Vercauteren - 2011

Symmetric cryptosystems

- Fault Analysis + Differential Side-Channel Analysis
- Differential Behavioral Analysis: attack on non-masked AES
  - Robisson, Manet - 2007
- Attack on masked AES but not FA-protected. Reduce the DPA countermeasure of one order.
  - Clavier, Feix, Gagnerot, Rousselet - 2010
- Attack on AES FA-protected and with masking of any order
  - Roche, Lomné, Khalfallah - 2011

Roche et al. combined attack

- Principle:
  1. Repeatable fault on the 16 bytes of key state of round 9
  2. Record the power consumption curve
  3. Find a first-order correlation on the computation of the faulted ciphertext

- Main relation:

$$\widetilde{C_i^j} = SB\left(SB^{-1}\left(C_i^j \oplus k_{10}^j\right) \oplus e_9^j\right) \oplus k_{10}^j \oplus e_{10}^j$$

- Complexity to retrieve the whole key:
  - N faults and $2^{28}A$
  - A = any DSCA statistical function on N curves

Efficiency

	Combined attack	High-order DSCA
Number of curves	Few and fixed	A lot and increasing with the order of masking
Complexity of key retrieval algorithm	$2^{28}A$	$2^{12}A$

Remarks on Roche et al.

- Requires fault on the 16 bytes of the key
  - Not practical in all AES implementations
  - Not trivial with all fault injection techniques

- If a *stuck-at* fault model is considered, a masked bit induces a repeatability divided by 2

- High complexity of the key retrieval algorithm

Outline

1. Combined attack
2. Related work on combined attacks
    1. Asymmetric cryptosystems
    2. Symmetric cryptosystems
    3. Roche et al.'s attack on AES
3. Combined attacks on AES key schedule
    1. Recursive structure of the key schedule
    2. RCON
    3. Affine transformation
4. Complexity of our attacks
5. Countermeasures
6. Conclusion

Combined attacks on AES key schedule

- Attacks based on two properties of the key schedule:
  - Recursive structure
  - Use of constant values

- Our propositions improve:
  - The number of faults
  - The complexity of the key retrieval algorithm

Recursive structure (1)

- Round key K_9:

$$K_9^0 = K_8^0 \oplus RCON_9 \oplus SB(K_8^{13})$$
$$K_9^1 = K_8^1 \oplus SB(K_8^{14})$$
$$K_9^2 = K_8^2 \oplus SB\left(K_8^{15}\right)$$
$$K_9^3 = K_8^3 \oplus SB(K_8^{12})$$
$$K_9^j = K_8^j \oplus K_9^{j-4} \text{ for } 4 \leq j \leq 15$$

- Relations between faults on K_9
- Ex: fault e_9^0 in K_9^0 ➔ same fault on bytes 4, 8 and 12
- Relations between faults on K_{10}
- Ex: fault e_9^0 in K_9^0 ➔ $e_9^0 = e_{10}^0 = e_{10}^8$ and $e_{10}^4 = e_{10}^{12} = 0$

Recursive structure (2)

- Needs $4N$ faults
- Improvements on the key retrieval algorithm
- To retrieve K_{10}^0
  - Loop only on k_{10}^0 and e_9^0 as $e_{10}^0 = e_9^0$
  - Complexity for this byte: $2^{16}A$
- Once e_9^0 is found → e_9^4, e_9^8 and e_9^{12} are deduced
  - Simple loop on k_{10}^j for $j = 4,8,12$
  - Complexity for each of these 3 bytes: 2^8A
- Same method for K_9^1, K_9^2 and K_9^3
- Complexity for the whole key:

$$4 \times (2^{16} + 3 \times 2^8)A$$
$$= (2^{20} + 3 \times 2^{10})A$$

RCON (1)

- First column of K_9

$$K_9^0 = K_8^0 \oplus RCON_9 \oplus SB(K_8^{13})$$
$$K_9^4 = K_8^4 \oplus K_9^0$$
$$K_9^8 = K_8^8 \oplus K_9^4$$
$$K_9^{12} = K_8^{12} \oplus K_9^8$$

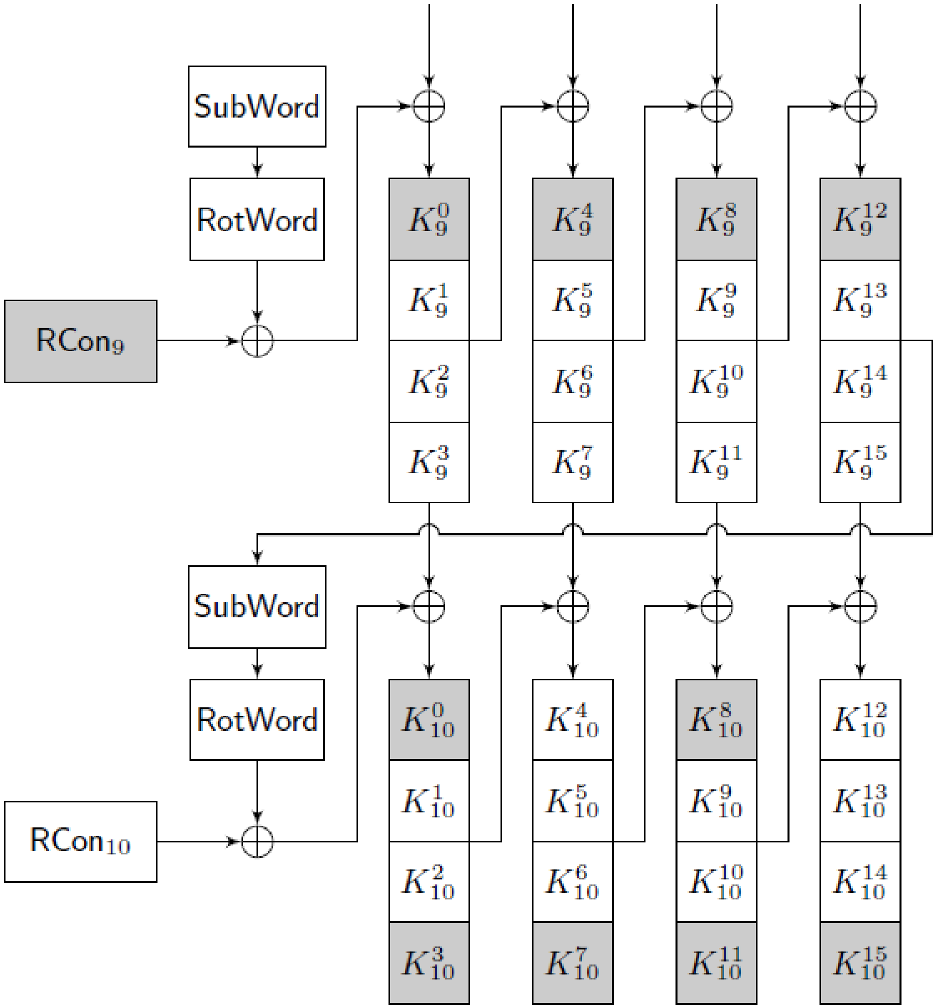
- One fault on $RCON_9$ affects 4 bytes of K_9 in the same way

- The fault can have a permanent effect

- Complexity similar to previous attack for 4 bytes:

$$(2^{16}+3 \times 2^8)A$$

RCON (2)



SubWord
RotWord
RCon9
K_9^0 K_9^1 K_9^2 K_9^3
K_9^4 K_9^5 K_9^6 K_9^7
K_9^8 K_9^9 K_9^{10} K_9^{11}
K_9^{12} K_9^{13} K_9^{14} K_9^{15}
SubWord
RotWord
RCon10
K_{10}^0 K_{10}^1 K_{10}^2 K_{10}^3
K_{10}^4 K_{10}^5 K_{10}^6 K_{10}^7
K_{10}^8 K_{10}^9 K_{10}^{10} K_{10}^{11}
K_{10}^{12} K_{10}^{13} K_{10}^{14} K_{10}^{15}

Attacking known constant values

- If the fault setup is characterized…

- $RCON_9 = 0\text{x}1\text{B}$
- Ex: if single bit *stuck-at* 0 or 1 model, only 4 possible values for $RCON_9$ ($0\text{x}1A, 0\text{x}19, 0\text{x}13, 0\text{x}0\text{B}$ if *stuck-at* 0)

- Lower complexity for key retrieval algorithm (4 bytes):

$$2^{10}A$$

- Whether *stuck-at* or *bit-flip* model, a fault on a constant will be XOR-ed → No impact on the repeatability

inside SECURE

Affine transformation (1)

- Most DSCA countermeasures compute the SubBytes as

$$SB(X) = \Omega \cdot \mathrm{Inv}_{F_{2^8}}(X) \oplus \Delta$$

where Ω is the matrix of the affine transformation and Δ is the vector.

- Different attack scenarios are possible depending on the implementation

Affine transformation (2)

1. Transient fault on Δ:
   - Same case as before
   - Complexity: $4N$ faults and $(2^{18} + 3 \times 2^{10})A$

2. Permanent fault. Different Δ_{SW} and Δ_{SB} for the SubWord and SubBytes
   - A fault e_{SW} on Δ_{SW} affects round 9 and 10
   - Faulted round 9 key is $\widetilde{K_9^j} = K_9^j \oplus e_{SW}$ for $0 \leq j \leq 15$
   - Relations between errors on K_{10}

$$e_{10}^{j+4} = e_{10}^{j+12} = e_{10}^{j} \oplus e_{SW}$$
$$e_{10}^{j+8} = e_{10}^{j} \text{ for } j = 0,1,2,3$$

   - Complexity: N faults and $(2^{24} + 3 \times 2^{16} + 3 \times 2^{10})A$

Affine transformation (3)

3. Permanent fault. Same Δ for SubWord and SubBytes
   - Same complexity as previous scenario
   - Data path modified → relation of key retrieval becomes

$$SB\left(SB^{-1}\left(C_i^j \oplus k_{10}^j\right) \oplus e_9^j\right) \oplus e_9^j \oplus k_{10}^j \oplus e_{10}^j$$

- If the fault setup is characterized, we can lower the complexity
  1. Transient fault:
     $4N$ faults and $2^{12}A$ (same complexity as classical DSCA)
  2. Permanent fault:
     N faults and $(2^{20} + 3 \times 2^{10})A$

Outline

Complexity of our attacks

Attack	# faults	# A
Key state K_9 (Roche et al.) - Transient on 16 bytes	N	2^{28}
Key state K_9 (Roche et al.) - Transient on 1 byte	$16N$	2^{20}
Key schedule - Transient 1 byte	$4N$	$2^{18} + 3 \times 2^{10}$
RCON - Transient known on 1 byte - Transient random on 1 byte - Permanent known on 1 byte - Permanent random on 1 byte	 N N 1 1	 2^{10} $2^{16} + 3 \times 2^{8}$ 2^{10} $2^{16} + 3 \times 2^{8}$
Affine transformation - Transient known on 1 byte - Transient random on 1 byte - Permanent known on 1 byte - Permanent random on 1 byte	 $4N$ $4N$ N N	 2^{12} $2^{18} + 3 \times 2^{10}$ $2^{20} + 3 \times 2^{10}$ $2^{24} + 3 \times 2^{16} + 3 \times 2^{10}$

inside SECURE

Outline

Countermeasures

- Masked coherence check:
  1. Store $C \oplus M_1$ and $C \oplus M_2$ two ciphertexts of the same message masked with M_1 and M_2
  2. Check $(C \oplus M_1) \oplus M_2 =^? (C \oplus M_2) \oplus M_1$
  3. If no fault, demask and output the ciphertext C

- Does not detect a permanent fault on $RCON_9$. Needs a known answer test or integrity check on $RCON_9$

Outline

1. Combined attack
2. Related work on combined attacks
    1. Asymmetric cryptosystems
    2. Symmetric cryptosystems
    3. Roche et al.'s attack on AES
3. Combined attacks on AES key schedule
    1. Recursive structure of the key schedule
    2. RCON
    3. Affine transformation
4. Complexity of our attacks
5. Countermeasures
6. Conclusion

Conclusion

- Combined attacks are a real threat to most current crypto implementations
- We propose different attack paths on AES that lower the complexity of previous combined attacks
- Repeatability of our attacks on AES constants do not depend on a *stuck-at* or *bit-flip* fault
- Needs additional countermeasure to protect against an attack on $RCON_9$

inside SECURE

Thank you for your attention !

Contact : avenelli@insidefr.com