Differential Fault Analysis against AES-192 and AES-256 with Minimal Faults

Chong Hee KIM

Information Security Group
Université Catholique de Louvain, Belgium

August 21, 2010

Outline

Outline

Differential fault analysis

DFA (Differential fault analysis)

- DFA uses differential information between correct and faulty ciphertexts to figure out the secret key
- Normally attacker gets faulty ciphertexts by giving external impact with voltage variation, glitch, laser, etc
- The first DFA: against DES by Biham and Shamir, 1997

DFA against AES-128

- Piret and Quisquater (2003)
  - 2 pairs, practical fault model (random byte error)
- Fukunaga and Takahashi: 1 pair with 2^{32} exhaustive search (8-35 minutes at Core2 Duo 3.0GHz PC)
- Tunstall and Mukhopadhyay: 1 pair with 2^{8} exhaustive search

Differential fault analysis

DFA (Differential fault analysis)

- DFA uses differential information between correct and faulty ciphertexts to figure out the secret key
- Normally attacker gets faulty ciphertexts by giving external impact with voltage variation, glitch, laser, etc
- The first DFA: against DES by Biham and Shamir, 1997

DFA against AES-128

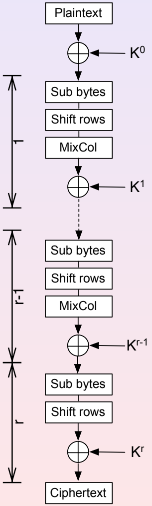
- Piret and Quisquater (2003)
  - 2 pairs, practical fault model (random byte error)
- Fukunaga and Takahashi: 1 pair with 2^{32} exhaustive search (8-35 minutes at Core2 Duo 3.0GHz PC)
- Tunstall and Mukhopadhyay: 1 pair with 2^{8} exhaustive search

Differential fault analysis

DFA against AES-192 and AES-256

- Application of Piret and Quisquter's: 4 pairs
- 2009, Li et al.: 16 or 3000 pairs
- 2010, Barenghi et al.: 16 pairs
- 2010, Takahashi and Fukunaga: 3 pairs for AES-192, 4 pairs for AES-256 (2 faulty plaintexts)
- Proposed methods: 2 pairs for AES-192, 3 pairs for AES-256

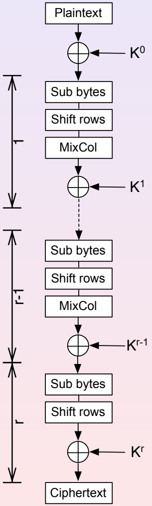
AES



- Intermediate result, called *State*, is represented as a two-dimensional byte array with 4 rows and 4 columns

$S_{(0,0)}$	$S_{(0,1)}$	$S_{(0,2)}$	$S_{(0,3)}$
$S_{(1,0)}$	$S_{(1,1)}$	$S_{(1,2)}$	$S_{(1,3)}$
$S_{(2,0)}$	$S_{(2,1)}$	$S_{(2,2)}$	$S_{(2,3)}$
$S_{(3,0)}$	$S_{(3,1)}$	$S_{(3,2)}$	$S_{(3,3)}$

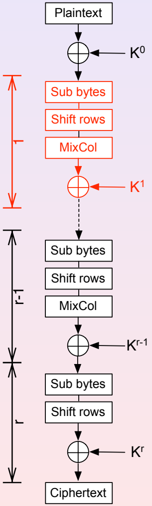
AES



- Each round is composed of 4 transformations except the last round:
  - SubBytes: 16 identical 8×8 S-boxes, non-linear byte substitution
  - ShiftRows: Each row is cyclically shifed over different offsets
  - MixColumns: A linear transformation to each column
  - AddRoundKey: A bitwise XOR with a round key
- Number of rounds

	Key length	Number of rounds r
AES-128	128	10
AES-192	192	12
AES-256	256	14

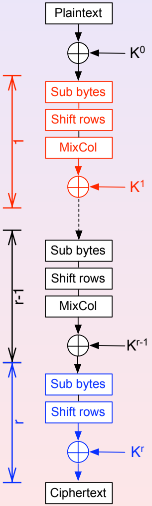
AES



- Each round is composed of 4 transformations except the last round:
  - SubBytes: 16 identical 8×8 S-boxes, non-linear byte substitution
  - ShiftRows: Each row is cyclically shifed over different offsets
  - MixColumns: A linear transformation to each column
  - AddRoundKey: A bitwise XOR with a round key
- Number of rounds

	Key length	Number of rounds r
AES-128	128	10
AES-192	192	12
AES-256	256	14

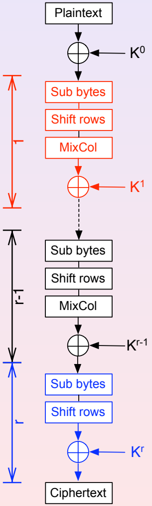
AES



- Each round is composed of 4 transformations except the last round:
  - SubBytes: 16 identical 8×8 S-boxes, non-linear byte substitution
  - ShiftRows: Each row is cyclically shifed over different offsets
  - MixColumns: A linear transformation to each column
  - AddRoundKey: A bitwise XOR with a round key
- Number of rounds

	Key length	Number of rounds r
AES-128	128	10
AES-192	192	12
AES-256	256	14

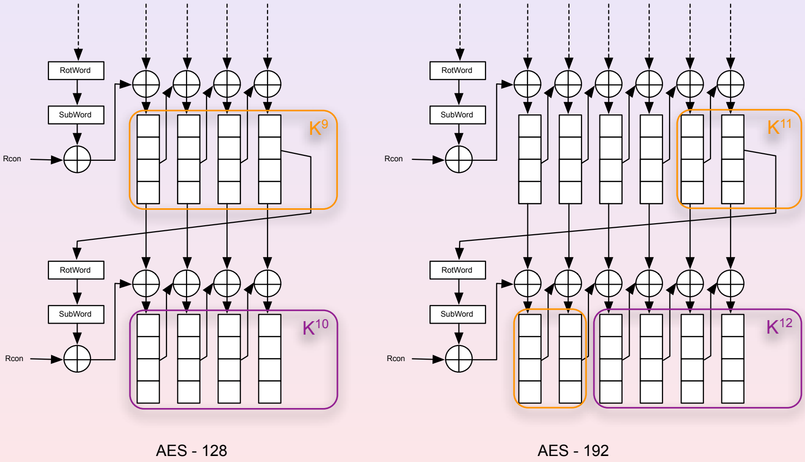
AES



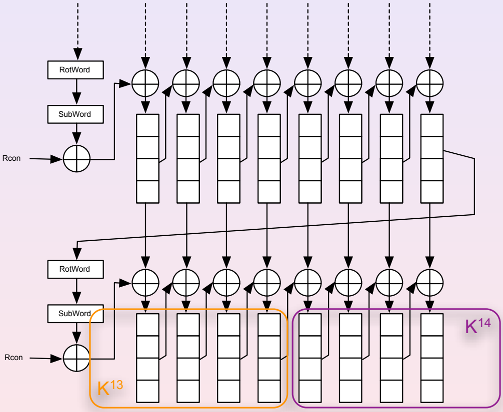
- Each round is composed of 4 transformations except the last round:
  - SubBytes: 16 identical 8×8 S-boxes, non-linear byte substitution
  - ShiftRows: Each row is cyclically shifed over different offsets
  - MixColumns: A linear transformation to each column
  - AddRoundKey: A bitwise XOR with a round key
- Number of rounds

	Key length	Number of rounds r
AES-128	128	10
AES-192	192	12
AES-256	256	14

AES key scheduling



AES key scheduling



AES - 256

Outline

Fault model

- We assume that
  - a byte of the AES intermediate state is corrupted by fault injection
  - the corrupted value is random and unkonw to the attacker
- Location of corrupted byte among 16 bytes
  - may be known to the attacker:
    ex) in [6], it was shown that precise control of fault injection was possible
  - may be not:
    perform 16 independent equivalent analysis
  - we assume that the attacker knows the location
- We assume that the attacker can get a pair of correct and faulty ciphertexts

Fault model

- We assume that
  - a byte of the AES intermediate state is corrupted by fault injection
  - the corrupted value is random and unkonw to the attacker
- Location of corrupted byte among 16 bytes
  - may be known to the attacker:
    ex) in [6], it was shown that precise control of fault injection was possible
  - may be not:
    perform 16 independent equivalent analysis
  - we assume that the attacker knows the location
- We assume that the attacker can get a pair of correct and faulty ciphertexts

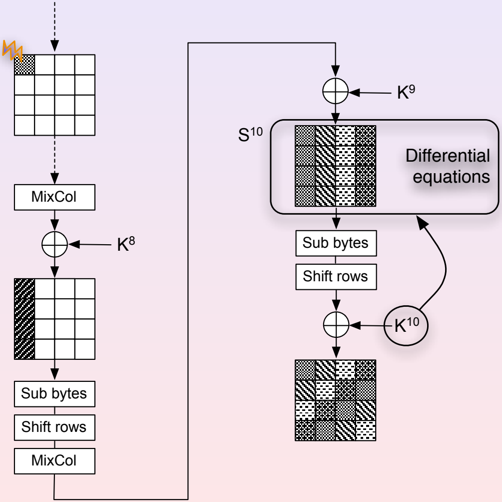
Fault model

- We assume that
  - a byte of the AES intermediate state is corrupted by fault injection
  - the corrupted value is random and unkonw to the attacker
- Location of corrupted byte among 16 bytes
  - may be known to the attacker:
    ex) in [6], it was shown that precise control of fault injection was possible
  - may be not:
    perform 16 independent equivalent analysis
  - we assume that the attacker knows the location
- We assume that the attacker can get a pair of correct and faulty ciphertexts

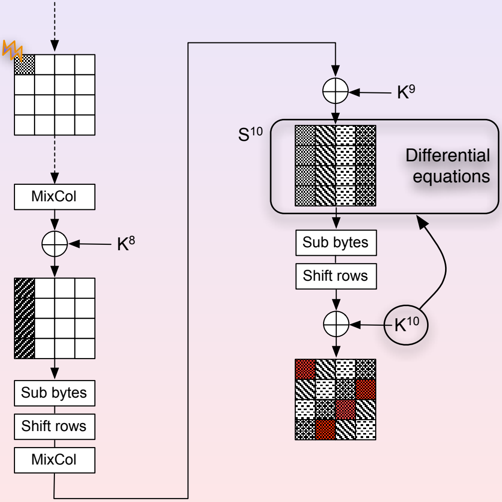
Basic concept of DFA against AES-128

- Based on Piret and Quisquater's method
  + recent improvement
- A 1-byte fault between *MixColumns* of rounds 7th and 8th

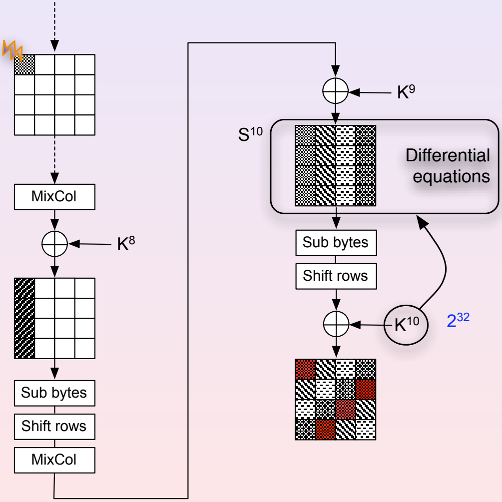
Basic concept of DFA against AES-128



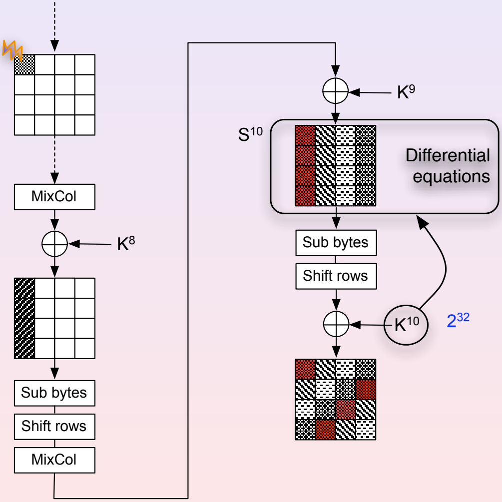
Basic concept of DFA against AES-128



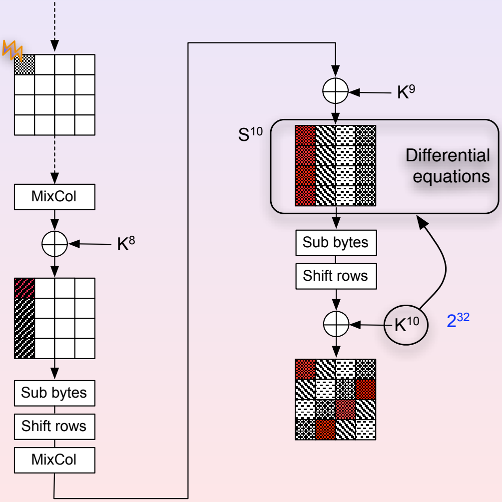
Basic concept of DFA against AES-128



Basic concept of DFA against AES-128



Basic concept of DFA against AES-128

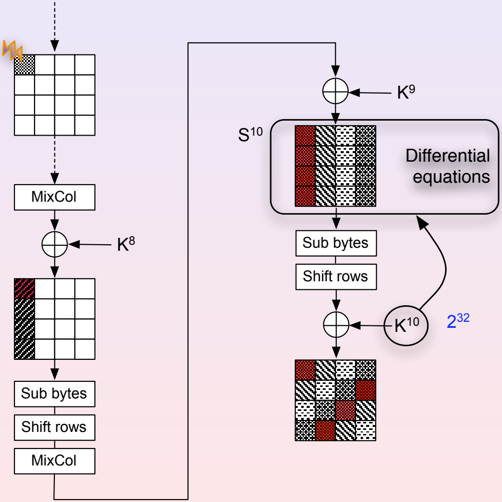


$$\Delta S^{10}_{(0,0)} = 2\sigma,$$
$$\Delta S^{10}_{(1,0)} = \sigma,$$
$$\Delta S^{10}_{(2,0)} = \sigma,$$
$$\Delta S^{10}_{(3,0)} = 3\sigma.$$

Basic concept of DFA against AES-128

$$
\begin{aligned}
\mathbf{SB}^{-1}(C_{0,0} \oplus K^{10}_{0,0}) \oplus \mathbf{SB}^{-1}(C^{*}_{0,0} \oplus K^{10}_{0,0}) &= 2\sigma, \\
\mathbf{SB}^{-1}(C_{1,3} \oplus K^{10}_{1,3}) \oplus \mathbf{SB}^{-1}(C^{*}_{1,3} \oplus K^{10}_{1,3}) &= \sigma, \\
\mathbf{SB}^{-1}(C_{2,2} \oplus K^{10}_{2,2}) \oplus \mathbf{SB}^{-1}(C^{*}_{2,2} \oplus K^{10}_{2,2}) &= \sigma, \\
\mathbf{SB}^{-1}(C_{3,1} \oplus K^{10}_{3,1}) \oplus \mathbf{SB}^{-1}(C^{*}_{3,1} \oplus K^{10}_{3,1}) &= 3\sigma.
\end{aligned}
$$

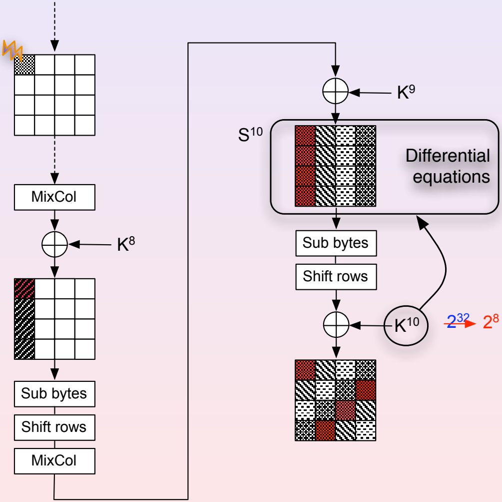
Basic concept of DFA against AES-128



$$\Delta S^{10}_{(0,0)} = 2\sigma,$$
$$\Delta S^{10}_{(1,0)} = \sigma,$$
$$\Delta S^{10}_{(2,0)} = \sigma,$$
$$\Delta S^{10}_{(3,0)} = 3\sigma.$$

Among 2^{32} candidates, in average 2^8 candidates satisfy equations.

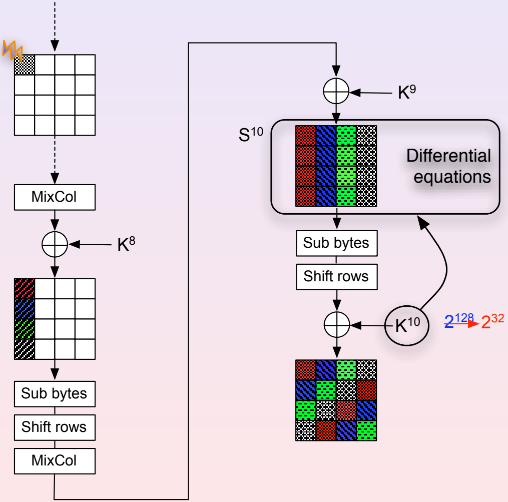
Basic concept of DFA against AES-128



$$\Delta S^{10}_{(0,0)} = 2\sigma,$$
$$\Delta S^{10}_{(1,0)} = \sigma,$$
$$\Delta S^{10}_{(2,0)} = \sigma,$$
$$\Delta S^{10}_{(3,0)} = 3\sigma.$$

Among 2^{32} candidates, in average 2^8 candidates satisfy equations.

Basic concept of DFA against AES-128

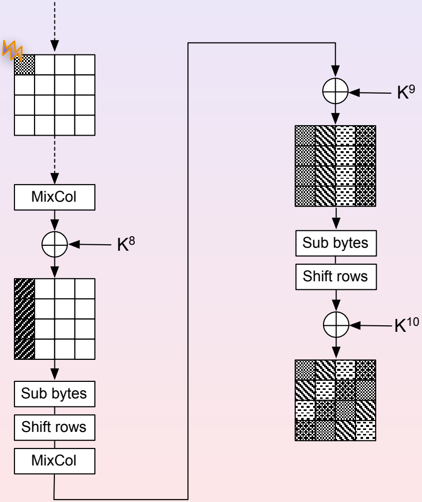


For other columns we construct similiar equations.

We have 2^{32} candidates for K^{10}.

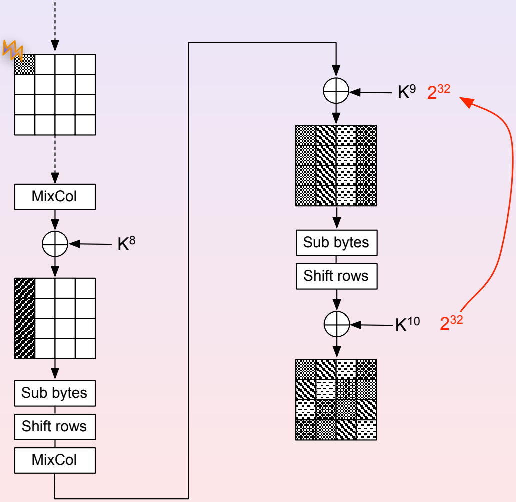
With 2 pairs, we have the correct key K^{10}.

Basic concept of DFA against AES-128



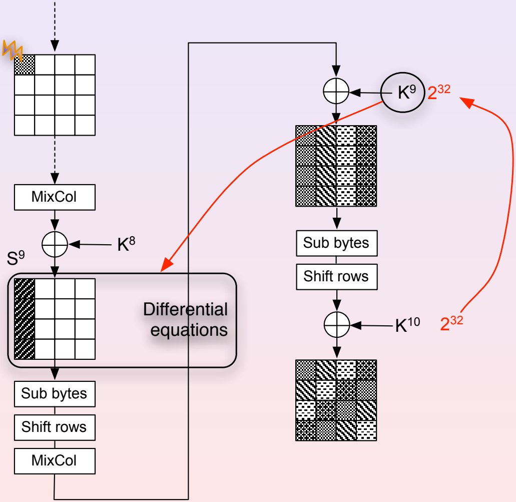
According to [12], we can further reduce the number of candidates to 2^8.

Basic concept of DFA against AES-128



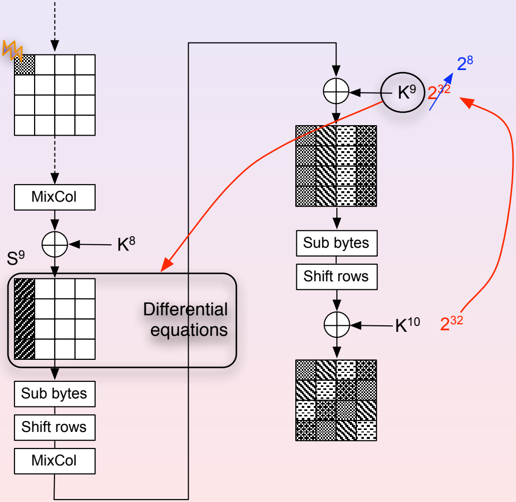
According to [12], we can further reduce the number of candidates to 2^8.

Basic concept of DFA against AES-128



According to [12], we can further reduce the number of candidates to 2^8.

Basic concept of DFA against AES-128



According to [12], we can further reduce the number of candidates to 2^8.

Outline

Objective

- With a current normal PC, an exhaustive search of 2^{32} can be done within tens of minutes.
- Therefore we try to minimize the required number of faults with up to 2^{32} exhaustive search.

DFA against AES-192: Method 1

Attack procedure

1. Obtain 2 pairs of (C_1, C_1^*) and (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Find K^{12}.
3. Find the left-half of K^{11} with key schedule.
4. Find 2^{32} candidates for the right-half of K^{11}.
5. Find the master secret key with an exhaustive search of 2^{32}.

DFA against AES-192: Method 1

Attack procedure

1. Obtain 2 pairs of (C_1, C_1^*) and (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Find K^{12}.
3. Find the left-half of K^{11} with key schedule.
4. Find 2^{32} candidates for the right-half of K^{11}.
5. Find the master secret key with an exhaustive search of 2^{32}.

DFA against AES-192: Method 1

Attack procedure

1. Obtain 2 pairs of (C_1, C_1^*) and (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Find K^{12}.
3. Find the left-half of K^{11} with key schedule.
4. Find 2^{32} candidates for the right-half of K^{11}.
5. Find the master secret key with an exhaustive search of 2^{32}.

DFA against AES-192: Method 1

Attack procedure

1. Obtain 2 pairs of (C_1, C_1^*) and (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Find K^{12}.
3. Find the left-half of K^{11} with key schedule.
4. Find 2^{32} candidates for the right-half of K^{11}.
5. Find the master secret key with an exhaustive search of 2^{32}.

DFA against AES-192: Method 1

Attack procedure

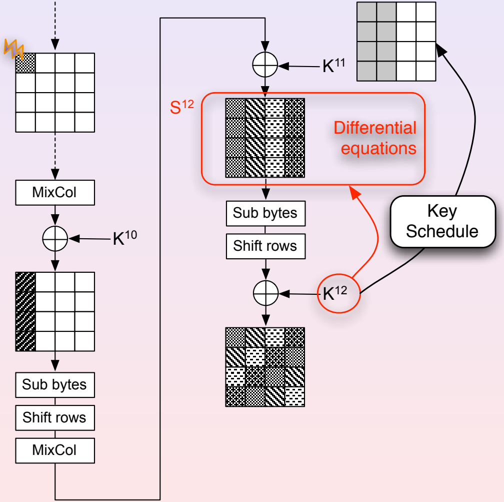
1. Obtain 2 pairs of (C_1, C_1^*) and (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Find K^{12}.
3. Find the left-half of K^{11} with key schedule.
4. Find 2^{32} candidates for the right-half of K^{11}.
5. Find the master secret key with an exhaustive search of 2^{32}.

DFA against AES-192: Method 1

Attack procedure

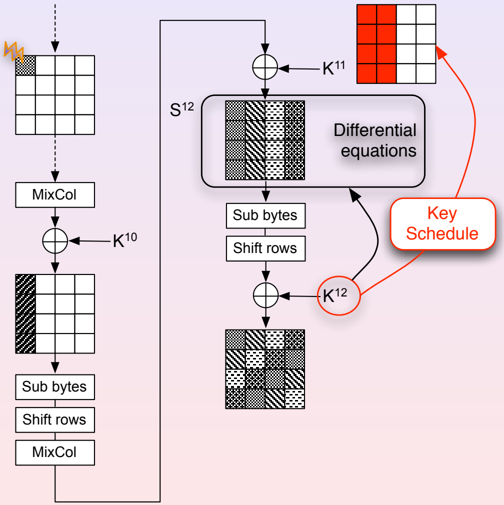
1. Obtain 2 pairs of (C_1, C_1^*) and (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Find K^{12}.
3. Find the left-half of K^{11} with key schedule.
4. Find 2^{32} candidates for the right-half of K^{11}.
5. Find the master secret key with an exhaustive search of 2^{32}.

DFA against AES-192: Method 1



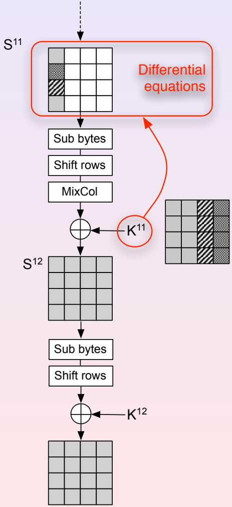
1. Find K^{12} with 2 pairs
2. Find the left-half of K^{11} with key schedule
3. Find 2^{32} candidates for the right-half of K^{11}
4. Find the master secret key with an exhaustive search of 2^{32}

DFA against AES-192: Method 1



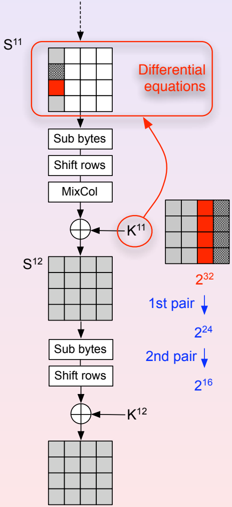
1. Find K^{12} with 2 pairs
2. Find the left-half of K^{11} with key schedule
3. Find 2^{32} candidates for the right-half of K^{11}
4. Find the master secret key with an exhaustive search of 2^{32}

DFA against AES-192: Method 1



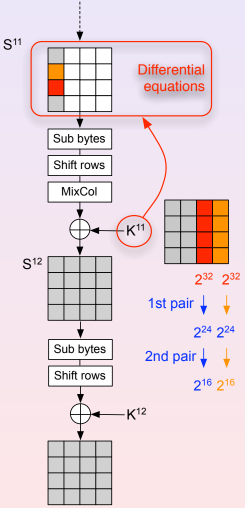
1. Find K^{12} with 2 pairs
2. Find the left-half of K^{11} with key schedule
3. Find 2^{32} candidates for the right-half of K^{11}
4. Find the master secret key with an exhaustive search of 2^{32}

DFA against AES-192: Method 1



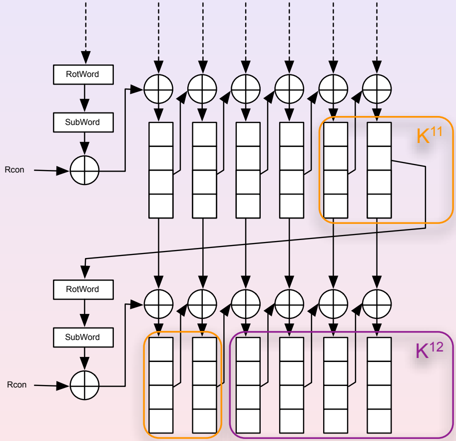
1. Find K^{12} with 2 pairs
2. Find the left-half of K^{11} with key schedule
3. Find 2^{32} candidates for the right-half of K^{11}
4. Find the master secret key with an exhaustive search of 2^{32}

DFA against AES-192: Method 1



1. Find K^{12} with 2 pairs
2. Find the left-half of K^{11} with key schedule
3. Find 2^{32} candidates for the right-half of K^{11}
4. Find the master secret key with an exhaustive search of 2^{32}

DFA against AES-192: Method 1



AES - 192

1. Find K^{12} with 2 pairs
2. Find the left-half of K^{11} with key schedule
3. Find 2^{32} candidates for the right-half of K^{11}
4. Find the master secret key with an exhaustive search of 2^{32}

DFA against AES-192: Method 2

Attack procedure

1. Obtain a pair of (C_1, C_1^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Obtain a pair of (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 8 and 9
3. Find 2^{32} candidates for K^{12} with (C_1, C_1^*).
4. Compute the 2^{32} for left-half of K^{11} with key schedule.
5. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
6. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).
7. Find the 2^8 candidates for right-half of K^{11} with (C_2, C_2^*).
8. Find the $MC^{-1}(K^{11})$ with (C_1, C_1^*).
9. Compute master secret key.

DFA against AES-192: Method 2

Attack procedure

1. Obtain a pair of (C_1, C_1^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Obtain a pair of (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 8 and 9
3. Find 2^{32} candidates for K^{12} with (C_1, C_1^*).
4. Compute the 2^{32} for left-half of K^{11} with key schedule.
5. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
6. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).
7. Find the 2^8 candidates for right-half of K^{11} with (C_2, C_2^*).
8. Find the $MC^{-1}(K^{11})$ with (C_1, C_1^*).
9. Compute master secret key.

DFA against AES-192: Method 2

Attack procedure

1. Obtain a pair of (C_1, C_1^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Obtain a pair of (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 8 and 9
3. Find 2^{32} candidates for K^{12} with (C_1, C_1^*).
4. Compute the 2^{32} for left-half of K^{11} with key schedule.
5. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
6. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).
7. Find the 2^8 candidates for right-half of K^{11} with (C_2, C_2^*).
8. Find the $MC^{-1}(K^{11})$ with (C_1, C_1^*).
9. Compute master secret key.

DFA against AES-192: Method 2

Attack procedure

1. Obtain a pair of (C_1, C_1^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Obtain a pair of (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 8 and 9
3. Find 2^{32} candidates for K^{12} with (C_1, C_1^*).
4. Compute the 2^{32} for left-half of K^{11} with key schedule.
5. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
6. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).
7. Find the 2^8 candidates for right-half of K^{11} with (C_2, C_2^*).
8. Find the $MC^{-1}(K^{11})$ with (C_1, C_1^*).
9. Compute master secret key.

DFA against AES-192: Method 2

Attack procedure

1. Obtain a pair of (C_1, C_1^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Obtain a pair of (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 8 and 9
3. Find 2^{32} candidates for K^{12} with (C_1, C_1^*).
4. Compute the 2^{32} for left-half of K^{11} with key schedule.
5. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
6. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).
7. Find the 2^8 candidates for right-half of K^{11} with (C_2, C_2^*).
8. Find the $MC^{-1}(K^{11})$ with (C_1, C_1^*).
9. Compute master secret key.

DFA against AES-192: Method 2

Attack procedure

1. Obtain a pair of (C_1, C_1^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Obtain a pair of (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 8 and 9
3. Find 2^{32} candidates for K^{12} with (C_1, C_1^*).
4. Compute the 2^{32} for left-half of K^{11} with key schedule.
5. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
6. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).
7. Find the 2^8 candidates for right-half of K^{11} with (C_2, C_2^*).
8. Find the $MC^{-1}(K^{11})$ with (C_1, C_1^*).
9. Compute master secret key.

DFA against AES-192: Method 2

Attack procedure

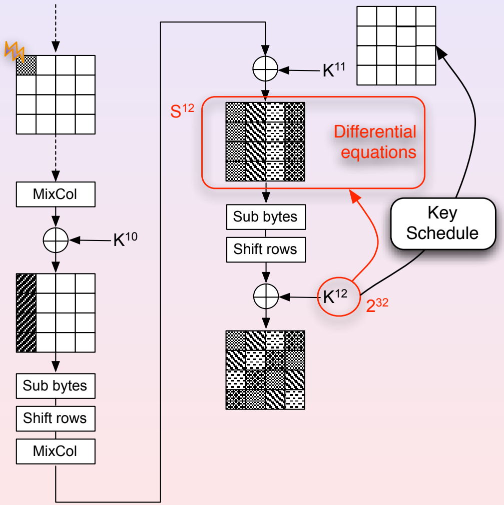
1. Obtain a pair of (C_1, C_1^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Obtain a pair of (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 8 and 9
3. Find 2^{32} candidates for K^{12} with (C_1, C_1^*).
4. Compute the 2^{32} for left-half of K^{11} with key schedule.
5. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
6. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).
7. Find the 2^8 candidates for right-half of K^{11} with (C_2, C_2^*).
8. Find the $MC^{-1}(K^{11})$ with (C_1, C_1^*).
9. Compute master secret key.

DFA against AES-192: Method 2

Attack procedure

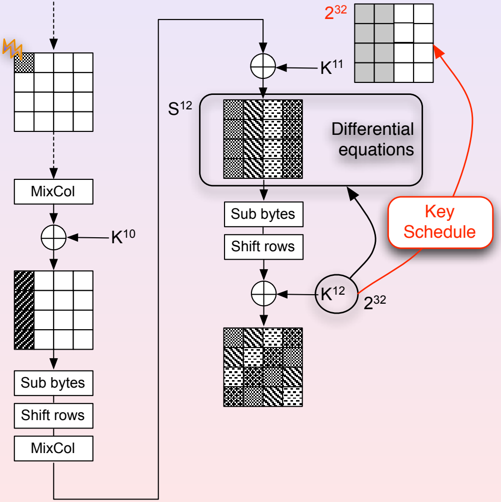
1. Obtain a pair of (C_1, C_1^*). Where the faults are injected between *MixColumns* of round 9 and 10.
2. Obtain a pair of (C_2, C_2^*). Where the faults are injected between *MixColumns* of round 8 and 9
3. Find 2^{32} candidates for K^{12} with (C_1, C_1^*).
4. Compute the 2^{32} for left-half of K^{11} with key schedule.
5. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
6. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).
7. Find the 2^8 candidates for right-half of K^{11} with (C_2, C_2^*).
8. Find the $MC^{-1}(K^{11})$ with (C_1, C_1^*).
9. Compute master secret key.

DFA against AES-192: Method 2



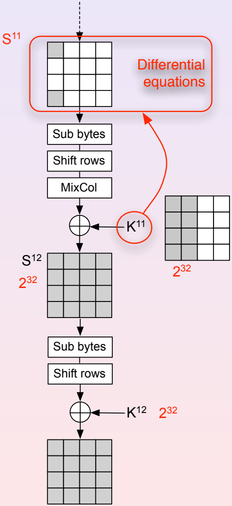
1. Find 2^{32} candidates for K^{12} with (C_1, C_1^*)
2. Compute the 2^{32} candidates for left-half of K^{11} with key schedule.
3. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
4. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).

DFA against AES-192: Method 2



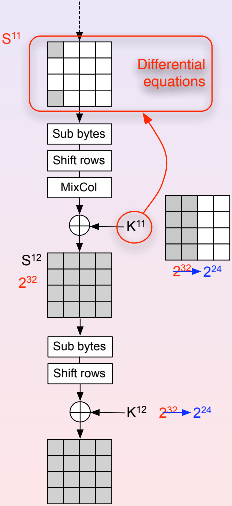
1. Find 2^{32} candidates for K^{12} with (C_1, C_1^*)
2. Compute the 2^{32} candidates for left-half of K^{11} with key schedule.
3. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
4. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).

DFA against AES-192: Method 2



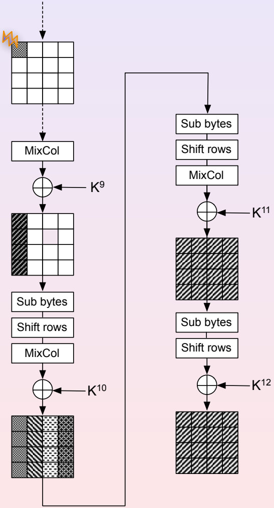
1. Find 2^{32} candidates for K^{12} with (C_1, C_1^*)
2. Compute the 2^{32} candidates for left-half of K^{11} with key schedule.
3. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
4. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).

DFA against AES-192: Method 2



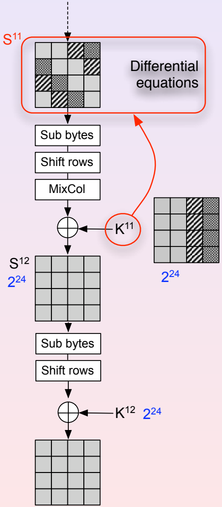
1. Find 2^{32} candidates for K^{12} with (C_1, C_1^*)
2. Compute the 2^{32} candidates for left-half of K^{11} with key schedule.
3. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
4. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).

DFA against AES-192: Method 2



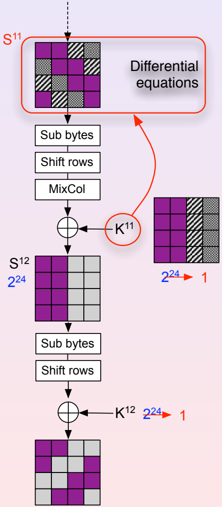
1. Find 2^{32} candidates for K^{12} with (C_1, C_1^*)
2. Compute the 2^{32} candidates for left-half of K^{11} with key schedule.
3. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
4. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).

DFA against AES-192: Method 2



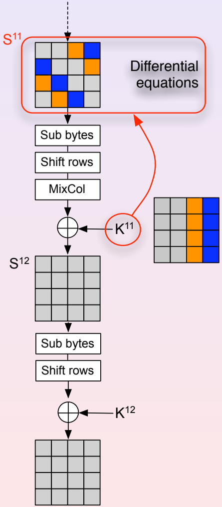
1. Find 2^{32} candidates for K^{12} with (C_1, C_1^*)
2. Compute the 2^{32} candidates for left-half of K^{11} with key schedule.
3. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
4. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).

DFA against AES-192: Method 2



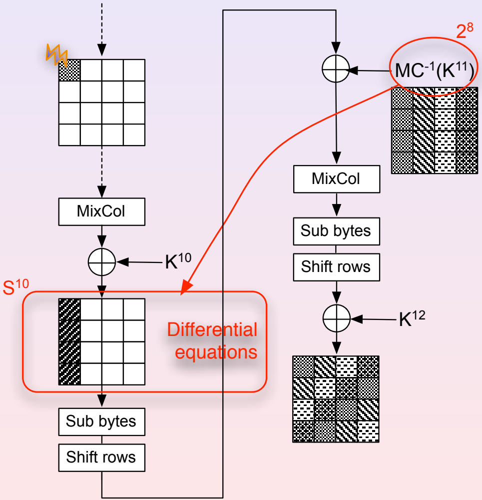
1. Find 2^{32} candidates for K^{12} with (C_1, C_1^*)
2. Compute the 2^{32} candidates for left-half of K^{11} with key schedule.
3. Reduce the candidates for K^{12} and the left-half of K^{11} to 2^{24}.
4. Find the left-half of K^{11} and K^{12} with (C_2, C_2^*).

DFA against AES-192: Method 2



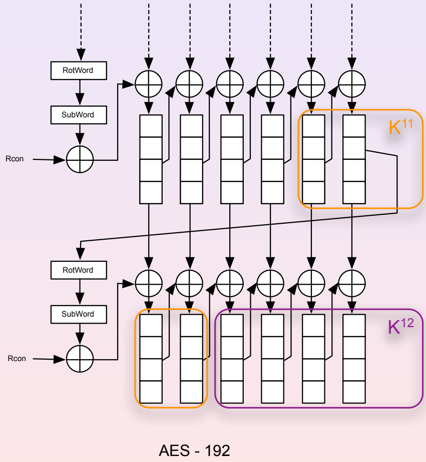
5. Find the 2^8 candidates for right-half of K^{11} with (C_2, C_2^*).
6. Find the $MC^{-1}(K^{11})$ with (C_1, C_1^*).
7. Compute the master secret key.

DFA against AES-192: Method 2



5. Find the 2^8 candidates for right-half of K^{11} with (C_2, C_2^*).
6. Find the $MC^{-1}(K^{11})$ with (C_1, C_1^*).
7. Compute the master secret key.

DFA against AES-192: Method 2



5. Find the 2^8 candidates for right-half of K^{11} with (C_2, C_2^*).
6. Find the $MC^{-1}(K^{11})$ with (C_1, C_1^*).
7. Compute the master secret key.

DFA against AES-256

Attack procedure

1. Obtain two pairs of correct and faulty ciphertexts (C_1, C_1^*) and (C_2, C_2^*) by giving faults between *MixColumns* of round 11 and 12.
2. Obtain a pair of correct and faulty ciphertexts (C_3, C_3^*) by giving faults between *MixColumns* of round 10 and 11.
3. Find K^{14} with (C_1, C_1^*) and (C_2, C_2^*).
4. Find 2^{32} candidates for $MC^{-1}(K^{13})$ with (C_3, C_3^*).
5. Find K^{13} with (C_1, C_1^*) and (C_2, C_2^*).
6. Find the master secret key with key scheduling.

DFA against AES-256

Attack procedure

1. Obtain two pairs of correct and faulty ciphertexts (C_1, C_1^*) and (C_2, C_2^*) by giving faults between *MixColumns* of round 11 and 12.
2. Obtain a pair of correct and faulty ciphertexts (C_3, C_3^*) by giving faults between *MixColumns* of round 10 and 11.
3. Find K^{14} with (C_1, C_1^*) and (C_2, C_2^*).
4. Find 2^{32} candidates for $MC^{-1}(K^{13})$ with (C_3, C_3^*).
5. Find K^{13} with (C_1, C_1^*) and (C_2, C_2^*).
6. Find the master secret key with key scheduling.

DFA against AES-256

Attack procedure

1. Obtain two pairs of correct and faulty ciphertexts (C_1, C_1^*) and (C_2, C_2^*) by giving faults between *MixColumns* of round 11 and 12.
2. Obtain a pair of correct and faulty ciphertexts (C_3, C_3^*) by giving faults between *MixColumns* of round 10 and 11.
3. Find K^{14} with (C_1, C_1^*) and (C_2, C_2^*).
4. Find 2^{32} candidates for $MC^{-1}(K^{13})$ with (C_3, C_3^*).
5. Find K^{13} with (C_1, C_1^*) and (C_2, C_2^*).
6. Find the master secret key with key scheduling.

DFA against AES-256

Attack procedure

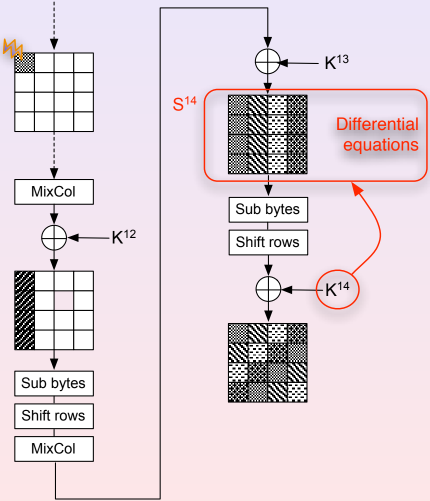
1. Obtain two pairs of correct and faulty ciphertexts (C_1, C_1^*) and (C_2, C_2^*) by giving faults between *MixColumns* of round 11 and 12.
2. Obtain a pair of correct and faulty ciphertexts (C_3, C_3^*) by giving faults between *MixColumns* of round 10 and 11.
3. Find K^{14} with (C_1, C_1^*) and (C_2, C_2^*).
4. Find 2^{32} candidates for $MC^{-1}(K^{13})$ with (C_3, C_3^*).
5. Find K^{13} with (C_1, C_1^*) and (C_2, C_2^*).
6. Find the master secret key with key scheduling.

DFA against AES-256

Attack procedure

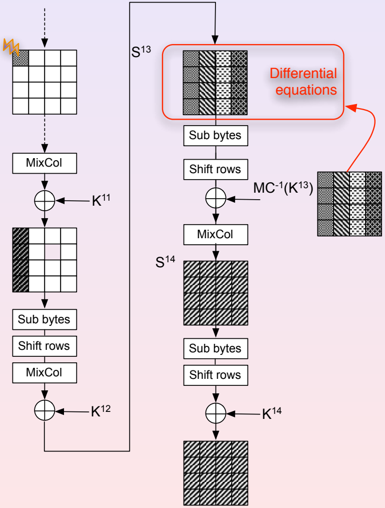
1. Obtain two pairs of correct and faulty ciphertexts (C_1, C_1^*) and (C_2, C_2^*) by giving faults between *MixColumns* of round 11 and 12.
2. Obtain a pair of correct and faulty ciphertexts (C_3, C_3^*) by giving faults between *MixColumns* of round 10 and 11.
3. Find K^{14} with (C_1, C_1^*) and (C_2, C_2^*).
4. Find 2^{32} candidates for $MC^{-1}(K^{13})$ with (C_3, C_3^*).
5. Find K^{13} with (C_1, C_1^*) and (C_2, C_2^*).
6. Find the master secret key with key scheduling.

DFA against AES-256



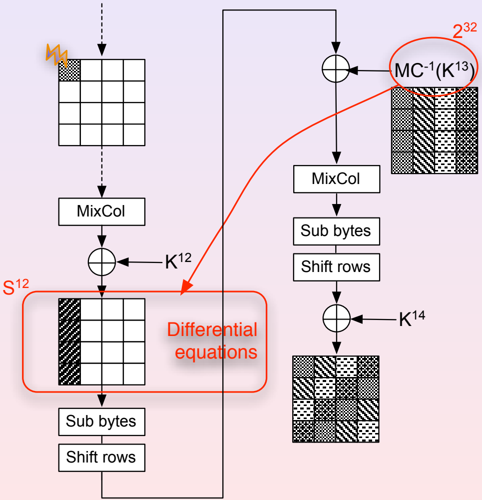
1. Find K^{14} with (C_1, C_1^*) and (C_2, C_2^*).
2. Find 2^{32} candidates for $MC^{-1}(K^{13})$ with (C_3, C_3^*).
3. Find K^{13} with (C_1, C_1^*) and (C_2, C_2^*).
4. Find the master secret key with key scheduling.

DFA against AES-256



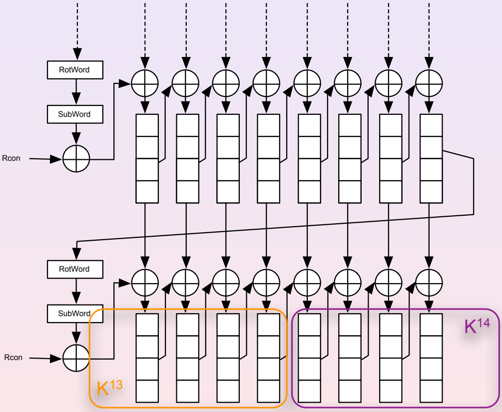
1. Find K^{14} with (C_1, C_1^*) and (C_2, C_2^*).
2. Find 2^{32} candidates for $MC^{-1}(K^{13})$ with (C_3, C_3^*).
3. Find K^{13} with (C_1, C_1^*) and (C_2, C_2^*).
4. Find the master secret key with key scheduling.

DFA against AES-256



1. Find K^{14} with (C_1, C_1^*) and (C_2, C_2^*).
2. Find 2^{32} candidates for $MC^{-1}(K^{13})$ with (C_3, C_3^*).
3. Find K^{13} with (C_1, C_1^*) and (C_2, C_2^*).
4. Find the master secret key with key scheduling.

DFA against AES-256



AES - 256

1. Find K^{14} with (C_1, C_1^*) and (C_2, C_2^*).
2. Find 2^{32} candidates for $MC^{-1}(K^{13})$ with (C_3, C_3^*).
3. Find K^{13} with (C_1, C_1^*) and (C_2, C_2^*).
4. Find the master secret key with key scheduling.

Outline

Comparisons with existing DFA's against AES-192

Reference	Fault model	No. of faults	Exhaustive search
Piret and Quisquater	1 byte	4	1
Li et al. method 1	1-4 bytes	$12^{\dagger}$	1
Li et al. method 2	4 bytes	$3000^{\dagger}$	1
Barenghi et al.	1 byte	$16^{\dagger}$	1
Takahashi and Fukunaga	1 byte	3	2^8
Our attack 1	1 byte	2	2^{32}
Our attack 2	1 byte	2	1

†: with same plaintext

Comparisons with existing DFA's against AES-256

Reference	Fault model	No. of faults	Exhaustive search
Piret and Quisquater	1 byte	4	1
Li et al. method 1	1-4 bytes	$12^{\dagger}$	1
Li et al. method 2	4 bytes	$3000^{\dagger}$	1
Barenghi et al.	1 byte	$16^{\dagger}$	1
Takahashi and Fukunaga	1 byte	$4^{\ddagger}$	2^{13}
Our attack	1 byte	3	1

†: with same plaintext

‡: 2 faulty plaintexts and 2 faulty ciphertexts

Questions and answers

- Thank you!
- Questions?