SiVenture
A division of NDS

DFA against AES Key Expansion

David Peacham & Byron Thomas

(SiVenture)

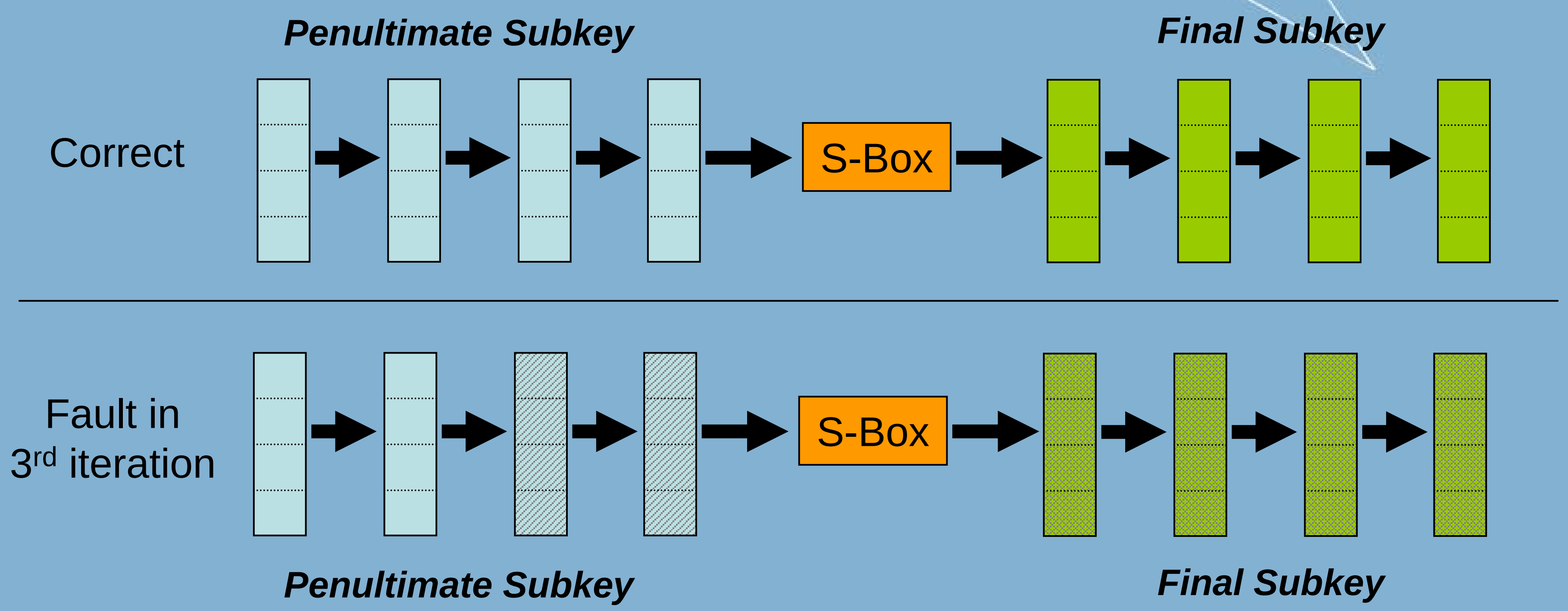
CHES 06 Rump Session

Background

- Several existing DFA attacks on AES:
  - [DUSART et al] Corrupt state bytes in penultimate round
  - [GIRAUD] Corrupt single bytes in the penultimate and antepenultimate subkeys
    - Requires a slow final search (2 32-bit exhaustive searches)
    - 250 true / corrupted ciphertext pairs
  - [Chen-Yen] improves upon Giraud’s attack to make the analysis easier
    - 24-bits of search; 1 true ciphertext with 22 corrupted texts

Our Attack : fault model

- Corrupt the values of a single *iteration* of the key expansion for the penultimate round subkey



Our Attack : methodology

- First corrupt last key iteration for penultimate round (rightmost column of key)
- Express the state prior to the penultimate key add as an equation for each byte involving some ciphertext bytes and some penultimate round key bytes
  - E.g. *Column 0, row 0: P0^S'[P0^R10^S[P13]^Z0]*

Our attack: Methodolgy

- Rewrite these in the presence of faults
  - E.g. *Column 0, row 0: P0^S'[P0^R10^S[P13^p13]^Z0^z0]*
- But these two must be equal so
  - *P0^S'[P0^R10^S[P13]^Z0] = P0^S'[P0^R10^S[P13^p13]^Z0^z0*
  - *S[P13]^S[P13^p13]^z0 = 0*
- Get similar equations for other bytes

Our attack: Methodolgy

- Now we just need to solve simultaneously, e.g.:
  1. *S[P14]^S[P14^p14]^z1 = 0*
  2. *S[P14]^S[P14^p14]^p13^z13 = 0*

  1. *S[P14]^S[P14^p14] = z1*
  2. *z1^p13^z13 = 0*
- This can be solved directly to find *p13* since *z1* and *z13* are the output differences in the 1st and 13th bytes
- *S[P13]^S[P13^p13]^z0 = 0* so can use our *p13* and perform an 8-bit search to find the key byte P13

Our Attack: finishing up

- Note: using one fault, we find two possible values (the true value and the corrupted value).
- Need 3 pairs per key iteration to derive key bytes definitively.
- Using the same set of faults (last column of the key) can find P14, P15, P2 ^ P6, P1 ^ P5 ^ P9, P3 ^ S[P12]

SiVenture
A division of NDS

Our Attack: finishing up

- Then we target one column to the left and rewrite the equations to find more key bytes
- And then continue back…
- After causing faults in the first column of the key, can derive all 16 bytes of P
- The unknowns of each equation are found using either an 8-bit or a 16-bit search

SiVenture
A division of NDS

Our attack: Advantages

- Naïve reverse-calculation countermeasures are unlikely to detect the fault
- The attack requires 12 pairs of correct and faulty ciphertexts (3 per key iteration)
- Only a single round key has to be faulted
- The fault model applies to schedule-on-demand
- It is simple to understand
- It is efficient: several 16-bit and 8-bit searches