A Differential Fault Attack Technique Against SPN Structures and the AES

G. Piret*, J.-J. Quisquater

CHES 2003 Workshop

Cologne, Germany

Outline of the Talk

1. General Context.
   → Introduction
   → Cipher Structure
   → Framework of the Attack
2. The Attack.
   → Sketch of an Attack
   → A Practical Attack
   → Dealing with Wrong-Located Faults
3. Application to the AES.
   → About the Linear Transform of the AES
   → The Basic Attack
   → An Improved Attack
   → Implementation on a PC
4. Conclusion.

Introduction: Fault attacks

- First suggestion in 1997: Boneh, DeMillo, Lipton. Fault Attack on RSA-CRT.
- Application to block ciphers, especially DES: Biham, Shamir 1997.
- Several papers about DFA on the AES: BS02, DLV03, G03, …

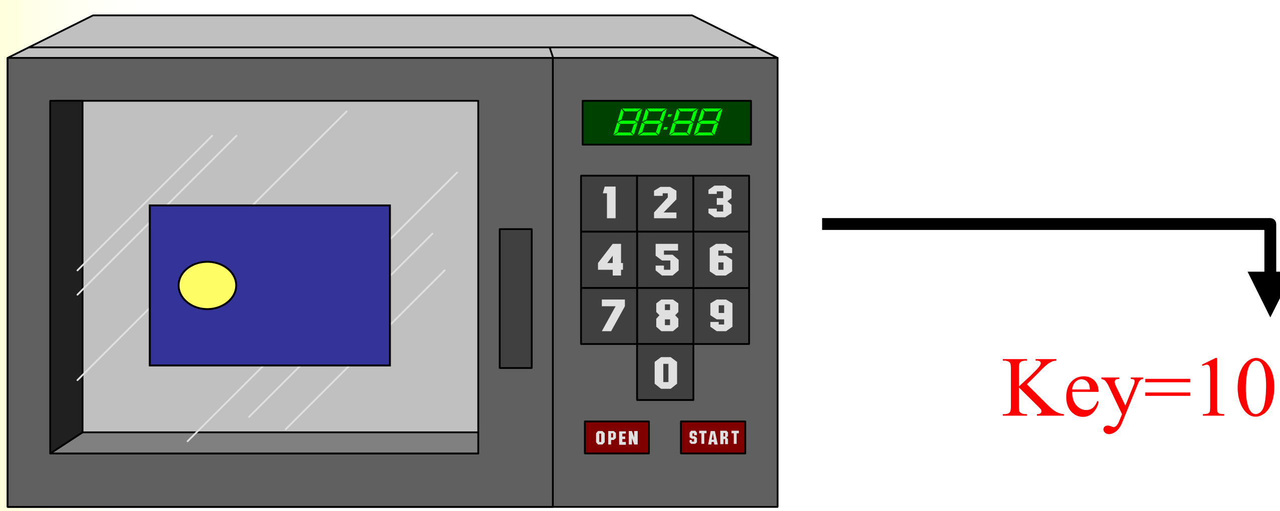
Fault Attacks : Principle

→ Induce faults during cryptographic computation.

- By changing power supply voltage.
- By increasing frequency of the external clock.
- By applying radiations.

→ Outputs faulty results.

→ Use them to recover the secret key stored in the card.



Key=1010110...

Framework of our Attack

- Faults occurring on **bytes.**
- A faulty ciphertext results from one unique fault.
- Cipher Structure: Substitution-Permutation Network.

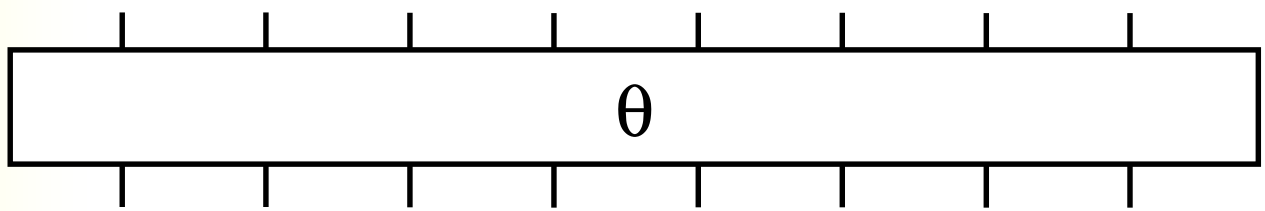
- Countermeasure: Double encryption.

Substitution-Permutation Network (SPN)

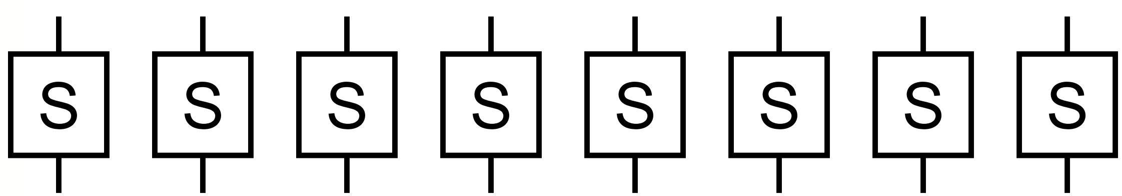
A round with structure $\sigma[K^r]\circ\theta\circ\gamma$ is iterated several times:

- $\sigma[K^r]$ = Key addition $\quad\sigma[k](a)=b \Leftrightarrow b=a\oplus k$

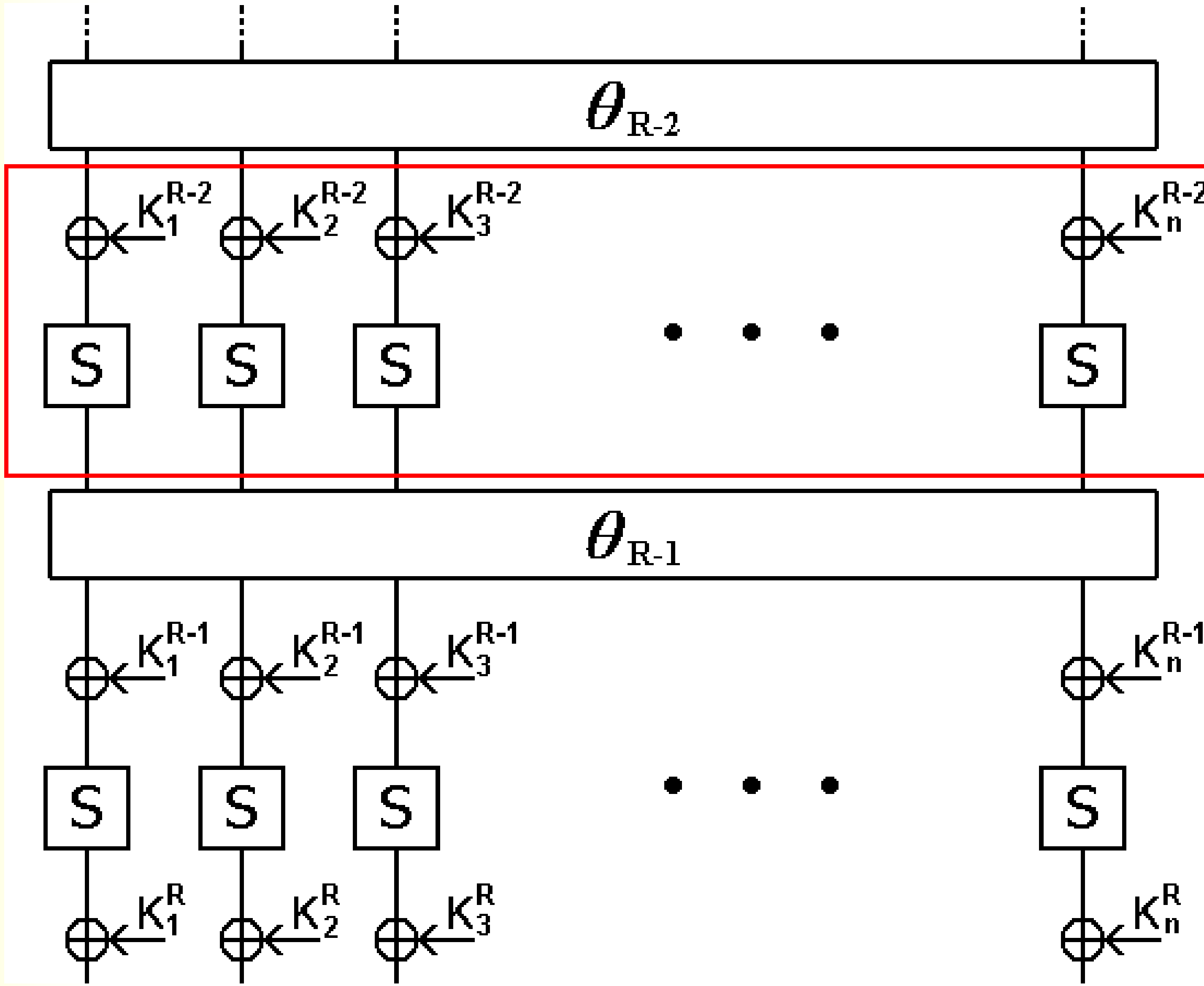
- θ = Linear diffusion layer



- γ = Non-linear layer



Fault Location



Observation

- The difference before θ_{R-1} caused by a random fault between θ_{R-2} and θ_{R-1} is of the form:

$$(0,\dots,0,\alpha,0,\dots,0)$$

The number of such differences is 255n.

- There are 255n corresponding differences before the last S-box layer. They are of the form:

$$(\alpha_1,\dots,\alpha_n)$$

Sketch of an Attack

1. Compute a list $\mathcal{D}$ of the 255n possible differences after θ_{R-1}.
2. Consider a plaintext ***P***, the corresponding ciphertext ***C***, and the faulty ciphertext ***C****.
3. For each possible K^R, compute the difference:

$$\gamma_R^{-1} \circ \sigma[K^R](C) \oplus \gamma_R^{-1} \circ \sigma[K^R](C^*)$$

   If it is in $\mathcal{D}$, add K^R to the list $\mathcal{L}$ of possible candidates.

4. Consider a new plaintext ***P***, with corresponding ciphertexts ***C*** and ***C****. Apply step 3 to all candidates of $\mathcal{L}$.

Some Comments

- 2 pairs (C,C*) are enough to retrieve K^R, provided the linear layer θ is optimal.
- If K^R is not enough to retrieve the master key K, last round can be peeled off, and the attack repeated to retrieve K^{R-1}.
- Not practical: Time complexity 2^{8n}.

A Practical Attack

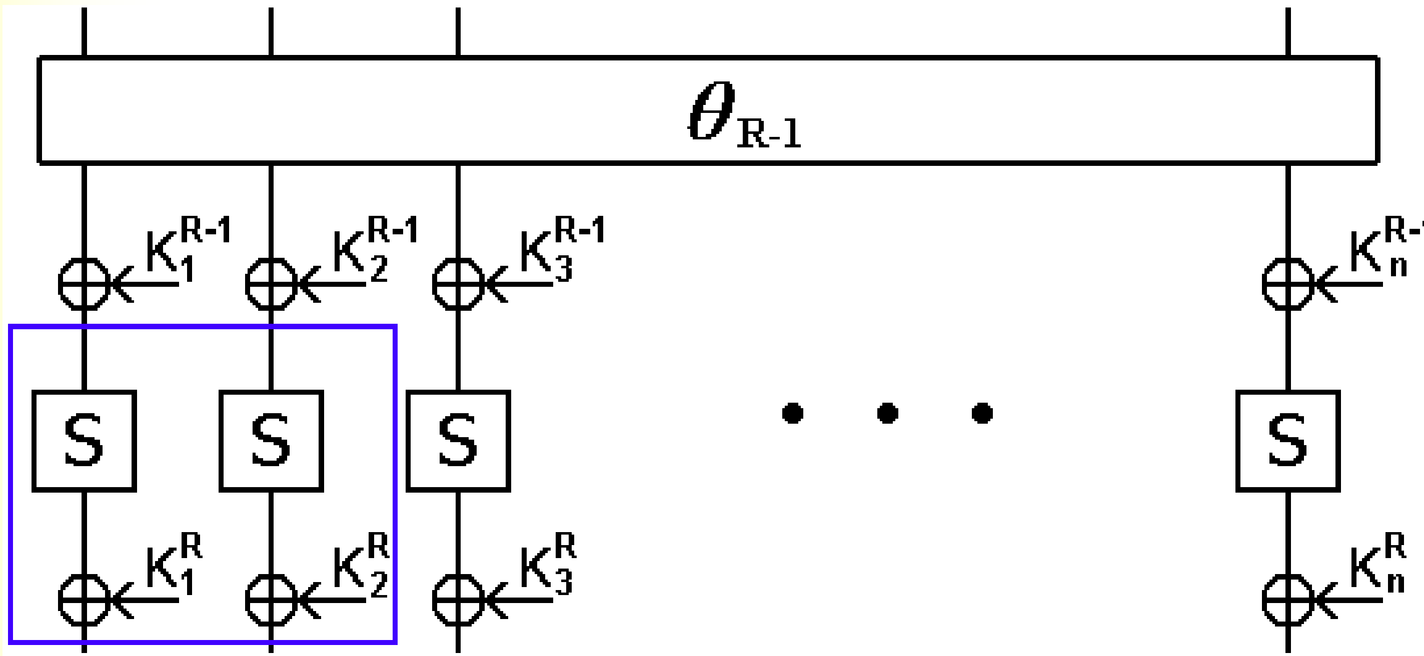
1. Compute the list $\mathcal{D}$ of possible differences before θ_{R-1}
2. Consider two pairs ***(C,C*)*** and ***(D,D*)***.
3. Consider the 2 left-most bytes of K^R. For each of the 2^{16} candidates, compute:

$$\gamma_R^{-1} \circ \sigma[\langle K_1^R, K_2^R \rangle](\langle C_1, C_2 \rangle) \oplus \gamma_R^{-1} \circ \sigma[\langle K_1^R, K_2^R \rangle](\langle C_1^*, C_2^* \rangle)$$

$$\gamma_R^{-1} \circ \sigma[\langle K_1^R, K_2^R \rangle](\langle D_1, D_2 \rangle) \oplus \gamma_R^{-1} \circ \sigma[\langle K_1^R, K_2^R \rangle](\langle D_1^*, D_2^* \rangle)$$

4. Compare the results with the 2 left-most bytes of the differences in $\mathcal{D}$. The $\langle K_1^R, K_2^R \rangle$ for which a match is found for both ciphertext pairs are stored in a list $\mathcal{L}$.

A Practical Attack



5. For each candidate of $\mathcal{L}$, try to extend it by one byte (computing both differences to check).
6. Keep extending candidates in $\mathcal{L}$ until they are n-bytes long. At this stage, only the right key is remaining.

Faults Occurring at a Wrong Location

- Usually the attacker has no control on the fault location.
- Problem: To distinguish pairs ***(C,C*)*** resulting from a fault occuring between θ_{R-2} and θ_{R-1} [*right pairs*] from other pairs [*wrong pairs*].

- If the diffusion layer θ_{R-1} is not optimal: Trivial.
- If θ_{R-1} is optimal, it is not possible to decide whether a single pair ***(C,C*)*** is a *right pair* or not.

Faults Occurring at a Wrong Location

- However if :
  - ***(C,C*)*** is a *right pair.*
  - ***(D,D*)*** is a ***wrong*** *pair.*

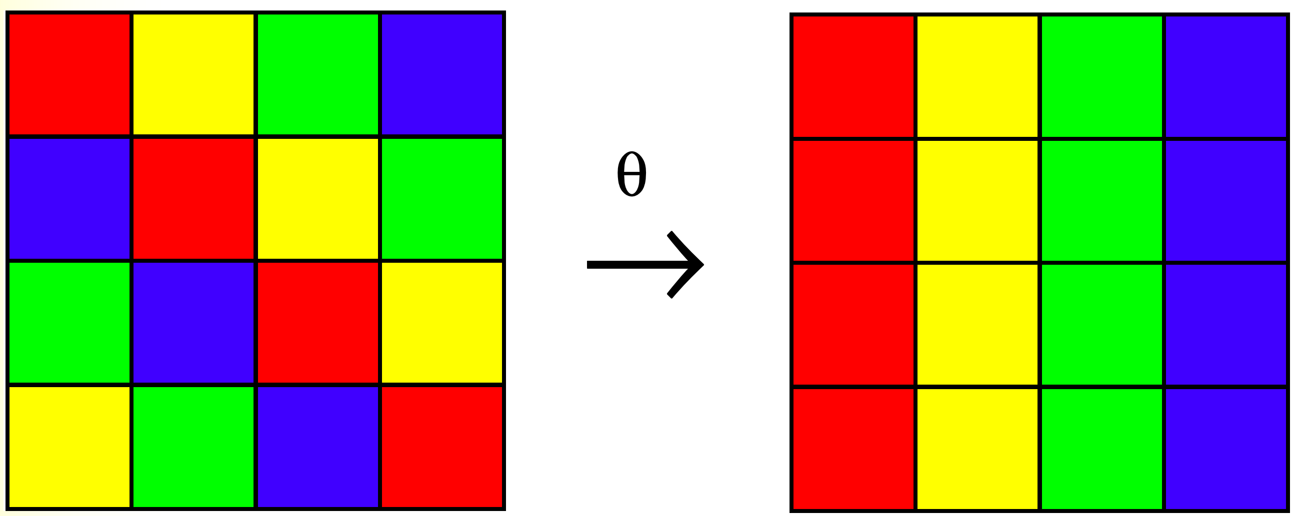
  Then applying the attack to these pairs
  ➔ no solution for K^R.

- Thus wrong pairs can be distinguished, by considering *pairs* of pairs (C,C*).

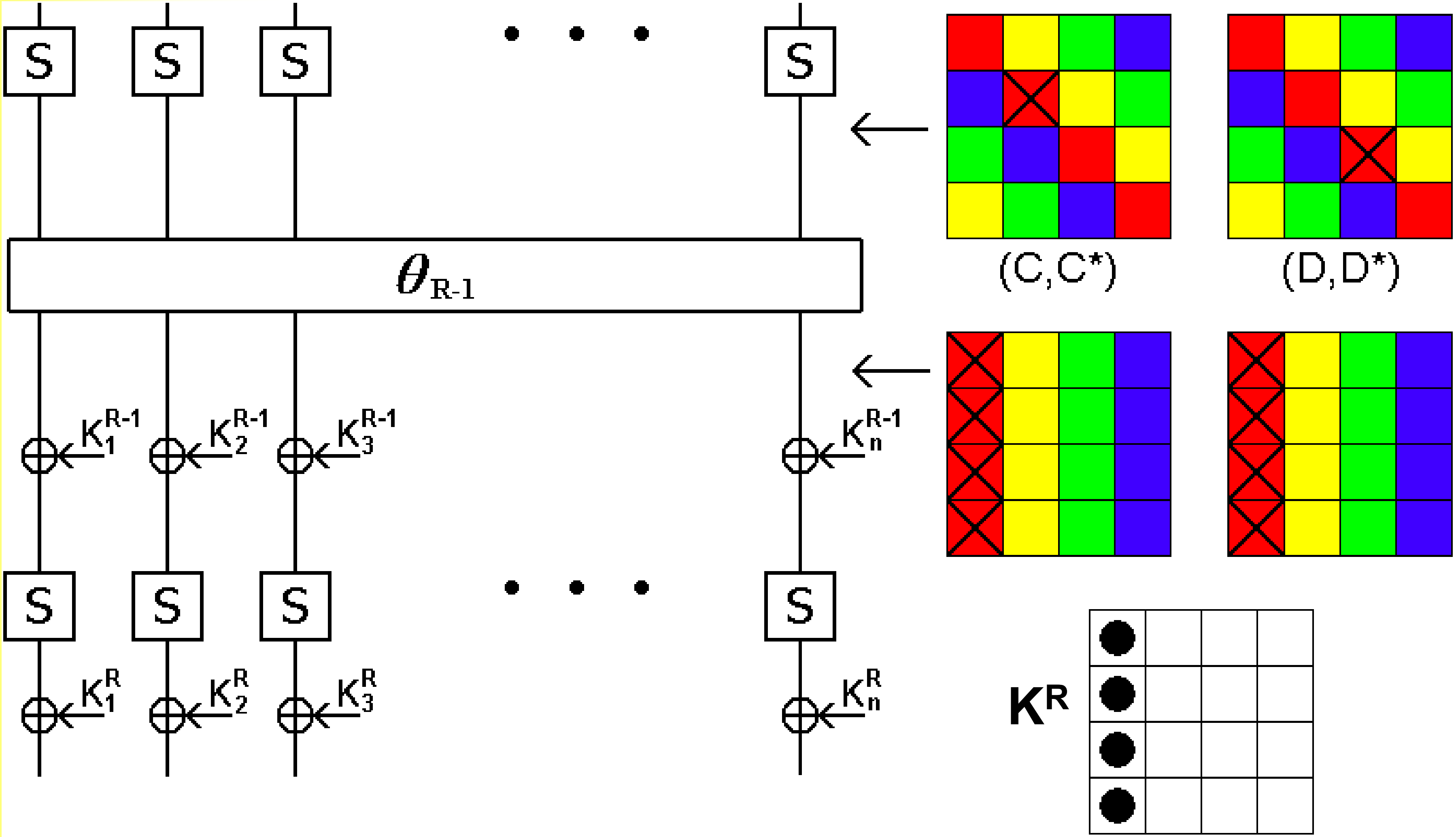
- Suppose 1 pair ***(C,C*)*** out of 50 is right.
  ➔ ~10000 (100*100) pairs **(*(C,C*);(D,D*)*)** need to be examined in order to find K^R. →**Feasible!**

The AES-128

- 128-bit block, 128-bit key variant. 10 rounds SP Network.
- Knowledge of K^R is enough to retrieve the master key.
- Non-optimal linear diffusion layer: Composition of 2 transformations, `ShiftRow` and `Mixcolumn`.



Basic Attack

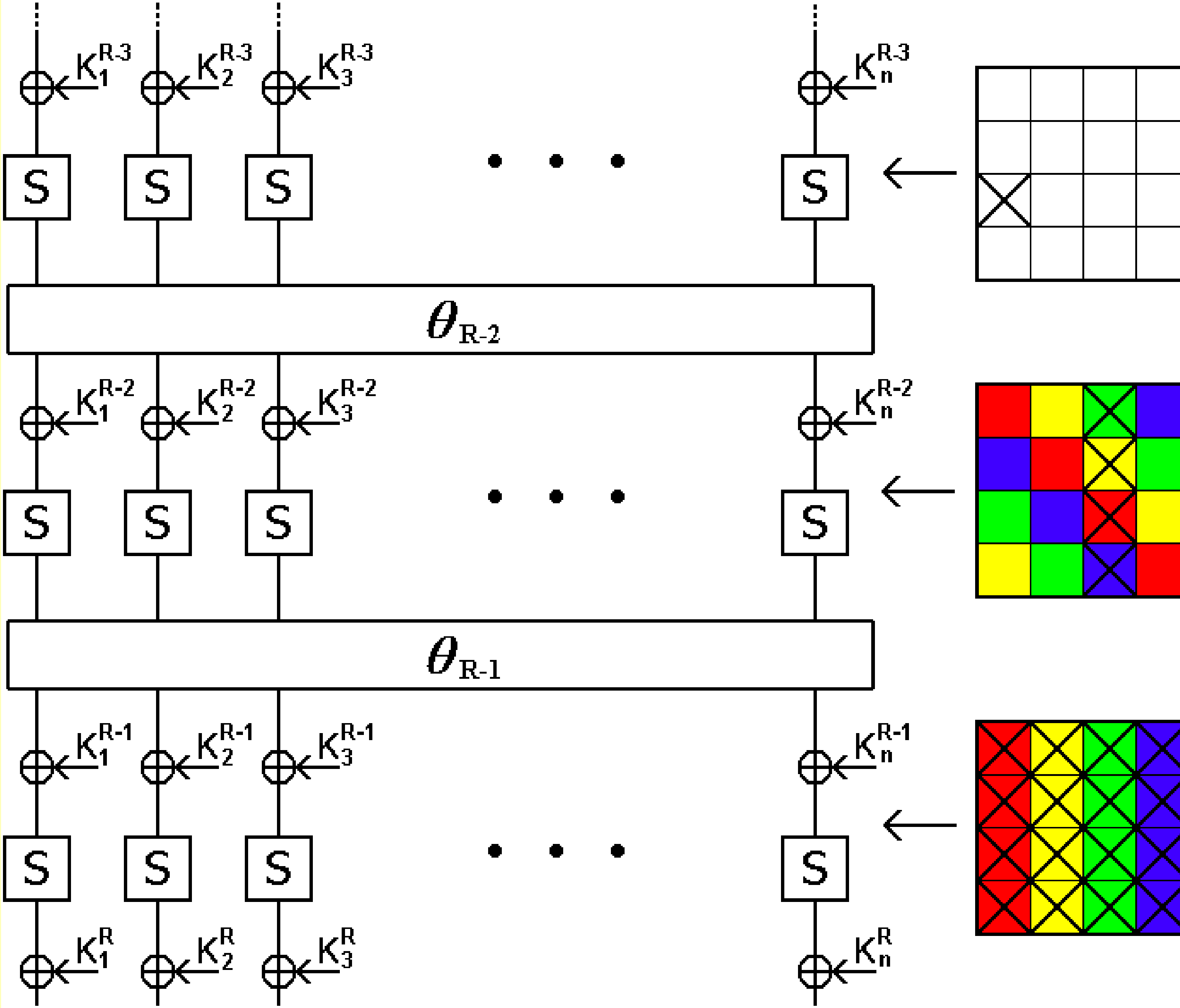


Basic Attack

- If the fault location can be chosen very precisely: 8=4*2 pairs ***(C,C*)*** are needed to retrieve K^R. (but in fact, 6 pairs are enough)
- If we cannot choose the byte where the fault occurs: ~15 pairs are needed.

An Improved Attack

- It is possible to do better if we deal with faults occuring between θ_{R-3} and θ_{R-2} (instead of between θ_{R-2} and θ_{R-1}).


K1R-3
K2R-3
K3R-3
KnR-3
S
S
S
S
θR-2
K1R-2
K2R-2
K3R-2
KnR-2
S
S
S
S
θR-1
K1R-1
K2R-1
K3R-1
KnR-1
S
S
S
S
K1R
K2R
K3R
KnR

Implementation on a PC

- Using 2 right pairs ***(C,C*)***, with fault occurring between θ_{R-3} and θ_{R-2}:
  → Takes a few seconds.
  → Unique candidate retrieved in 77% of the cases.
  → Number of candidates never exceeds 16.

- Applying the attack to 2 pairs one of which is wrong (i.e. corresponds to a fault occuring before θ_{R-3}), the obtained set of solutions was always empty.
  $\Rightarrow$ We can indeed reject wrong pairs !!

Conclusion

- Attack exploits faults on bytes.
- If fault location can be chosen:
  - →Requires only **2** faulty ciphertexts.
  - →Takes a few seconds.
- If fault location **cannot** be chosen:
  - →Requires ~100 faulty ciphertexts
  - →Completes in a few hours.

- Applicable to other ciphers: Khazad, Noekeon, Serpent,…
- The simple and elegant structure of SPNs makes such an efficient attack possible.