Implementation Security In Cryptography

Lecture 11: Fault Attacks

Recap

- Till the last lecture
  - Masking…

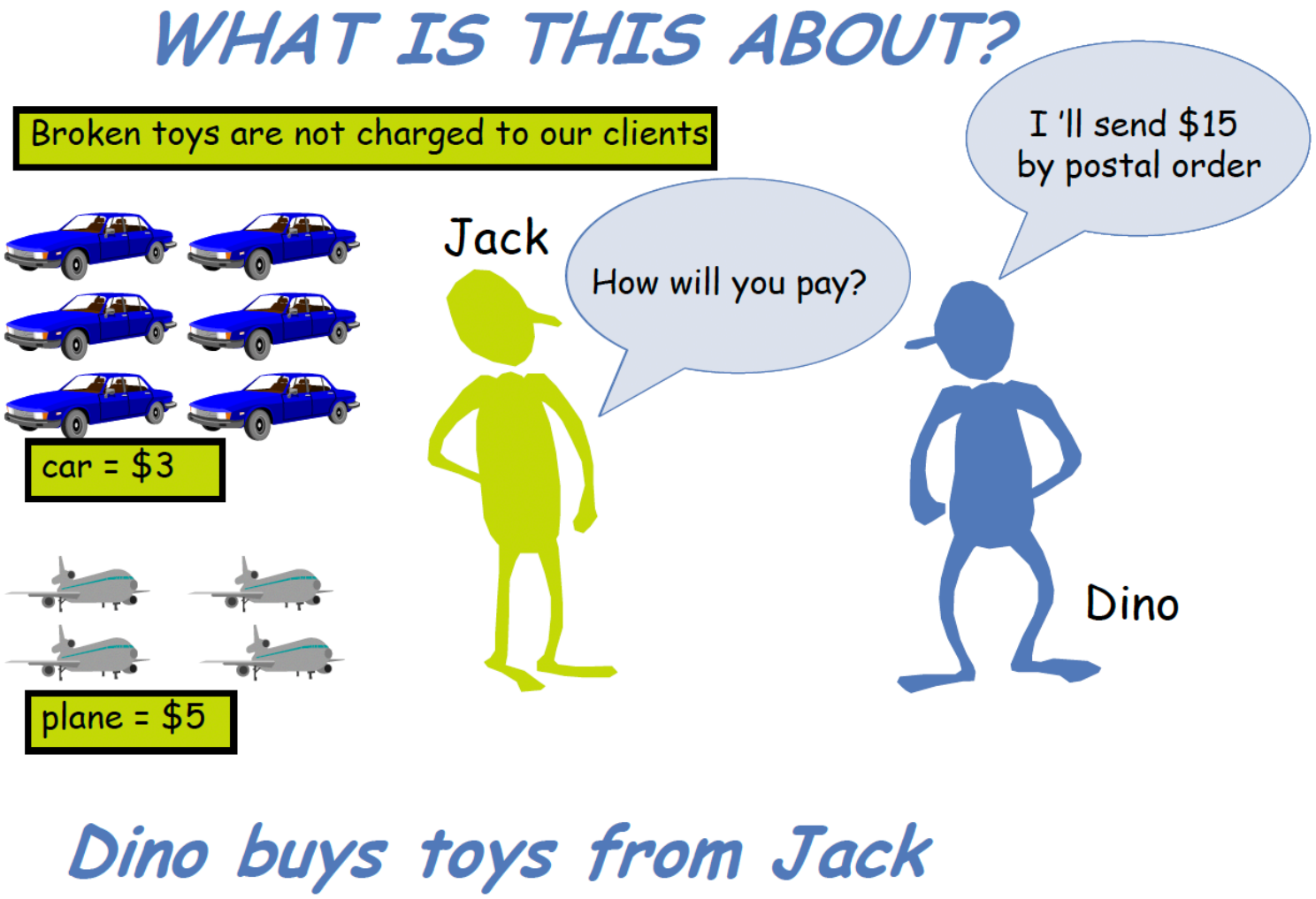
Today

- Fault Attacks

Faults in Our Life…

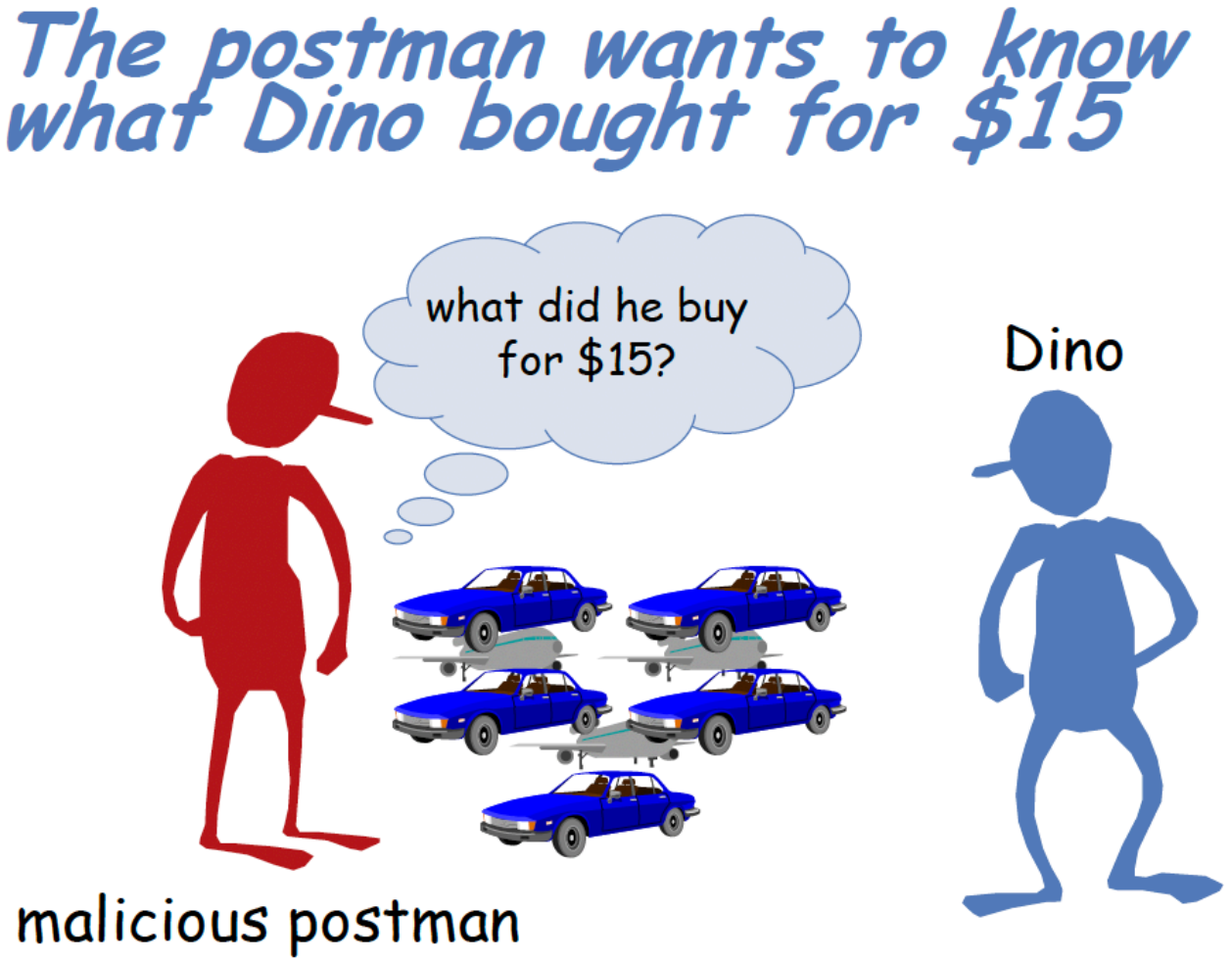
- Happens….
- We are all human being (Hope there is no AI agent in my class) :P
- We all learn from our faults….
  - The learning is what I am looking for here

Faults in Our Life…



- The Sorcerer's Apprenctice's Guide to Fault Attacks, FDTC 2006

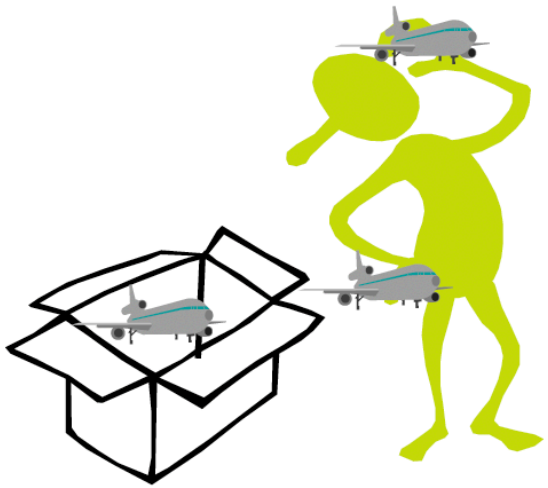
Faults in Our Life…



- The Sorcerer’s Apprenctice’s Guide to Fault Attacks, FDTC 2006

Faults in Our Life…

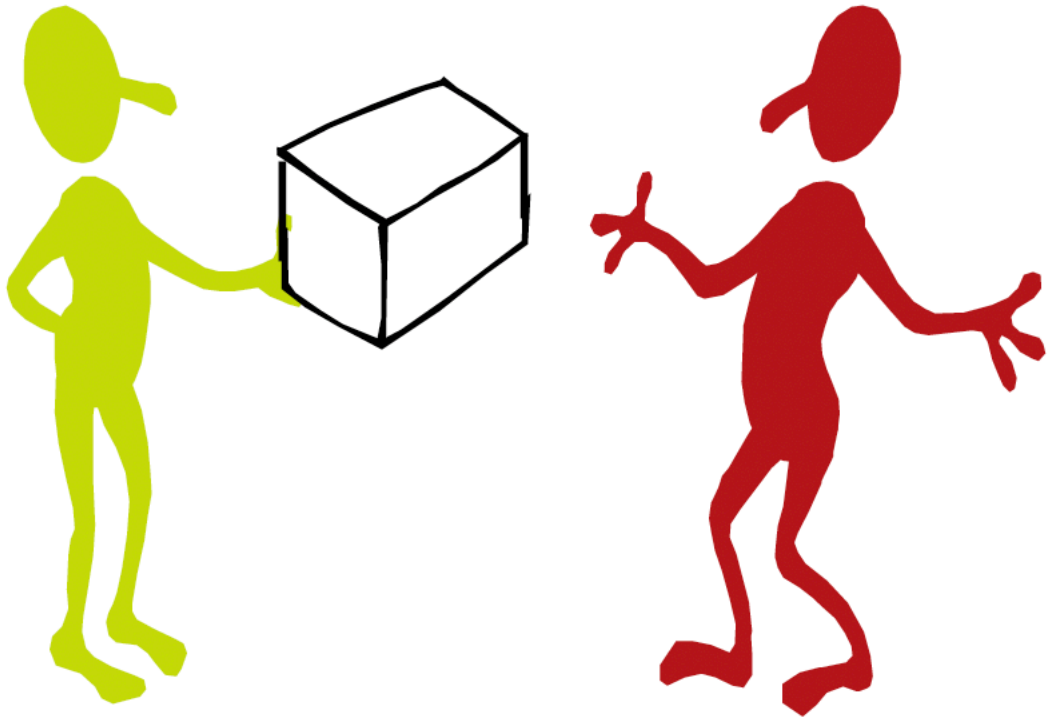
In the meanwhile Jack prepares the DHL



- The Sorcerer’s Apprenctice’s Guide to Fault Attacks, FDTC 2006

Faults in Our Life…

and gives it to the postman

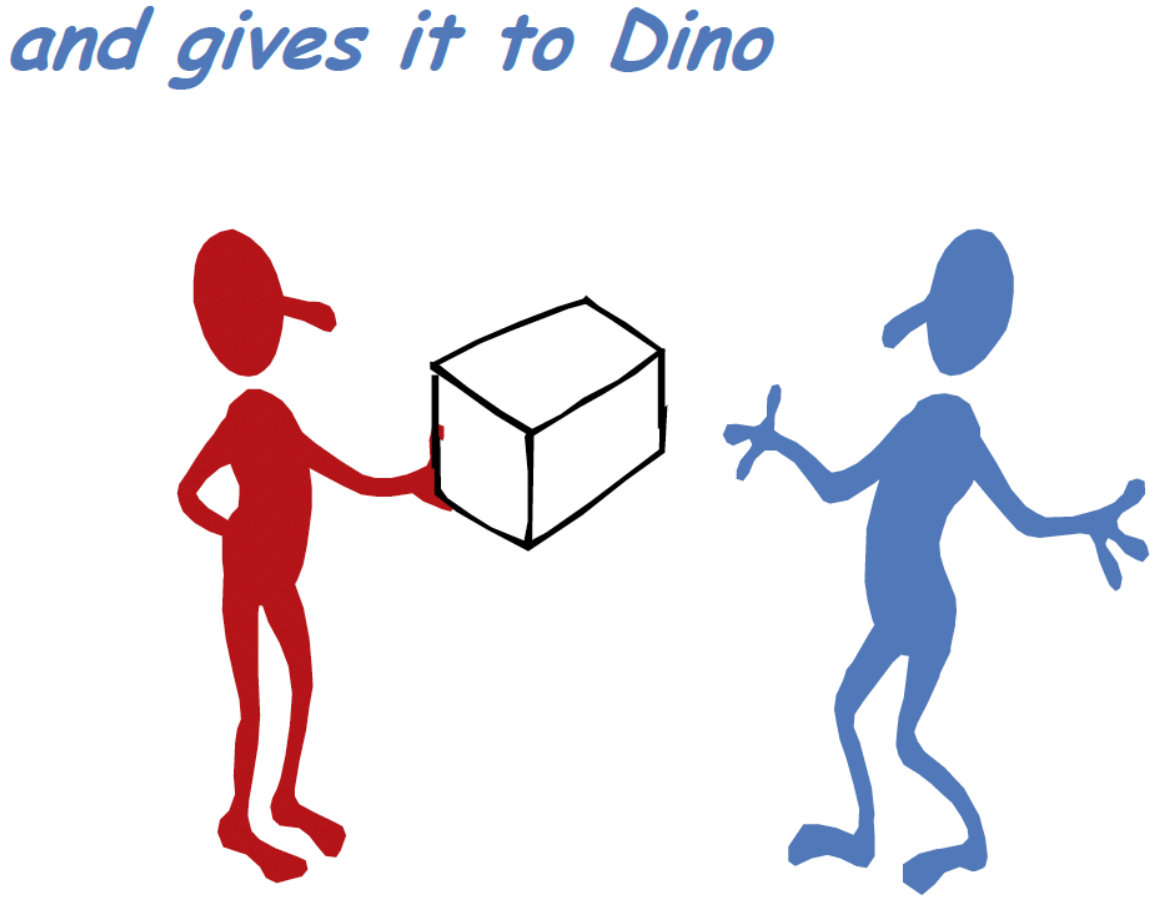


- The Sorcerer’s Apprenctice’s Guide to Fault Attacks, FDTC 2006

Faults in Our Life…

- The Sorcerer’s Apprenctice’s Guide to Fault Attacks, FDTC 2006

Faults in Our Life…



- The Sorcerer’s Apprenctice’s Guide to Fault Attacks, FDTC 2006

Faults in Our Life...

a week later he monitors Dino's postal order...

= 4 × 3 = $12

= 2 × 5 = $10

Lesson learned: **Fault attacks** can also extract secrets from tokens!

Hardware faults can have various sources:
voltage glitches, light beams, laser beams...

- The Sorcerer's Apprenctice's Guide to Fault Attacks, FDTC 2006

Faults in Our Crypto…

- Introduction of faults in the normal execution of cryptographic algorithms and analysis of faulty output to obtain the key
- First conceived in 1996 by Boneh, Demillo and Lipton
- E. Biham developed Differential Fault Analysis (DFA) of DES
- Today there are numerous examples of fault analysis of block ciphers such as AES under a variety of fault models and fault injection techniques
- Popular Fault Injection Techniques – Clock Glitches, Voltage Glitches, EM and Optical Injection Techniques

What Faults are Up To?

Serious Security: Rowhammer returns to gaslight your computer

Gaslights produce a telltale flicker when nearby lamps are lit; DRAM values do something similar when nearby memory cells are accessed.

Written by Paul Ducklin

JULY 10, 2023

NAKED SECURITY | DATA LEAKAGE | ROWHAMMER | SERIOUS SECURITY

You're probably familiar with the word *gaslighting*, used to refer to people with the odious habit of lying not merely to cover up their own wrongdoing, but also to make it look as though someone else is at fault, even to the point of getting the other person to doubt their own memory, decency and sanity.

You might not know, however, that the term comes from a 1930s psychological thriller play called *Gas Light* (spoiler alert) in which a manipulative and murderous husband pretends to spend his evenings out on the town

Technology

Using just a $25 device a researcher hacked into Elon Musk's Starlink system

What will the technology mogul have to say about this?

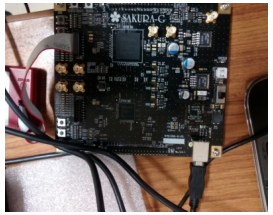
Faults in Our Crypto…

Small World

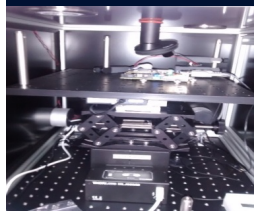
Voltage-Glitch

EM-FI

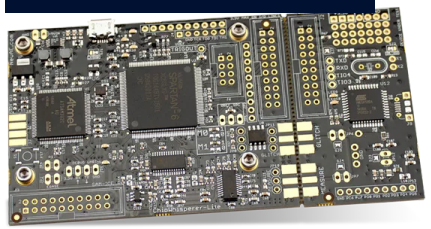
Clock-Glitch



Laser-FI



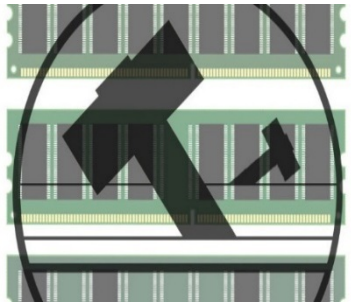
Chipwhisperer



- Tight control on injection timing
  - You can hit the variable you want at a specific instant
- Multi-bit — but can be made single-bit too
- Biased fault distribution
- Mainly transient
- Multiple injection challenging

Big World

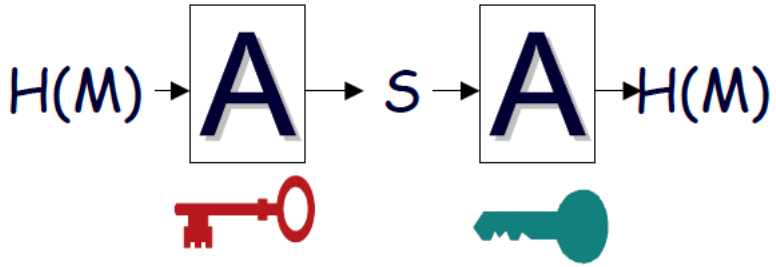
Row-Hammer



- Timing can be controlled but precision is less
  - Target variable must be in RAM for quite sometime
- Single-bit — fairly well controllable
- Mainly persistent
- Multiple injection is easier

A Break From AES...RSA Signature

- Let's talk RSA
- Public key algorithm, — Encryption and Signature
  - We talk about signature.
- **Simple Idea**:
  - Alice generates a secret and a public key.
  - Gives the public key to "Public"
  - Signs a message M with secret key and generates signature C
  - Everyone with the public key can verify that:
    - The C is a valid signature of M
    - C is generated by Alice only and nobody else



RSA Signature in Brief…

- Alice generates two large primes p, q and computes $N = pq$
- It also finds e and d, where $ed \equiv 1 \ mod \ \phi(N)$, this $\phi(N)$ is called Eular's totient function and $\phi(N) = (p-1)(q-1)$, as p, q are primes.
- p, q, d is secret key
- N, e is the public key
- **Sign**: $m \in \mathbb{Z}_n^*$, compute $s = m^d \ mod \ N$, and returns $s \,||\, m$
- **Verify**: Check if $m' == m$, where $m' = s^e \mod N$
- I have omitted some crucial details here for simplicity. But that's not for what we are going to explain..
  - E.g, m is not the message but is a hash of the message..also there are some paddings needed for security…

RSA-CRT

- CRT stands for **Chinese Reminder Theorem** — check it out!!!
- RSA-CRT is a performance optimisation trick…

$$\begin{cases} a \equiv 1 \pmod{p} \\ a \equiv 0 \pmod{q} \end{cases} \text{ and } \begin{cases} b \equiv 0 \pmod{p} \\ b \equiv 1 \pmod{q} \end{cases} \qquad \begin{aligned} d_p &= d \pmod{p-1} \\ d_q &= d \pmod{q-1} \end{aligned} \qquad \begin{aligned} s_p &= m^{d_p} \pmod{p} \\ s_q &= m^{d_q} \pmod{q} \end{aligned}$$

$$s = a \times s_p + b \times s_q \pmod{N}$$

Attacking RSA-CRT…

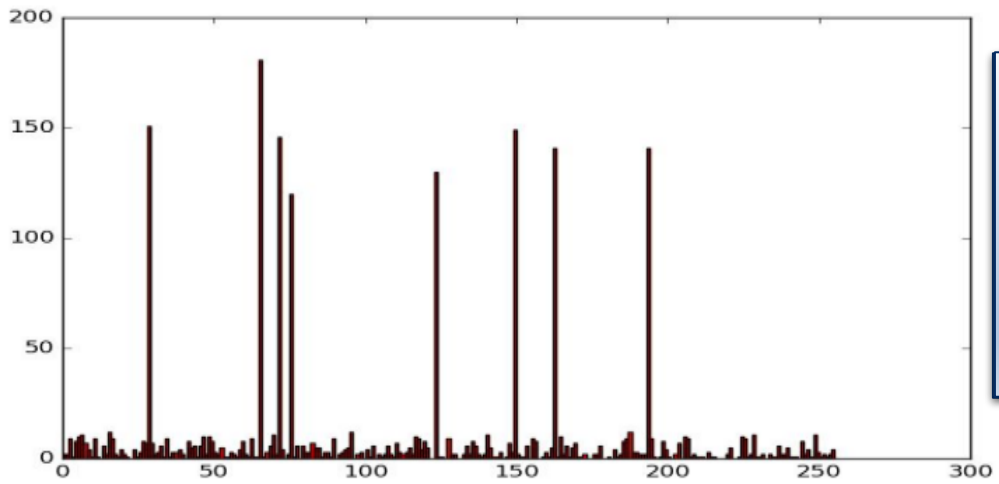
- Again, we want to find the secret key…So attack the signing process..
- Let’s say, for a message m, I can repeat the signature generation process.
- What I do:
  - Generate the correct signature $s = \text{sign}(m, d)$
  - On the same message generate a faulty signature $\hat{s} = \text{sign}(m, d)$
    - Fault happens during the computation of the sign.
    - The fault only corrupts the computation of s_p or s_q, but not both

Attacking RSA-CRT…

- Let $\hat{s} = a \times s_p + b \times \hat{s}_q \ mod \ N$
- Let $s = a \times s_p + b \times s_q \ mod \ N$
- Let $\Delta = s - \hat{s} = b(s_q - \hat{s}_q)$
- Now, observe that, since $b \equiv 0 \ mod \ p$
  - $\Delta = b(s_q - \hat{s}_q)$ is divisible by p
  - So, $\Delta = kp$, and $gcd(s - \hat{s}, n) = p$
- Ok, you got p, you know N, so you know q — game over!!!
- Attack is also possible if you do not use CRT — analysis is slightly different

Let There Be Faults: Fault Model

- Key component of a fault attack
- Attack procedure changes according to the fault model
- Random localized faults
  - Bit/nibble/byte fault
  - Most general model
- Biased faults
  - Device-dependent model
- Instruction skip/modify.
- Constant fault
  - Stuck-at-0/1
- Persistant fault

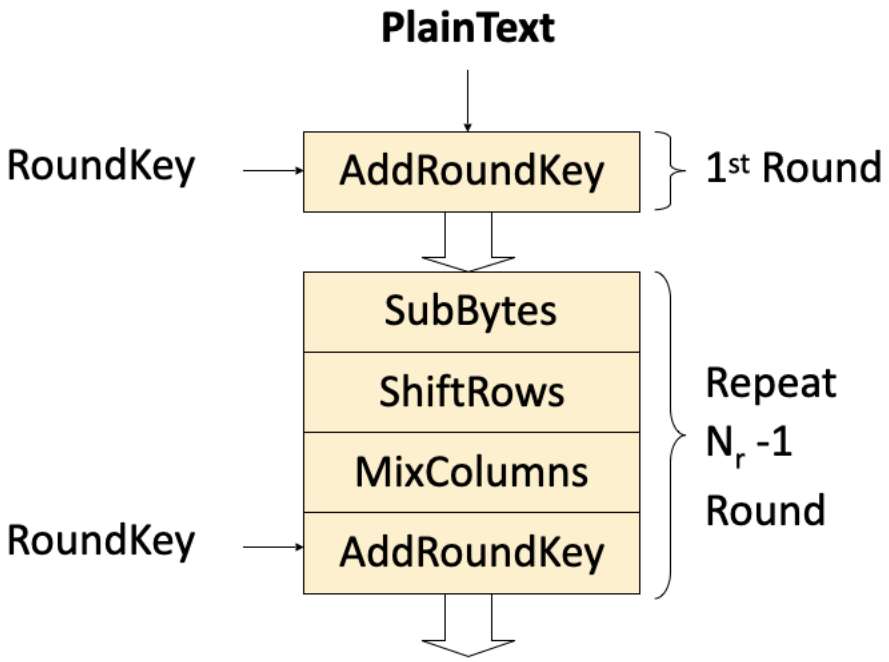


In a fault space of size 256, only 8 faults occur in 98% of the total injections!!!

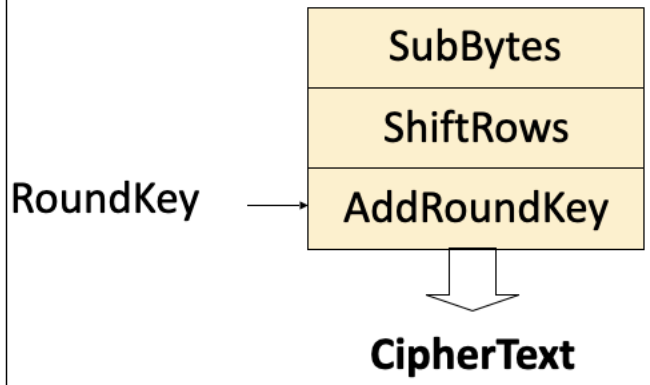
```
ldi   r1  0;
ld    r1  #M1
ldi   r2  0
ld    r2  #M2
add   r1  r2
str   r1
```

```
ldi   r1  0;
nop
ldi   r2  0
ld    r2  #M2
add   r1  r2
str   r1
```

The AES Storv...



First 9 Rounds



Last Round

Looking Inside AES Once Again

State

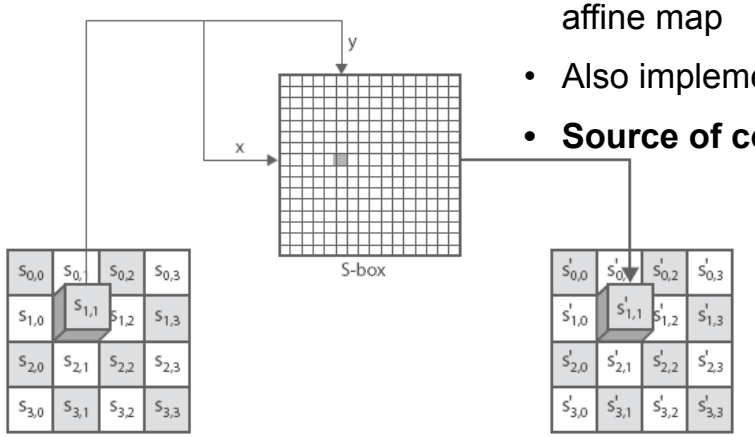
1 byte

s00	s01	s02	s03
s10	s11	s12	s13
s20	s21	s22	s23
s20	s31	s32	s33

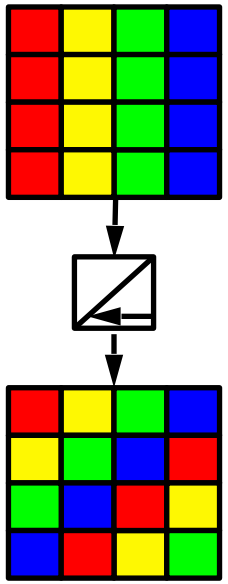
k00	k01	k02	k03
k10	k11	k12	k13
k20	k21	k22	k23
k20	k31	k32	k33

1. SubBytes

- Nonlinear Boolean Function
- Finite field inversion followed by affine map
- Also implemented as a table
- **Source of confusion**

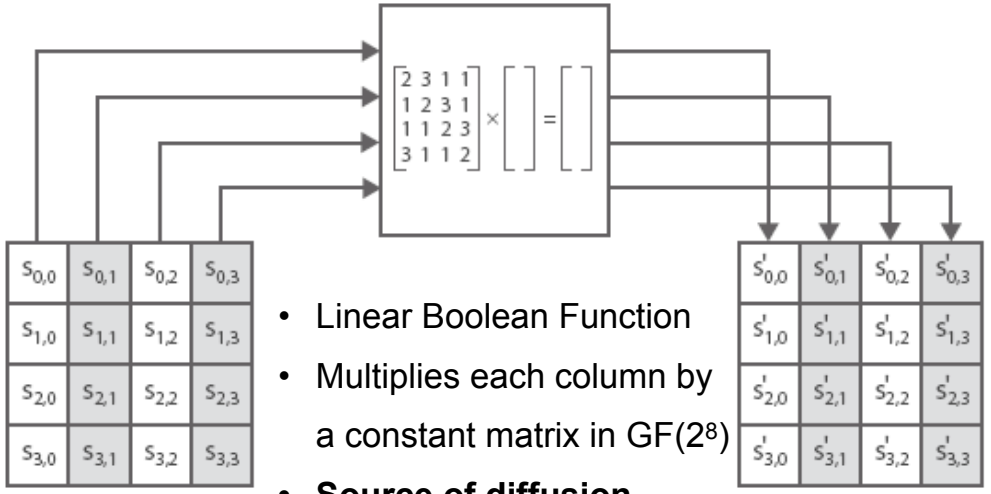


2. ShiftRows



- Linear Boolean Function
- Left circular shift of rows
- **Source of diffusion**

3. MixColumns



- Linear Boolean Function
- Multiplies each column by a constant matrix in GF(2^8)
- **Source of diffusion**

4. AddRoundKey

s00	s01	s02	s03
s10	s11	s12	s13
s20	s21	s22	s23
s20	s31	s32	s33

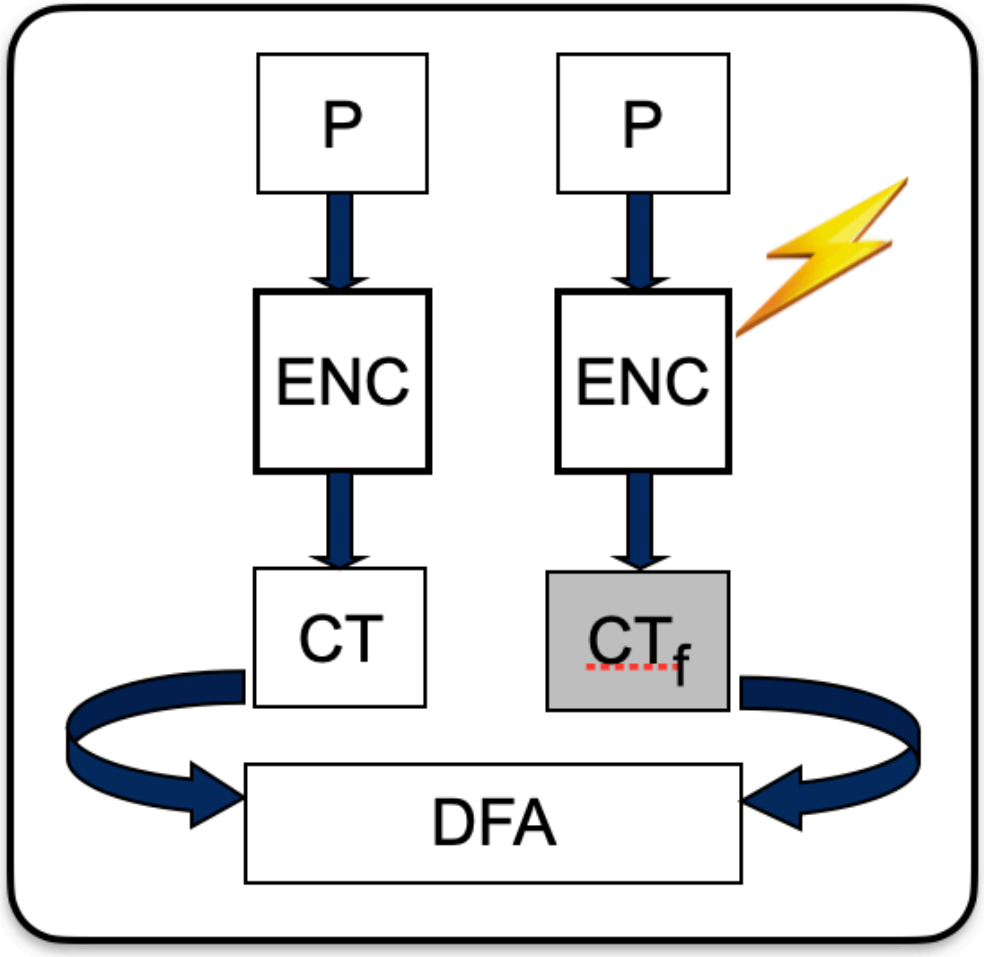
$\oplus$

k00	k01	k02	k03
k10	k11	k12	k13
k20	k21	k22	k23
k20	k31	k32	k33

- Linear Boolean Function
- XOR the state with a round key

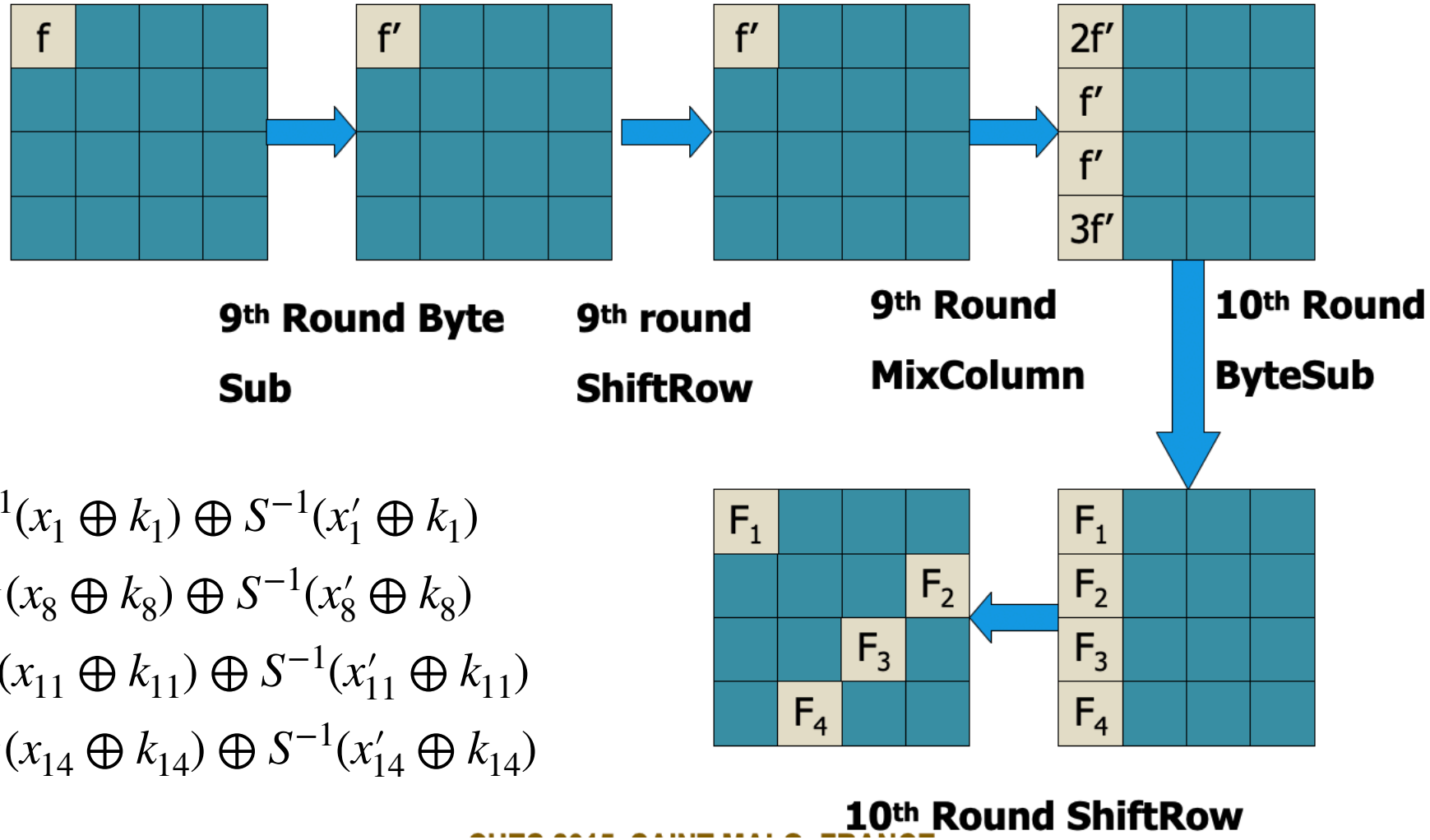
The AES Story…

- The attacker can corrupt one specific round operation of AES.
- One of multiple bytes of the AES state gets corrupted.
- Then what happens?
- We also assume that we have both correct and faulty ciphertext for the same plaintext…
- **Where to inject the fault**:
  - In round functions
  - In key schedule



Let there be a fault at 9th round

Let the correct ciphertext be
$x = (x_1, x_2, \cdots x_{16})$
The faulty ciphertext be
$x' = (x'_1, x'_2, \cdots x'_{16})$



$$2f' = S^{-1}(x_1 \oplus k_1) \oplus S^{-1}(x'_1 \oplus k_1)$$
$$f' = S^{-1}(x_8 \oplus k_8) \oplus S^{-1}(x'_8 \oplus k_8)$$
$$f' = S^{-1}(x_{11} \oplus k_{11}) \oplus S^{-1}(x'_{11} \oplus k_{11})$$
$$3f' = S^{-1}(x_{14} \oplus k_{14}) \oplus S^{-1}(x'_{14} \oplus k_{14})$$

Let there be a fault at 9th round

$$2f' = S^{-1}(x_1 \oplus k_1) \oplus S^{-1}(x_1' \oplus k_1)$$

$$f' = S^{-1}(x_8 \oplus k_8) \oplus S^{-1}(x_8' \oplus k_8)$$

$$f' = S^{-1}(x_{11} \oplus k_{11}) \oplus S^{-1}(x_{11}' \oplus k_{11})$$

$$3f' = S^{-1}(x_{14} \oplus k_{14}) \oplus S^{-1}(x_{14}' \oplus k_{14})$$

- On an average there is one solution to the equation: $S^{-1}(X) \oplus S^{-1}(X + \alpha) = \beta$ — why?
  - Two possible solutions can be $X = 0$ and $X = \alpha$. In that case $\beta = \alpha^{-1}$.
  - If not, then we can transform this to $\beta x^2 + \alpha\beta x + \alpha = 0$ — which (may) have 2 more solutions
  - If $X = 0$ or $X = \alpha$, then the equation can have two more solutions in $GF(2^n)$, depending on n is even or odd. Solutions have form $\{0, \alpha, e\alpha, e^2\alpha\}$, with e being a field ($GF(2^n)$) element. — depends on some deep finite field tricks…

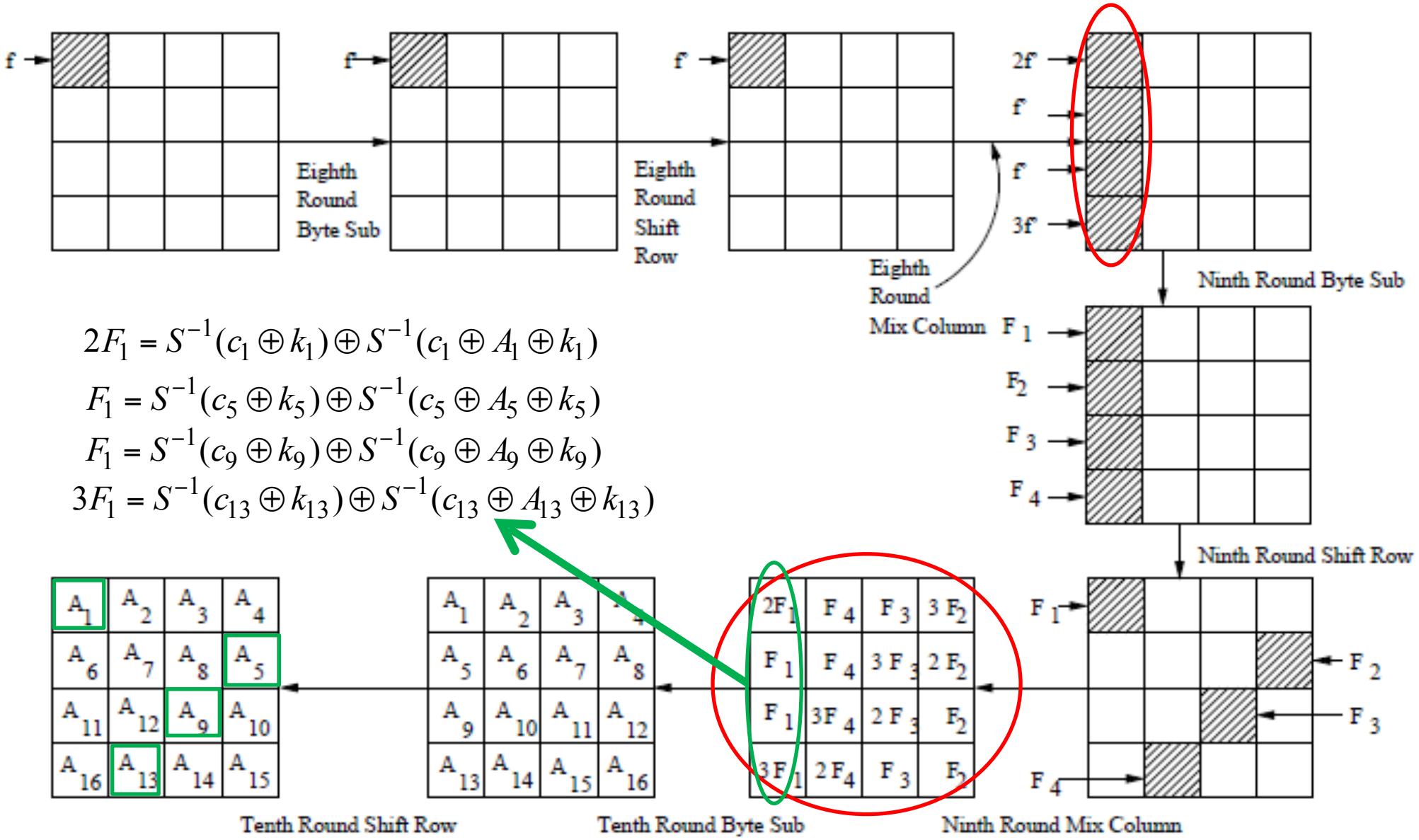
  So, it can have 0, 2, or 4 solutions — on average 1 solutions, we observed….
- Thus for one value of f' there is 1 value for k_1, k_8, k_{11}, k_{14} which satisfies the equations.
- Thus for all the 2^8 values of f', there are 2^8 values for k_1, k_8, k_{11}, k_{14}.
- Thus the total size of AES key is 2^{32}

Let there be a fault at 9th round

- With one faulty cipher text:
  - Number of possible values per 4 bytes of the key is around 2^8.
  - There are 2^{32} possible candidates for 128 bits of the AES key.
  - Brute force key is thus possible!

$$2f' = S^{-1}(x_1 \oplus k_1) \oplus S^{-1}(x'_1 \oplus k_1)$$
$$f' = S^{-1}(x_8 \oplus k_8) \oplus S^{-1}(x'_8 \oplus k_8)$$
$$f' = S^{-1}(x_{11} \oplus k_{11}) \oplus S^{-1}(x'_{11} \oplus k_{11})$$
$$3f' = S^{-1}(x_{14} \oplus k_{14}) \oplus S^{-1}(x'_{14} \oplus k_{14})$$

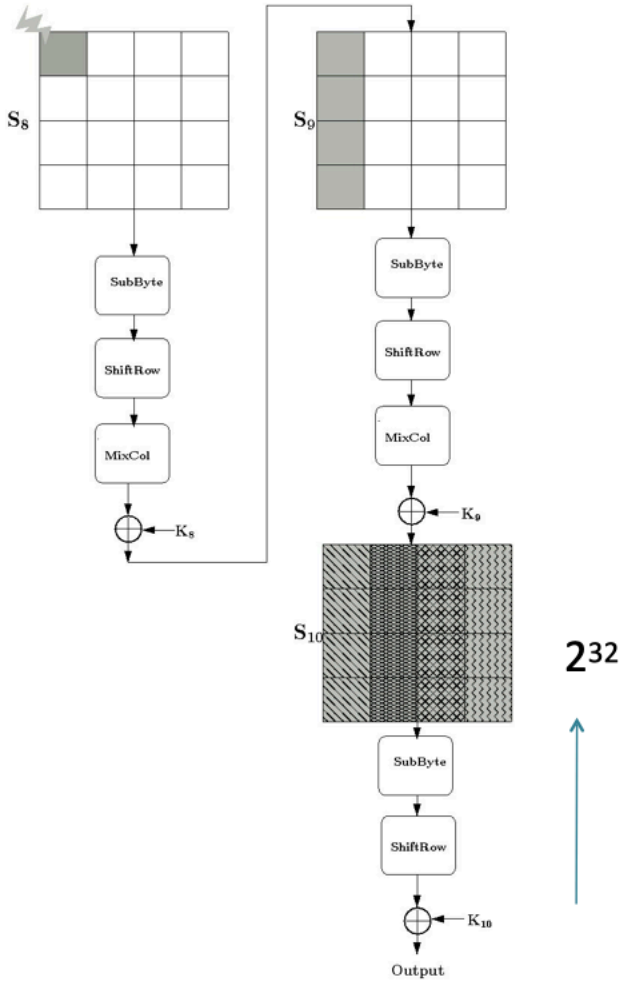
Let's Improve…8th Round Injection



Let’s Improve…8th Round Injection…

- Search space reduced in two phases.
- First phase:
  - ➢Find the 2^{32} candidates of 10^{th} round key.
- Second phase
  - ➢ Deduce four differential equation from differences {2f’,f’,f’,3f’}.
  - ➢ Reduce the 2^{32} candidates to 2^{8} using the four differential equation.

Let’s Improve…8th Round Injection…



2^{32}

2^{128}

- Find 2^{32} candidates K_{10}

Differential Equation

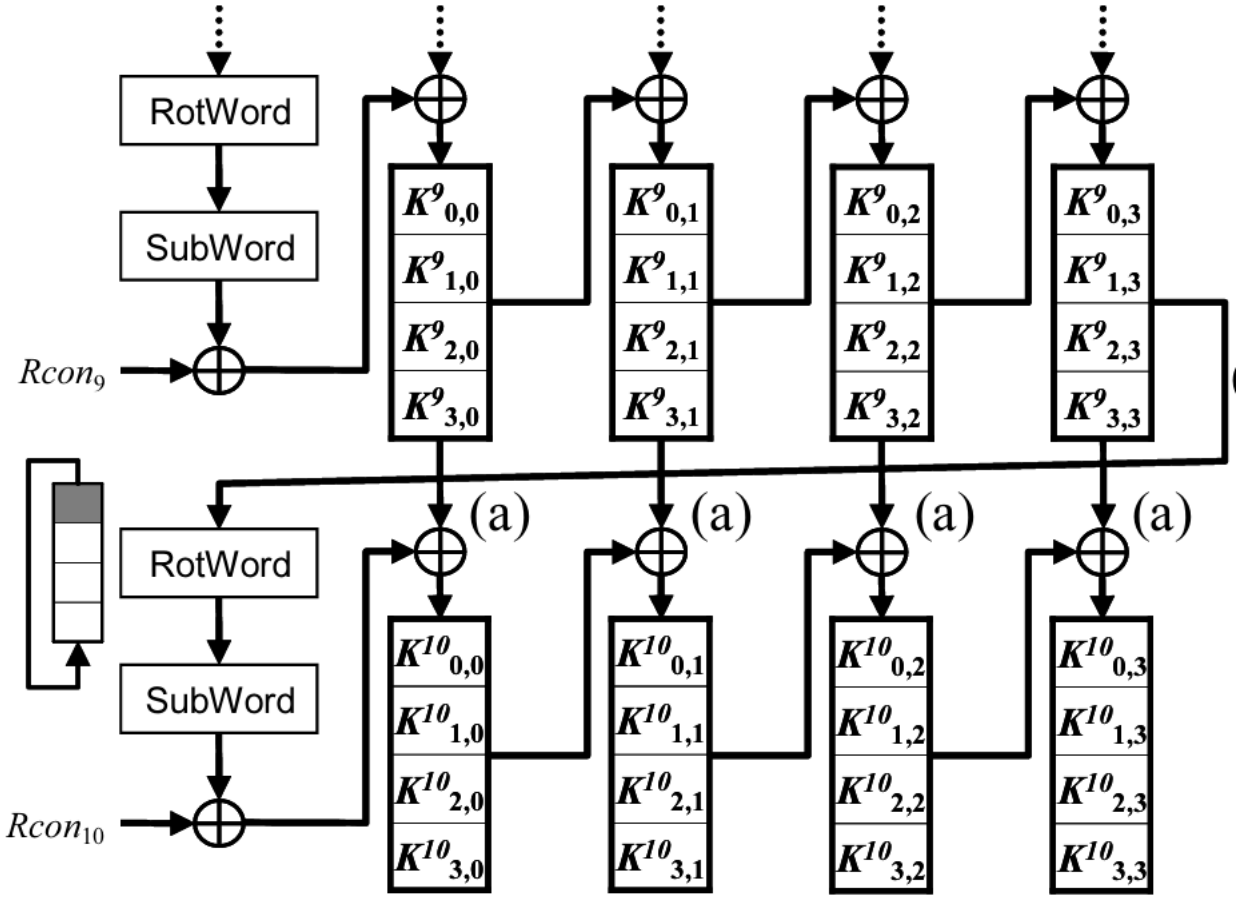
$$2F_1 = S^{-1}(x_0 \oplus k_0) \oplus S^{-1}(x'_0 \oplus k_0)$$

$$F_1 = S^{-1}(x_{13} \oplus k_{13}) \oplus S^{-1}(x'_{13} \oplus k_{13})$$

$$F_1 = S^{-1}(x_{10} \oplus k_{10}) \oplus S^{-1}(x'_{10} \oplus k_{10})$$

$$3F_1 = S^{-1}(x_7 \oplus k_7) \oplus S^{-1}(x'_7 \oplus k_7)$$

AES Key Schedule



Rcon

Algorithm 2: The AES-128 `KeySchedule` function.

Input: $(r-1)^{th}$ round key ($X = x_i$ for $i \in \{1, \ldots, 16\}$).
Output: r^{th} round key X.

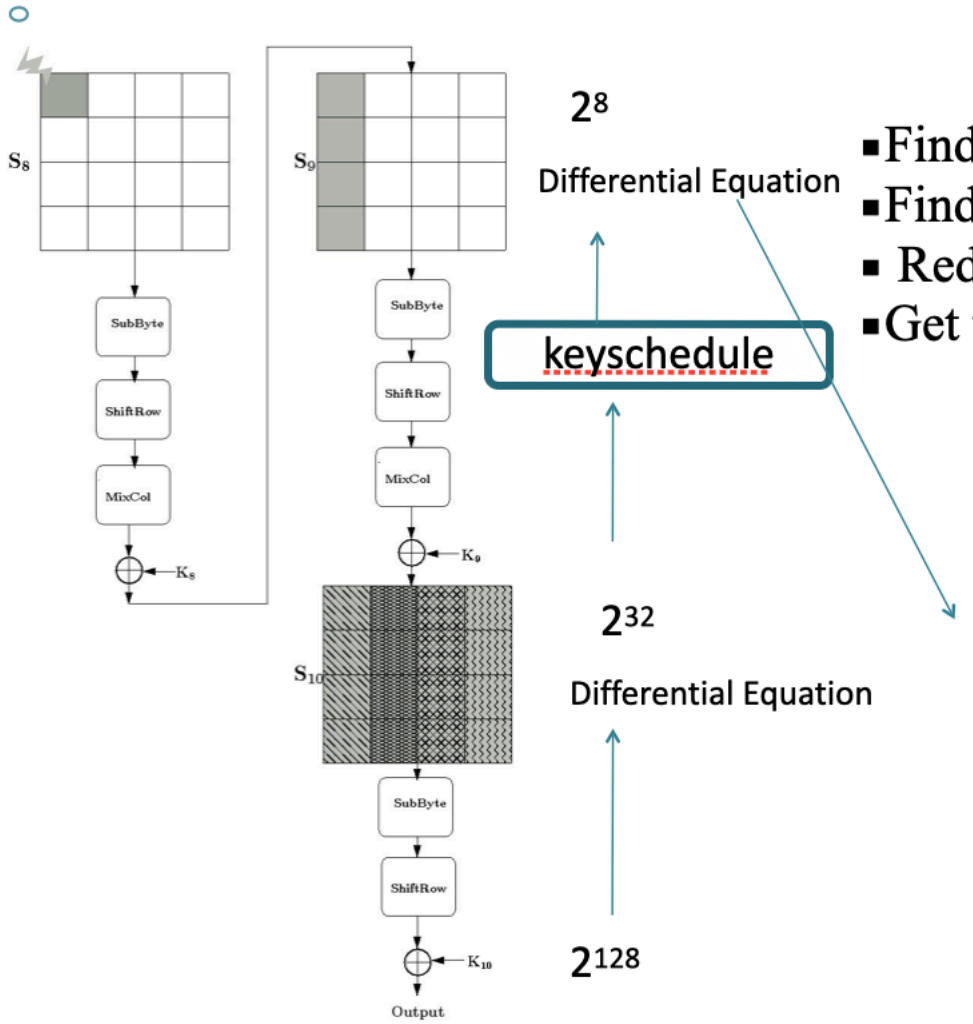
for $i \leftarrow 0$ **to** 3 **do**
  $x_{(i<<2)+1} \leftarrow x_{(i<<2)+1} \oplus S(x_{(((i+1)\wedge 3)<<2)+4})$
end
$x_1 \leftarrow x_1 \oplus h_r$
for $i \leftarrow 1$ **to** 16 **do**
  if $(i-1) \mod 4 \neq 0$ **then**
    $x_i \leftarrow x_i \oplus x_{i-1}$
  end
end
return X

Let's Improve…8th Round Injection…

$$\mathbf{K}^9 = \begin{pmatrix} k_1 \oplus S(k_{14} \oplus k_{10}) \oplus h_{10} & k_5 \oplus k_1 & k_9 \oplus k_5 & k_{13} \oplus k_9 \\ k_2 \oplus S(k_{15} \oplus k_{11}) & k_6 \oplus k_2 & k_{10} \oplus k_6 & k_{14} \oplus k_{10} \\ k_3 \oplus S(k_{16} \oplus k_{12}) & k_7 \oplus k_3 & k_{11} \oplus k_7 & k_{15} \oplus k_{11} \\ k_4 \oplus S(k_{13} \oplus k_9) & k_8 \oplus k_4 & k_{12} \oplus k_8 & k_{16} \oplus k_{12} \end{pmatrix} \cdot \quad \mathbf{K}^{10} = \{k_1, k_2, \cdots k_{16}\}$$

- AES key schedule is invertible

Let's Improve…8th Round Injection…



2^8

Differential Equation

keyschedule

2^{32}

Differential Equation

2^{128}

- Find 2^{32} candidate K^{10}
- Find 2^{32} Candidates of K^9 using keyschedule
- Reduce K^9 to 2^8 candidates
- Get the master key by 2^8 brute-force search

$$2f = \Delta S_9^{(0,0)}$$

$$f' = \Delta S_9^{(0,1)}$$

$$f' = \Delta S_9^{(0,2)}$$

$$3f' = \Delta S_9^{(0,3)}$$

8th Round Injection…How does the Equation Look?

$$
\begin{aligned}
2f' = {} & S^{-1}\Big(14\left(S^{-1}(x_1 \oplus k_1) \oplus k_1'\right) \oplus 11\left(S^{-1}(x_{14} \oplus k_{14}) \oplus k_2'\right) \oplus \\
& 13\left(S^{-1}(x_{11} \oplus k_{11}) \oplus k_3'\right) \oplus 9\left(S^{-1}(x_8 \oplus k_8) \oplus k_4'\right)\Big) \oplus \\
& S^{-1}\Big(14\left(S^{-1}(x_1' \oplus k_1) \oplus k_1'\right) \oplus 11\left(S^{-1}(x_{14}' \oplus k_{14}) \oplus k_2'\right) \oplus \\
& 13\left(S^{-1}(x_{11}' \oplus k_{11}) \oplus k_3'\right) \oplus 9\left(S^{-1}(x_8' \oplus k_8) \oplus k_4'\right)\Big) \\
= {} & S^{-1}\Big(14\left(S^{-1}(x_1 \oplus k_1) \oplus ((k_1 \oplus S(k_{14} \oplus k_{10}) \oplus h_{10}))\right) \oplus \\
& 11\left(S^{-1}(x_{14} \oplus k_{14}) \oplus (k_2 \oplus S(k_{15} \oplus k_{11}))\right) \oplus \\
& 13\left(S^{-1}(x_{11} \oplus k_{11}) \oplus (k_3 \oplus S(k_{16} \oplus k_{12}))\right) \oplus \\
& 9\left(S^{-1}(x_8 \oplus k_8) \oplus (k_4 \oplus S(k_{13} \oplus k_9))\right)\Big) \oplus \\
& S^{-1}\Big(14\left(S^{-1}(x_1' \oplus k_1) \oplus ((k_1 \oplus S(k_{14} \oplus k_{10}) \oplus h_{10}))\right) \oplus \\
& 11\left(S^{-1}(x_{14}' \oplus k_{14}) \oplus (k_2 \oplus S(k_{15} \oplus k_{11}))\right) \oplus \\
& 13\left(S^{-1}(x_{11}' \oplus k_{11}) \oplus (k_3 \oplus S(k_{16} \oplus k_{12}))\right) \oplus \\
& 9\left(S^{-1}(x_8' \oplus k_8) \oplus (k_4 \oplus S(k_{13} \oplus k_9))\right)\Big)
\end{aligned}
$$

- We have 4 such equations

Let’s Improve…8th Round Injection…

- Time complexity of previous attack: $O(2^{32})$
- Time complexity of this attack: $O(2^{8})$
- Just with one fault injection!!!