

Fault Attacks

Kostas Papagiannopoulos
University of Amsterdam
kostaspap88@gmail.com // kpcrypto.net

Contents

Introduction

Differential Fault Analysis of DES

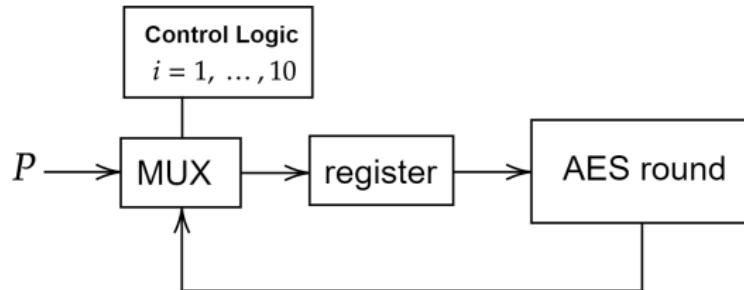
Introduction

Introduction

- ▶ So far we have used **passive** attacks
 - e.g. differential cryptanalysis observes plaintext/ciphertext, timing attacks measure processing time, power attacks measure chip consumption, etc.
- ▶ We will now move to **active** attacks where the adversary does not just observe but can also modify the computation

Introduction

Fault Injection: Data Corruption



- ▶ The round-based hardware implementation of AES iterates for 10 rounds
- ▶ A control logic circuit keeps track of the round i
- ▶ What would happen if the logic circuit is faulted and gets stuck to $i = 1$?
- ▶ We would perform a very weak encryption

Introduction

Fault Injection: Instruction Skip

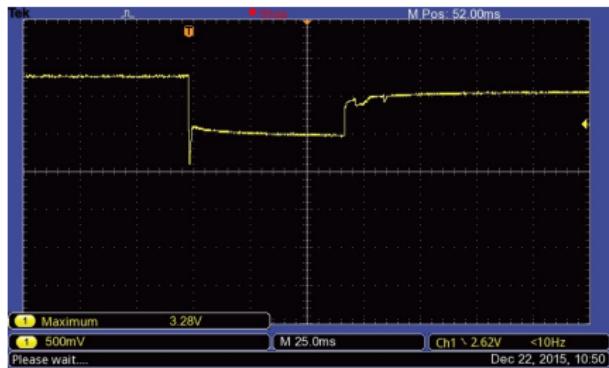
- ▶ Assume that a PIN check gets compiled to the following assembly code

```
pushq %rax;  
pushq %rbx;  
callq .PIN_Check_function;  
movq %rax, %rdx;  
...
```

- ▶ What happens if the CPU gets faulted when it is about to call the PIN Check function?
- ▶ If the instruction opcode gets altered it could result in an unknown opcode
- ▶ The CPU treats unknown opcodes as nop instructions
- ▶ Thus the PIN code check gets skipped

Introduction

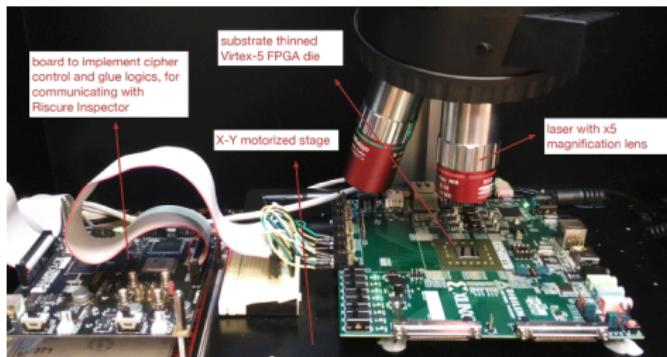
- ▶ How to inject faults on a device?
- ▶ Decrease or increase the power supply for a very small time
- ▶ **Voltage glitching** is one of the earliest fault attacks yet remains simple, effective and applicable to numerous devices. It is a **non-invasive technique** since it does not need device modifications.



- ▶ On the downside voltage glitches result often in coarse faults

Introduction

- ▶ How to inject precise faults on a device?
- ▶ Target circuits with a focused laser beam
- ▶ **Laser-based fault injection** is a more complex process that can result in fine-grained control over the injected fault. It is a **semi-invasive** technique since it typically needs chip decapsulation



- ▶ On the downside, achieving precise faults may require a large parameter search effort, trying to locate the exact circuit spot, the right laser intensity, etc.

Introduction

Fault Model:

- ▶ Various methods will produce different types of faults, thus many fault categories exist
- ▶ **Granularity:** the fault can alter a single-bit or multiple-bits or several bytes
- ▶ **Modification:** the faulted value is affected by random bitflips or is now biased according to some statistical distribution. Similarly the faulted value can be stuck-at-zero or stuck-at-one.
- ▶ **Control:** The fault can be injected on a large chip region or on a small part of the surface. Likewise the fault can be injected with low and high precision in time
- ▶ **Duration:** The fault can be transient (the faulted value reverts), persistent (we need to reset the device to revert the faulted value), permanent (the faulted value cannot be altered)

Differential Fault Analysis of DES

DFA

- ▶ Original attack by Biham and Shamir (1997)
- ▶ To perform DFA, we encrypt the same plaintext P twice: once in a fault-free manner and once while faulting the encryption algorithm

fault-free encryption: $C = \text{enc}(P, K)$

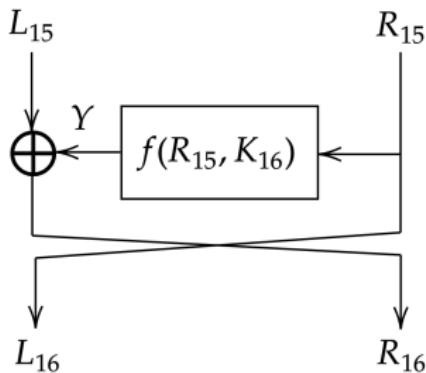
faulted encryption: $C' = \text{enc}^{\frac{1}{2}}(P, K)$

- ▶ Thus DFA assumes that the attacker can control and fix the plaintext P and can also observe the ciphertexts C, C'
- ▶ The attack exploits the difference between C and C' (i.e. $C \oplus C'$) to obtain information about the secret key K
- ▶ For the attack to work, we must generate several plaintexts P , capture the respective ciphertext pairs (C, C') and use their difference to gradually recover the constant key K

DFA

Fault model and propagation:

- ▶ DES has an internal state of 64 bits, split in the left half L and the right half R
- ▶ The attacker is able to flip randomly bits of the right half R
- ▶ The fault is injected before the beginning of the final DES round i.e. we fault R at the beginning of round 16. Thus value R_{15} is faulted.
- ▶ The injected fault (random bitflips on R_{15}) is transient



fault-free DES round 16

- ▶ In the Feistel construction with round function $f(\cdot)$ it holds that:

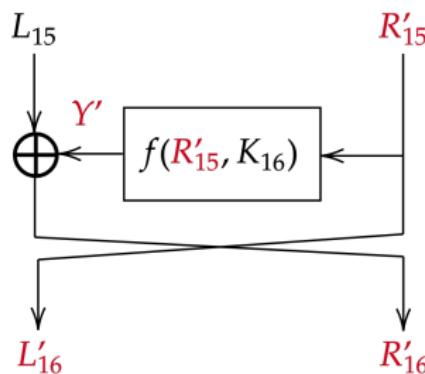
$$L_{16} = R_{15}$$

$$R_{16} = L_{15} \oplus f(R_{15}, K_{16})$$

- ▶ To produce the DES ciphertext C we apply the final permutation $FP(\cdot)$ to the 16th round output:

$$C = FP([L_{16}, R_{16}])$$

DFA



faulted DES round 16

- ▶ A fault is injected in R_{15} causing bitflips

$$R'_{15} = R_{15} \oplus \epsilon$$

$$\epsilon \in_R \{0, 1\}^{32}$$

- ▶ The Feistel construction propagates the fault to the output:

$$L'_{16} = R'_{15}$$

$$R'_{16} = L_{15} \oplus f(R'_{15}, K_{16})$$

$$C' = FP([L'_{16}, R'_{16}])$$

DFA

Attack Idea:

- ▶ The fault-free and faulted ciphertexts (C, C') are available to the attacker.
DFA applies the inverse final permutation $FP(\cdot)$ of DES to compute $[L_{16}, R_{16}]$ and $[L'_{16}, R'_{16}]$ from (C, C')

$$[L_{16}, R_{16}] = FP^{-1}(C), \quad [L'_{16}, R'_{16}] = FP^{-1}(C')$$

- ▶ **Fault differential.** We define the following XOR-difference between the fault-free value R_{16} and the faulted value R'_{16}

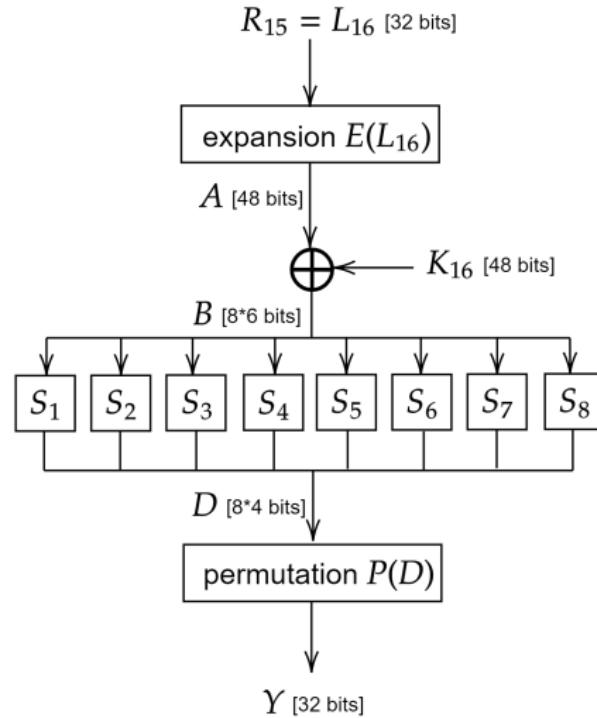
$$\Delta R_{16} \stackrel{\text{def}}{=} R_{16} \oplus R'_{16} = (L_{15} \oplus f(R_{15}, K_{16})) \oplus (L_{15} \oplus f(R'_{15}, K_{16})) =$$

$$f(R_{15}, K_{16}) \oplus f(R'_{15}, K_{16}) = f(L_{16}, K_{16}) \oplus f(L'_{16}, K_{16})$$

- ▶ The fault differential ΔR_{16} , together with values L_{16} and L'_{16} will be used to recover information about the round key K_{16}

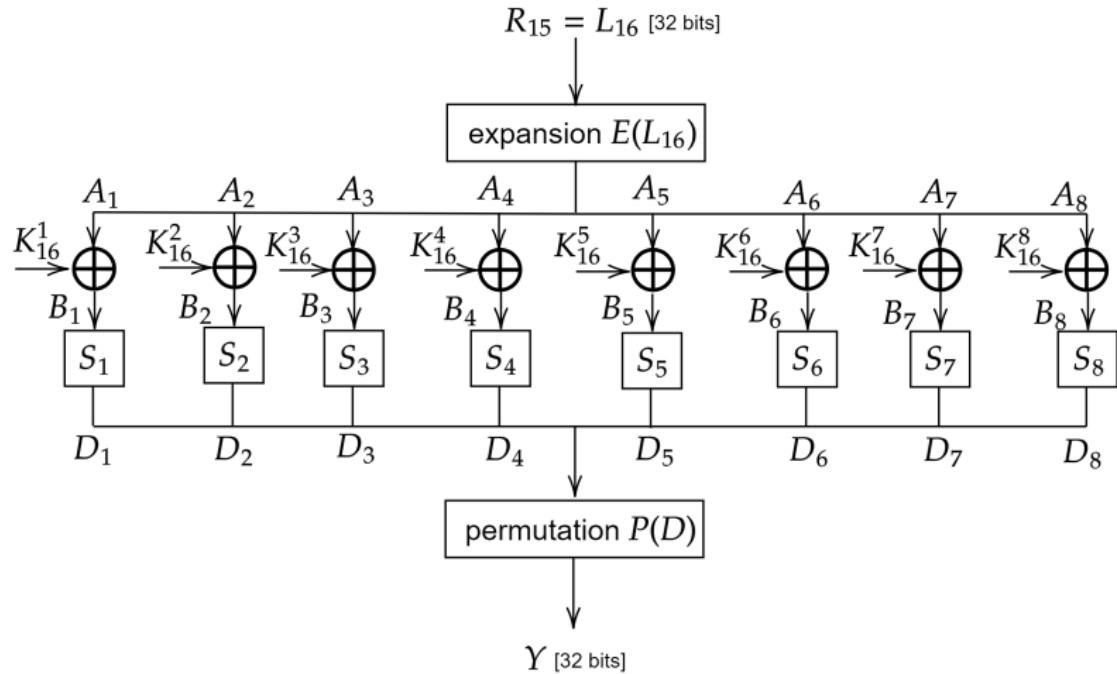
DFA

- ▶ Reminder of the DES function $f(\cdot)$ on round 16

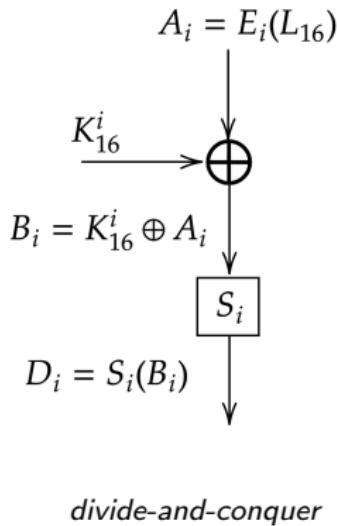


DFA

- ▶ Divide-and-conquer approach



DFA



- We isolate the i th sbox $S_i(\cdot)$ of the DES function $f(\cdot)$

$$i = 1, 2, \dots, 8$$

- The 32-bit input L_{16} gets expanded to the 48-bit value A . We isolate the i th 6-bit part of A and index it as A_i

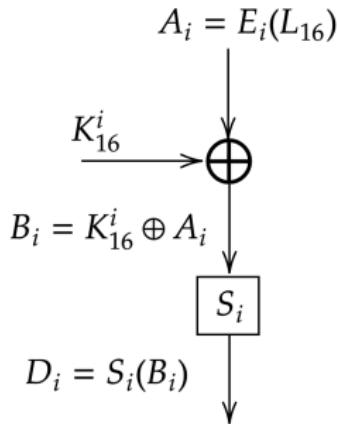
$$A = E(L_{16}), \quad A_i = E_i(L_{16})$$

Notation: $E_i(L_{16})$ reads as: “we compute the 48-bit value $E(L_{16})$ and select the i th 6-bit part of the result”

- The value A_i gets XORed with the 6-bit key part K_{16}^i . The result is the 6-bit value B_i

$$B_i = K_{16}^i \oplus A_i$$

DFA



- ▶ The i th sbox of DES $S_i(\cdot)$ is applied to B_i
The result is the 4-bit value D_i

$$D_i = S_i(B_i)$$

- ▶ Putting these steps together we get:

$$D_i = S_i(B_i) \iff$$

$$D_i = S_i(K_{16}^i \oplus A_i) \iff$$

$$D_i = S_i(K_{16}^i \oplus E_i(L_{16}))$$

divide-and-conquer

DFA

- ▶ Using the fault differential ΔR_{16} and the structure of $f(\cdot)$ we have:

$$\Delta R_{16} = f(L_{16}, K_{16}) \oplus f(L'_{16}, K_{16}) \iff$$

$$\Delta R_{16} = Y \oplus Y' \iff \Delta R_{16} = P(D) \oplus P(D')$$

- ▶ The permutation $P(\cdot)$ is a linear operation:

$$\Delta R_{16} = P(D \oplus D')$$

- ▶ We apply the inverse permutation $P^{-1}(\cdot)$ to both sides of the equation:

$$P^{-1}(\Delta R_{16}) = P^{-1}(P(D \oplus D')) \iff P^{-1}(\Delta R_{16}) = D \oplus D'$$

- ▶ We isolate the i th DES sbox:

$$P_i^{-1}(\Delta R_{16}) = D_i \oplus D'_i \iff$$

$$P_i^{-1}(\Delta R_{16}) = S_i(E_i(L_{16}) \oplus K_{16}^i) \oplus S_i(E_i(L'_{16}) \oplus K_{16}^i)$$

Using the **DFA equation** we are able to recover the 6-bit key part K_{16}^i

DFA

Putting the DFA attack together:

1. Acquire faulty ciphertexts: generate n random plaintexts P and repeat the following process. Store the n ciphertext pairs (C, C') .

```
1 for j=1 until n do
2   instructions;
3    $P \xleftarrow{R} \{0, 1\}^{64}$ 
4    $C = enc(P, K)$ 
5    $C' = enc^{\frac{1}{2}}(P, K)$ 
6 end
```

– We try to recover the key using the 1st pair (C, C') . The process will be repeated for all ciphertext pairs (n in total).

2. Invert the DES final permutation $FP(\cdot)$ for the ciphertext pair (C, C')

$$FP^{-1}(C), \quad FP^{-1}(C')$$

3. Split the pair $(FP^{-1}(C), FP^{-1}(C'))$ to left and right parts

$$[L_{16}, R_{16}] \leftarrow FP^{-1}(C), \quad [L'_{16}, R'_{16}] \leftarrow FP^{-1}(C')$$

4. Compute the fault differential ΔR_{16} and apply the inverse permutation $P^{-1}(\cdot)$

$$\Delta R_{16} = R_{16} \oplus R'_{16}, \quad P^{-1}(\Delta R_{16})$$

5. Compute the expansion $E(\cdot)$ of values L_{16} and L'_{16}

$$E(L_{16}), \quad E(L'_{16})$$

- The DES function $f(\cdot)$ consists of 8 sboxes $S_i(\cdot)$ and operates on 8 6-bit key parts K_{16}^i , where $i = 1, 2, \dots, 8$
- We try to recover the 1st 6-bit key part (K_{16}^1) i.e. we focus on sbox $i = 1$. The same process will be repeated for all sboxes (8 in total).

DFA

7. For all possible values of K_{16}^1 i.e. for $k \in \{0, 1, 2, \dots, 2^6 - 1\}$

7.1 Isolate the values related to sbox S_1

$$P_1^{-1}(\Delta R_{16}) \xleftarrow{i=1} \Delta R_{16}, \quad E_1(L_{16}) \xleftarrow{i=1} E(L_{16}), \quad E_1(L'_{16}) \xleftarrow{i=1} E(L'_{16})$$

7.2 Construct the DFA equation

$$P_1^{-1}(\Delta R_{16}) = S_1(E_1(L_{16}) \oplus k) \oplus S_1(E_1(L'_{16}) \oplus k)$$

7.3 If the DFA equation holds, then k is a valid candidate for K_{16}^1
If it does not, then we discard k

DFA

DFA on DES round 16:

```
1 for  $j = 1$  until  $n$  do
2    $(C, C') \leftarrow j$ th ciphertext pair
3   -compute  $FP^{-1}(C), FP^{-1}(C')$ 
4   -split  $[L_{16}, R_{16}] \leftarrow FP^{-1}(C)$  and  $[L'_{16}, R'_{16}] \leftarrow FP^{-1}(C')$ 
5   -compute  $P^{-1}(\Delta R_{16}), E(L_{16}), E(L'_{16})$ 
6   for  $i = 1$  until 8 do
7     candidates =  $\emptyset$ 
8     for  $k = 0$  until  $2^6 - 1$  do
9        $P_i^{-1}(\Delta R_{16}) \leftarrow P^{-1}(\Delta R_{16}), E_i(L_{16}) \leftarrow E(L_{16}), E_i(L'_{16}) \leftarrow E(L'_{16})$ 
10      check =  $P_i^{-1}(\Delta R_{16}) == S_i(E_i(L_{16}) \oplus k) \oplus S_i(E_i(L'_{16}) \oplus k)$ 
11      if check then
12        | candidates = candidates  $\cup$   $k$ 
13      end
14    end
15    if  $j == 1$  then
16      |  $K_{16}^i = candidates$ 
17    else
18      |  $K_{16}^i = K_{16}^i \cap candidates$ 
19    end
20  end
21 end
```

Final notes on DFA:

- ▶ Very strong attack that requires only a small amount of fault injections
- ▶ Requires observable ciphertext and the ability to keep the plaintext constant for a fault-free and a faulted encryption
- ▶ DFA has many variants: various fault models can be injected on different DES rounds or other ciphers

Countermeasures:

- ▶ **Redundancy.** Run the encryption with the same plaintext input more than once and compare the ciphertexts. If they match then no fault was injected.
- ▶ **Error detection/correction.** Enhance the cipher with a scheme that detects and/or corrects tampering of internal values during computation
- ▶ **Sensors.** Implement sensors on the device that detect e.g. voltage fluctuations or laser injections.
- ▶ Still, ciphertext comparison can also be faulted, certain faults (ineffective faults) can bypass redundancy/error detection and sensors may accidentally render the device useless