RECALL OF LAST SESSION

- **Fault attacks**
  - Test bench setup (means / cartography…)
  - Example of single bit switch on RSA key (ECC mult)
  - Attack on CRT (Bellcore)
  - Attacks on the code (PIN auth)
  - Attacks on symmetric cipher (DES / AES)

- **Safe error attacks**
  - Example on a Square & Multiply always (with dumies)
  - Example on a clear/set key register

SIDE CHANNEL ATTACKS

- Means
- Timing attacks
- SPA
- DPA
- CPA
- Template Attacks

SIDE CHANNEL ATTACKS- HISTORY

- Kocher in 1996
- Measure the computation time of an algorithm
- Example on RSA
- Example on PIN verification

TIMING ATTACKS (1/2)

PIN verification

```
correctPIN = {1,2,3,4}
bool isPinOk(char* presentedPin){
   for(i=0 to 3){
   if (presentedPIN[i] != correctPIN[i]){
     return false
   }
   return true
}
```

TIMING ATTACKS (2/2)

- **Attack on previous code**
  - Try 10 values for presentedPIN[0] from 0 to 9
  - Measure each computation for the 10 tries
  - The correct value is the one where the time is the longer
  - Do it again for the 3 other searched digits

- **Cost of the attack: 40 tries only instead of the 10^4 theoretical ones**

SIDE CHANNEL ATTACKS : TIMING ANALYSIS

- Other examples
  - Square and multiply for RSA gives the Hamming Weight of d
  - A naïve exponentiation would give d
  - Cache-timing attacks
    - The value of a secret can invalidate CPU cache => this is slower
- Countermeasures
  - Balanced code (no dependency on a secret value)

```
correctPIN = {1,2,3,4}
isPinOk = true;
bool isPinOk(char* presentedPin){
    for(i=0 to 3){
         isPinOk &= (presentedPIN[i] == correctPIN[i])
    }
    return isPinOk
}
```

SPA

- **Simple Power Analysis**
- **Measure the power consumption / EM of the chip during a command execution**
- **Aim**
  - Discover easily the secret key
  - See conditional branches
  - Locate different part of the algorithm

SPA DES

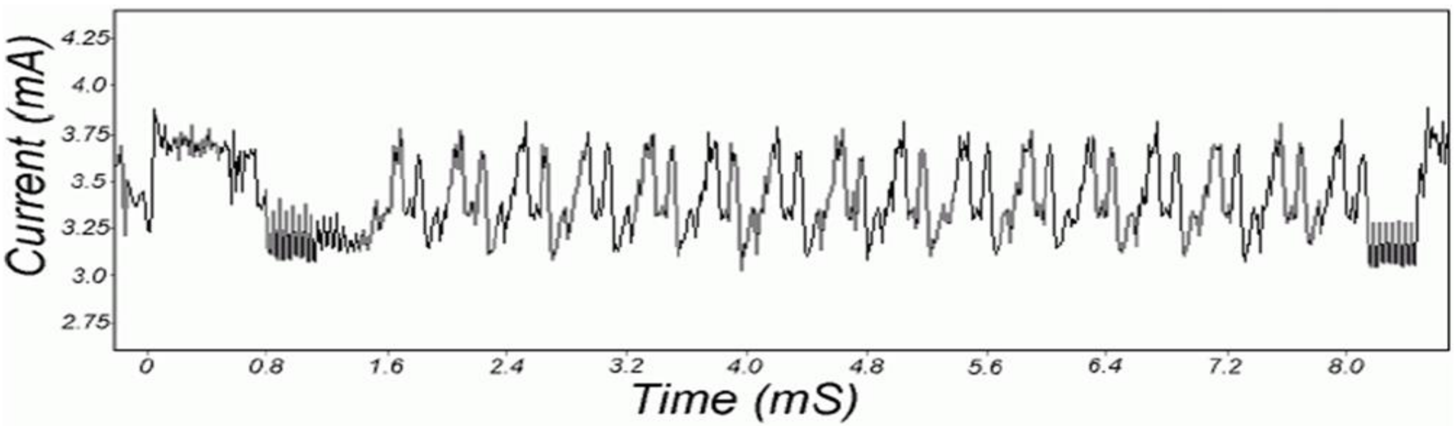


Figure 1: SPA trace showing an entire DES operation.

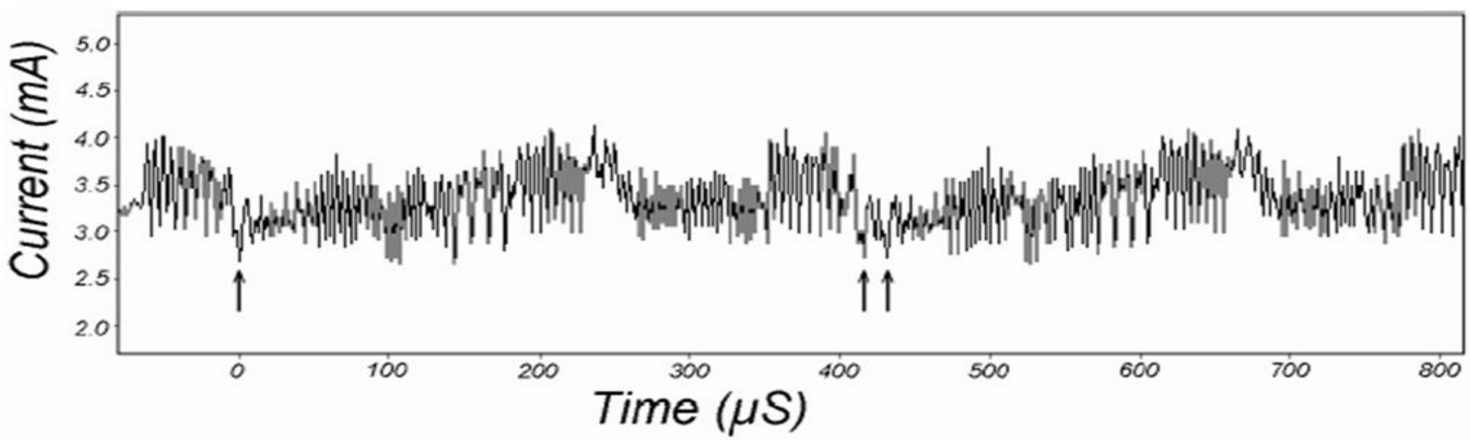


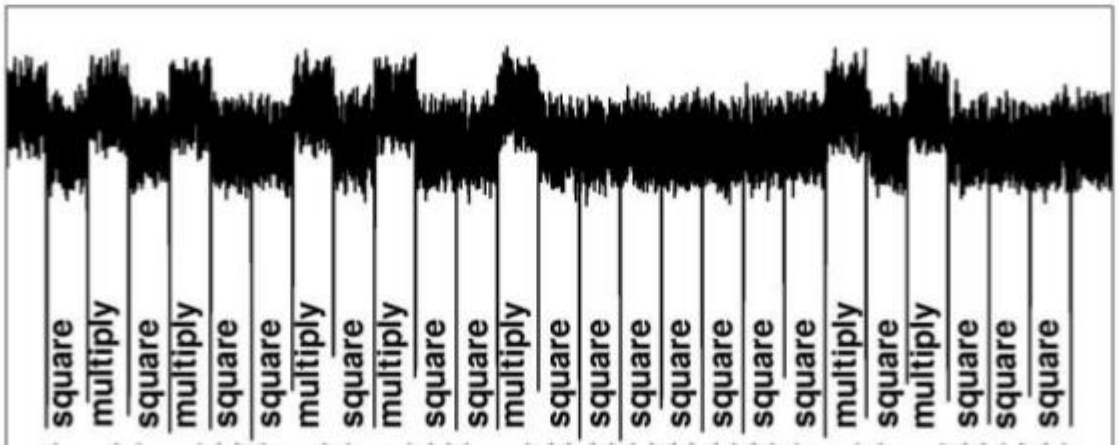
Figure 2: SPA trace showing DES rounds 2 and 3.

SPA: EXAMPLE ON ASYMMETRIC ALGORITHM

- Signature generation or decryption
- $S = m^d \bmod n$
  - m is the message to sign
  - n is the modulus (p*q where p and q are primes numbers) of size 1024 bits for example
  - d is the secret key of the size of the modulus (e*d = 1 mod (p-1)*(q-1), where e is the public exponent)
- Aim: find d

SPA ON RSA

- RSA S&M



- Power consumption depends on the data manipulated. For example the power consumption of a register is linear with the HW of the data written

CONCLUSION ON SPA

Use of representative patterns

Attack strategy

- Know the used algorithm
- Make hypotheses on the implementation

Countermeasures

- Software: balanced code (whole or parts of the algorithm), no conditions
- Hardware: modification of the signal shape, jitter (desynchronisation)

DPA: INTRODUCTION

- Evolution of SPA
- Requirements
  - EM or power consumption measure
  - Knowledge of the algorithm
  - Several curves
  - Knowledge of plaintexts or ciphertexts

DPA PRINCIPLES

- **Information leaks**
- **Power consumption depends on**
  - Manipulated data
  - Executed instruction
- **Leakage models**
  - Hamming weight of data, address or OpCode
  - Weight of transitions (bit inversion on bus state)
  - Others depending on chip

DPA THE RECIPE

- 1. Acquizition
- 2. Selection Function
- 3. Statistical attack
- 4. retrieve the key

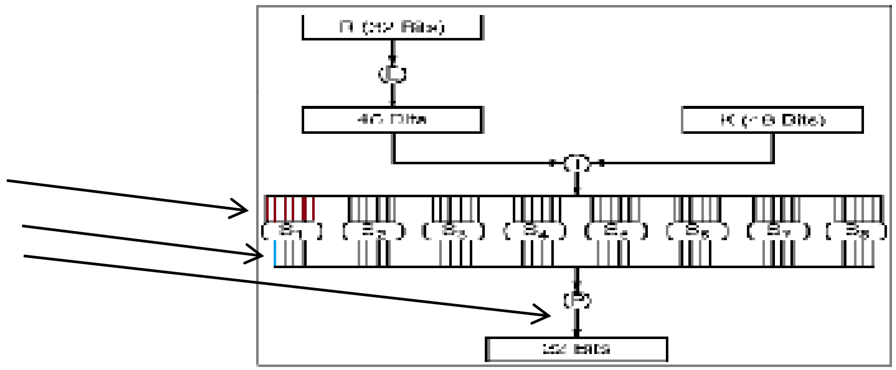
DPA RECIPE

- 1. Acquizition
  - feed the smartcard with a known plaintext (or get the output ciphertext)
  - Capture the EM/power consumption
  - Save the trace
- This operation is repeated several times with a new plaintext

DPA RECIPE

- 2. Choose a selection function
  - This is an intermediate value which depends on a part of the key and another **known** data
  - knowing this intermediate data and the known data => part of the key
  - Sel(M_s,K_s)

Target



DPA : THE RECIPE

- 3. The attack
  - $Sel(M_s,K_s)$ is 1-bit long
    - Let's say M_s and K_s are n-bit long
  - The set of captured traces are classed into 2^n depending on the value of Ms. Means are computed
  - => This leads to 2^n traces.

Each trace represent one possible M_s :

$$\{T_0,\ldots, T_{2^n}\}$$

DPA: THE RECIPE

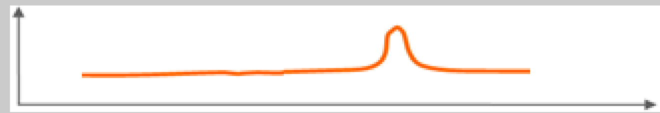
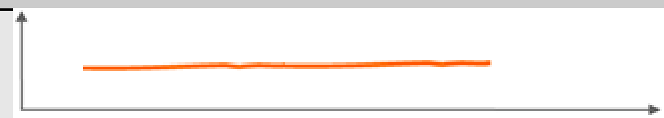
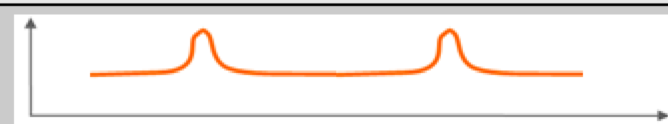
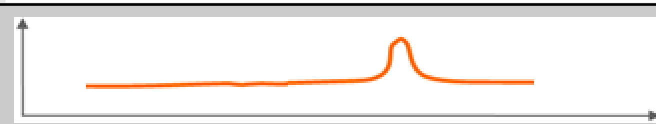
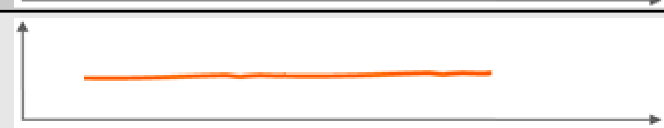
```
for each possible value of Ks: Ks,j
    Hj = nulltrace for hypothesis Ks,j
    for each trace Ti  -- corresponding to Ms,i
        seltmp = Sel(Ms,i,Ks,j)  -- is computed
        If(seltmp == 1)
            Hj += Ti
        else Hj -= Ti
```

- 2^n hypothesis traces are obtained.
  - The one with the correct hypothesis
    - added traces for which the intermediate bit value is 1
    - substracted the traces for which the intermediate bit value is 0
  - The 2^n – 1 hypothesis traces mixed up the traces adding and substracting traces

DPA: ONE EXAMPLE

- Here is a simple encryption algorithm
  - m: 3 bit, k: 3bit, C(m,k) : 3 bits
  - C(m,k) = Sbox(m xor k)
  - Sbox[] = [0x3, 0x7, 0x2, 0x0, 0x6, 0x1, 0x5, 0x4]
- A set of traces are captured
- Means are computed for each possible value of m (=0..7)

DPA: ONE EXAMPLE

Value of m	Means of traces for the corresponding m
0x0 = $(000)_b$	
0x1 = $(001)_b$	
0x2 = $(010)_b$	
0x3 = $(011)_b$	
0x4 = $(100)_b$	
0x5 = $(101)_b$	
0x6 = $(110)_b$	
0x7 = $(111)_b$	

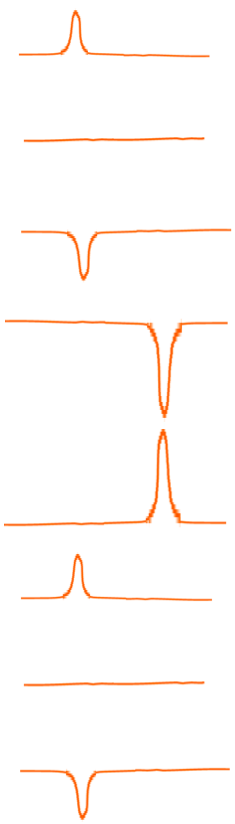
DPA: ONE EXAMPLE

k\m	000	001	010	011	100	101	110	111
000	011	111	010	000	110	001	101	100
001	111	011	000	010	001	110	100	101
010	010	000	011	111	101	100	110	001
011	000	010	111	011	100	101	001	110
100	110	001	101	100	011	111	010	000
101	001	110	100	101	111	011	000	010
110	101	100	110	001	010	000	011	111
111	100	101	001	110	000	010	111	011

$C(m,k)$

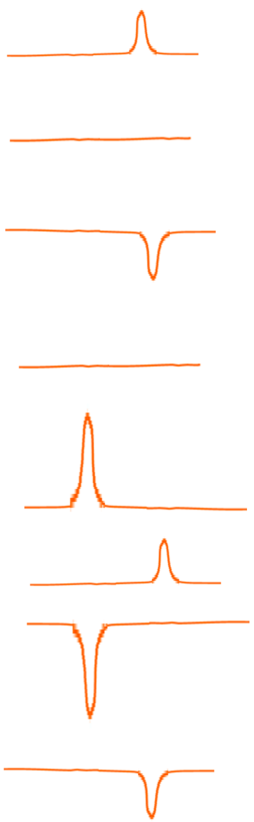
DPA: ONE EXAMPLE BIT 1

k\m	000	001	010	011	100	101	110	111
000	011	111	010	000	110	001	101	100
001	111	011	000	010	001	110	100	101
010	010	000	011	111	101	100	110	001
011	000	010	111	011	100	101	001	110
100	110	001	101	100	011	111	010	000
101	001	110	100	101	111	011	000	010
110	101	100	110	001	010	000	011	111
111	100	101	001	110	000	010	111	011



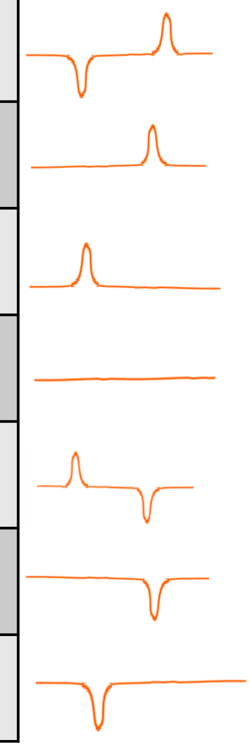
DPA: ONE EXAMPLE BIT 2

k\m	000	001	010	011	100	101	110	111
000	011	111	010	000	110	001	101	100
001	111	011	000	010	001	110	100	101
010	010	000	011	111	101	100	110	001
011	000	010	111	011	100	101	001	110
100	110	001	101	100	011	111	010	000
101	001	110	100	101	111	011	000	010
110	101	100	110	001	010	000	011	111
111	100	101	001	110	000	010	111	011



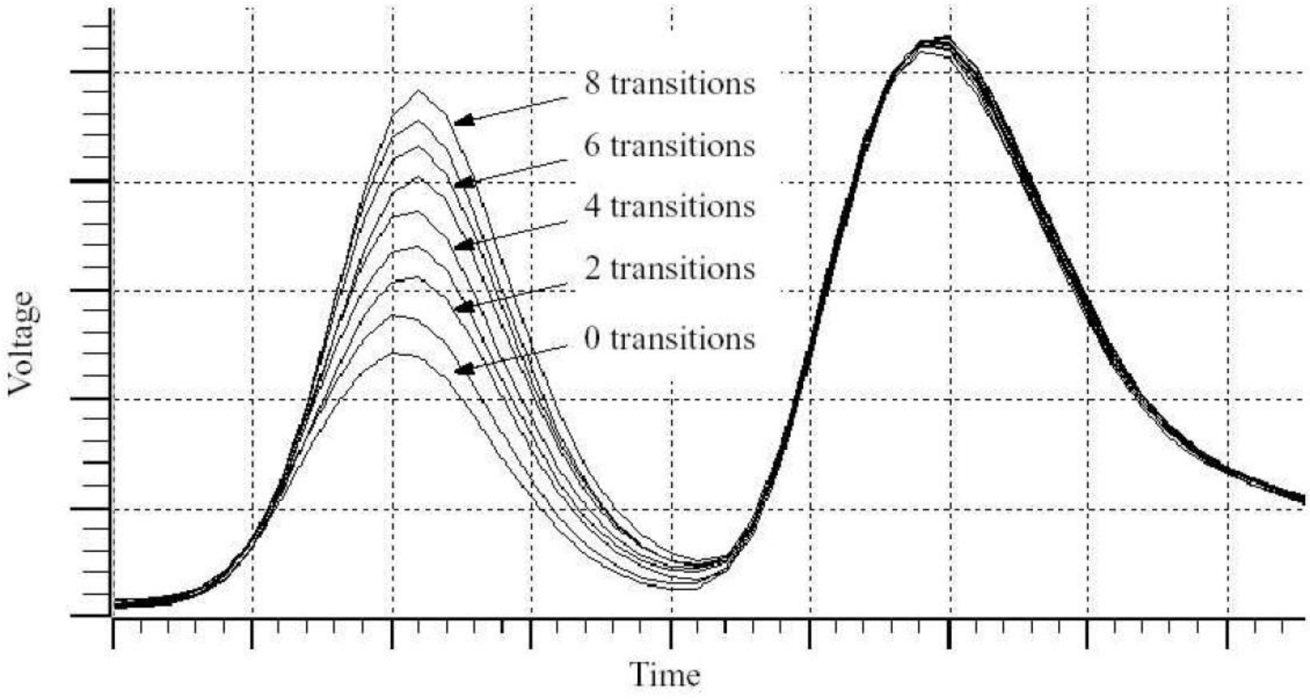
DPA: ONE EXAMPLE BIT 0

k\m	000	001	010	011	100	101	110	111
000	011	111	010	000	110	001	101	100
001	111	011	000	010	001	110	100	101
010	010	000	011	111	101	100	110	001
011	000	010	111	011	100	101	001	110
100	110	001	101	100	011	111	010	000
101	001	110	100	101	111	011	000	010
110	101	100	110	001	010	000	011	111
111	100	101	001	110	000	010	111	011



CPA: PRINCIPLES

- $W = a.H(D) + b$
  - D: handling data
  - H: Hamming weight function
  - a, b: constants



CPA: PRINCIPLES

$$C(t) = \sum_{i=1}^{m} (1 - \alpha_i)\beta_i c_{01}(t) + \alpha_i(1 - \beta_i)c_{10}(t) + Crest(t)$$

- m bit size
- c_{01} power consumption of a bit switching from 0 to 1
- c_{10} power consumption of a bit switching from 1 to 0
- α_i and β_i bit values at instants $t-1$ and t
- $Crest$ power consumption independent of the data

Source:

- R. Bevan, E. Knudsen: *Ways to Enhance DPA.* ICISC 2002

CPA: PRINCIPLES

for i from 1 to N :

- Estimate the power consumption : $M_{W,i}$
- Measure the real power consumption : $W_i(t)$
- Compute the correlation factor between M_W and $W(t)$ for a lot of experiments

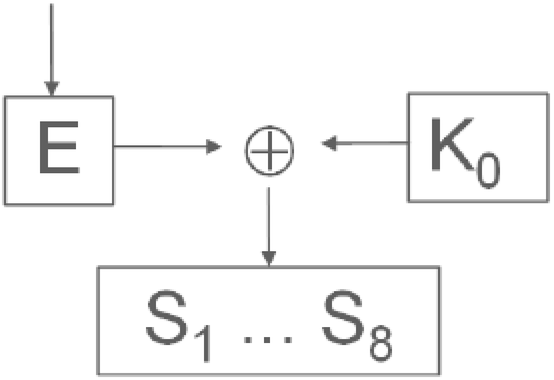
$$\rho(t) = \rho(W(t), M_W) = \frac{Cov(W(t), M_W)}{\sigma_{W(t)}.\sigma_{M_W}}$$

$$\hat{\rho}(t) = \hat{\rho}(W(t), M_W) = \frac{N.\sum(W_{i(t)}.M_{W,i}) - \sum W_{i(t)}.\sum M_{W,i}}{\sqrt{N.\sum W_{i(t)}^2 - (\sum W_{i(t)})^2}.\sqrt{N.\sum M_{W,i}^2 - (\sum M_{W,i})^2}}$$

$\Rightarrow$ Nearest to 1 the factor is, better is the estimation

CPA: PRINCIPLES

$(e_1, \dots, e_8)$ output of E expansion
$(k_1, \dots, k_8)$ subkey K_0
$(s_1, \dots, s_8)$ output of the Sboxes $S_1 \dots S_8$



Simple modelling:

Hamming weight of an output Sbox bit: b_{ij}

For all known e_i and all unknown k_i, we compute:

$$M(e_i, k_i, j) = b_{ij} = S_i(k_i \oplus e_i)_j$$

and

$$\rho(e_i \mapsto W(e_i), e_i \mapsto M(e_i, k_i, j))$$

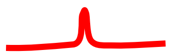
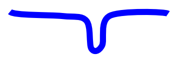
SCA COUNTERMEASURES

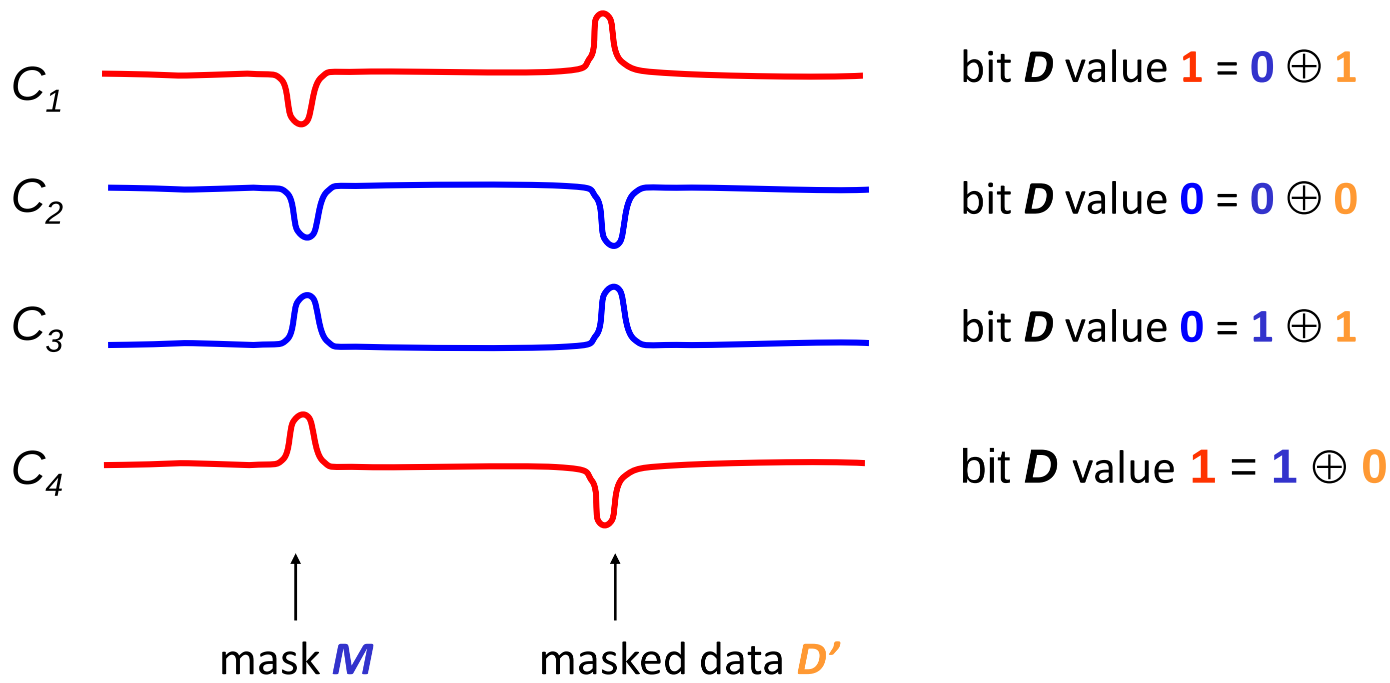
- Software/Hardware
  - Data masking → ⚠ 2^{nd} order DPA !
  - Dummy operations
  - Random delays
  - Dual rail
  - Jittering: Dynamic random clock / Dummy operations
  - Leakage reduction
  - Noise addition

SCA COUNTERMEASURES

- Message blinding
  - random r
    - compute r_2 = r^e mod N
  - compute M * r_2
  - Sign M * r_2 = > S=(M * r_2) ^d mod N = M^d mod N
  - Intermediate value depends on an unknown mask r2
- Exponent blinding
  - Compute d’ = d + k*phi(N)

SCA COUNTERMEASURES (DATA MASKING)

Handle a mask ***M*** and masked data ***D'***= ***M*** ⊕ ***D***
➔ the bit D is not anymore computed by the controller
➔ no more  or  correlated with D (the power consumption is not correlated with D



PROFILED SCA (TEMPLATES) :

- Statistical attack
- Characterization of the noise
- 2 phases
  - Building phase
  - Matching phase
- Advantage comparing to DPA: less traces
- Same countermeasures as for DPA

PROFILED SCA: 1ST PHASE (BUILDING DB)

- Acquire traces: different data & keys
- Choose a model
  - Value of a byte
  - Hamming weight
- Choose Points Of Interest (POI)
- Classify traces depending on possible model values
- Template(class) = (mean(class), cov(class))

PROFILED SCA : 1ST PHASE (BUILDING DB)

- p traces t_i
- Mean $\bar{t} = M = \frac{1}{p}\sum_{j=1}^{p} t_j$
- Covariance $\mathrm{cov}(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})$
- Covariance matrix $CM(u,v) = \mathrm{cov}(N_u, N_v)$
- $N_i = t_i - M$ (noise)

PROFILED SCA : 2ND PHASE (MATCHING)

- Acquire traces: different data & same key
- Choose same POI
- For each key guess k, for each trace j
  - Determine the class i = model(k, j)
  - Compute the probability of matching

  → maximum likelihood

$$p(N_i) = \frac{1}{\sqrt{(2\pi)^n |CM_i|}} \cdot e^{-\frac{1}{2} N_i^T CM_i^{-1} N_i}$$

- → retrieve key bits corresponding to the chosen model

End of lecture

Questions?