Attaques Physiques

PENHARD

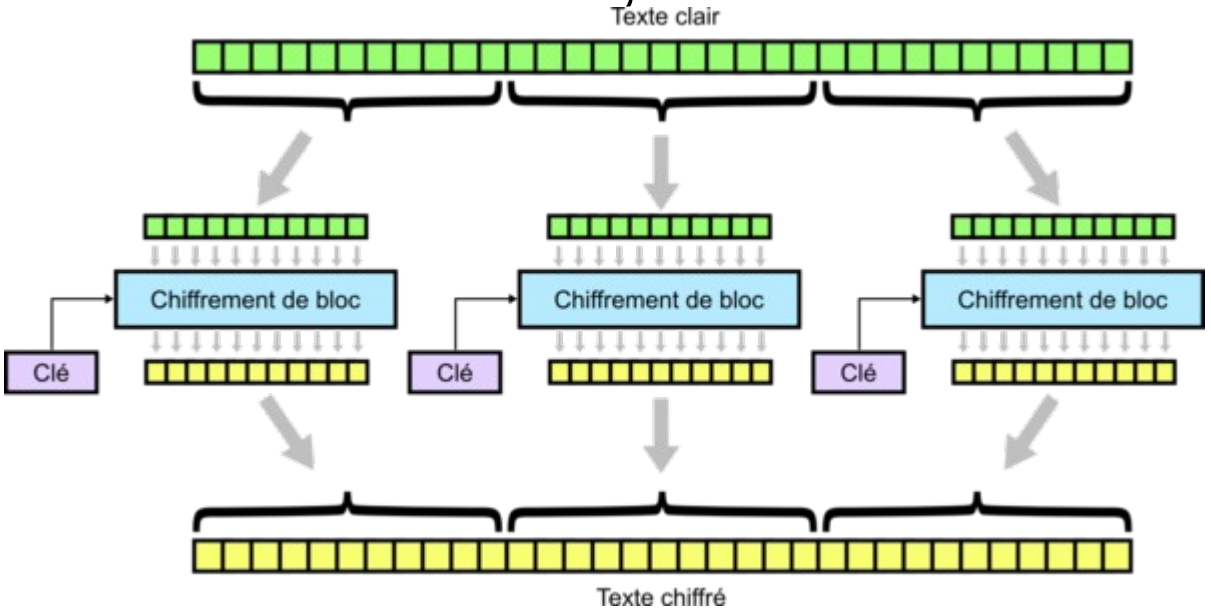
IUT de Béziers, dépt. R&T © 2019-2021
https://www.borelly.net/
Christophe.BORELLY@umontpellier.fr

Généralités

- L’attaque des supports physiques afin de déterminer des clés de chiffrement
  - Attaques non invasives (coût réduit)
    - Canaux cachés (xPA, xEMA, ...)
    - Fautes/glitch
    - Force brute
  - Attaques semi-invasives (coût moyen)
    - Fautes laser, Sondage laser, Émission lumière
  - Attaques invasives (coût élevé)
    - Reverse engineering, Probing, FIB

Cas d'étude

- Chiffrement symétrique par blocs
  - La même clé permet de chiffrer et déchiffrer
- **AES** : blocs de 16 octets (128 bits)
- Mode **ECB** (Electronic CodeBook)
  - Inconvénient : Même bloc, Même chiffré



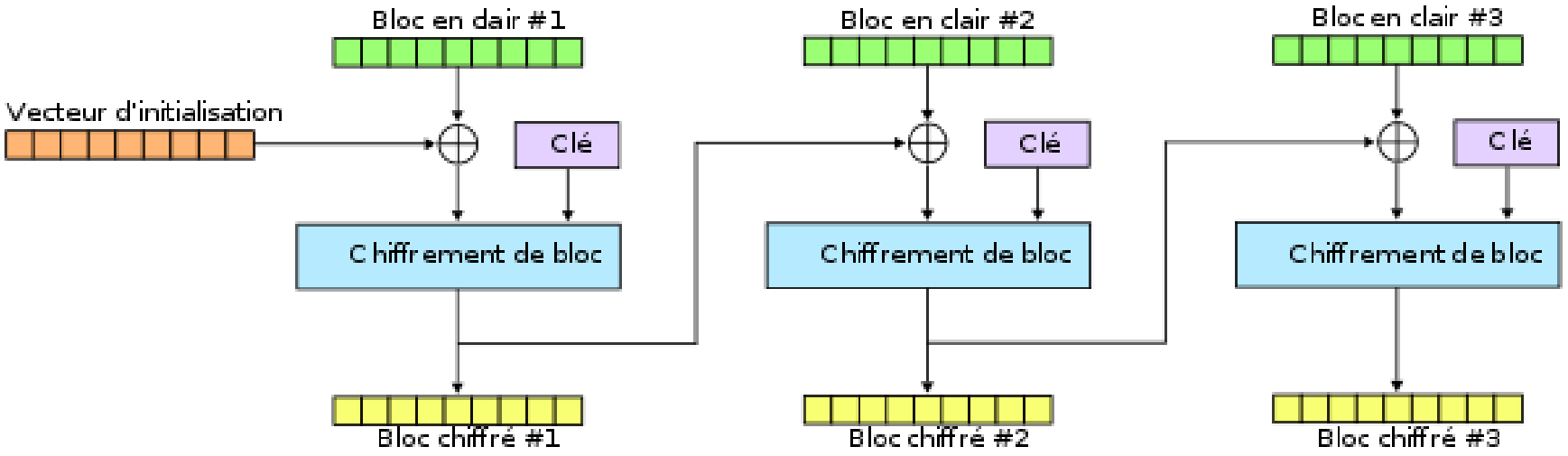
https://fr.wikipedia.org/wiki/Mode_d%27op%C3%A9ration_(cryptographie)

Padding : bourrage

- LEN = taille des données, PAD = 16 - (LEN%16)
- **PKCS#5** (RFC 8018) et **PKCS#7** (RFC 5652)
  - Public Key Cryptography Standards
- Exemples en hexadécimal :
  - Si il manque 4 octets dans le dernier bloc :
    - `xx xx xx xx xx xx xx xx xx xx xx xx 04 04 04 04`
  - Si il manque 7 octets dans le dernier bloc :
    - `xx xx xx xx xx xx xx xx xx 07 07 07 07 07 07 07`
  - Si il manque 15 octets dans le dernier bloc :
    - `xx 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F`

Mode CBC

- Cipher Block Chaining
- Vecteur d'initialisation pour éviter le problème de ECB
- Sensible aux attaques de type « Padding Oracle »



https://fr.wikipedia.org/wiki/Mode_d%27op%C3%A9ration_(cryptographie)

Notation AES

- Bloc de 16 octets sous forme de carré

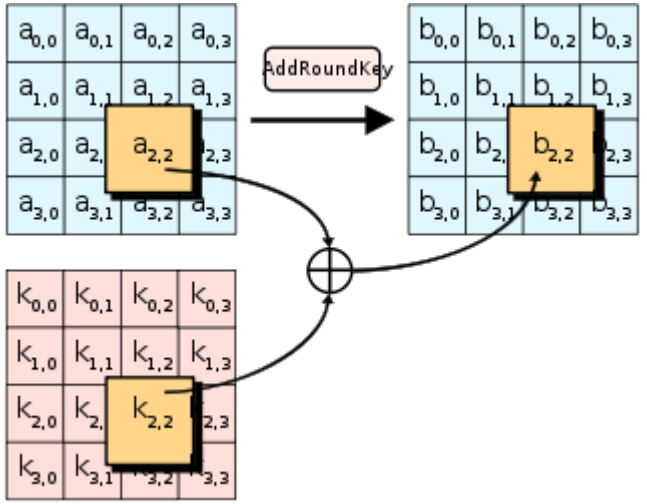
0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

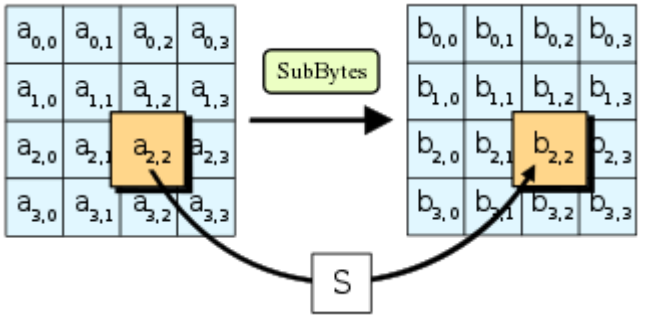
- KEY = Clé AES
- PTI = Plain Text Input
- CTO = Cipher Text Output
- `CTO=`**`AES128-ECB`**`(KEY,PTI)`

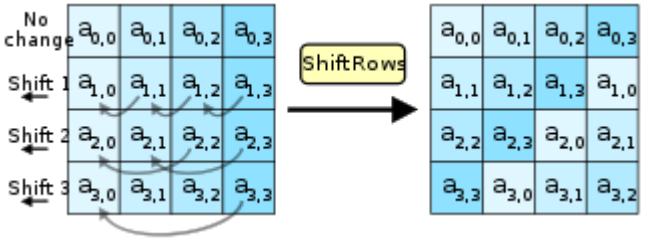
Calcul AES (1)

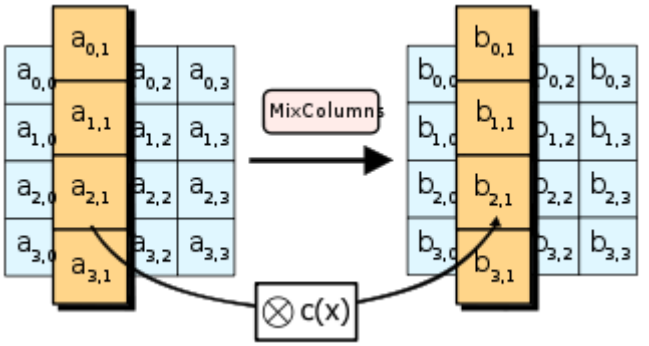
- 4 fonctions de base :
  - AddRoundKey
  - SubBytes (S-Box)
  - ShiftRows
  - MixColumns

Voir NIST-FIPS197



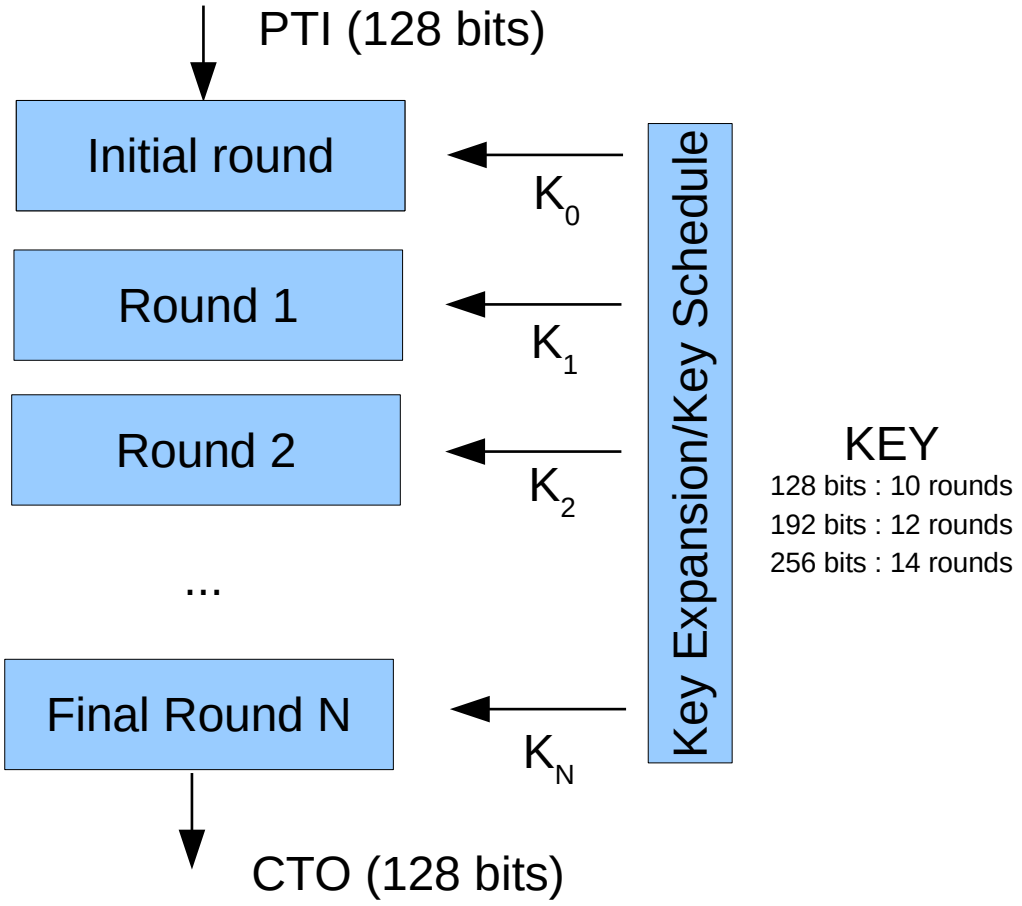






Calcul AES (2)

- Initial round
  - AddRoundKey
- Round
  - SubBytes + ShiftRows + MixColumns + AddRoundKey
- Final Round
  - SubBytes + ShiftRows + AddRoundKey



Key Expansion

- Voir page 24-25/51 NIST FIPS 197
- https://en.wikipedia.org/wiki/AES_key_schedule
- Découpage en mots de 32 bits (4 octets)

- $W_i = Key_i$ pour i<4
- $W_i = W_{i-4} \oplus subWord(RotWord(W_{i-1})) \oplus Rcon(i/4)$ pour i>=4 et i%4==0
- $W_i = W_{i-4} \oplus W_{i-1}$ sinon

Outils linux

```
$ echo -n Hello | xxd -p
48656c6c6f
$ echo 48656c6c6f | xxd -p -r
Hello
$ openssl enc -ciphers|grep aes-128
-aes-128-cbc  -aes-128-cfb -aes-128-cfb1
-aes-128-cfb8 -aes-128-ctr -aes-128-ecb
-aes-128-ofb  -aes-192-cbc -aes-192-cfb
$ man enc
 openssl enc -cipher [-help] [-list] [-ciphers] [-in filename]
   [-out filename] [-pass arg] [-e] [-d] [-a] [-base64] [-A]
   [-k password] [-kfile filename] [-K key] [-iv IV] [-S salt]
   [-salt] [-nosalt] [-z] [-md digest] [-iter count] [-pbkdf2]
   [-p] [-P] [-bufsize number] [-nopad] [-debug] [-none]
   [-rand file...] [-writerand file] [-engine id]
 openssl [cipher] [...]
```

Programmation PHP

```php
<?php
$msg='Hello';
$hex=bin2hex($msg);
$bin=pack('H*',$hex);
printf("%s\n",$hex);//48656c6c6f
printf("%s\n",$bin);//Hello
?>
```

```
openssl_encrypt($data,$algo,$key,$options=0,$iv='')
  Algo: Voir openssl_get_cipher_methods()
  Options: OPENSSL_RAW_DATA (output raw data)
           OPENSSL_ZERO_PADDING (no padding)
```

Programmation Python3

```
msg='Hello'
msgBytes=bytes(msg,'utf8')# msg.encode()
hex=msgBytes.hex()
bin=str(bytes.fromhex(hex),'utf8')
# bin=msgBytes.decode()
print(f'{hex}')#48656c6c6f
print(f'{bin}')#Hello
```

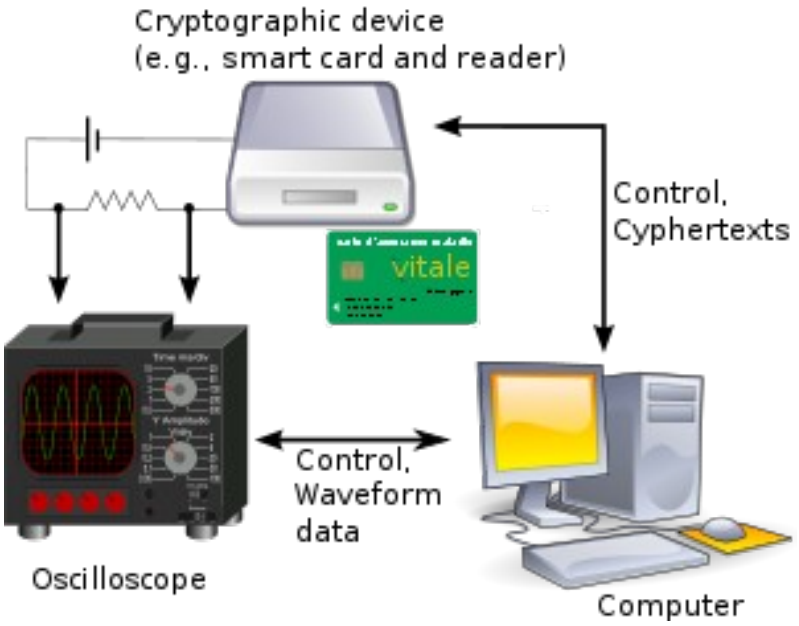
- Module pycryptodome :

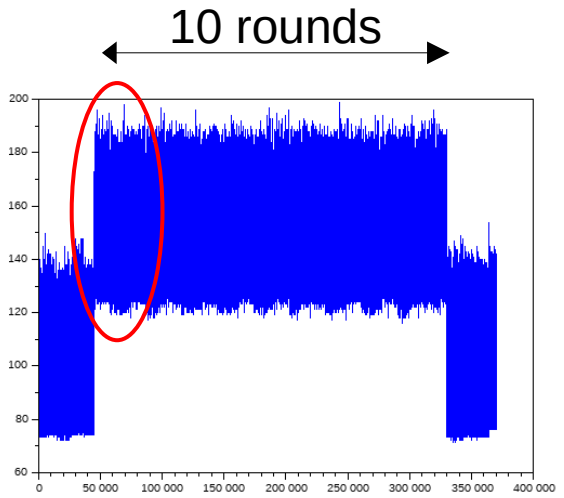
https://pycryptodome.readthedocs.io/en/latest/

=> src/cipher/aes.html

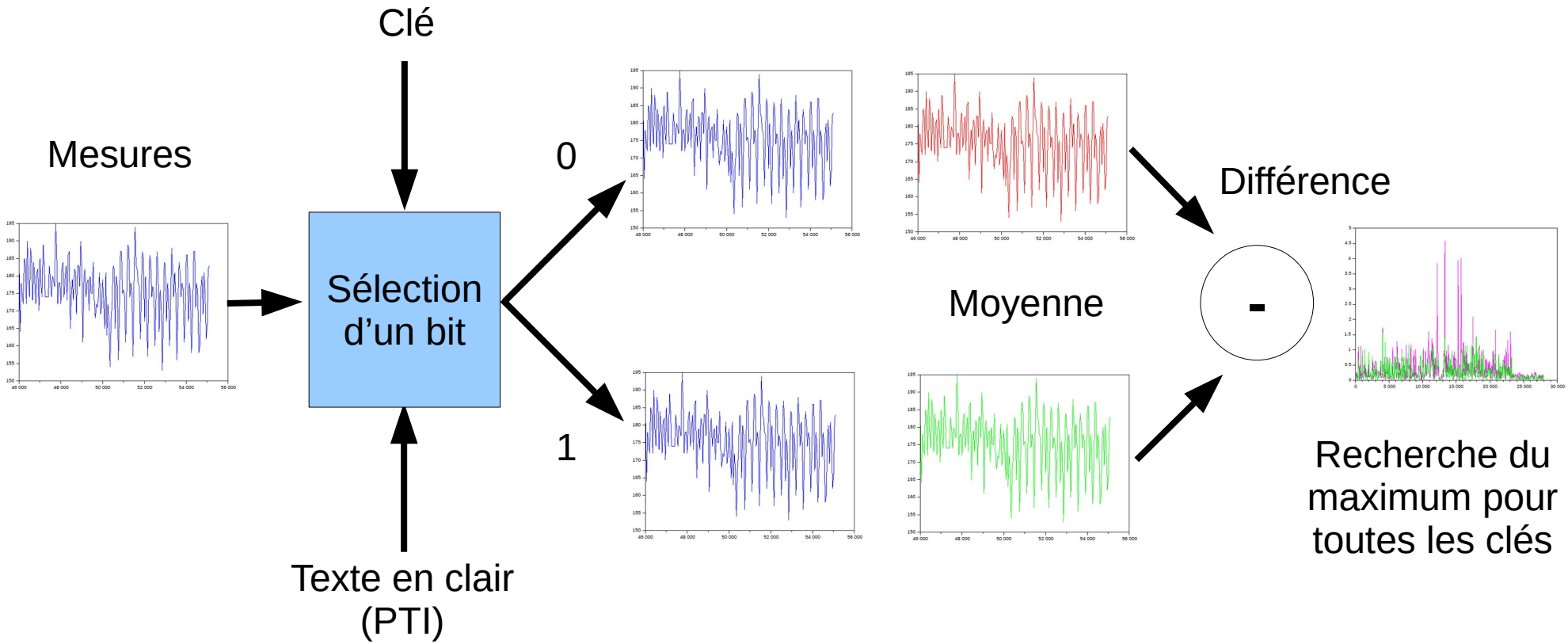
SPA et DPA

- KOCHER en 1999 (Attaque de DES)
- Simple/Differential Power Analysis
  - Pour **AES** : Sortie SubBytes du 1er round





DPA

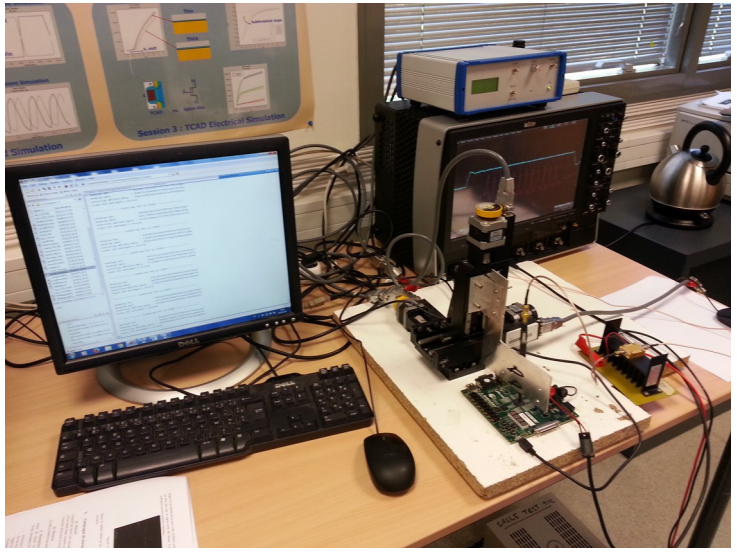

Clé
Mesures
Sélection d'un bit
0
1
Texte en clair (PTI)
Moyenne
Différence
-
Recherche du maximum pour toutes les clés

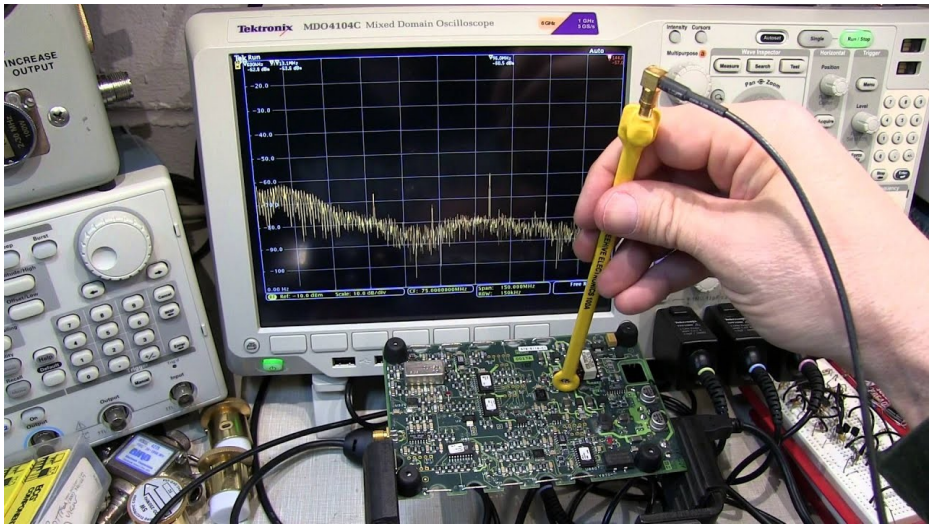
CPA

- Correlation Power Analysis (BRIER en 2004)
- Calcul de la distance de **Hamming**
  - Nombre de bits à 1
  - Estime le nombre de transitions
- Calcul de la matrice de corrélation (covariance)
- Beaucoup plus rapide !

CEMA

- Mesures Electro-Magnétiques
- Attaque juste avant le dernier round

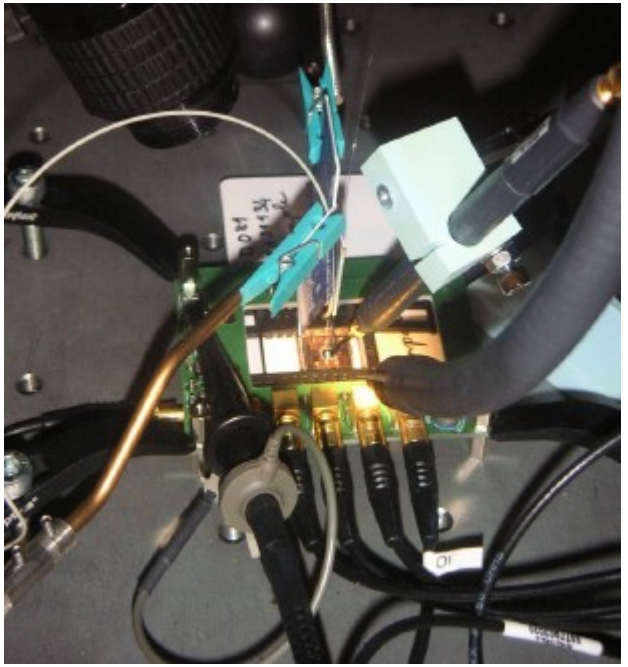




- On retrouve K10
- Il ne reste plus qu’à remonter jusqu’à la clé AES initiale...

DFA

- Differential Fault Analisys/Attack



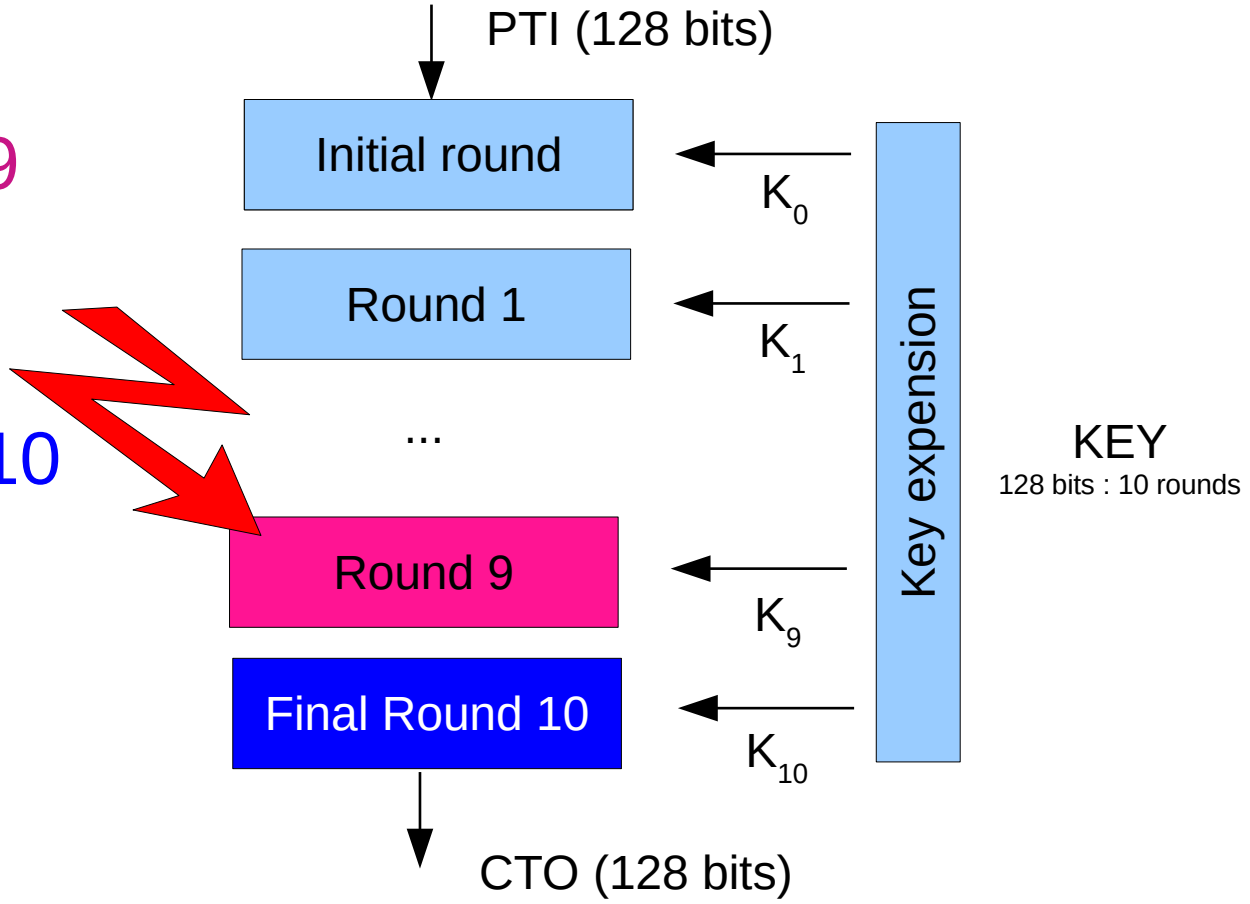
Réf. Introduction aux Attaques par Fautes - Jessy CLÉDIÈRE (CESTI-Léti)

Types d’attaques DFA

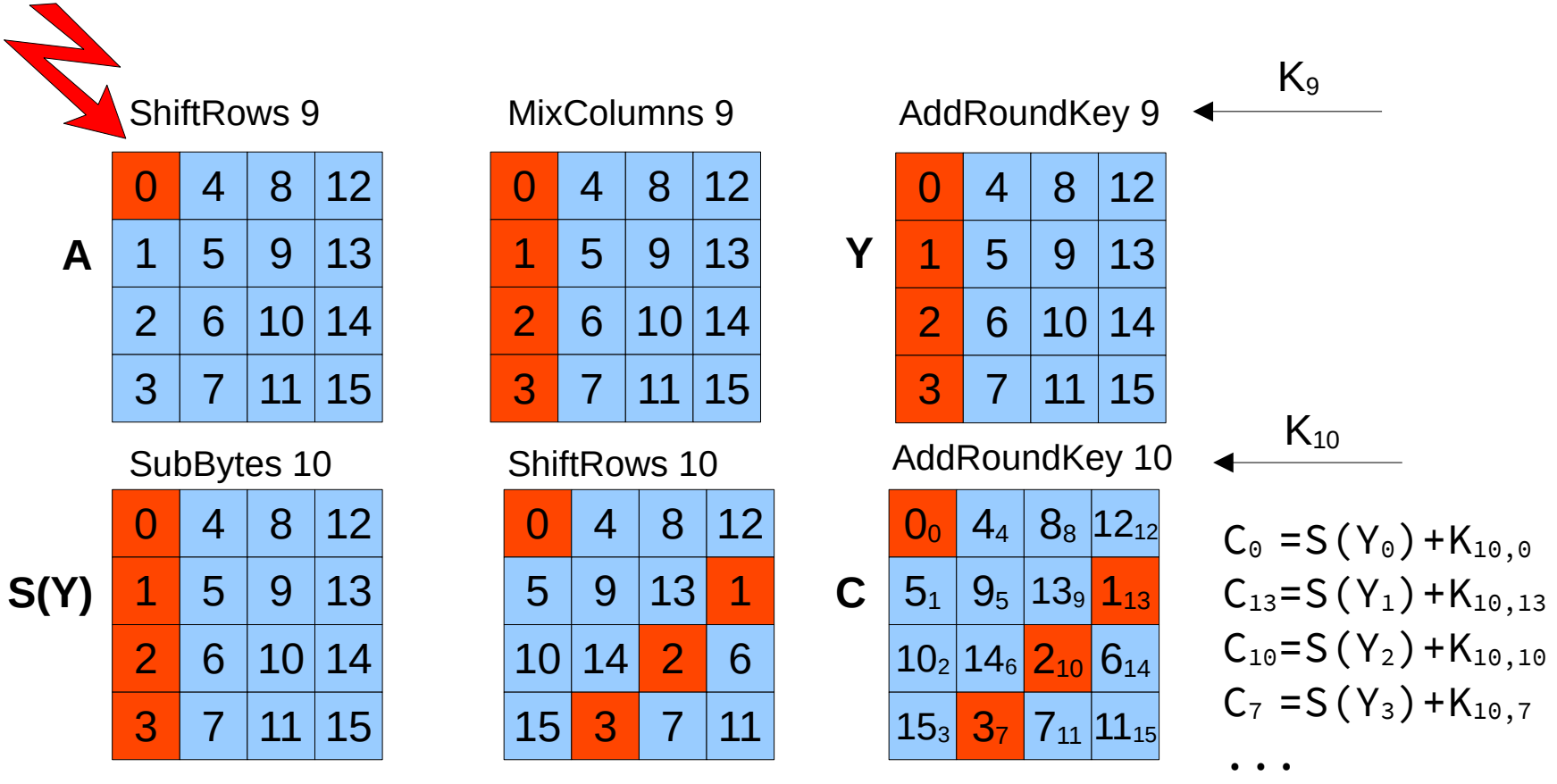
- Modification sur l’état interne de AES à un round donné (e.g. M9 ou M8)
- Modification sur le « key schedule » de AES à un round donné (ex. K9 ou K8)
  - 1 seul bit en erreur
  - Plusieurs bits
  - Plusieurs octets

DFA-1 bit Round 9-AES128 (1)

- ...
- MixColumns
- AddRoundKey K9
- SubBytes
- ShiftRows
- AddRoundKey K10



DFA-1 bit Round 9-AES128 (2)



DFA-1 bit Round 9-AES128 (3)

- Une erreur sur 1 bit juste avant le MixColumns génère 4 erreurs dans le chiffré final.
- En comparant le chiffré normal C et celui en erreur C^*, on obtient 4 relations avec 4 octets de Y (ou SY) et Z.
- Mais il y a plusieurs solutions, et il faut répéter l'opération 1 à 3 fois en pratique pour obtenir les 4 octets de SY puis ceux de K10.
- Avec de 8 à 12 erreurs, on peut donc retrouver toute la sous clé K10, puis remonter à la clé AES comme avec CEMA...

DFA-1 bit Round 9-AES128 (4)

- Si A est l'état avant le MixColumns du round 9
- Si Y est l'état après AddRoundKey9.
- $Z=A0+A^*0$ (l'erreur avec 1 seul bit à 1)

$$C0 + C^*0 = S(Y0)+S(2Z+Y0)$$
$$C13+C^*13=S(Y1)+S(Z+Y1)$$
$$C10+C^*10=S(Y2)+S(Z+Y2)$$
$$C7 + C^*7 = S(Y3)+S(3Z+Y3)$$
$$...$$

Références

- Formation CNFM au LIRMM par Florent BRUGUIER en 2017
- https://www.paulkocher.com/doc/DifferentialPowerAnalysis.pdf
- https://blog.quarkslab.com/differential-fault-analysis-on-white-box-aes-implementations.html (attention, il y a une petite erreur !)
- https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
- NIST – FIPS 197