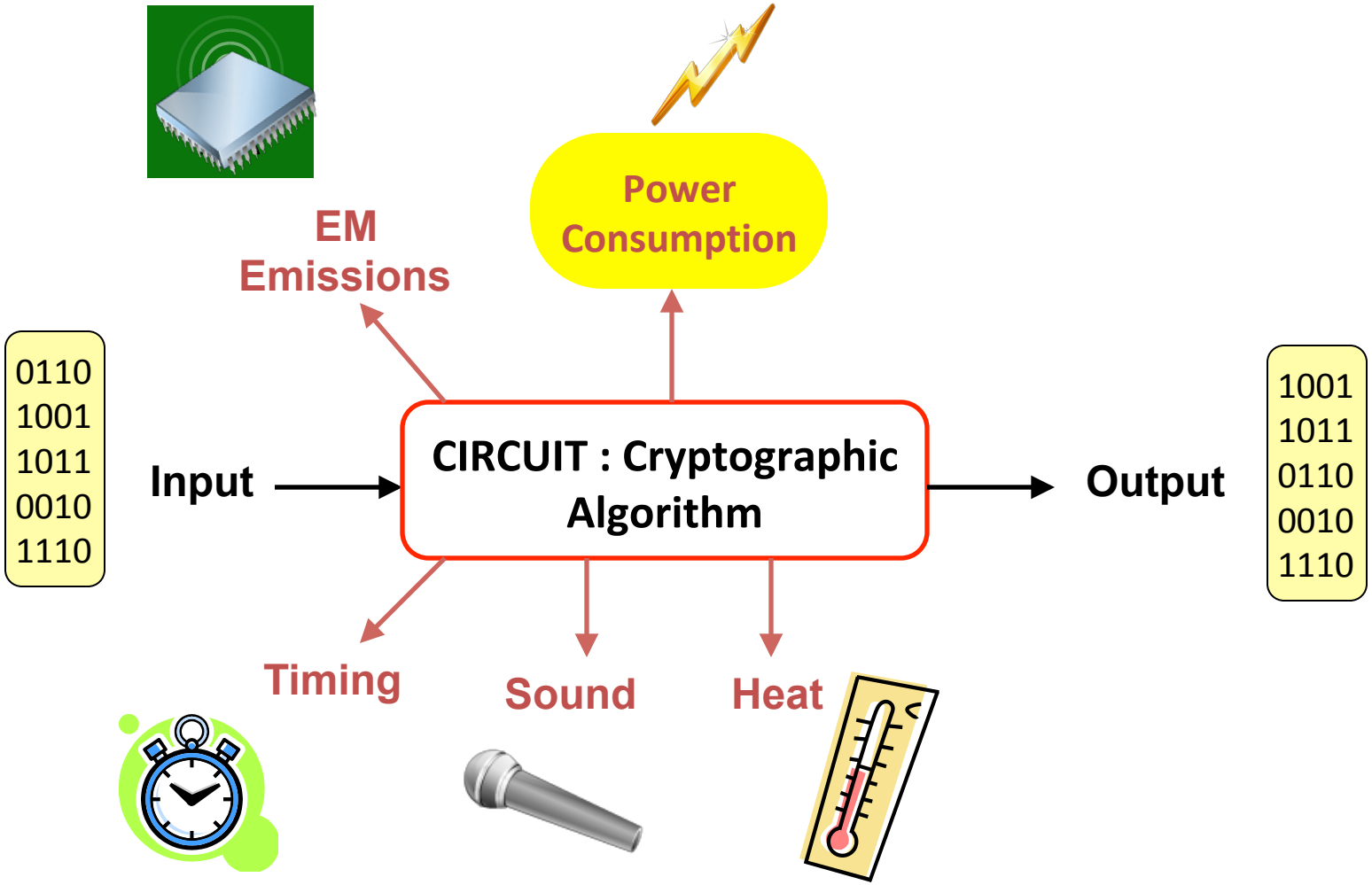
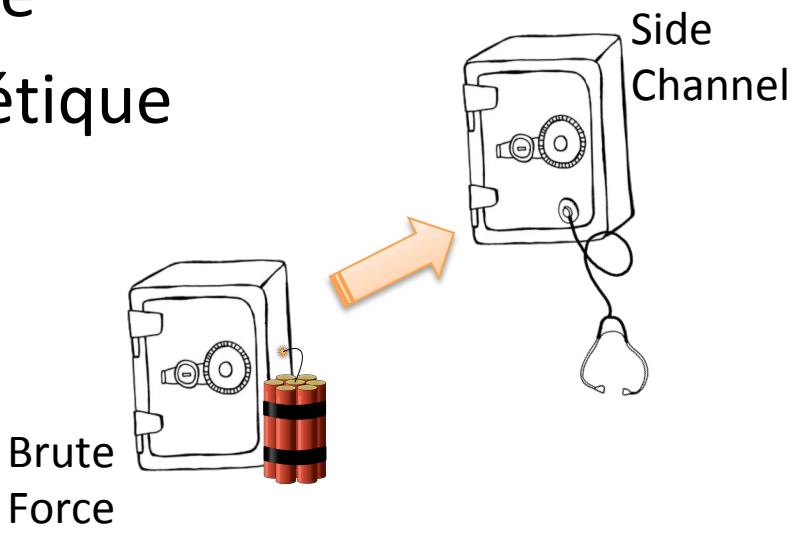
ATTAQUES PAR CANAUX CACHÉS
SIDE-CHANNEL ATTACKS

Side-Channel Attacks



Side-Channel Attacks

- Basées sur l'information récupérée sur l'implantation physique du crypto-système
  - Information temporelle
  - Consommation électrique
  - Emanation électromagnétique
  - Son
  - Lumière
  - …



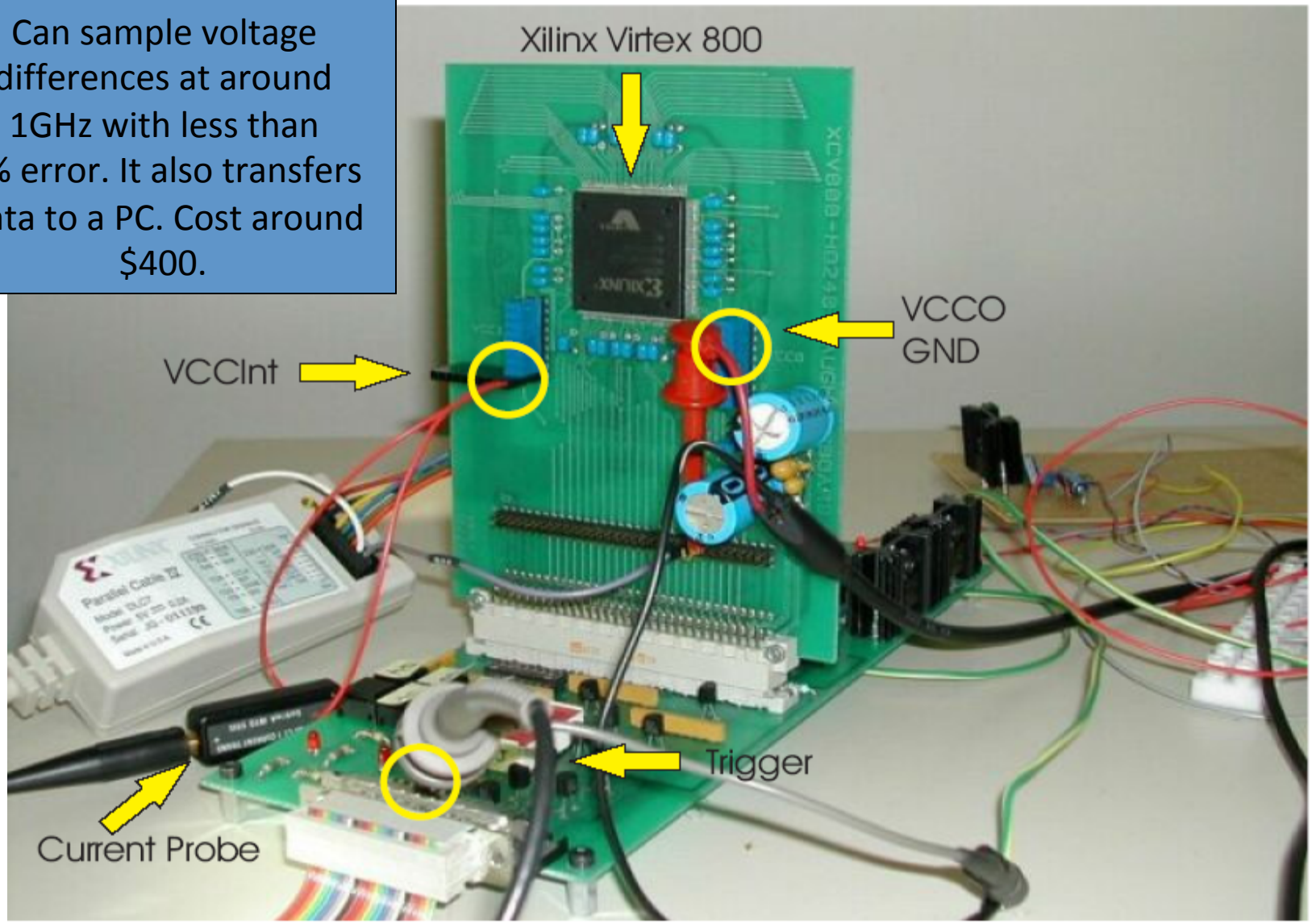
Exemple 1 : Attaques par analyse de courant

- La consommation du circuit est liée aux données.

- SPA : Simple Power Analysis
- DPA : Differential Power Analysis

Lab Set Up pour Analyse de Courant

Can sample voltage differences at around 1GHz with less than 1% error. It also transfers Data to a PC. Cost around $400.

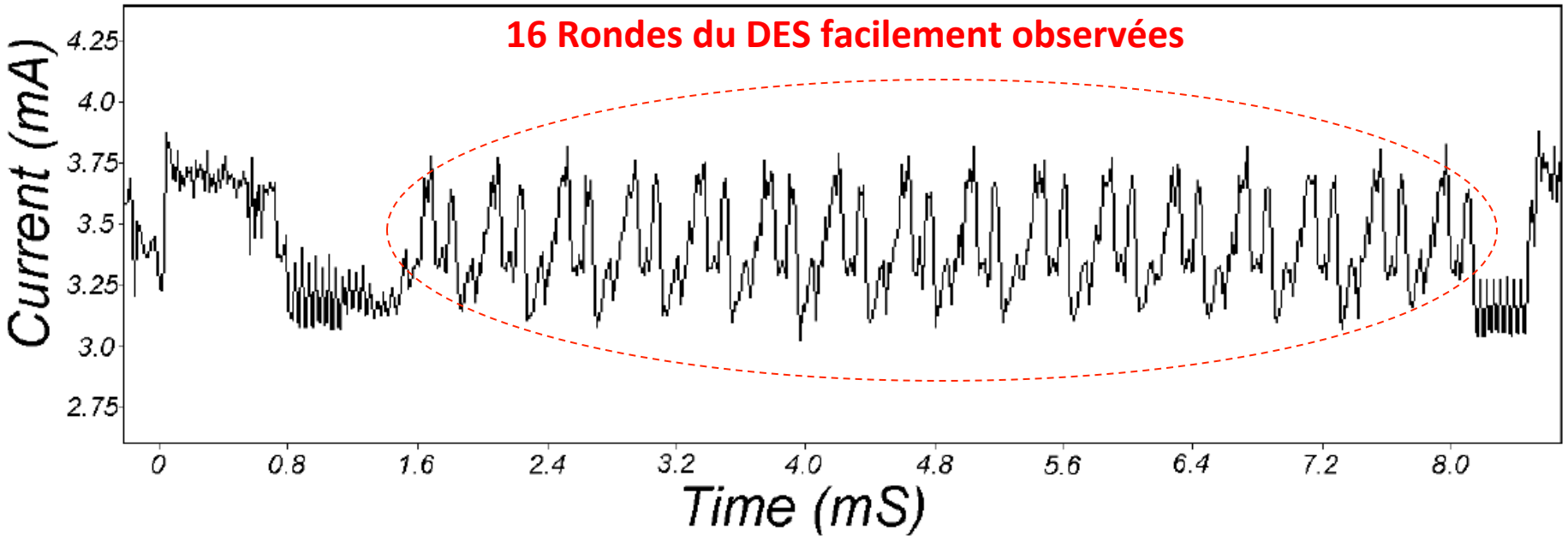


Courtesy: Side-Channel Analysis Lab, *Graz University of Technology*

Simple Power Analysis (SPA)

- Interprétation directe de la consommation du circuit
- Vise les opérations effectuées et la clef!
- Trace: courbe temporelle du courant
- Une opération à 1ms échantillonnée à 5MHz donne une trace avec 5000 points

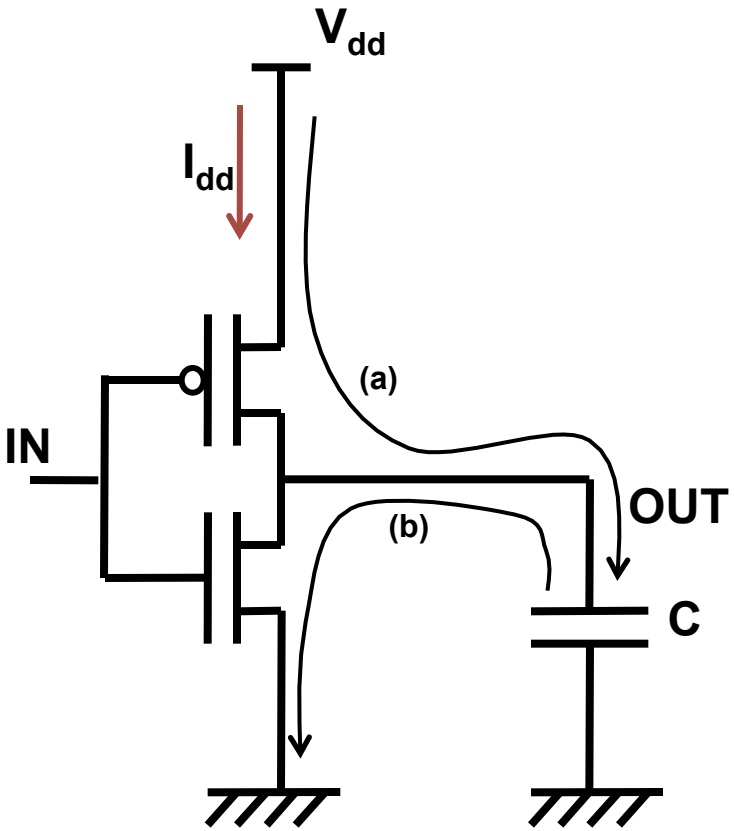
Traces de courant du DES

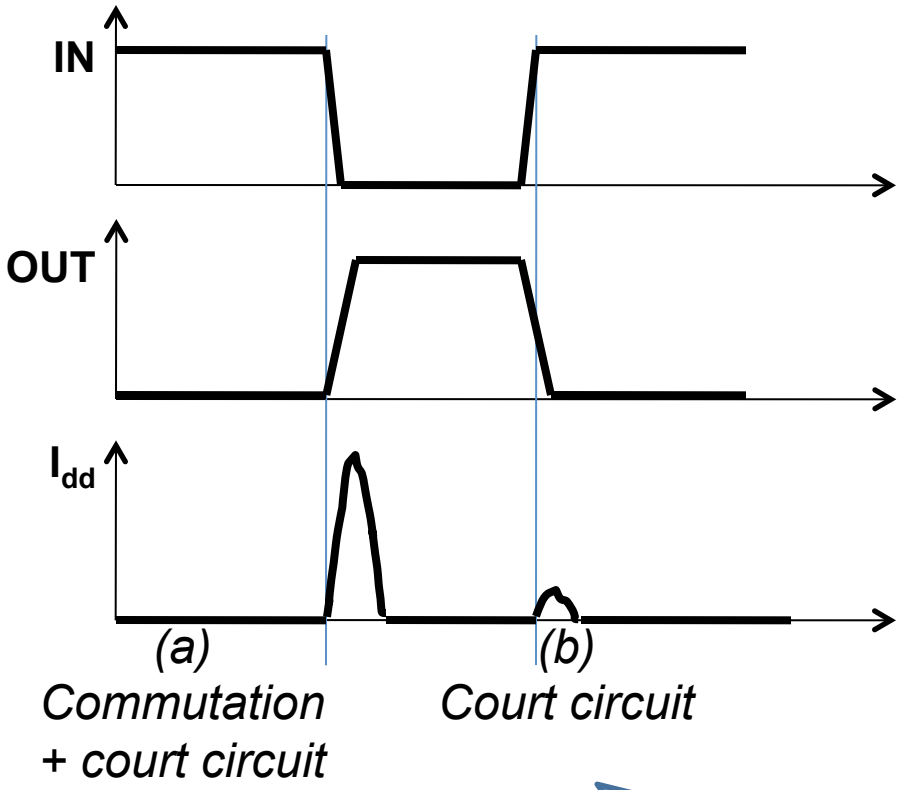


- 1er résultat : on identifie quand ont lieu les opérations crypto

- 2ème résultat : on arrive à différentier les différentes tâches

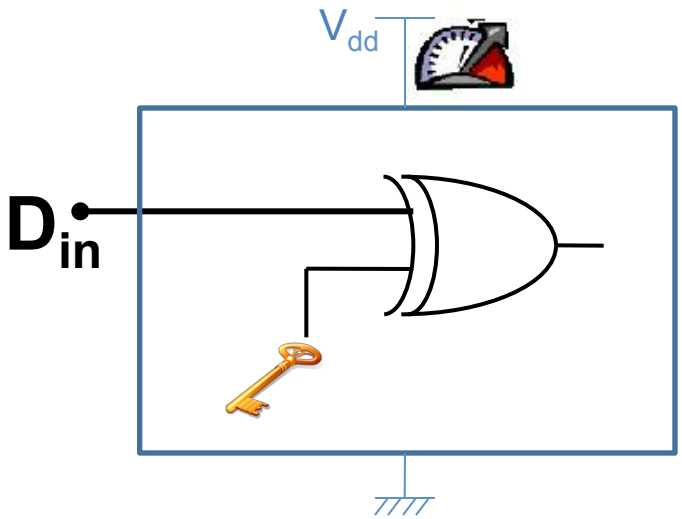
Consommation en CMOS

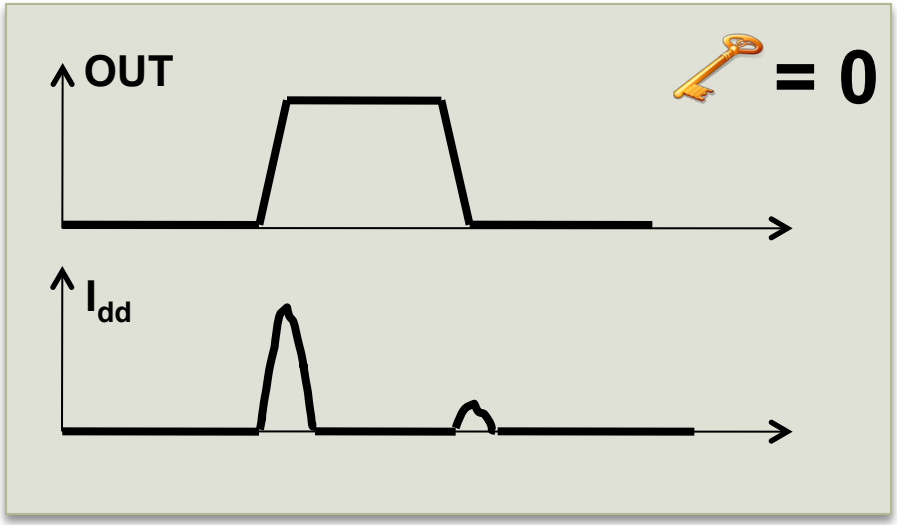


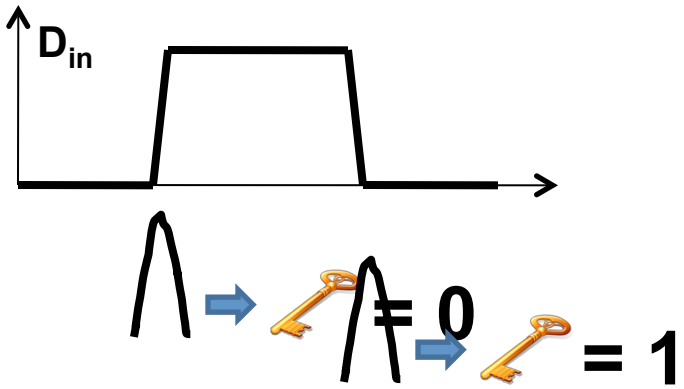


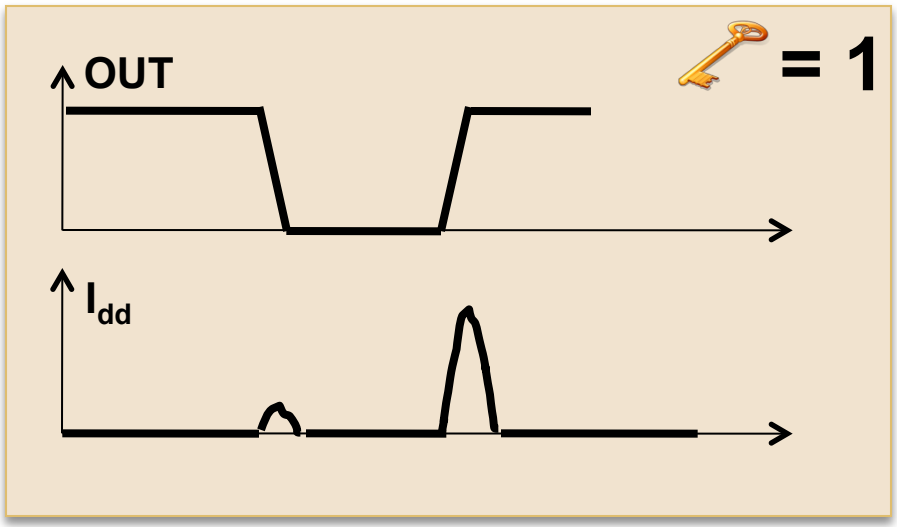
Courant de Vdd à GND quand le transistor N et le transistor P commutent en même temps lors de la transition (courant de court circuit)

SPA sur porte XOR









SPA sur RSA

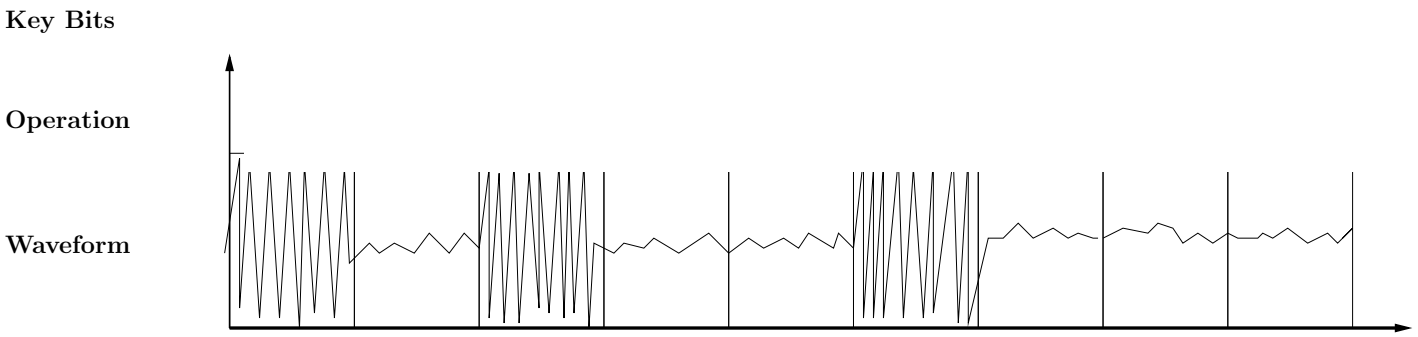
```
Input:    X, N, K=(k_{j-1}, …, k_1, k_0)_2
Output:   Z = X^K mod N

1:    Z = 1;
2:    for i=j-1 downto 0 {
3:        Z = Z * Z mod N //Square
4:        if (k_i==1) Z = Z * X mod N //Multiply
5:        }
6:    return(Z);
```

Ex : Calculer $X^{13} = X^{1101}$

1: Z=1
// k_3 =1
3: Z=1
4: Z=X
// k_2=1
3: Z= X^2
4: Z = X^3
// k_1=0
3: Z=X^6
// k_0=1
3: Z=X^{12}
4: Z=X^{13}

SPA sur RSA

```
Input:    X, N, K=(k_{j-1}, …, k_1, k_0)_2
Output:   Z = X^K mod N

1:    Z = 1;
2:    for i=j-1 downto 0 {
3:        Z = Z * Z mod N //Square
4:        if (k_i==1) Z = Z * X mod N //Multiply
5:        }
6:    return(Z);
```



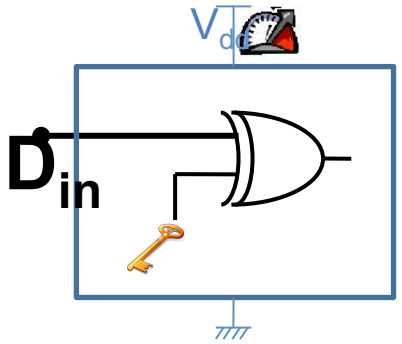
Contre-mesure sur RSA

- Faire en sorte que la consommation soit constante. Ex : introduire des opérations factices.

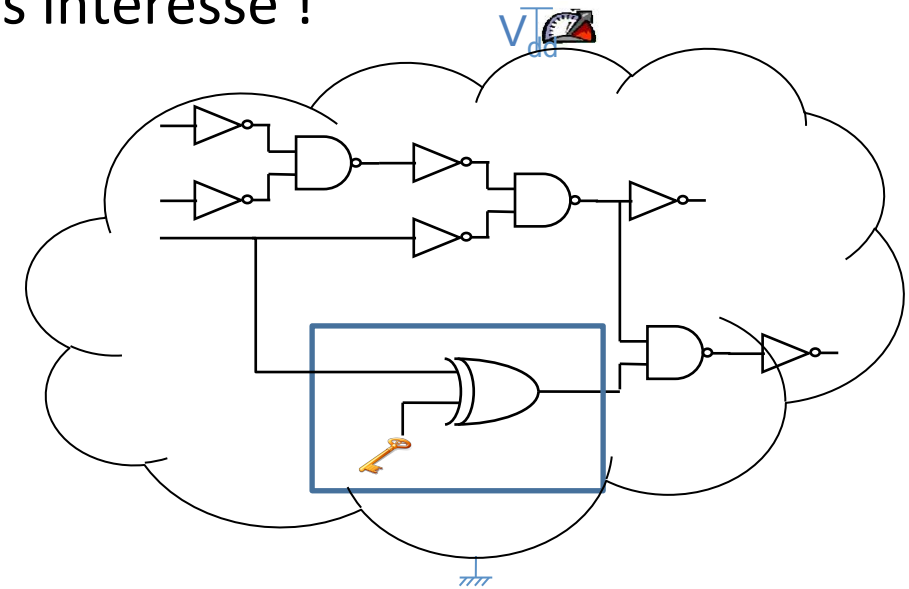
```
Input:    X, N, K=(k_{j-1}, …, k_1, k_0)_2
Output:   Z = X^K mod N

1:    Z = 1;
2:    for i=j-1 downto 0 {
3:        Z = Z * Z mod N //Square
4:        if (k_i==1) Z = Z * X mod N //Multiply
5:        else  U = Z * X mod N
6 :       }
6:    return(Z);
```

Differential power analysis (DPA)

- Idée 1 : on vise un bit lié à la clef et en mesurant la conso. on en déduit la clef.



- Pb : on ne peut mesurer que la conso. du circuit total pas de la porte qui nous intéresse !



Differential power analysis (DPA)

- Idée 2 :
  - On applique beaucoup de stimuli différents,
  - On calcule des moyennes
  - La conso liée à la partie non intéressante va être constante
- Utilisation de méthodes statistiques pour trouver de petites variations qui peuvent être masqués par le bruit ou des erreurs de mesures

DPA principe

Consommation de puissance différente suivant les états (0 ou 1), ou transitions (0->1 et 1->0)

1/ Phase d'acquisition de courbes (message clair 'aléatoire')

2/ Procédure

Hypothèse sur une clef

Pour chaque hypothèse de clef

Diviser les courbes en 2 groupes PA et PB pour un bit choisi suivant les valeurs observées

PA : le bit a consommé

PB : le bit n'a pas consommé

Calculer la moyenne des courbes de chaque groupe

Hypothèse incorrecte : moyennes égales

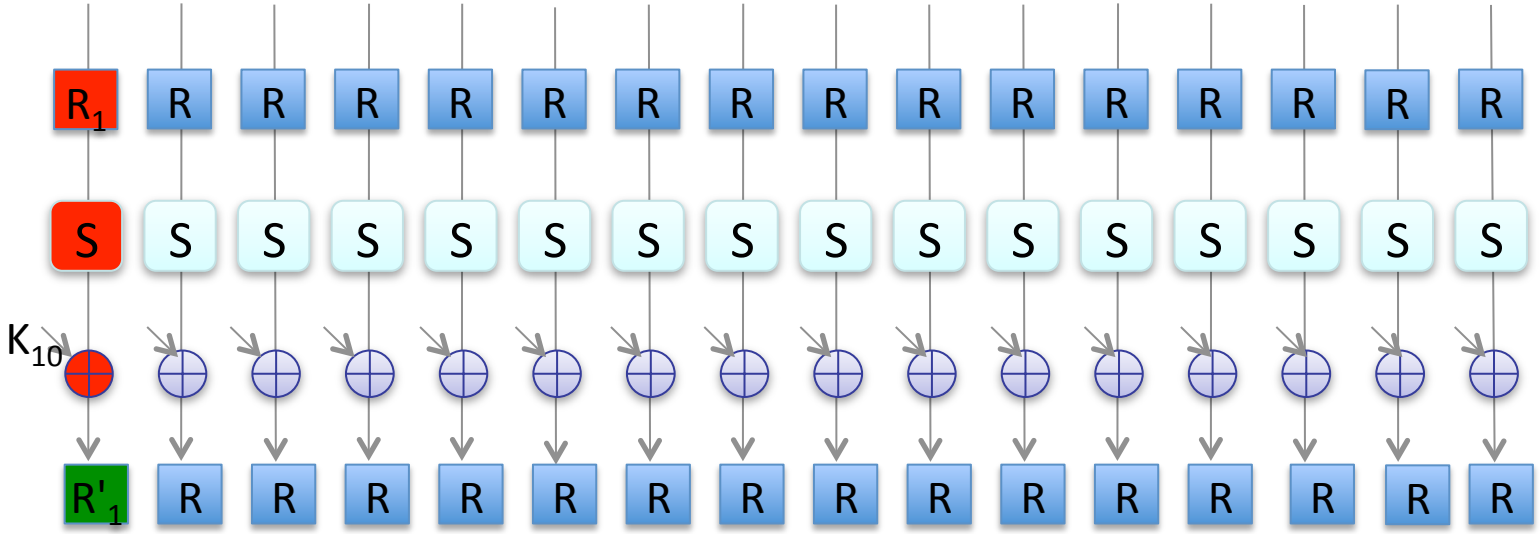
Hypothèse correcte → différence non négligeable

DPA principe

- On applique des données connues et on mémorise la courbe de consommation (globale)
- On vise un bit particulier (par exemple un bit de sortie de l'AES)
- On corrèle la valeur du bit au fait qu'il y a consommation sur ce bit ou pas selon une hypothèse sur la clef
- On extrait par des méthodes statistiques la consommation liée à ce bit de la conso. globale.
  - statiquement on "enlève" la conso liée aux autres bits
- On vérifie si il y a une corrélation.
- Si la corrélation est bonne, hypothèse sur la clef OK
- Sinon, hypothèse KO

DPA

- Hypothèse sur la clef ?
  - AES : clef 128 bits => $2^{128} \cong 10^{31}$hypothèses !!!!
- On ne s'intéresse qu'à une partie de la clef
- Cas de l'AES :
  - dernière ronde (SPA)
  - 8 bits de clef : 256 hypothèses



DPA : Procédure pour l'AES

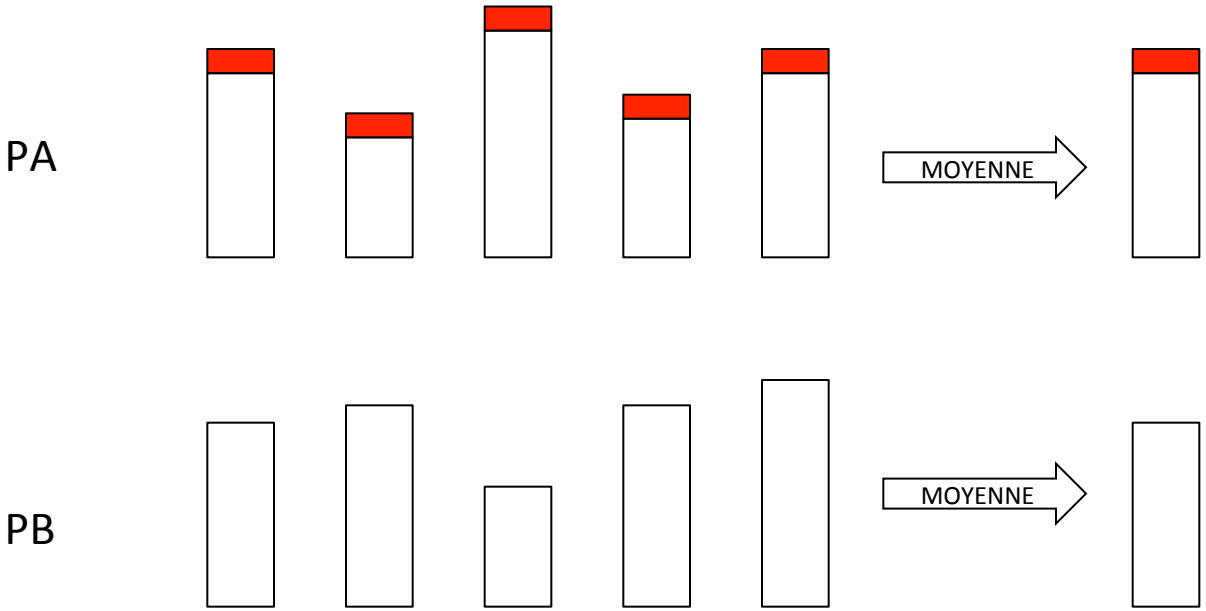
1. Faire la mesure de conso de 1000 opérations AES,
2. Hypothèse sur une clef associée à une S-box à la dernière ronde
    1. Calculer le premier bit d'entrée de la S-box pour chaque chiffré en se basant sur l'hypothèse de clef
    2. Diviser l'ensemble des courbes en 2 groupes PA et PB (entrée 0 et entrée 1)

    PA = contient les courbes où le bit "consomme de la puissance"

    PB = contient les courbes où le bit "ne consomme pas de la puissance"

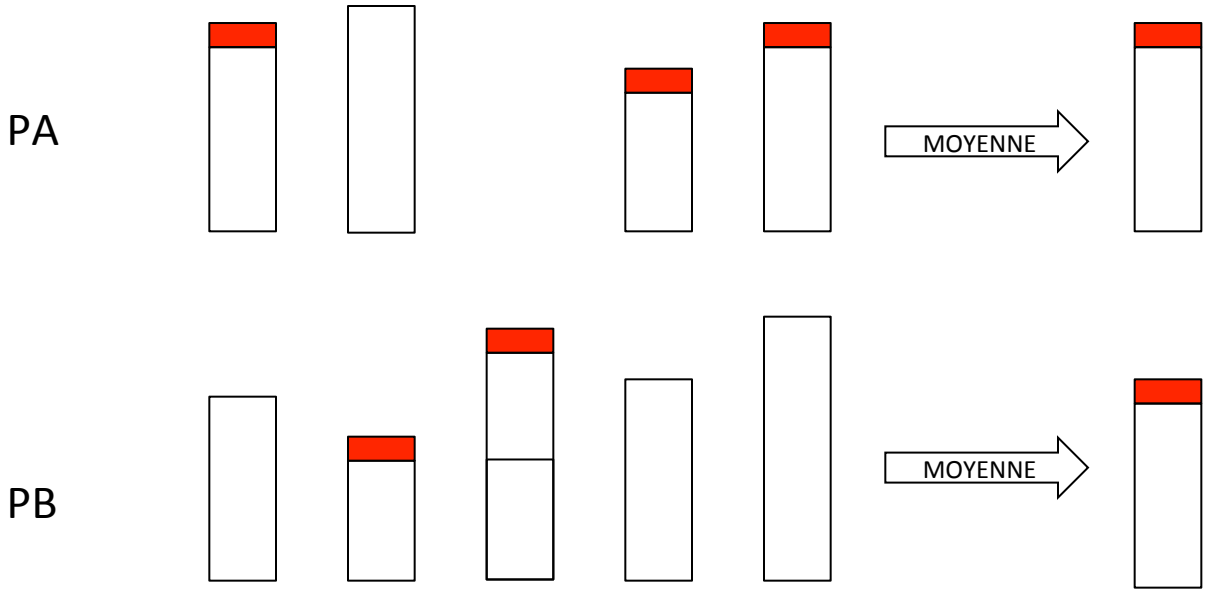
3. Calculer la courbe moyenne de chaque groupe : PA_m et PB_m
4. Calculer la différence des 2 courbes : $|Pa_m - PB_m|$
5. Si l'hypothèse est correcte → pics sur la courbe de différences
6. Répéter 2-5 pour les autres bits et autres S-boxes

Pour le bonne hypothèse de clef : PA contient toujours une composante de la puissance qui n'est pas dans PB.

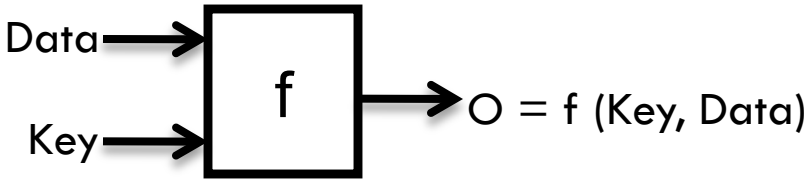


→ Conso(PA) > Conso(PB)

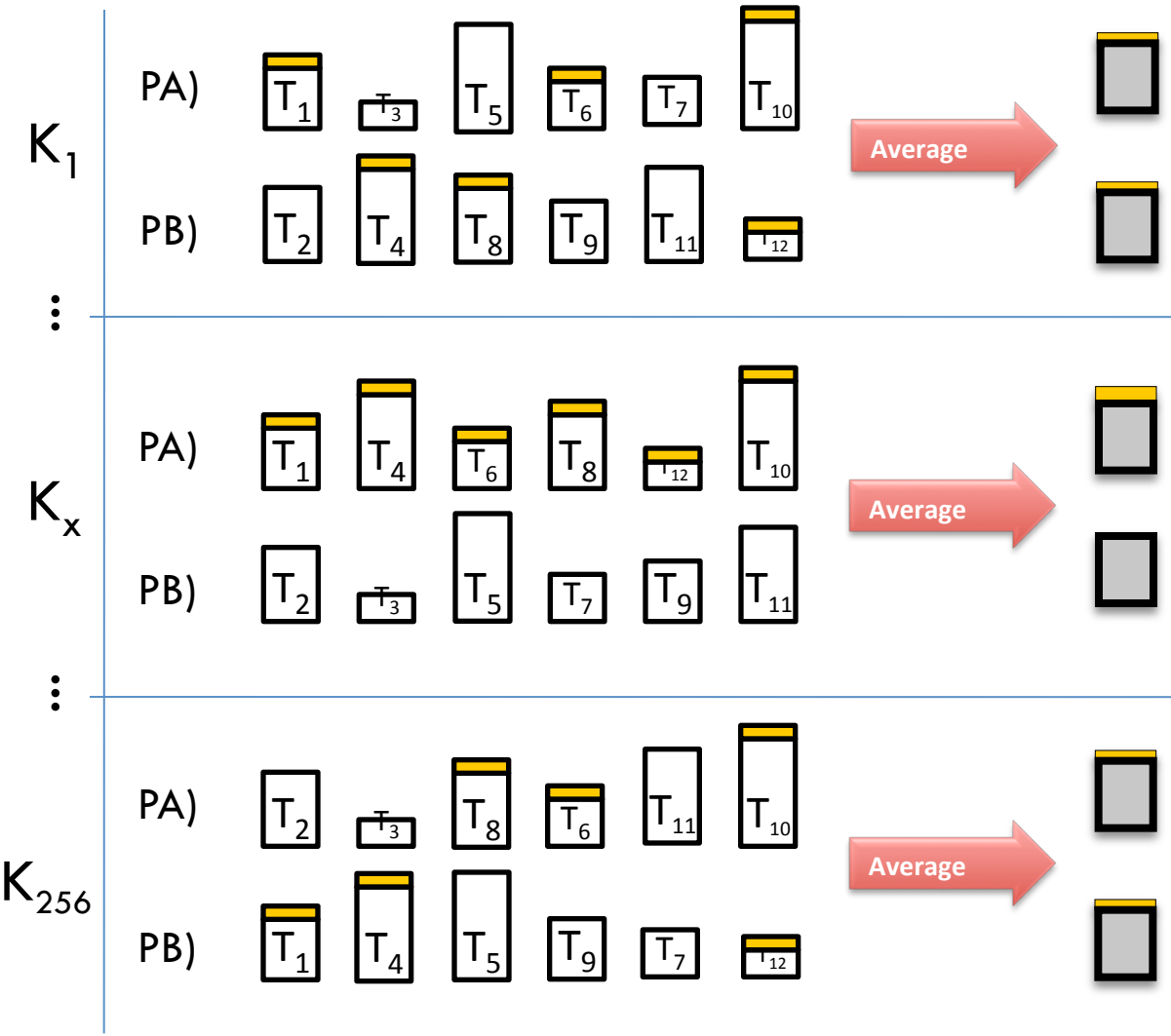
Pour une mauvaise hypothèse de clef : la consommation liée à ce bit se répartit entre PA et PB



→ Conso(PA) ≈ Conso(PB)



Inputs: T_1, T_2, …, T_{12}
Key: K_x (8 bits)



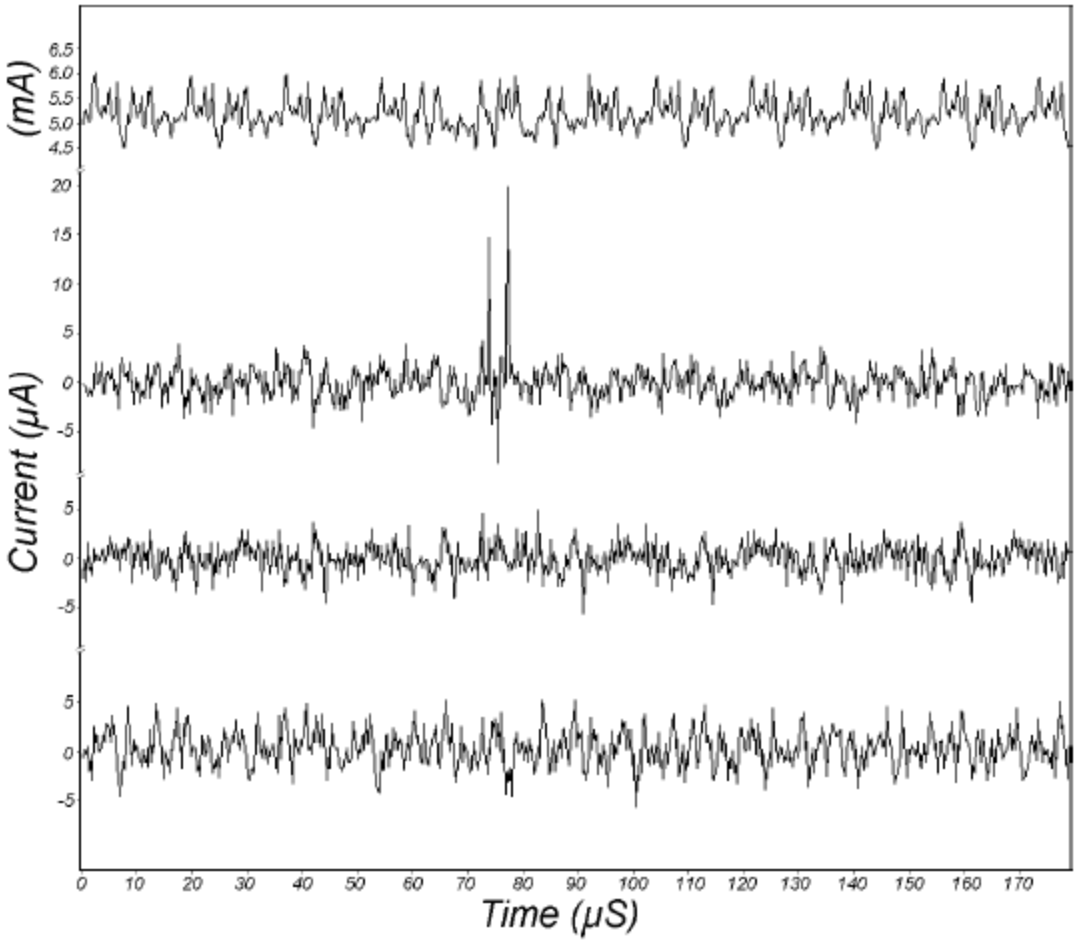
DPA AES : Exemple de résultat

Consommation de puissance moyenne

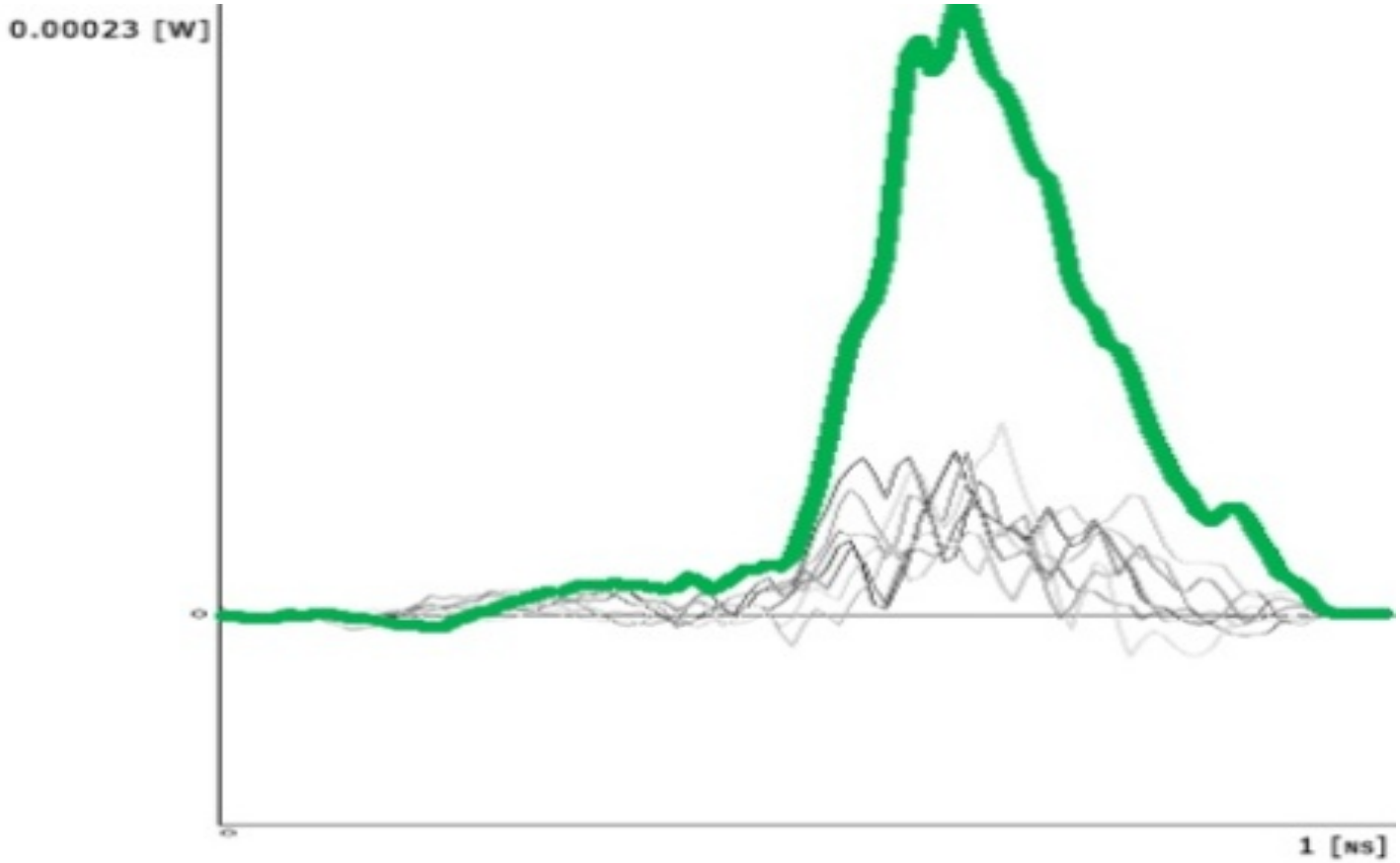
Courbe différentielle de puissance avec une hypothèse de clef <u>correcte</u>

Courbe différentielle de puissance avec une hypothèse de clef <u>incorrecte</u>

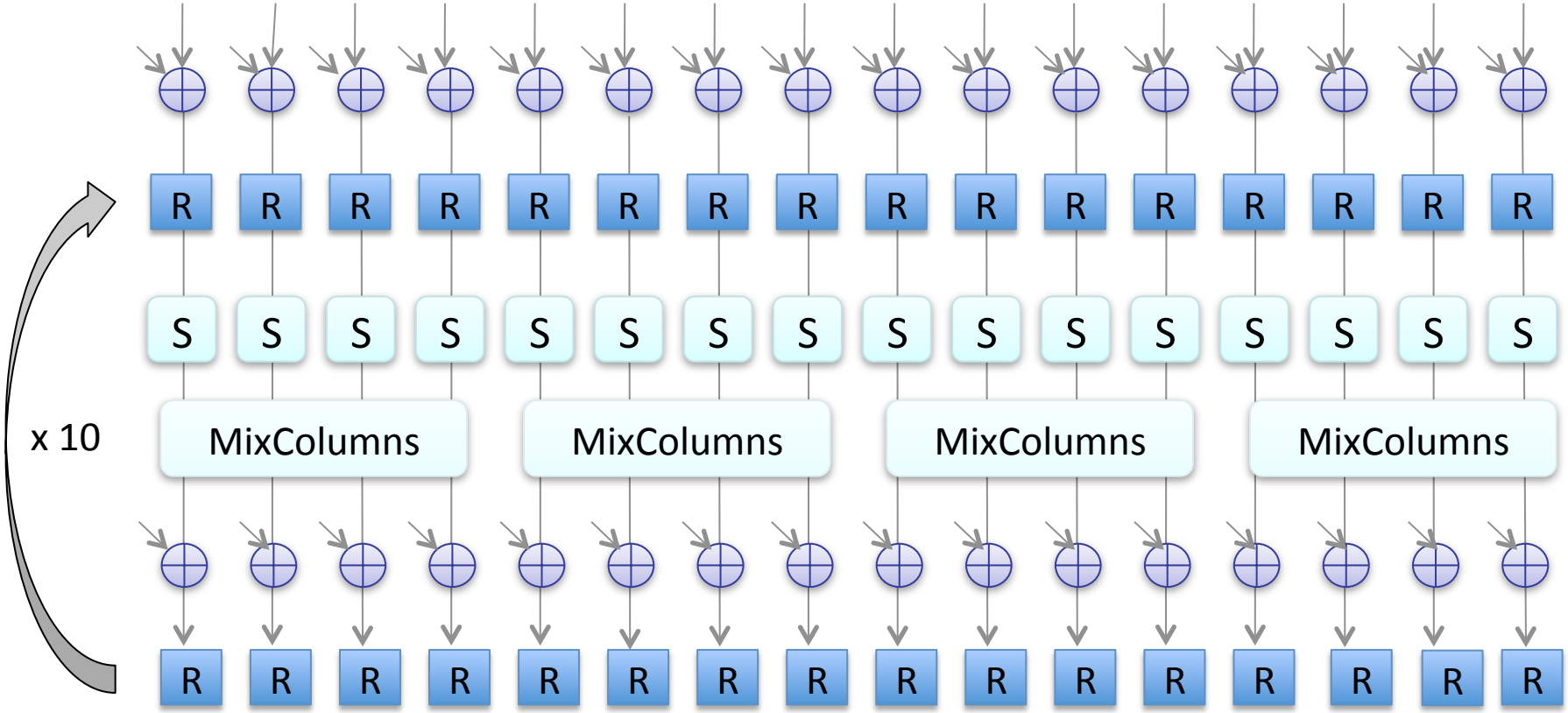
Courbe différentielle de puissance avec autre hypothèse de clef <u>incorrecte</u>



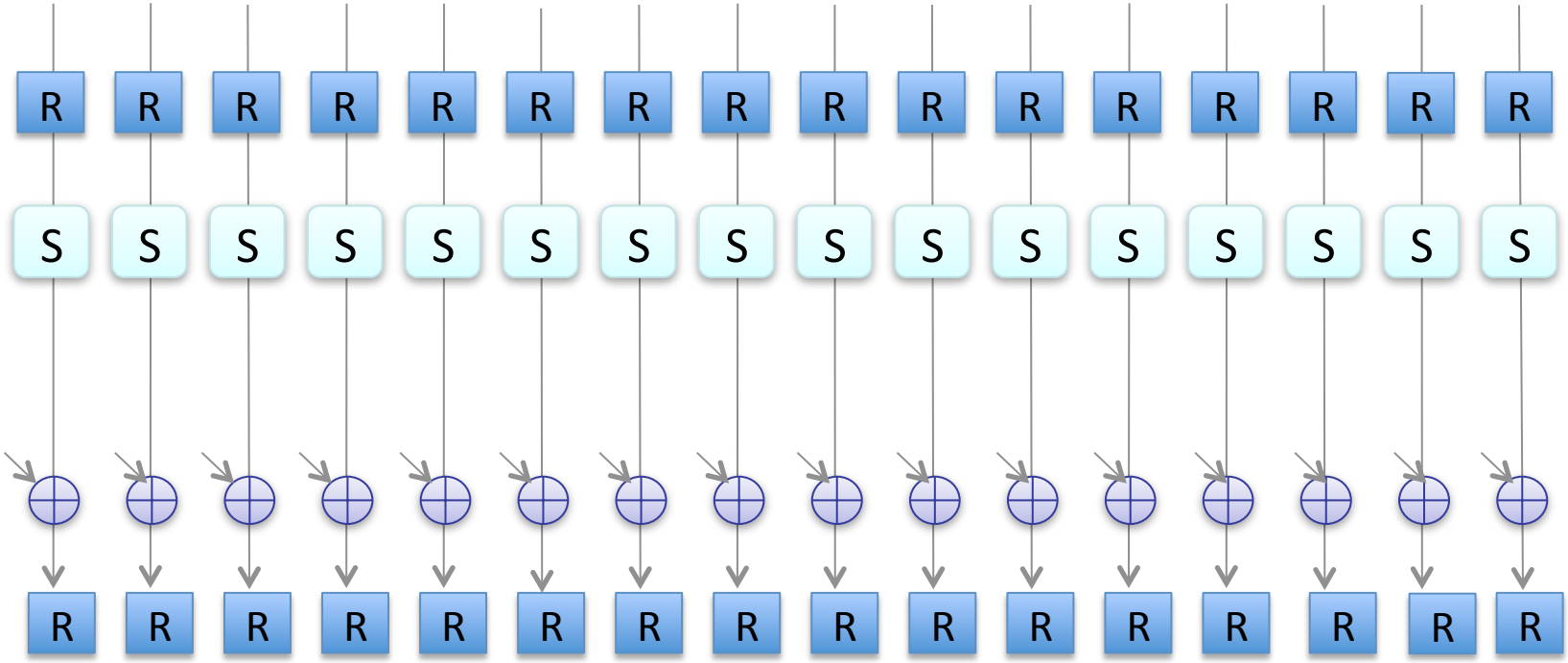
Exemple de résultat



DPA sur AES : Rappel

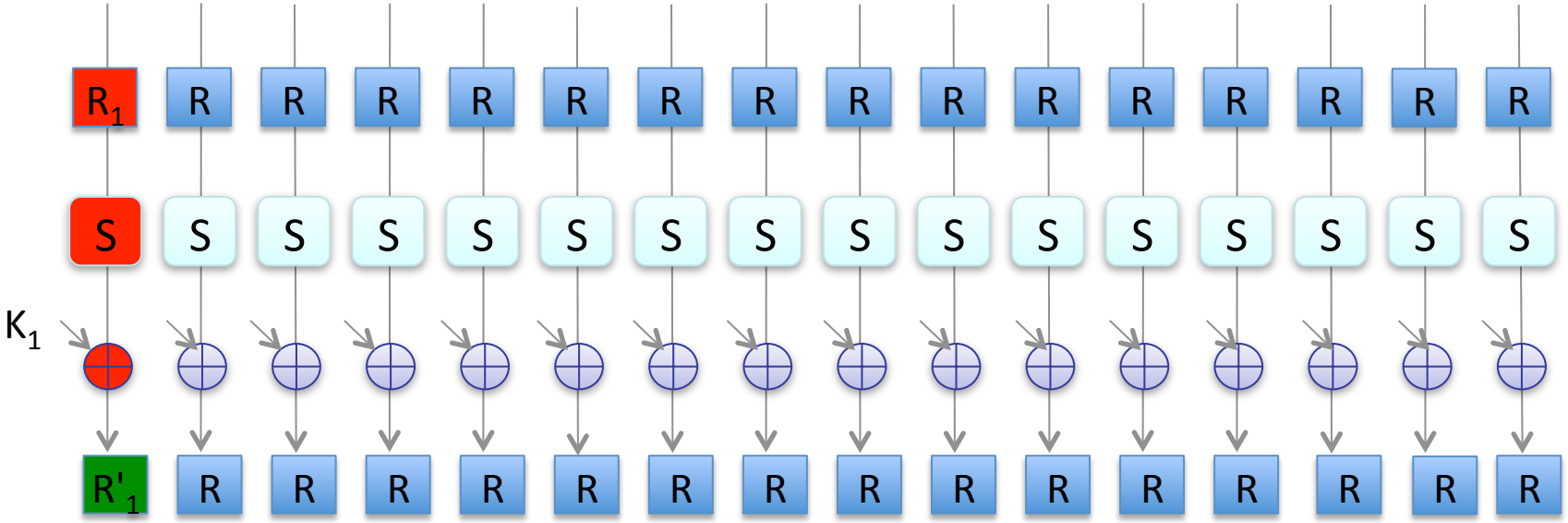


DPA sur AES : Dernière ronde



DPA sur AES : L'attaque

10000 messages clairs aléatoires

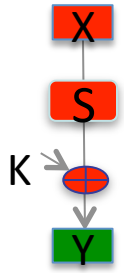


- Attaque sur 1 bit de R_1
- R_1 ne dépend que d'un octet de K
- La conso sur R2, ……, R16 se répartit sur PA et PB de façon homogène
- 256 hypothèses de clef

Exemple : SBOX AES

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	**04**	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Si on observe **04** en sortie d'une SBOX, l'entrée est 30



$Y = SB(X) \text{ xor } K \Leftrightarrow SB(X) = Y \text{ xor } K$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

<u>On s'intéresse au bit de gauche (par exemple)</u>

On observe 04 (00000100) en sortie pour une donnée d'entrée

Hypothèses de clef : 8 bits : 256 hypothèses

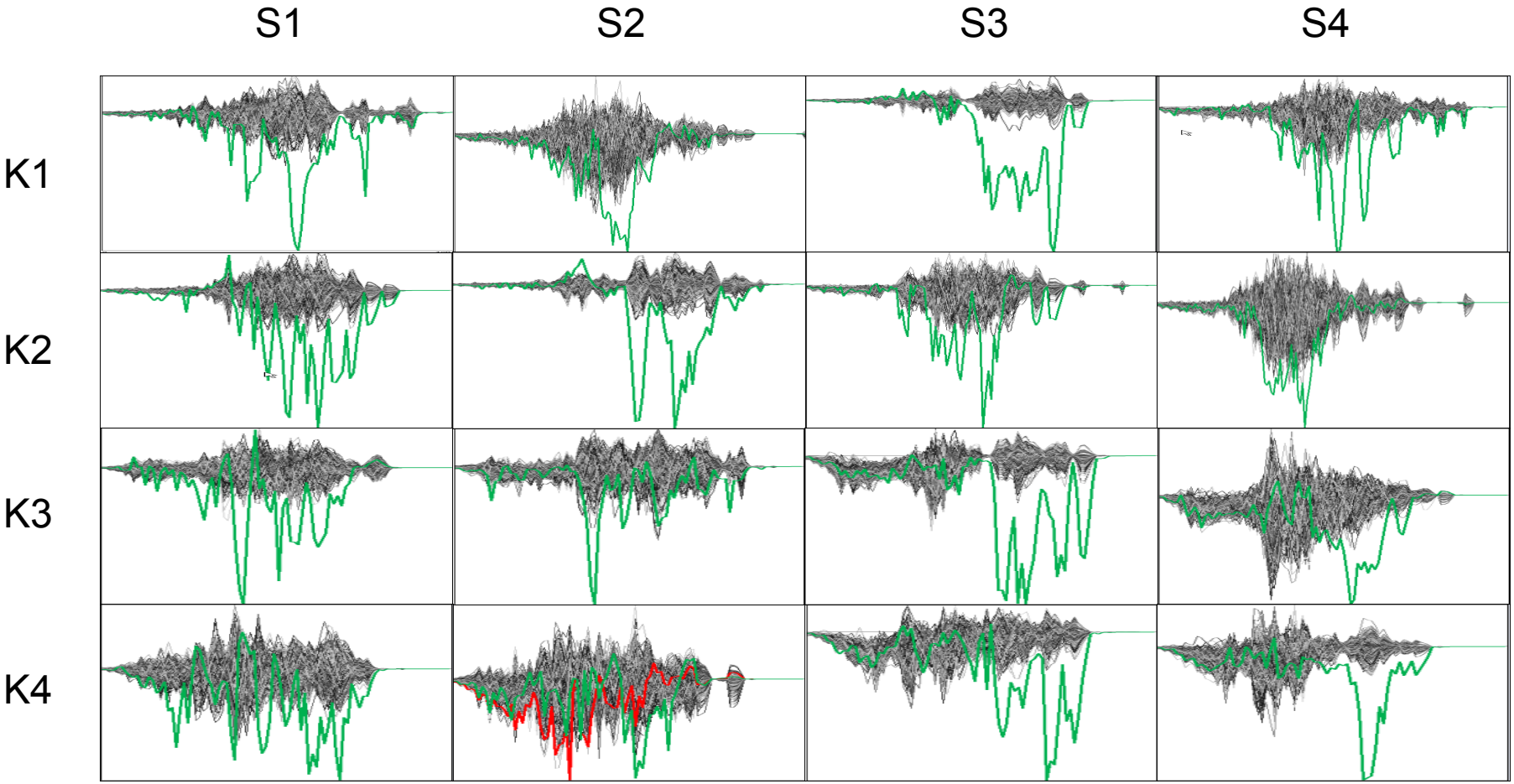
- Clef = 00 => sortie SB(X) = 04 xor 00 = 04 => X = 30 (00110000),
- passage de 0 à 0 => courbe dans paquet B de la clef 00
- Clef = 01 => sortie SB(X) = 04 xor 01 = 05 => X = 36 (00110110),
- passage de 0 à 0 => courbe dans paquet B de la clef 01
- Clef = 02 => sortie SB(X) = 04 xor 02 = 03 => X = D6 (11000110),
- passage de 1 à 0 => courbe dans paquet A de la clef 02
- ……
- Clef = FF => sortie SB(X) = 04 xor FF = FB => X = 63 (01100111),
- passage de 0 à 0 => courbe dans paquet B de la clef FF

On refait la même chose pour (beaucoup) d'autres messages

On fait la moyenne des courbes de PA et PB pour chaque hypothèse
On fait le différence des courbes $PA_{moyenne}$ et $PB_{moyenne}$ pour chaque hypothèse

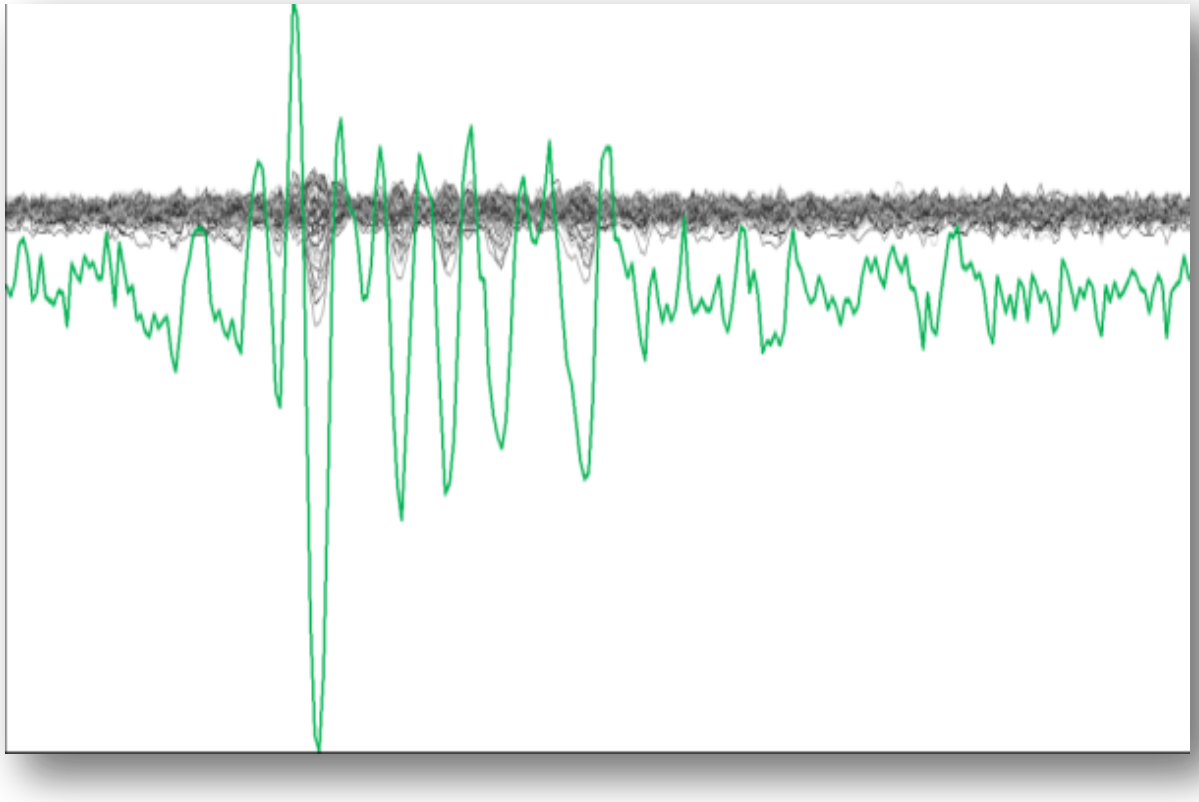
file://localhost/Users/brunorouzeyre/Bruno/Crypto/DPA.wmv

Résultats sur différentes implantations de SBOX



Resultats sur FPGA

- V1 implanté surFPGA
- S1



CPA

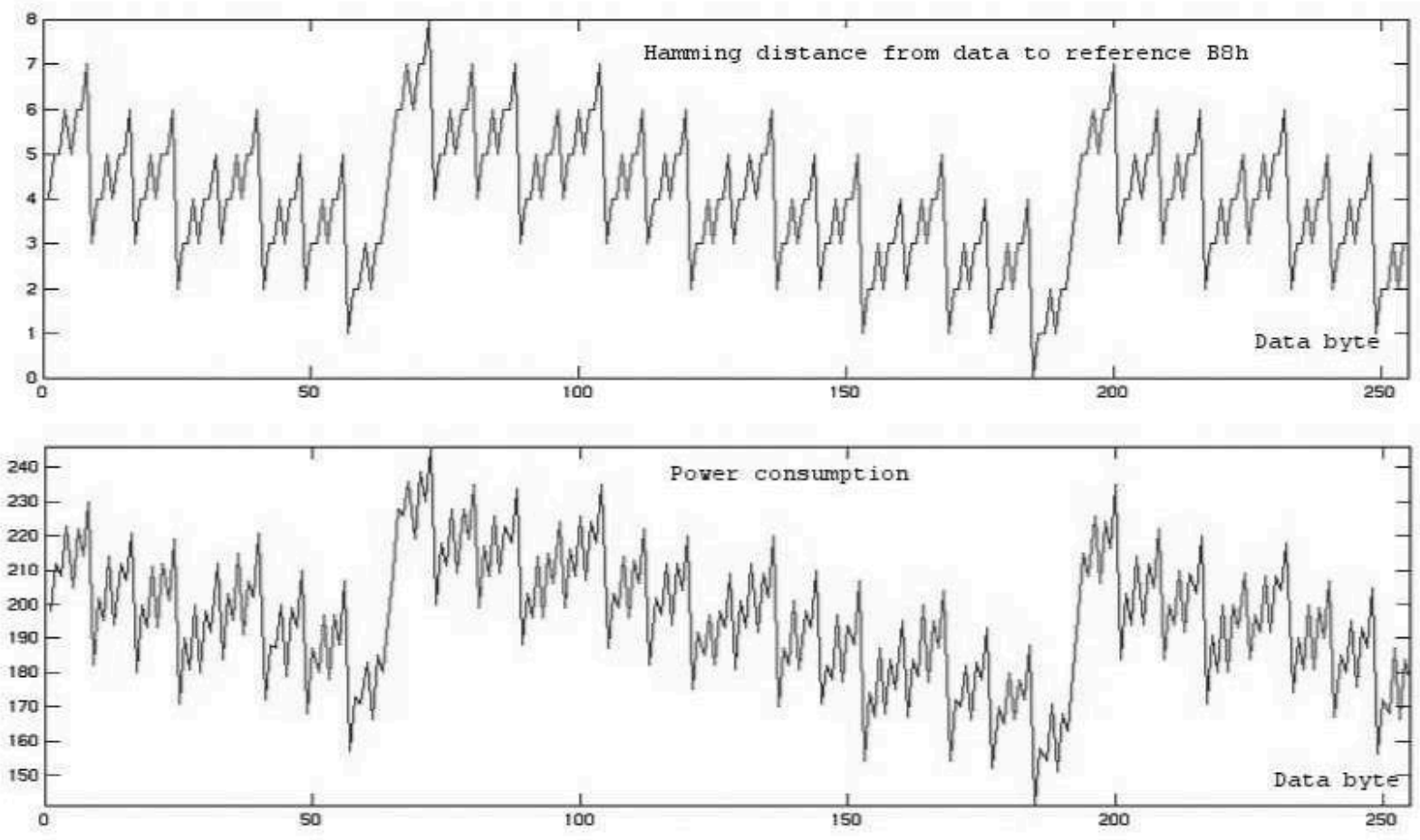
- Modèle poids de Hamming : La puissance W consommée est proportionnelle au poids de Hamming de la donnée
- Coefficient de corrélation : $W = aH(D) + b$

$$\rho_{WH} = \frac{Cov(W,H)}{\sigma_w.\sigma_h} = \frac{E\left((W-\mu_W)(H-\mu_H)\right)}{\sigma_W.\sigma_H}$$

- Le coefficient de corrélation indique comment deux variables aléatoires (Puissance et donnée) correspondent l'une à l'autre.

CPA

- Puissance dissipée par une opération proportionnelle au poids de Hamming



Analyse de corrélation

Coefficient de Pearson:

obtenu en divisant la covariance de 2 variables par leur écart type

$$\hat{\rho}_{WH} = \frac{E\left((W-\mu_W)(H-\mu_H)\right)}{\sigma_W.\sigma_H} = \frac{N\Sigma W_i H_i - \Sigma W_i \Sigma H_i}{\sqrt{N\Sigma W_i^2 - \Sigma W_i^2}\sqrt{N\Sigma H_i^2 - \Sigma H_i^2}}$$

Valeur entre -1 et +1 :

0 : aucune corrélation

DPA en pratique

- Même principe que la DPA:
  - Hypothèses sur les clefs
  - La valeur du coef de Pearson maximale donne la clef

CPA vs DPA

CPA

- Attaque utilisant la relation entre la consommation et les données
- Cherche la meilleure corrélation pour toutes les valeurs de clef
- Plus rapide (nombre de courbes) que la DPA

DPA

- Attaque utilisant la relation entre la consommation et les données
- Vise la différence des moyennes des courbes

Exemple sur DES

Sous-clefs : 24, 19, 8, 8, 5, 50, 43, 2

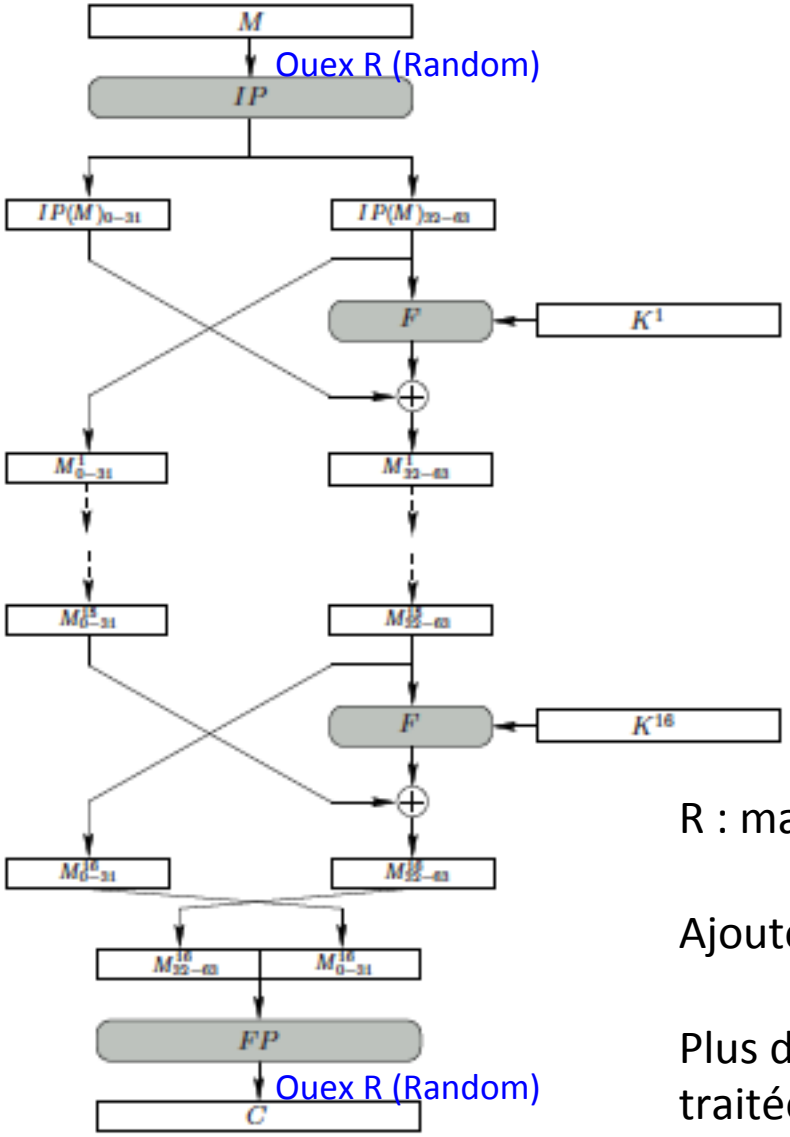
Hypothèses de clef

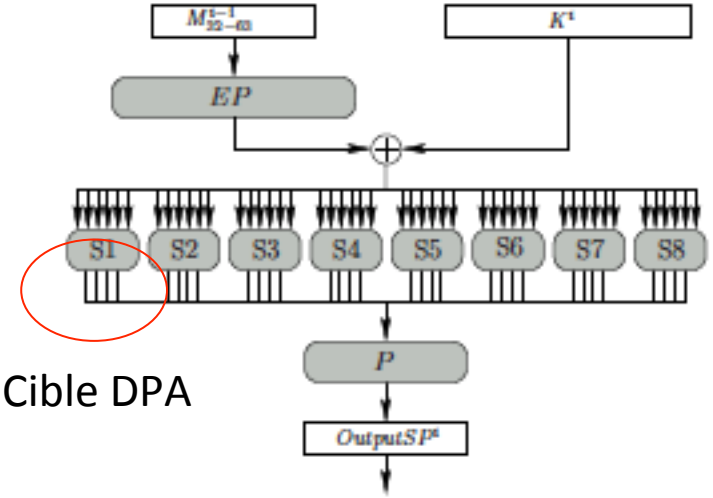
SBox_1		SBox_2		SBox_3		SBox_4		SBox_5		SBox_6		SBox_7		SBox_8	
K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}	K	ρ_{max}
24	92%	19	90%	8	87%	8	88%	5	91%	50	92%	43	89%	2	89%
48	74%	18	77%	18	69%	44	67%	32	71%	25	71%	42	76%	28	77%
01	74%	57	70%	05	68%	49	67%	25	70%	05	70%	52	70%	61	76%
33	74%	02	70%	22	66%	02	66%	34	69%	54	70%	38	69%	41	72%
15	74%	12	68%	58	66%	29	66%	61	67%	29	69%	0	69%	37	70%
06	74%	13	67%	43	65%	37	65%	37	67%	53	67%	30	68%	15	69%

Contre-mesures DPA (ou CPA)

- Masquage des données par une valeur aléatoire
- Techno duale (doubler les bus, implanter f et f')
- Désynchronisation (introduire des temps d'attente aléatoires, décalage des courbes)
- Ajout de bruit
- Changer la clef régulièrement.
- etc…..

Contre-mesure DPA par masquage DES





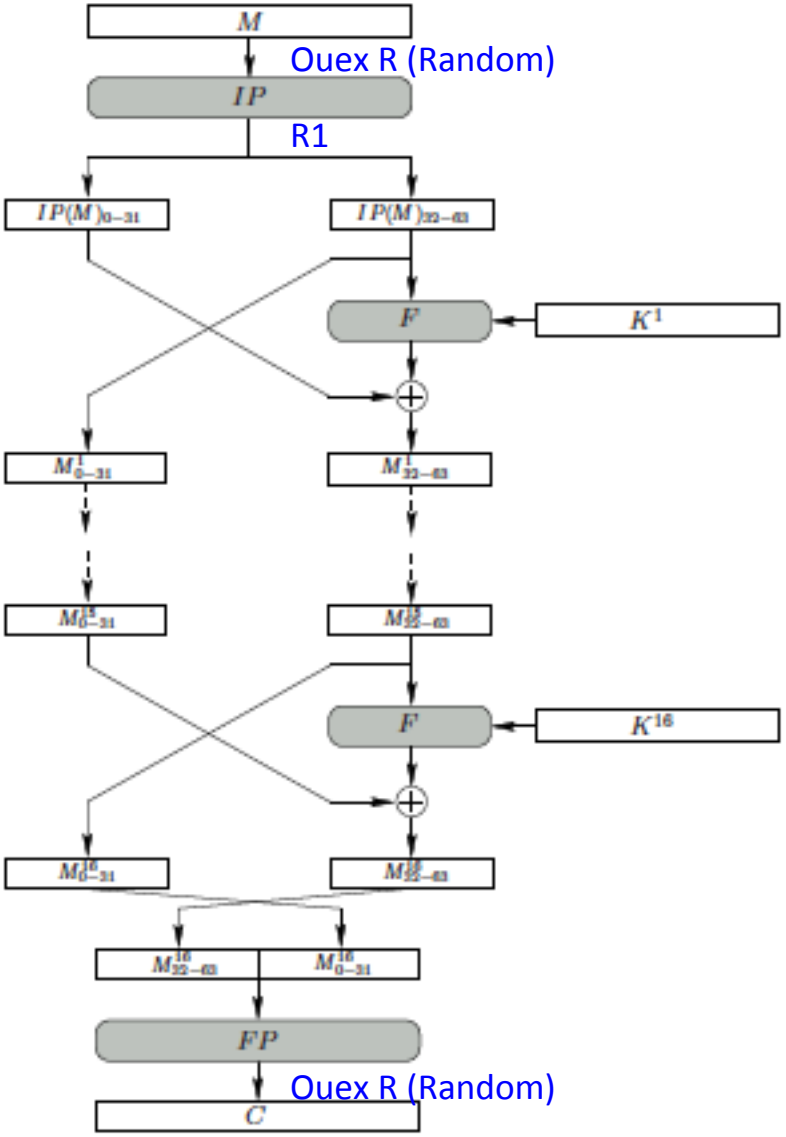
Cible DPA

R : masque

Ajouté en début et en fin de cryptage

Plus de corrélation entre la conso et les données traitées

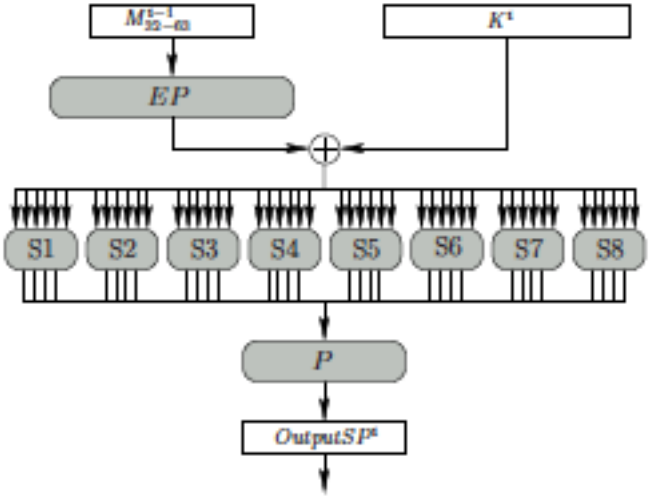
Contre-mesure DPA par masquage DES



R : masque

Mais toutes les opérations ne sont pas linéaires : DES (M+R) +R ≠ DES (M)

Dans le cas du DES : les SBOXs



Principe (C. Giraud)

- Ou-exclusif entre le message M et le masque R avant la permutation initiale IP.
- R1 =IP(R)
- Facile de suivre la propagation du masque lors des parties linéaires.
- En effet, on obtient avant les boites-S un résultat intermédiaire masqué avec $R2 = EP(R1_{32-63})$
- Cependant, il est très compliqué de connaitre la valeur du masque booléen en sortie des parties non linéaires du DES, c.-à-d. en sortie des boites-S.
- Afin de permettre de maitriser la valeur du masque en sortie de ces parties, on utilise des boites-S modifiées notées boîtes-SM.
- La méthode dans le cas du DES consiste à rétablir la valeur du masque R1 à la fin de chaque tour.
- En particulier, la partie droite de la sortie du tour doit avoir un masque correspondant à $R1_{32-63}$.
- Pour obtenir cela, on définit la boite-SM à partir de la boite S du DES par :

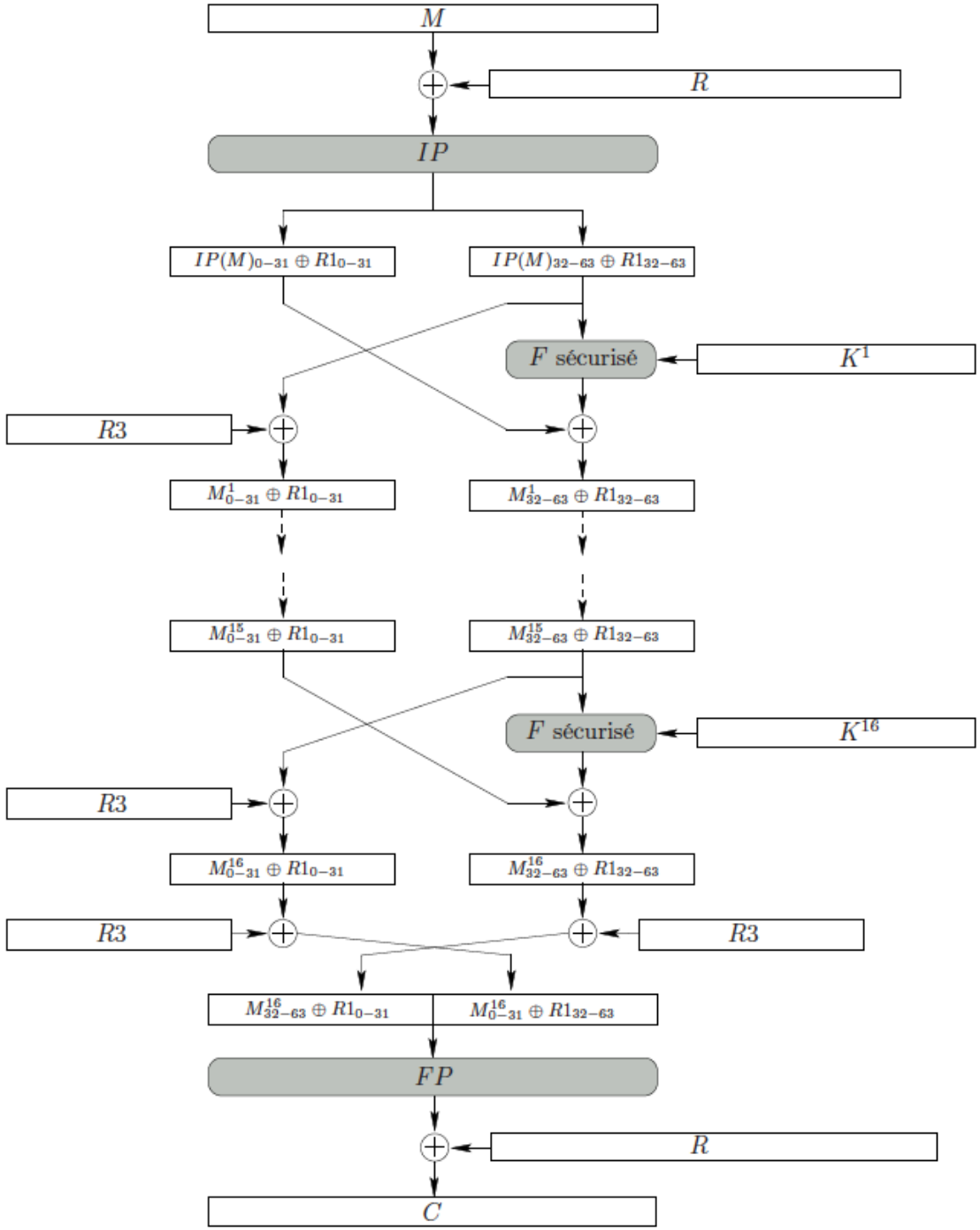
  $SM(A) = S(A \oplus R2) \oplus P^{-1}(R1_{0-31} \oplus R1_{32-63})$ où P^{-1} est l'inverse de la permutation P.
- Il est aussi nécessaire de modifier la partie gauche de l'entrée en appliquant un ou-exclusif avec $R3 = R1_{0-31} \oplus R1_{32-63}$. Le masque R1 sera donc rétabli à la fin du tour.
- Après le dernier tour, les deux blocs de 32 bits sont intervertis. Pour des raisons d'efficacité, on applique un ou-exclusif sur ces deux parties avec R3 avant la permutation finale FP. Ceci permet d'obtenir après FP le chiffré attendu masqué avec R.

Création des masques et calcul de la boîte-S modifiée

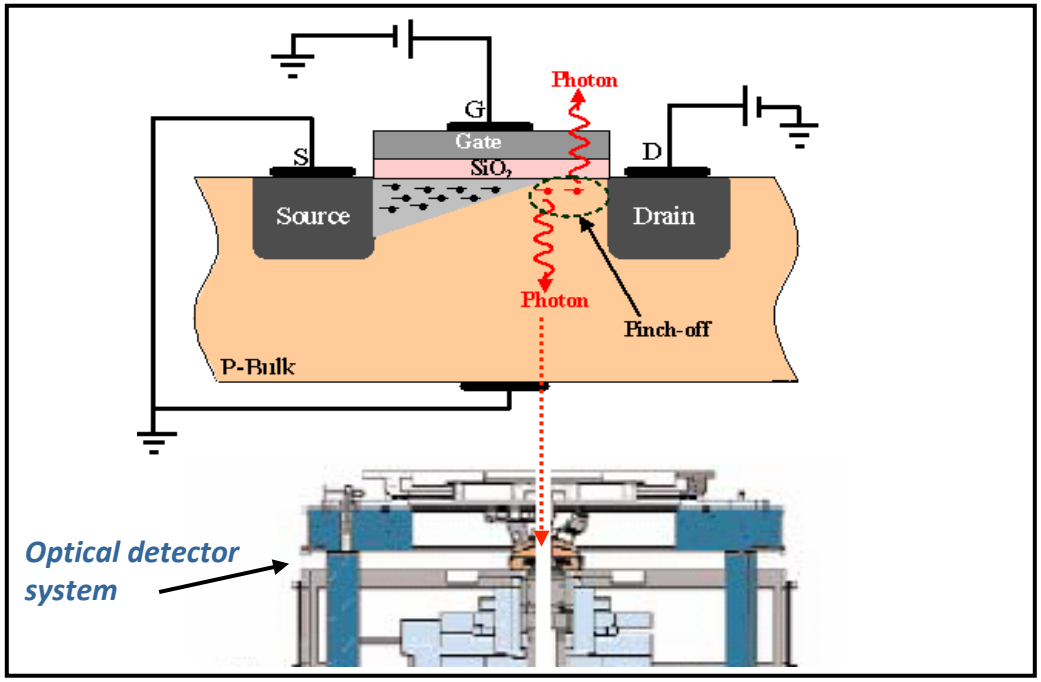
1. Tirer une valeur aléatoire R de 8 octets
2. R1 ← IP(R)
3. R2 ← EP($R1_{32-63}$)
4. R3 ← $R1_{0-31} \oplus R1_{32-63}$
5. Pour k de 1 à 8 faire
6. Pour i de 0 à 63 faire
7. $SM_k(i) = S_k(i \oplus R2_{[6(k-1)]-[6k-1]}) \oplus (P{-}1(R1_{0-31} \oplus R1_{32-63}))_{[4(k-1)]-[4k-1]}$

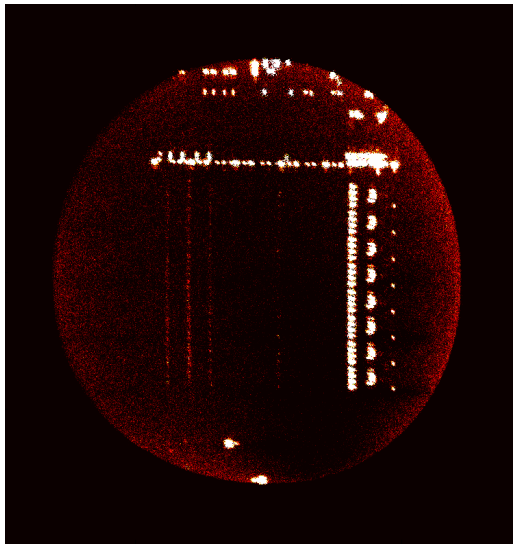
DES sécurisé

8. resultat ← M ⊕ R
9. resultat ← IP(resultat) *[resultat = IP(M) ⊕ R1]*
10. Pour i de 1 à 16 faire *Les 16 rondes du DES*
11. temp1 ← $resultat_{32-63} \oplus R3$ *[temp1 = Mi ⊕ $R1_{0-31}$]*
12. temp2 ← EP($result_{32-63}$) *[temp2 = EP(Mi−1_{32-63}) ⊕ R2]*
13. temp2 ← temp2 ⊕ Ki *[temp2 = EP(Mi−1_{32-63}) ⊕ R2 ⊕ Ki]*
14. Pour k de 1 à 8 faire
15. $temp2_{[6(k-1)]-[6k-1]} \leftarrow SM_k(temp2_{[6(k-1)]-[6k-1]})$ ***[temp2 = P^{-1}(OutputSPi ⊕ $R1_{0-31}$ ⊕ $R1_{32-63}$)]***
16. $resultat_{32-63} \leftarrow P(temp2)$ ***[$resultat_{32-63}$ = OutputSPi ⊕ $R1_{0-31}$ ⊕ $R1_{32-63}$]***
17. $resultat_{32-63} \leftarrow resultat_{32-63} \oplus resultat_{0-31}$ ***[$resultat_{32-63}$ = Mi_{32-63} ⊕ $R1_{32-63}$]***
18. $resultat_{0-31} \leftarrow temp1$ ***[$resultat_{0-31}$ = Mi_{0-31} ⊕ $R1_{0-31}$]***
19. $temp1 \leftarrow resultat_{0-31} \oplus R3$ ***[temp1 = $M16_{0-31}$ ⊕ $R1_{32-63}$]***
20. $resultat_{0-31} \leftarrow resultat_{32-63} \oplus R3$ ***[$resultat_{0-31}$ = $M16_{32-63}$ ⊕ $R1_{0-31}$]***
21. $resultat_{32-63} \leftarrow temp1$ ***[$resultat_{32-63}$ = $M16_{0-31}$ ⊕ $R1_{32-63}$]***
22. resultat ← FP(resultat) ***[resultat = C ⊕ R]***
23. resultat ← resultat ⊕ R ***[resultat = C]***
24. Retourner resultat

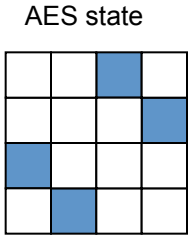

M
R
IP
$IP(M)_{0-31} \oplus R1_{0-31}$
$IP(M)_{32-63} \oplus R1_{32-63}$
F sécurisé
K^1
$R3$
$M^1_{0-31} \oplus R1_{0-31}$
$M^1_{32-63} \oplus R1_{32-63}$
$M^{15}_{0-31} \oplus R1_{0-31}$
$M^{15}_{32-63} \oplus R1_{32-63}$
F sécurisé
K^{16}
$R3$
$M^{16}_{0-31} \oplus R1_{0-31}$
$M^{16}_{32-63} \oplus R1_{32-63}$
$R3$
$R3$
$M^{16}_{32-63} \oplus R1_{0-31}$
$M^{16}_{0-31} \oplus R1_{32-63}$
FP
R
C

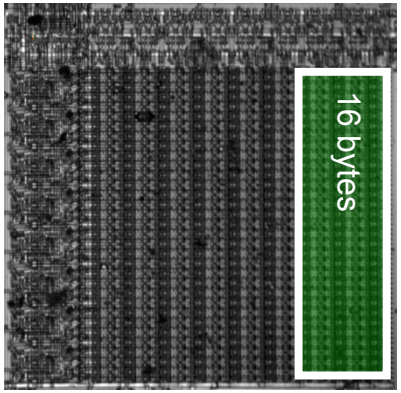
2/ Attaques en Emission de lumière



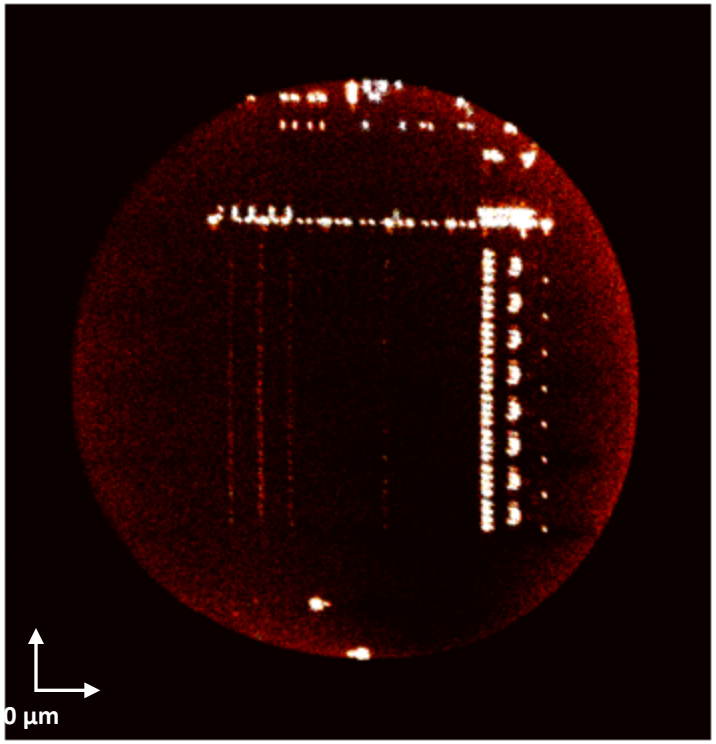


Coût : 2M€ !





PIC Internal RAM (20x; silicon thickness 40 µm)

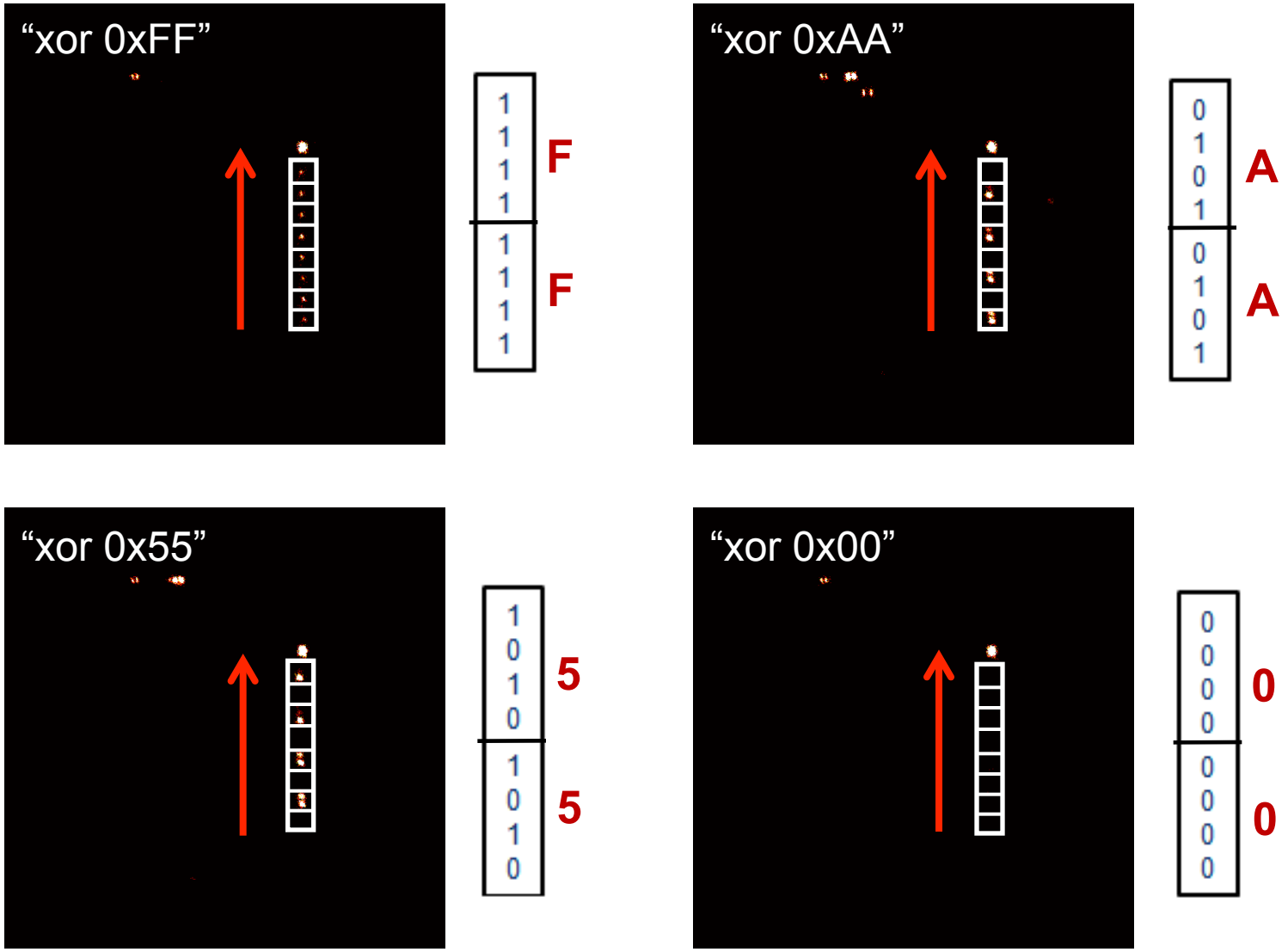


Monitor the changes on the bytes in State block during AES encryptions.

How? : **Dynamic light emission detection (PICA)**

Theory **:** byte flips => light is emitted

byte stays => just noise


“xor 0xFF”
1 1 1 1
1 1 1 1
F
F
“xor 0xAA”
0 1 0 1
0 1 0 1
A
A
“xor 0x55”
1 0 1 0
1 0 1 0
5
5
“xor 0x00”
0 0 0 0
0 0 0 0
0
0

Conclusion

- Algorithmes crypto sûrs mais implantation matérielle (vitesse, portabilité …)
- Attaques sur circuit par observation des valeurs physiques :
  - conso,
  - émission de lumière,
  - émanations EM (cf. P. Maurine)
  - …
- Corrélation entre valeurs physiques et données
- SPA, DPA, CPA
- Contre-mesures