Attaques par consommation de courant: collisions et codes LDPC.

Benoît Gérard

CryptoGroup - Université catholique de Louvain - Belgique

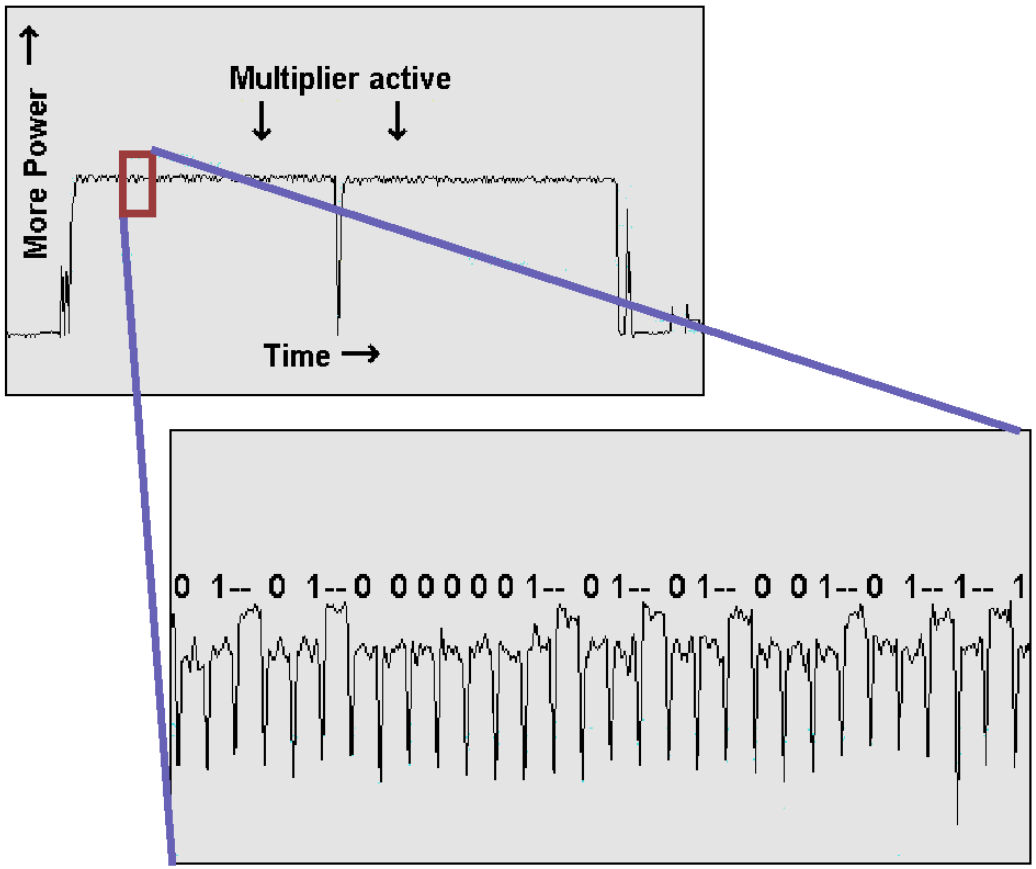
Séminaire GREYC - 29 mars 2012

Attaques par canaux auxiliaires

Attaques

- par consommation électrique;
- par radiations électromagnétiques;
- temporelles;
- par injection de fautes;
- “buffer overflow”;
- ...

Analyse par consommation électrique (1/4)



RSA (déchiffrement)

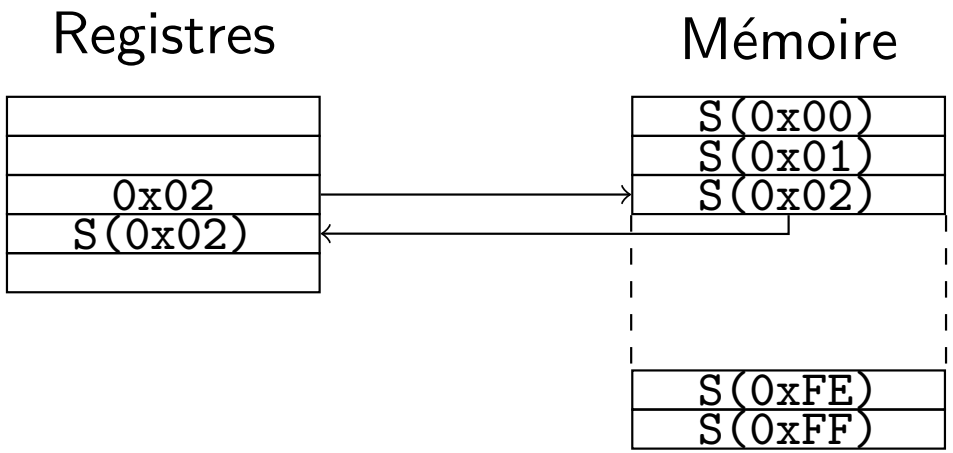
$$m = c^{s_k} \mod pq$$

- Restes Chinois
- Square & Multiply
- $I_{multi} > I_{square}$

Analyse par consommation électrique (2/4)

[Kocher, Jaffe et Jun, *CRYPTO'99*]: *In addition to large-scale power variations due to the instruction sequence, there are effects correlated to data values being manipulated.*

Pour un chiffrement symétrique, on cible les boîtes-S.



Boîte-S en table sur un microcontrôleur.

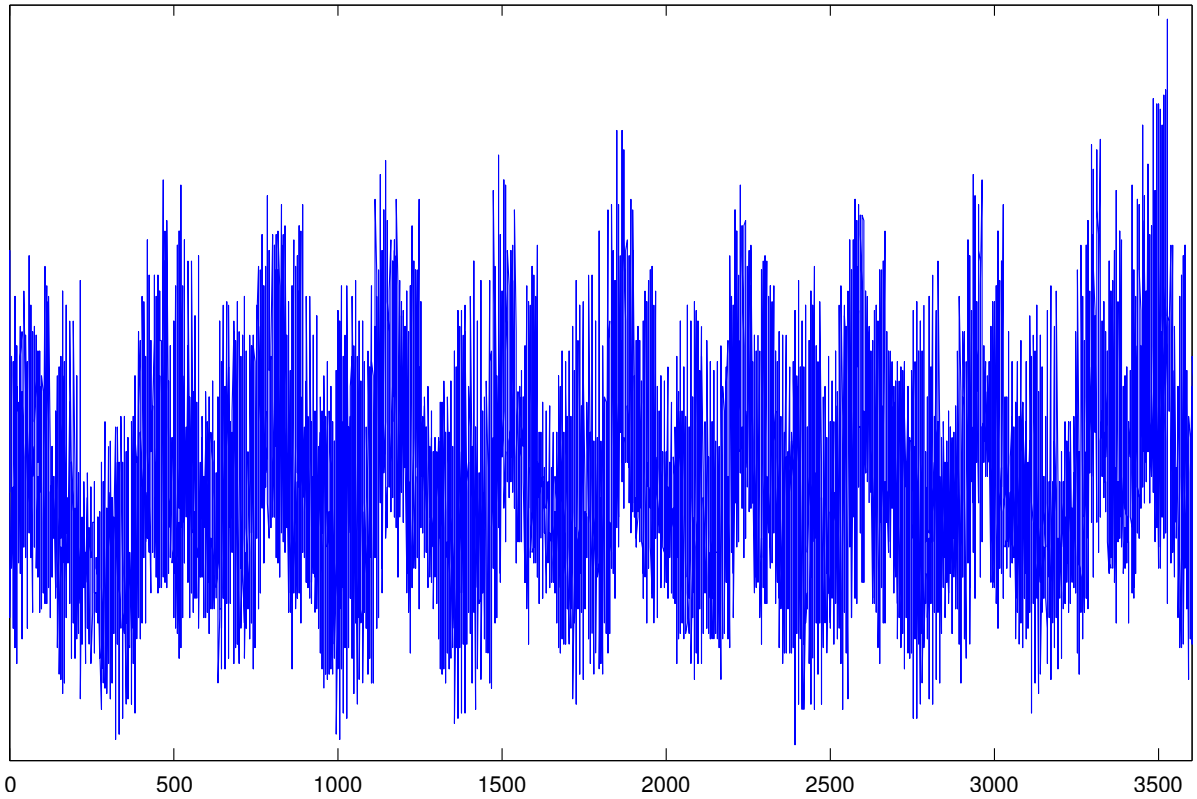
Analyse par consommation électrique (3/4)

1. Profilage:

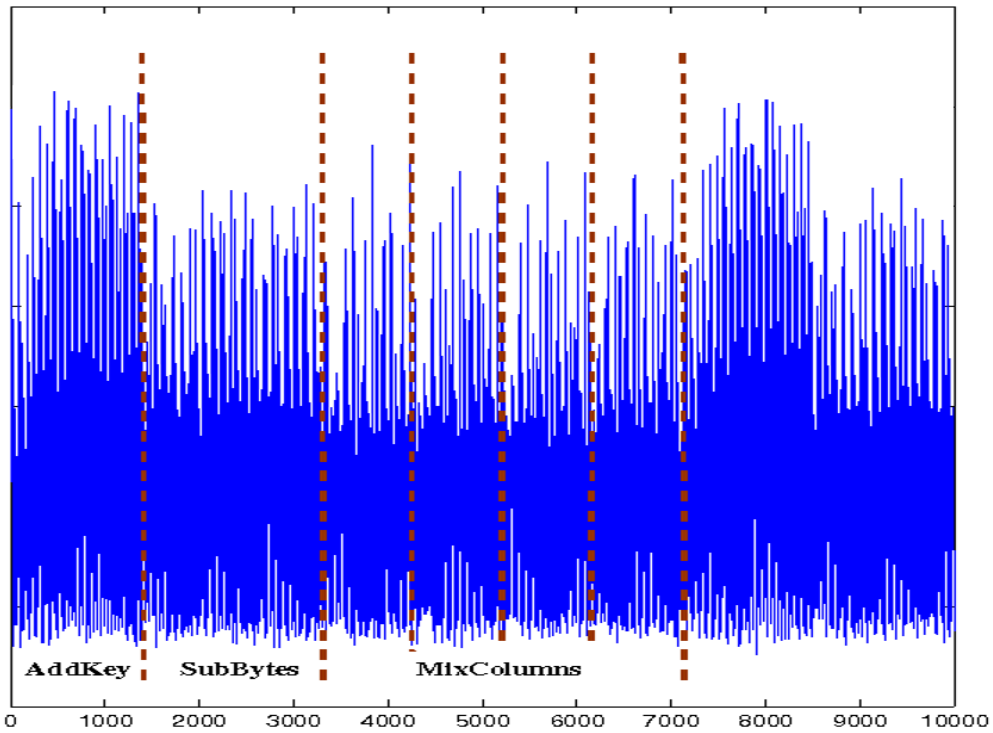
   Lien entre consommation et valeur déterminée expérimentalement hors ligne.

2. Intuition d'ingénieur $\longrightarrow$ modèle de fuite:
   - un fil à 1 consomme plus qu'à 0;
   - fonction du poids de Hamming d'une valeur;
   - combinaison linéaire des bits d'une valeur;
   - fonction de la distance de Hamming entre l'ancienne et la nouvelle valeur.

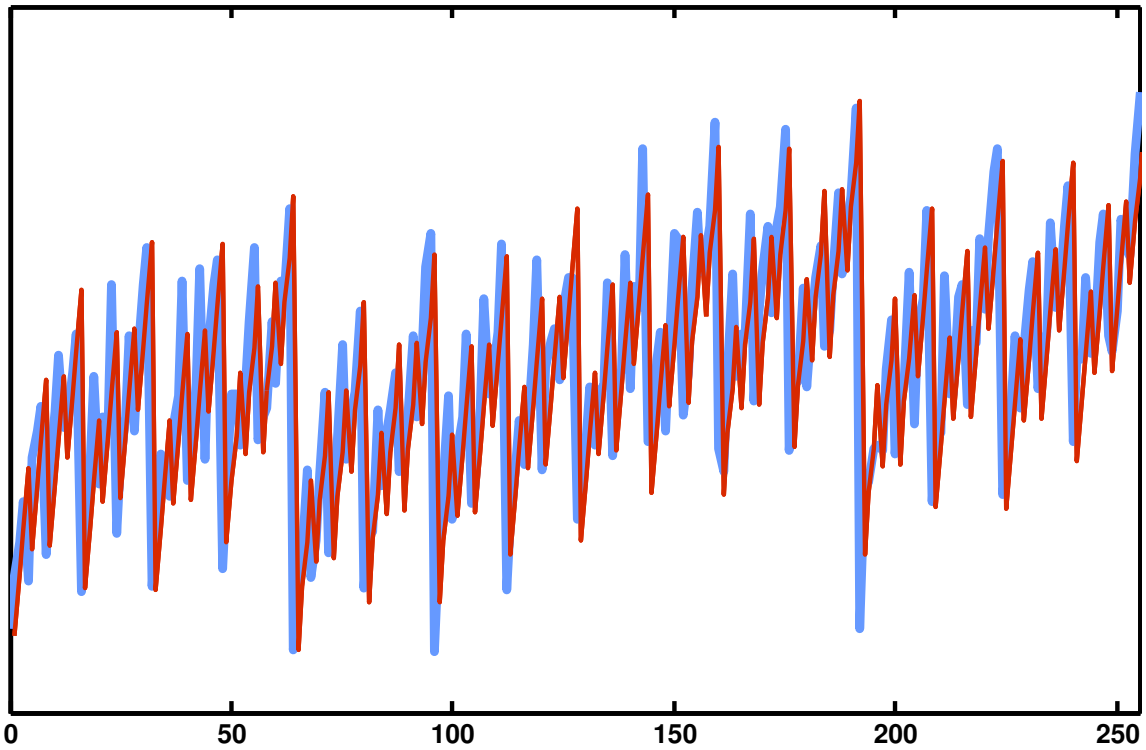
Analyse par consommation électrique (4/4)



Analyse par consommation électrique (4/4)



Analyse par consommation électrique (4/4)



Plan

Attaque par collisions linéaires

Vers une attaque plus “propre”

Résultats expérimentaux

Perspectives

Plan

Attaque par collisions linéaires

Vers une attaque plus “propre”

Résultats expérimentaux

Perspectives

Deux contextes d'attaque

1. Attaques profilées:
   - ✓ attaques "optimale";
   - x profilage = contraintes;
   - x variabilité d'un circuit à l'autre.
2. Attaques non-profilées:
   - ✓ peu de connaissance *a priori* du circuit;
   - x ↗ généralité $\Longrightarrow$ efficacité ↘;
   - x validité des modèles sur des technologies 45nm ($\rightarrow$11nm);

Deux contextes d'attaque

1. Attaques profilées:
   - ✓ attaques "optimale";
   - x profilage = contraintes;
   - x variabilité d'un circuit à l'autre.

2. Attaques non-profilées:
   - ✓ peu de connaissance *a priori* du circuit;
   - x ↗ généralité $\Longrightarrow$ efficacité ↘;
   - x validité des modèles sur des technologies 45nm ($\rightarrow$11nm);

S'affranchir du modèle de fuite

Boîtes-S en série:

- Software: liste instructions.
- Hardware: contrainte spatiale $>$ contrainte efficacité.

Hypothèse

Si $X = X'$, alors

$$T \approx T'.$$

S'affranchir du modèle de fuite

Boîtes-S en série:

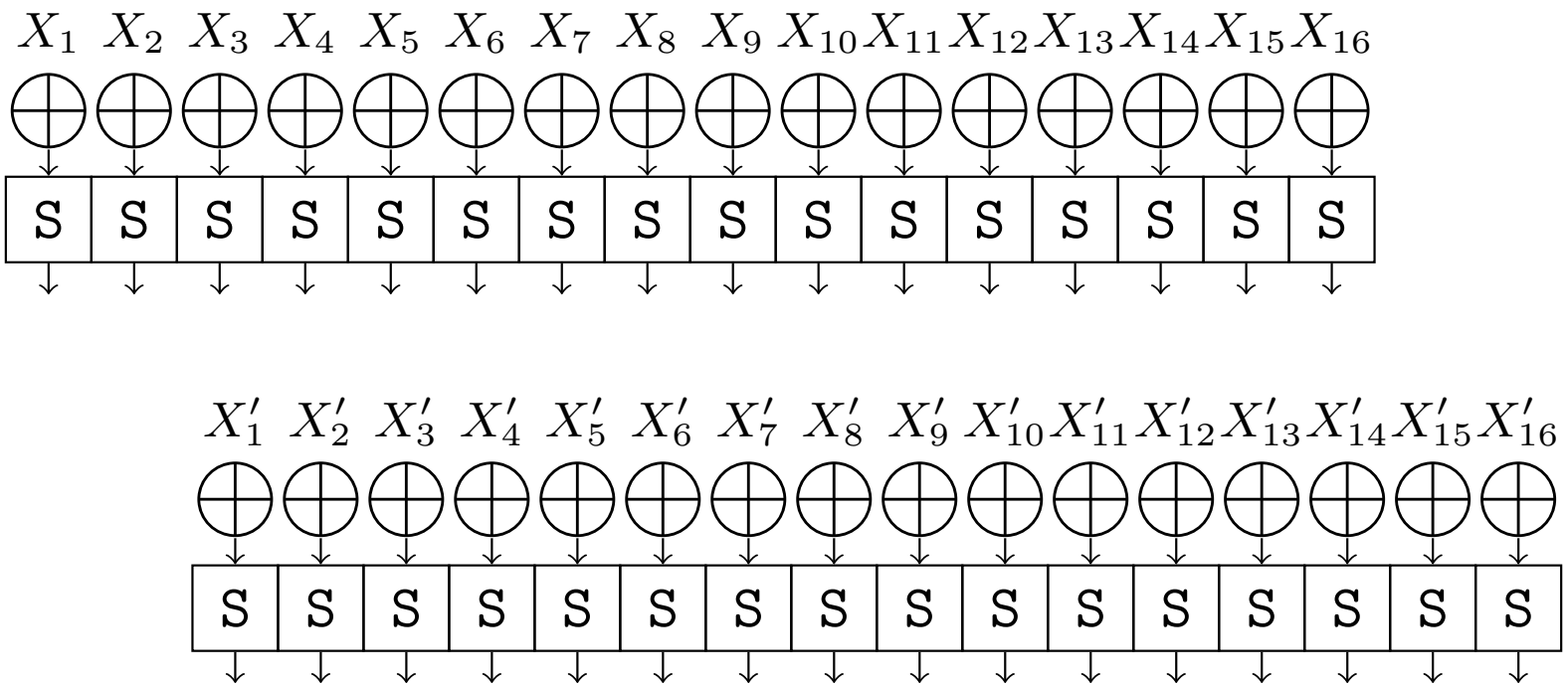
- Software: liste instructions.
- Hardware: contrainte spatiale $>$ contrainte efficacité.

Modèle

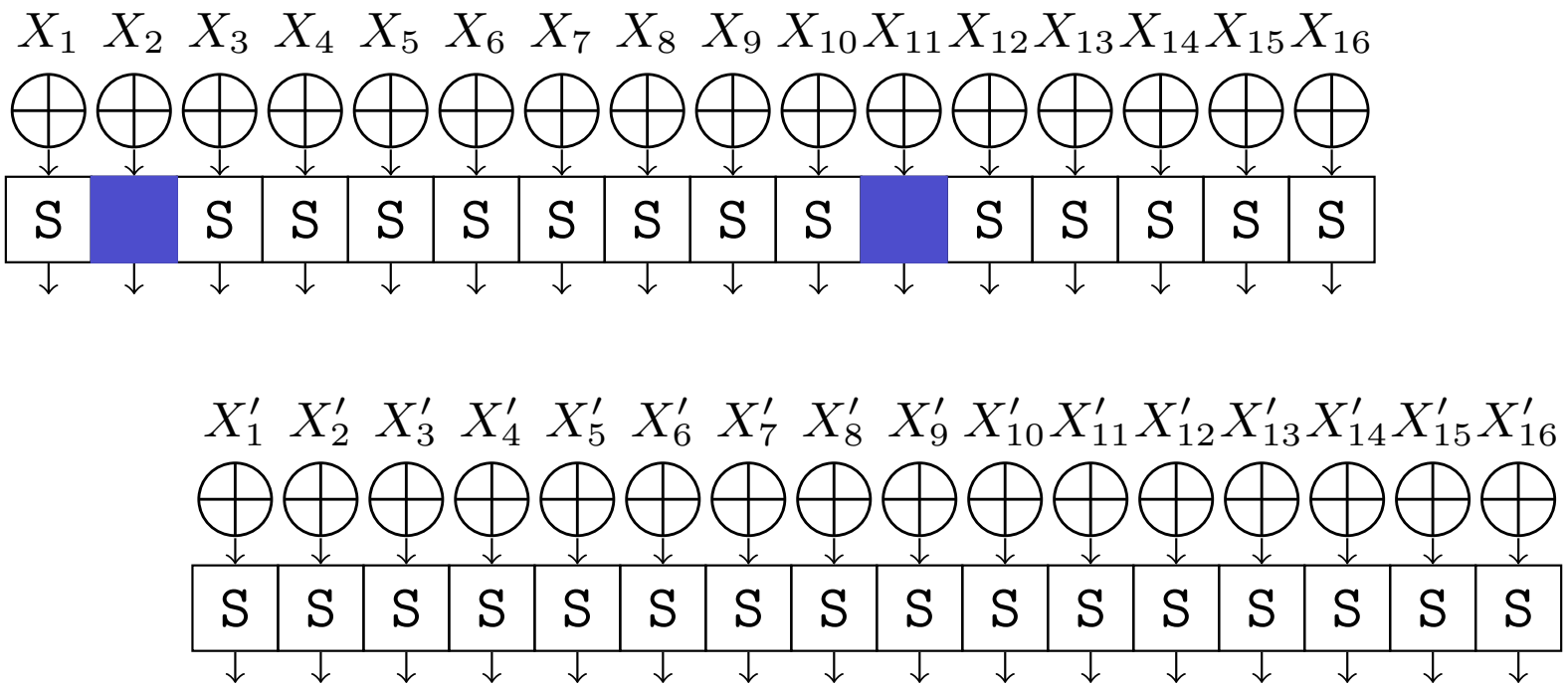
Il existe $S(\cdot,\cdot)$ et deux distributions $\mathcal{D}_{\mathrm{c}}$ et $\mathcal{D}_{\mathrm{nc}}$ telles que:

$$\Pr\left[S(T,T')=s\right]=\begin{cases}\Pr_{\mathcal{D}_c}\left[s\right] & \text{si } X=X',\\ \Pr_{\mathcal{D}_{nc}}\left[s\right] & \text{sinon.}\end{cases}$$

Attaques par collisions linéaires sur l'AES

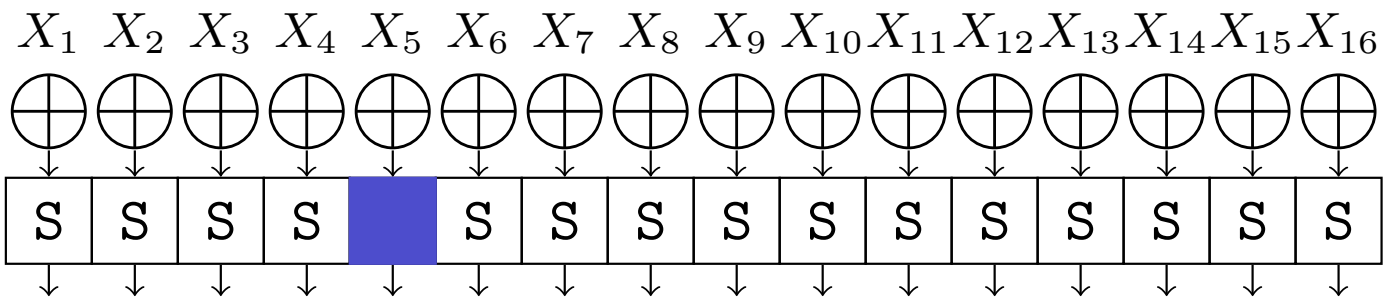


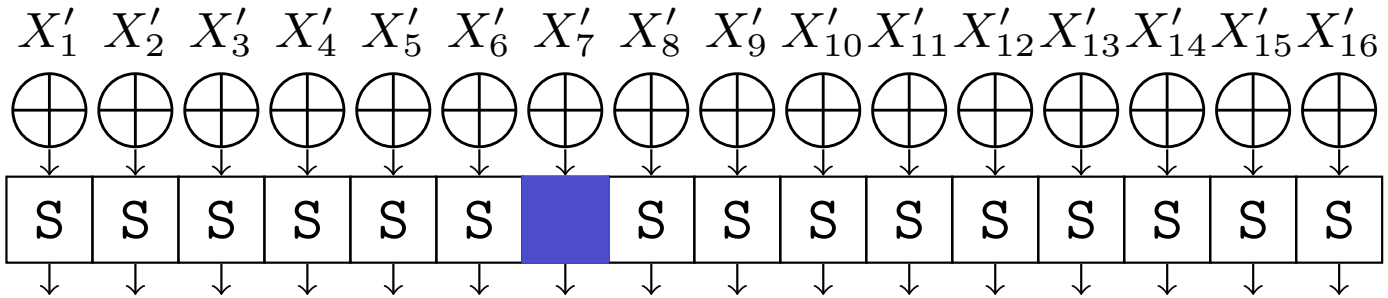
Attaques par collisions linéaires sur l'AES



$$X_2 \oplus K_2 = X_{11} \oplus K_{11} \quad \Longrightarrow \quad K_2 \oplus K_{11} = X_2 \oplus X_{11}$$

Attaques par collisions linéaires sur l'AES





$$X_5 \oplus K_5 = X'_7 \oplus K_7 \quad \Longrightarrow \quad K_5 \oplus K_7 = X_5 \oplus X'_7$$

Attaques par collisions linéaires sur l'AES

$$\begin{cases} K_1 \oplus K_5 &= \Delta_{1,5} \\ K_1 \oplus K_2 &= \Delta_{1,2} \\ K_2 \oplus K_8 &= \Delta_{2,8} \\ K_3 \oplus K_4 &= \Delta_{3,4} \\ K_6 \oplus K_7 &= \Delta_{6,7} \end{cases}$$

Attaques par collisions linéaires sur l'AES

$$\begin{cases} K_1 \oplus K_5 = \Delta_{1,5} \\ K_1 \oplus K_2 = \Delta_{1,2} \\ K_2 \oplus K_8 = \Delta_{2,8} \\ K_3 \oplus K_4 = \Delta_{3,4} \\ K_6 \oplus K_7 = \Delta_{6,7} \end{cases}$$

2^{24} valeurs pour (K_1, K_3, K_6)

$\Downarrow$

2^{24} clefs $(K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8)$
au lieu de 2^{64}.

Erreurs

Bruit $\Rightarrow$ erreurs!

Détection de collisions $\Longleftrightarrow$ Test binaire d'hypothèses

- Erreur de type I: collision non détectée (pas si grave).
- Erreur de type II: fausse collision détectée!!!

$$\begin{cases} K_1 \oplus K_2 &= \texttt{0x76} \\ K_2 \oplus K_3 &= \texttt{0x61} \\ K_4 \oplus K_5 &= \texttt{0x77} \end{cases}$$

Vote binaire/ternaire [Bogdanov 2008]

Ternaire

Binaire

$$T \approx T'$$

$$\Downarrow$$

$$(*) \begin{cases} \text{soit } T \approx T'' \text{ et } T' \approx T'', \\ \text{soit } T \not\approx T'' \text{ et } T' \not\approx T''. \end{cases}$$

Vote binaire/ternaire [Bogdanov 2008]

Ternaire

Binaire

$$T \approx T'$$

$$\Downarrow$$

$$(*) \begin{cases} \text{soit } T \approx T'' \text{ et } T' \approx T'', \\ \text{soit } T \not\approx T'' \text{ et } T' \not\approx T''. \end{cases}$$

$$\#\{T'' \text{ satisfaisant } (*)\}$$

Vote binaire/ternaire [Bogdanov 2008]

Ternaire

$$T \approx T'$$

$$\Downarrow$$

$$(*) \begin{cases} \text{soit } T \approx T'' \text{ et } T' \approx T'', \\ \text{soit } T \not\approx T'' \text{ et } T' \not\approx T''. \end{cases}$$

$$\#\{T'' \text{ satisfaisant } (*)\}$$

Binaire

$$\underbrace{\begin{cases} K_1 \oplus K_2 = \texttt{0x76} \\ K_1 \oplus K_2 = \texttt{0x12} \\ K_1 \oplus K_2 = \texttt{0x76} \\ K_1 \oplus K_2 = \texttt{0xA2} \\ K_1 \oplus K_2 = \texttt{0x13} \end{cases}}$$

Vote binaire/ternaire [Bogdanov 2008]

Ternaire

$$T \approx T'$$

$$\Downarrow$$

$$(*) \begin{cases} \text{soit } T \approx T'' \text{ et } T' \approx T'', \\ \text{soit } T \not\approx T'' \text{ et } T' \not\approx T''. \end{cases}$$

$$\#\{T'' \text{ satisfaisant } (*)\}$$

Binaire

$$\underbrace{\begin{cases} K_1 \oplus K_2 = \texttt{0x76} \\ K_1 \oplus K_2 = \texttt{0x12} \\ K_1 \oplus K_2 = \texttt{0x76} \\ K_1 \oplus K_2 = \texttt{0xA2} \\ K_1 \oplus K_2 = \texttt{0x13} \end{cases}}$$

$$\Downarrow$$

$$K_1 \oplus K_2 = \texttt{0x76}$$

Plus d'erreurs

Fonction de fuite: poids de Hamming.

$$\mathcal{L}(x) = HW(x).$$

Alors, par exemple

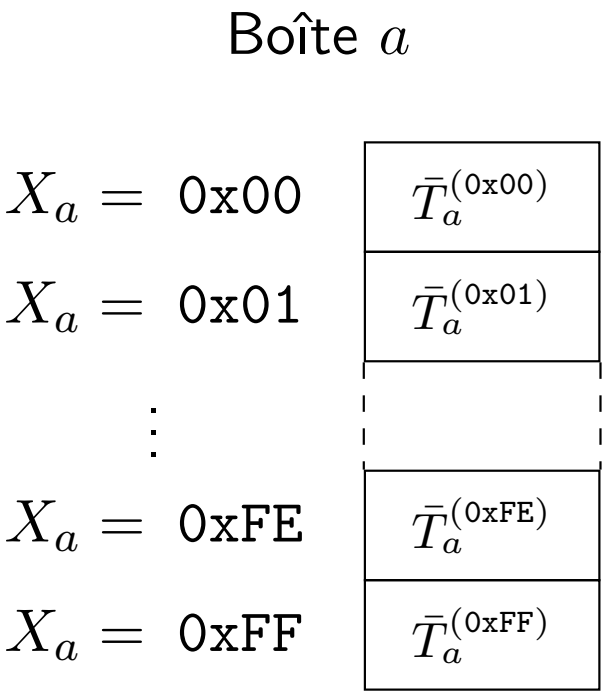
$$\mathcal{L}(\texttt{0x11}) = \mathcal{L}(\texttt{0x50}).$$

$\Rightarrow$ beaucoup de fausses collisions,

$\Rightarrow$ cas non résolu par les votes.

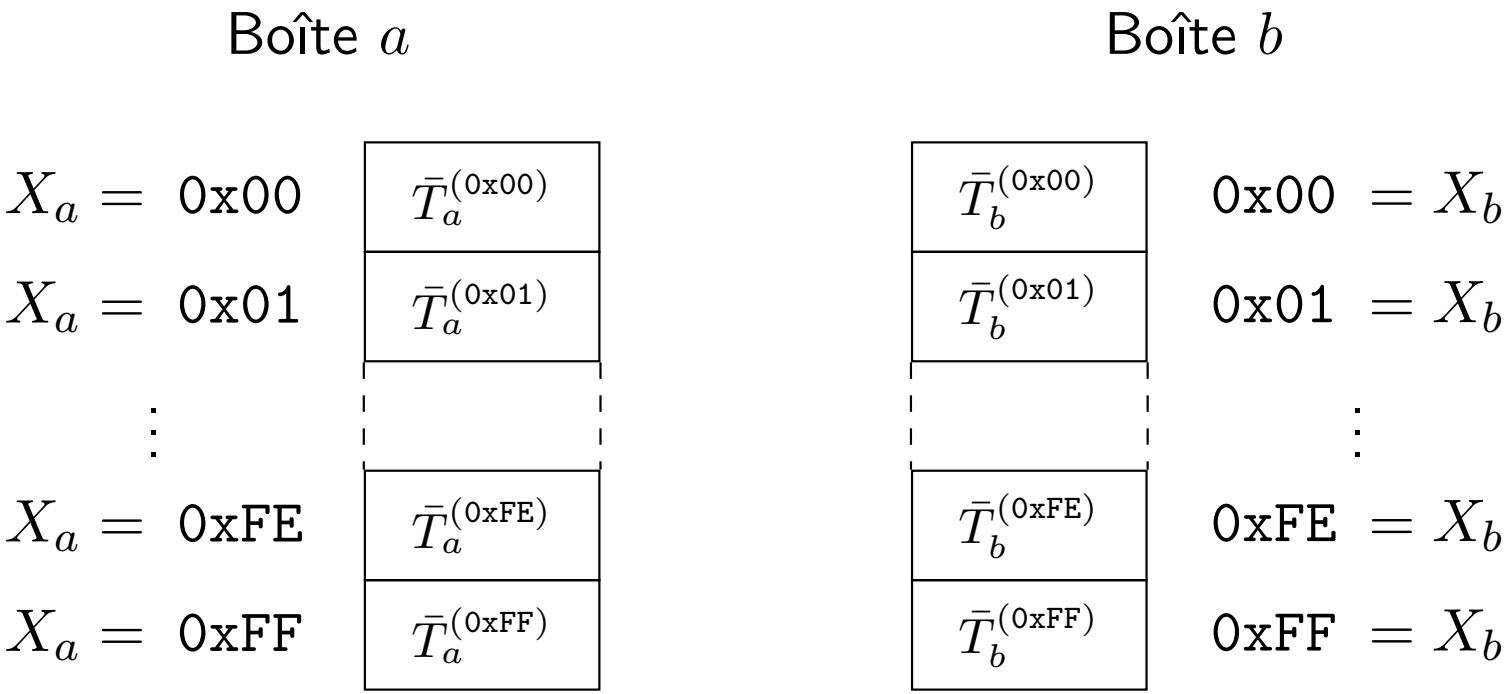
Approche alternative [Moradi et al. 2010]

Pour chaque boîte-S former un vecteur de traces moyennes.



Approche alternative [Moradi et al. 2010]

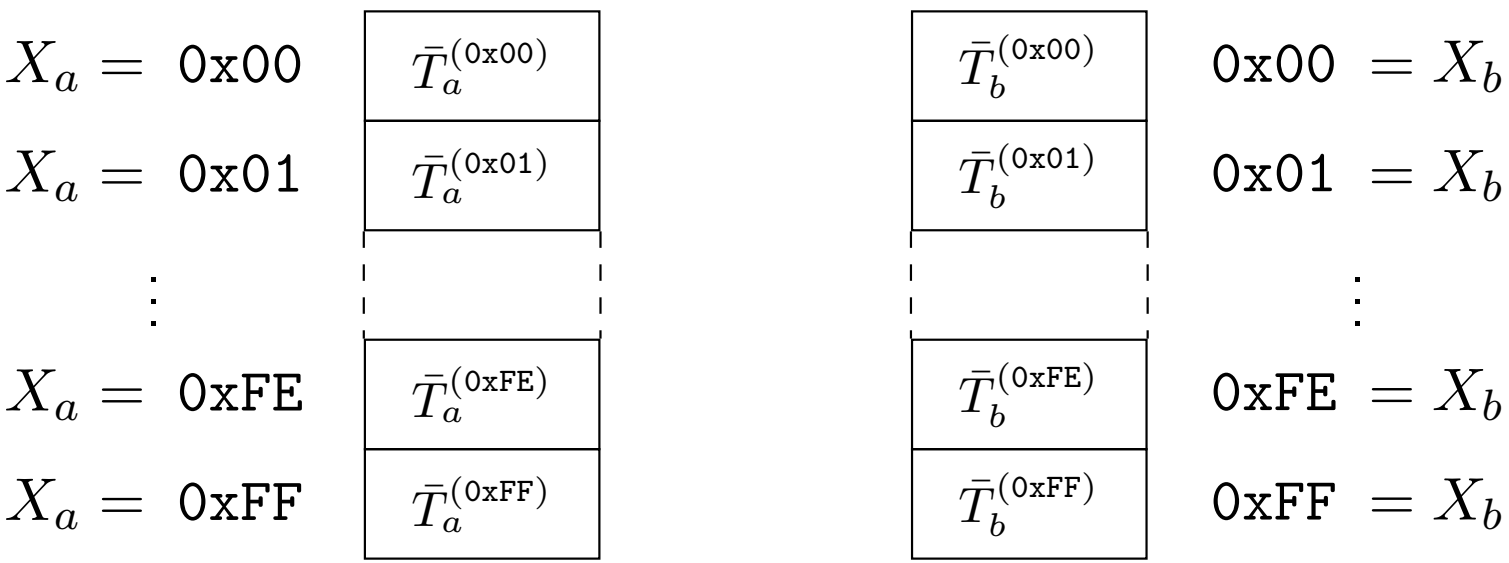
Pour chaque boîte-S former un vecteur de traces moyennes.



Approche alternative [Moradi et al. 2010]

Pour chaque boîte-S former un vecteur de traces moyennes.

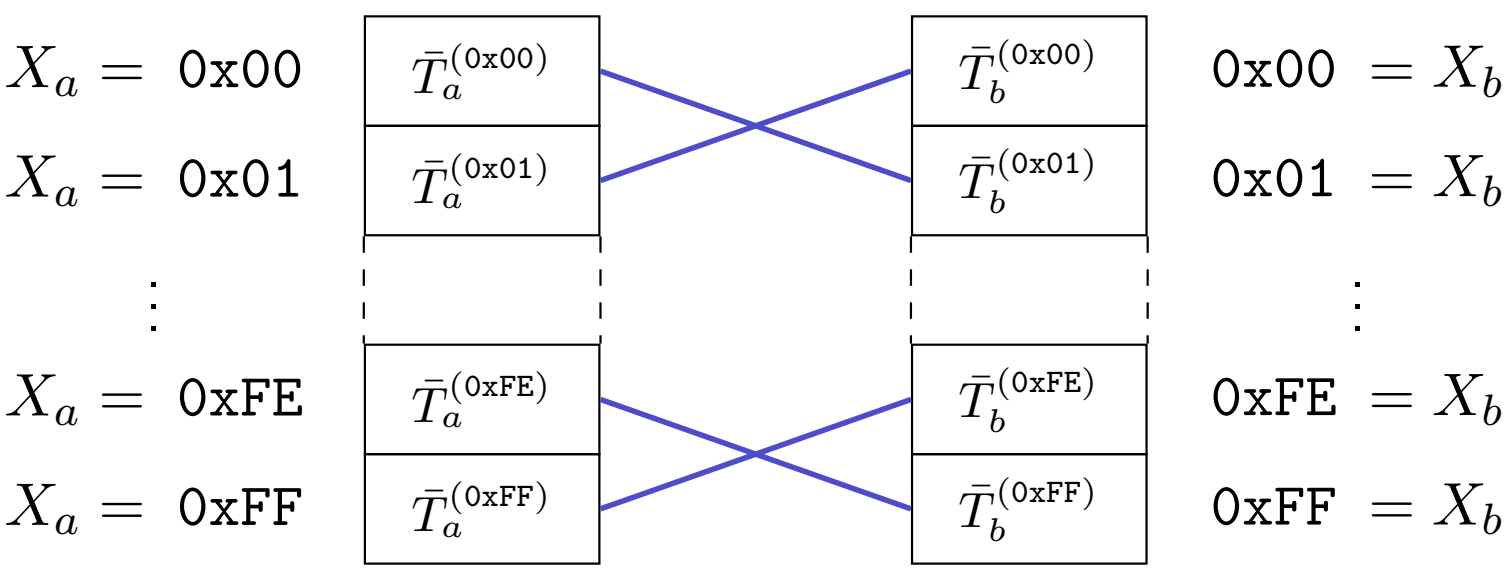
$$K_a \oplus K_b = \texttt{0x01}$$



Approche alternative [Moradi et al. 2010]

Pour chaque boîte-S former un vecteur de traces moyennes.

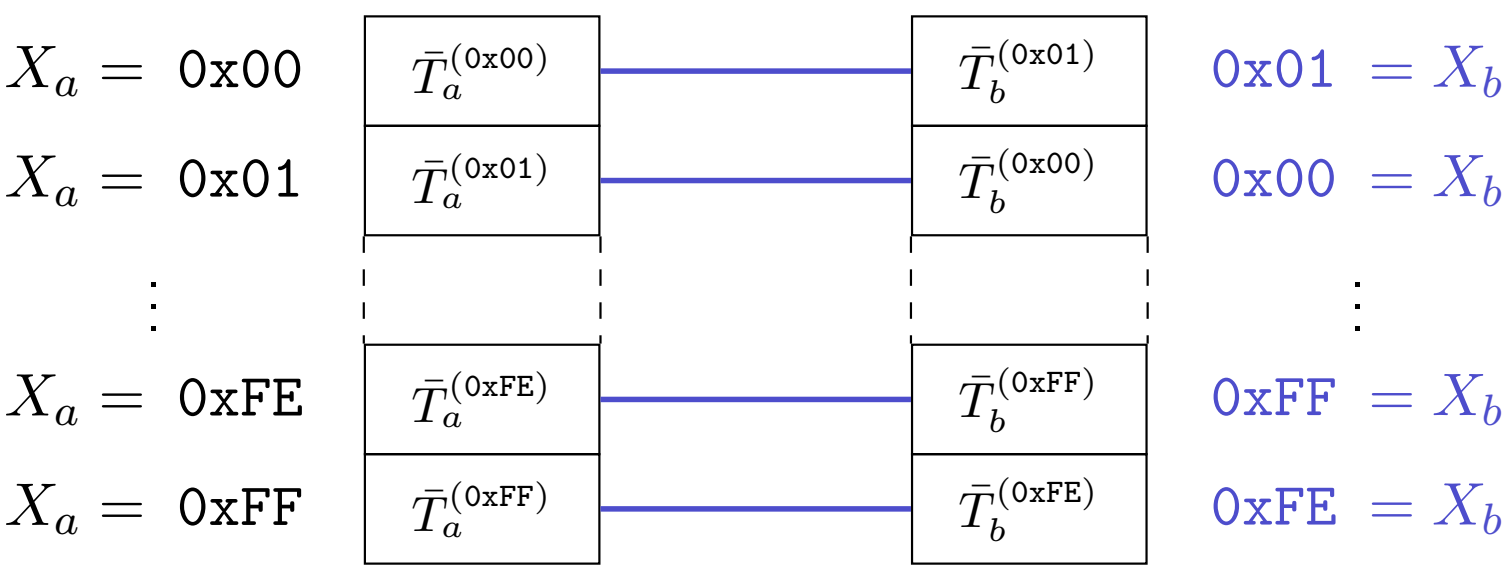
$$K_a \oplus K_b = \texttt{0x01}$$



Approche alternative [Moradi et al. 2010]

Pour chaque boîte-S former un vecteur de traces moyennes.

$$K_a \oplus K_b = \texttt{0x01}$$



Approche alternative [Moradi et al. 2010]

Pour chaque boîte-S former un vecteur de traces moyennes.

Pour chaque valeur δ possible pour $K_a \oplus K_b$:

- permutation du tableau b;
- calcul de la corrélation entre les deux tableaux;
- on garde la valeur ayant le meilleur score.

Plus vraiment de fausses collisions mais toujours des erreurs.

Plan

Attaque par collisions linéaires

Vers une attaque plus “propre”

Résultats expérimentaux

Perspectives

Sous-utilisation de l'information disponible

$$\Delta K_{a,b} \stackrel{\text{def}}{=} K_a \oplus K_b$$

Littérature:

1. donner des scores aux valeurs δ de $\Delta K_{a,b}$;
2. choisir $\delta_{\max}$ avec le meilleur score.

Information perdue:

- Information souple: $\delta_{\max}$ bien meilleur que les autres ?
- Redondance: $\Delta K_{a,b} \oplus \Delta K_{a,c} = \Delta K_{b,c}$.

Attaques par collisions $\Leftrightarrow$ problème de décodage

$$\Delta K_{a,b} \stackrel{\text{def}}{=} K_a \oplus K_b$$

$$\Delta K \stackrel{\text{def}}{=} (\Delta K_{1,2}, \ldots, \Delta K_{15,16})$$

Il y a 120 variables $\Delta K_{a,b} \in \mathbb{F}_{256}$.
Les vecteurs ΔK définissent un s.e.v. de dimension 15 de $\mathbb{F}_{256}^{120}$

$$\forall\, 1 \leq a < b < c \leq 16, \quad \Delta K_{a,b} \oplus \Delta K_{a,c} = \Delta K_{b,c}.$$

On a donc un code linéaire $\mathcal{C}_{\text{AES}}$ de rendement $1/8$:
$\rightarrow$ on devrait pouvoir arriver à corriger quelques erreurs ...

Attaques par collisions $\Leftrightarrow$ problème de décodage

Soit $\Delta \in \mathbb{F}_{256}^{120}$, $\Delta = (\Delta_{1,2}, \ldots, \Delta_{15,16})$, alors

$$\Delta \in \mathcal{C}_{\mathrm{AES}}$$

$$\Updownarrow$$

$$\forall\, 1 \leq a < b < c \leq 16, \quad \Delta_{a,b} \oplus \Delta_{a,c} \oplus \Delta_{b,c} = 0.$$

Matrice de contrôle H_{AES} tq.

$$\Delta \in \mathcal{C}_{\mathrm{AES}} \Leftrightarrow H_{\mathrm{AES}}.{}^{t}\Delta = 0$$

Attaque pas collision $\Leftrightarrow$ décodage LDPC

Matrice de contrôle H_{AES} formée par 105 relations parmi les 560

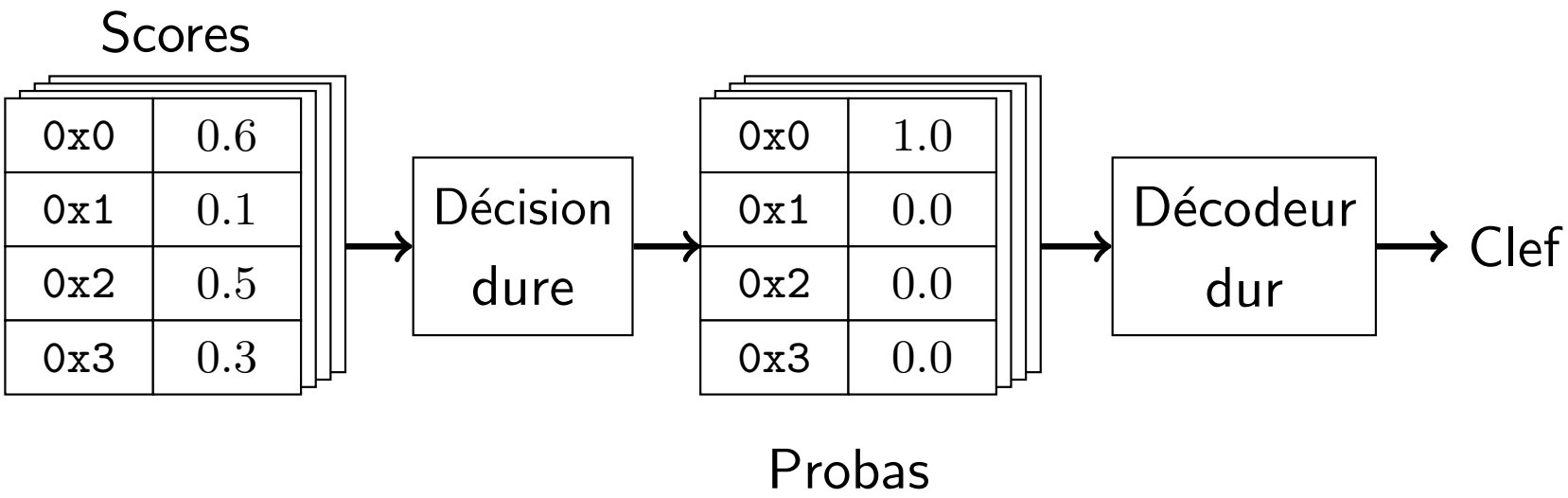
$$\forall\, 1 \leq a < b < c \leq 16, \quad \Delta_{a,b} \oplus \Delta_{a,c} \oplus \Delta_{b,c} = 0.$$

$\Rightarrow$ matrice de contrôle creuse avec trois 1 par ligne.

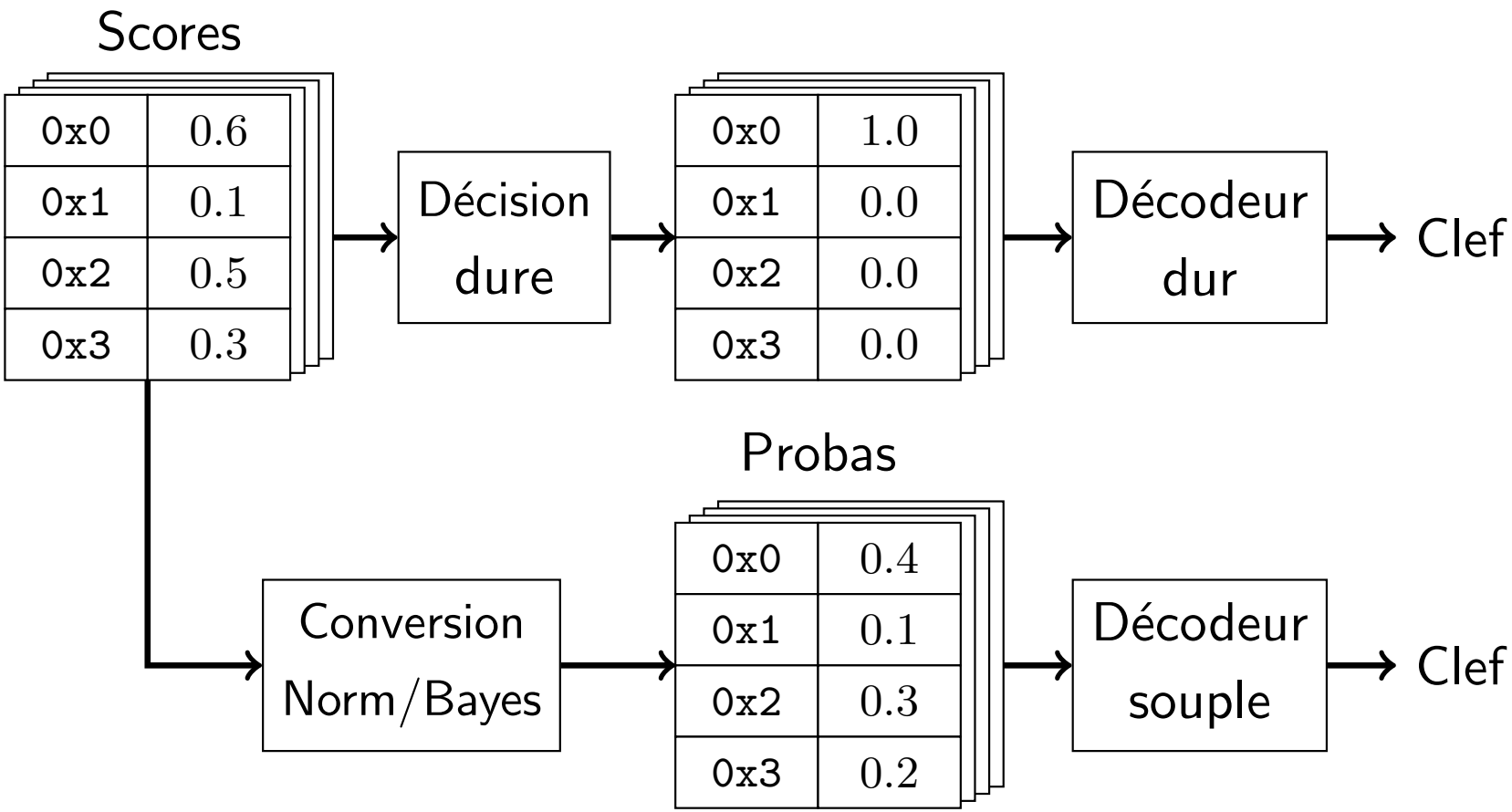
Codes LDPC: *Low Density Parity Check*

Codes LDPC: codes avec matrice de contrôle creuse.
Ici cas non binaire: corps de base $\mathbb{F}_{256}$.

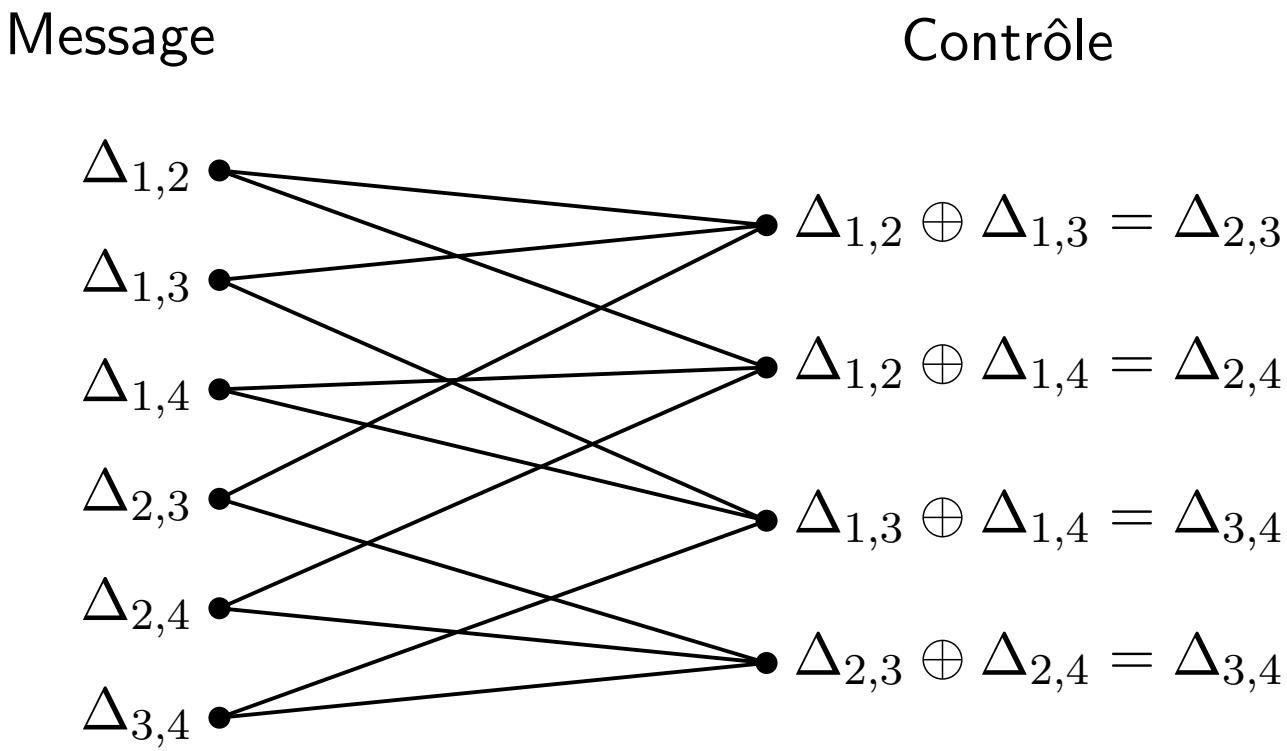
Décodage dur vs décodage souple



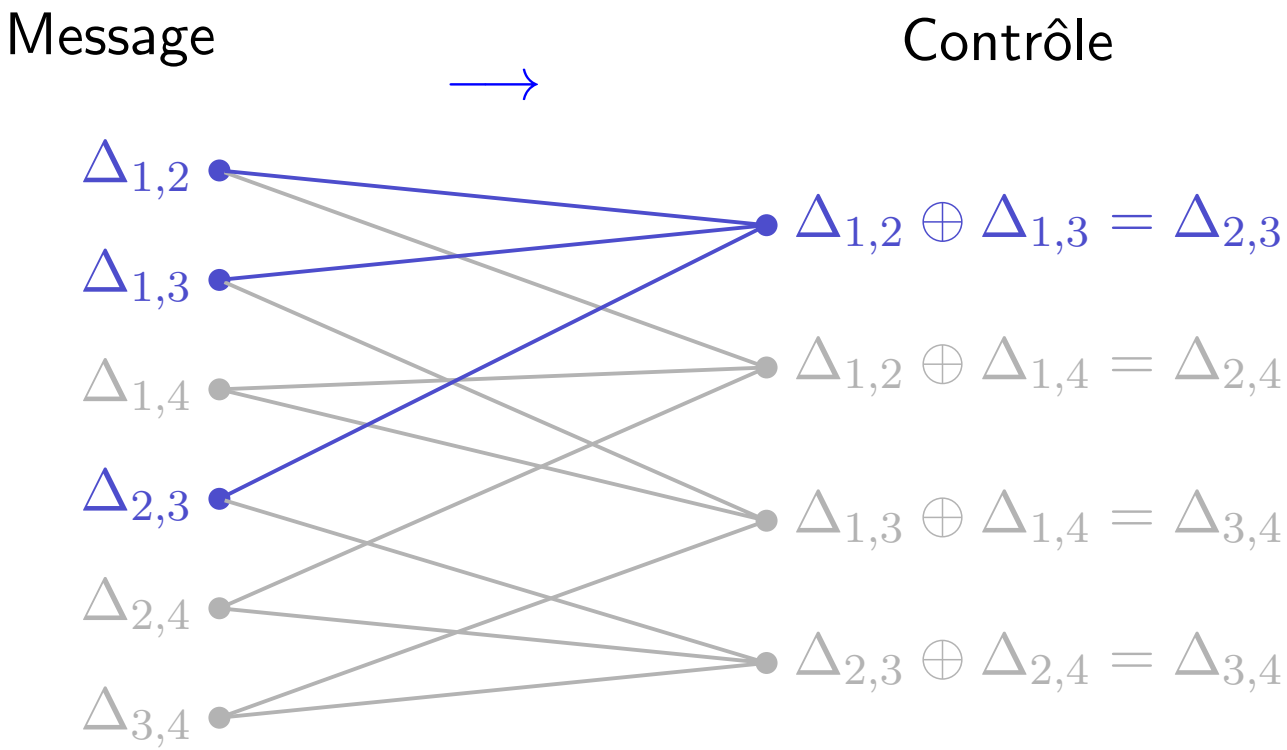
Décodage dur vs décodage souple



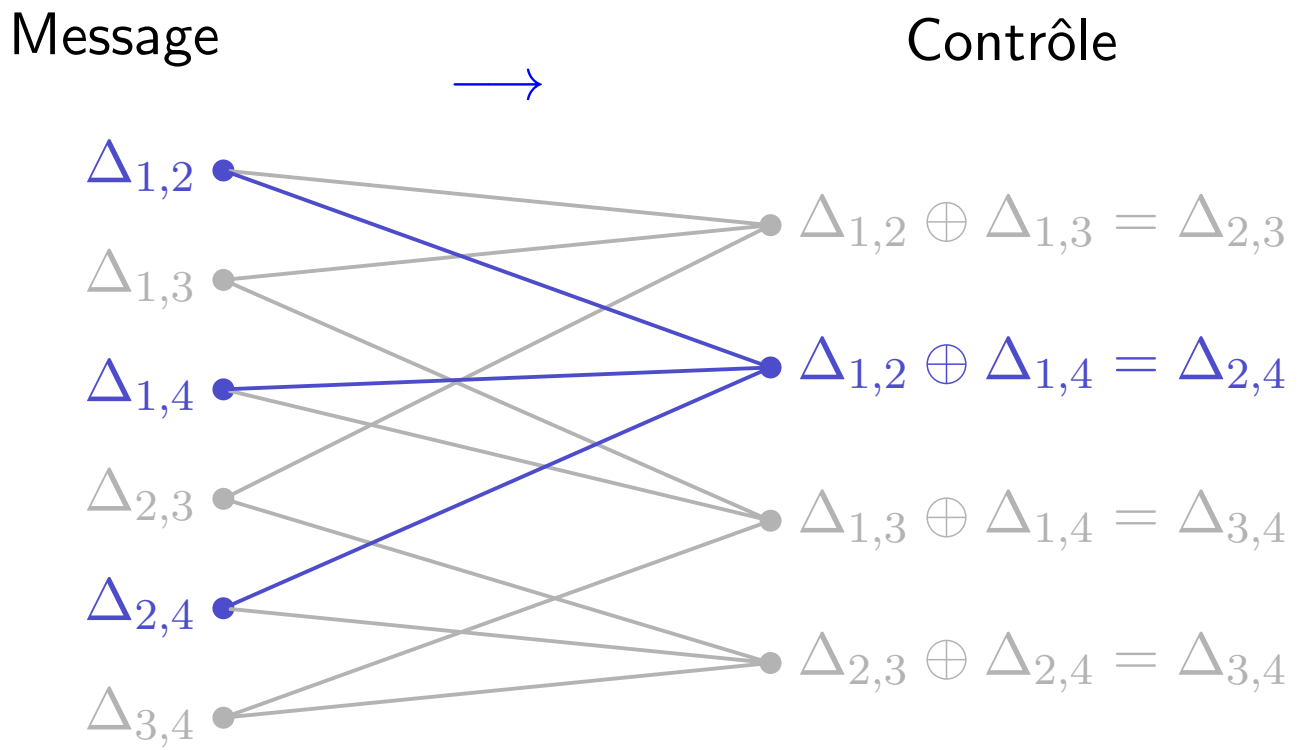
Décodage par propagation de croyance



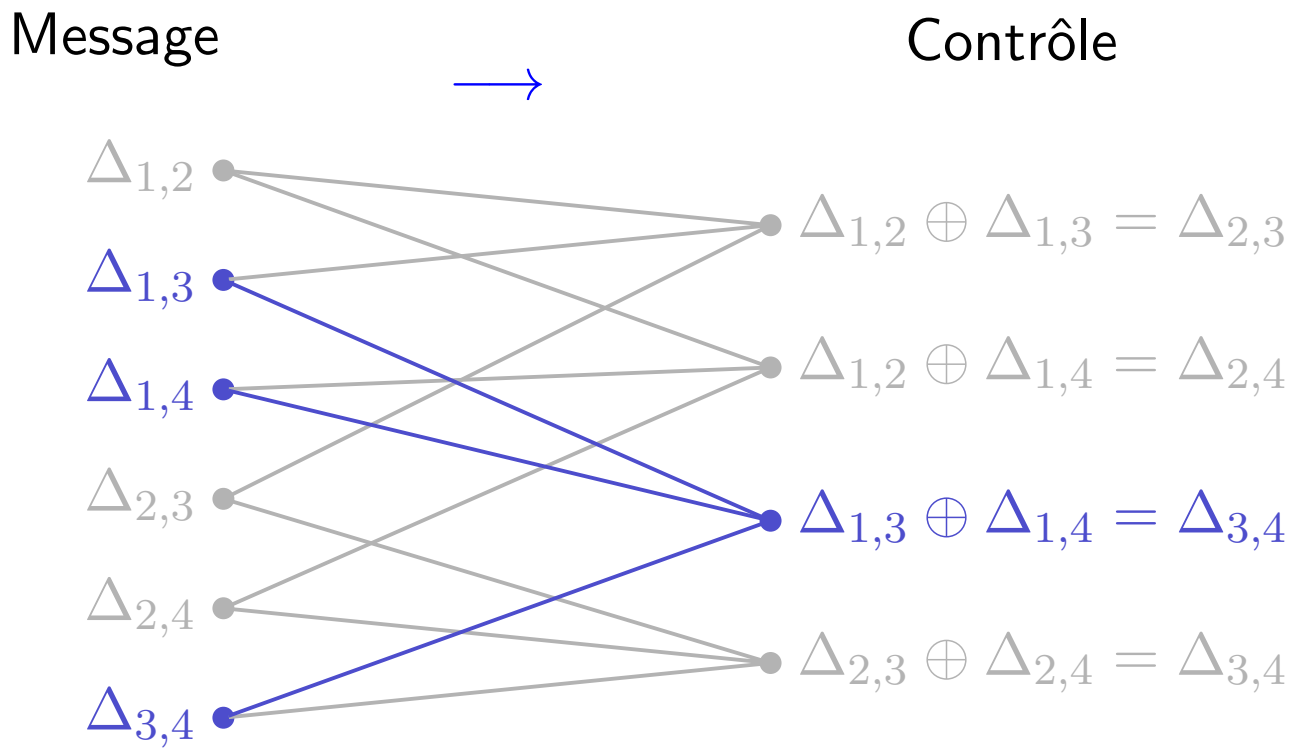
Décodage par propagation de croyance



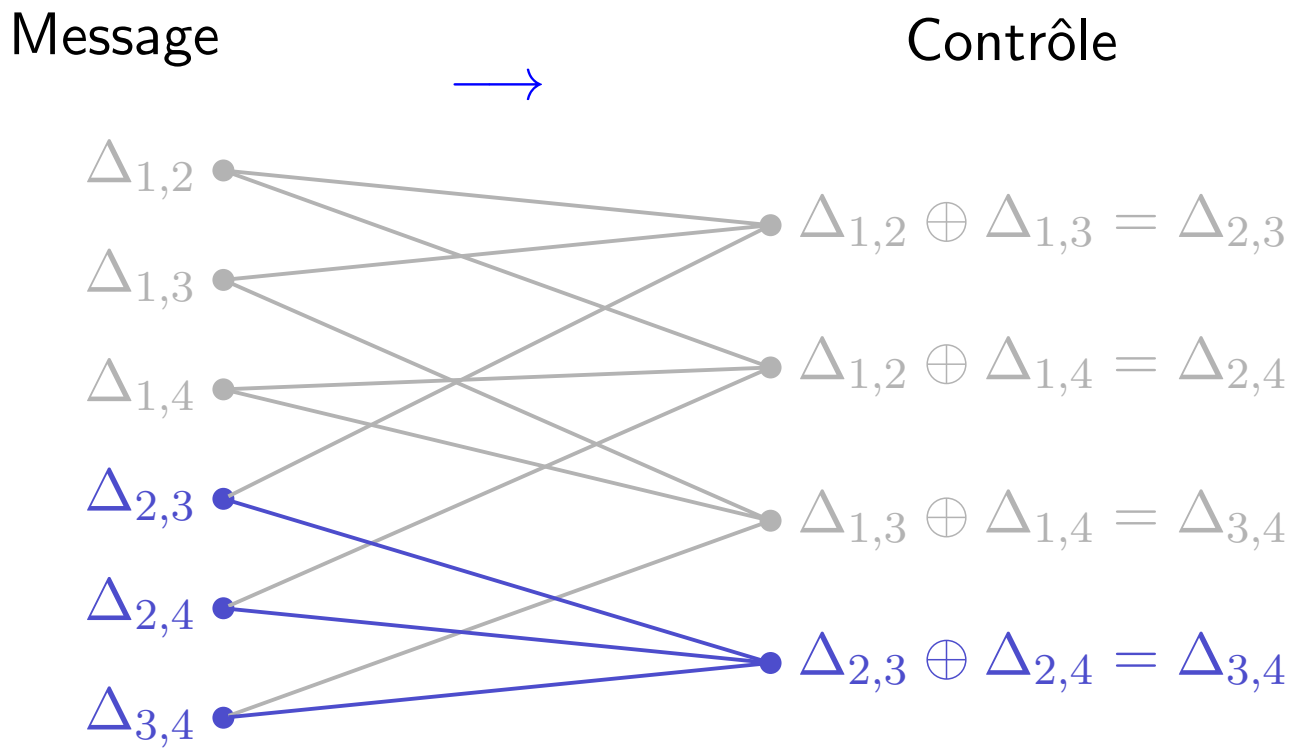
Décodage par propagation de croyance



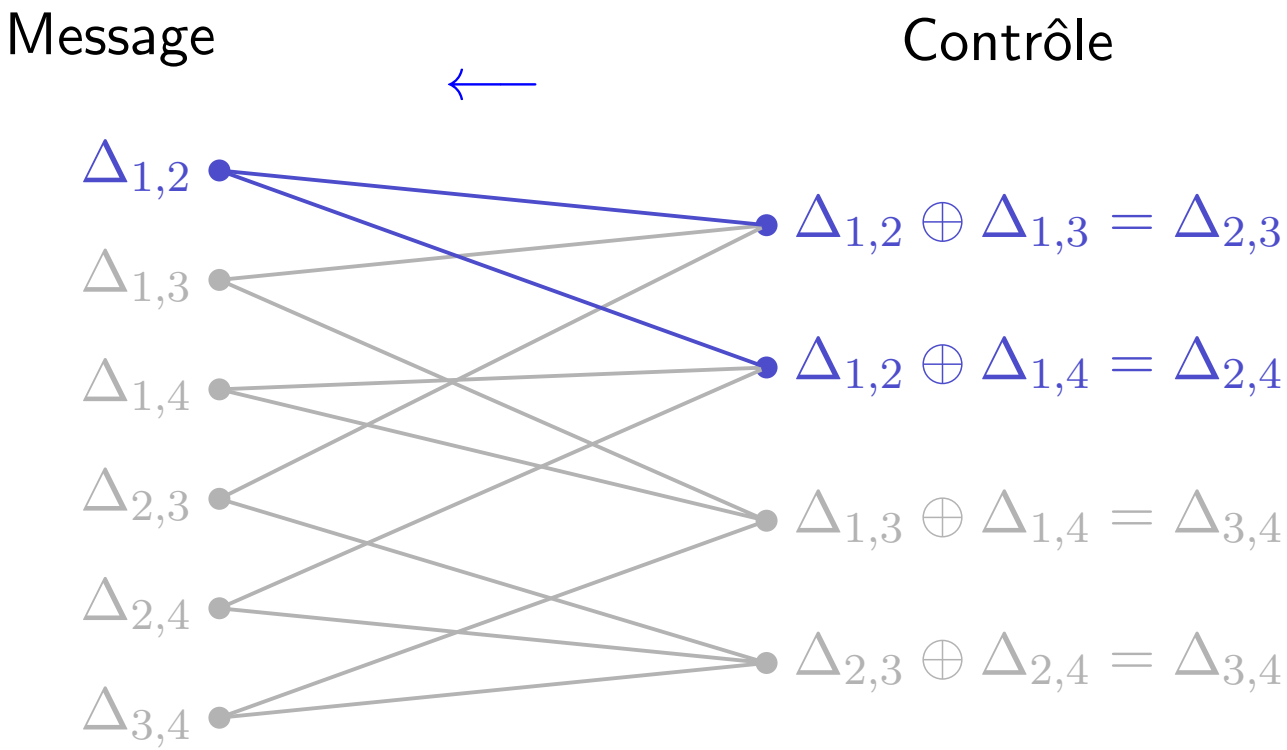
Décodage par propagation de croyance



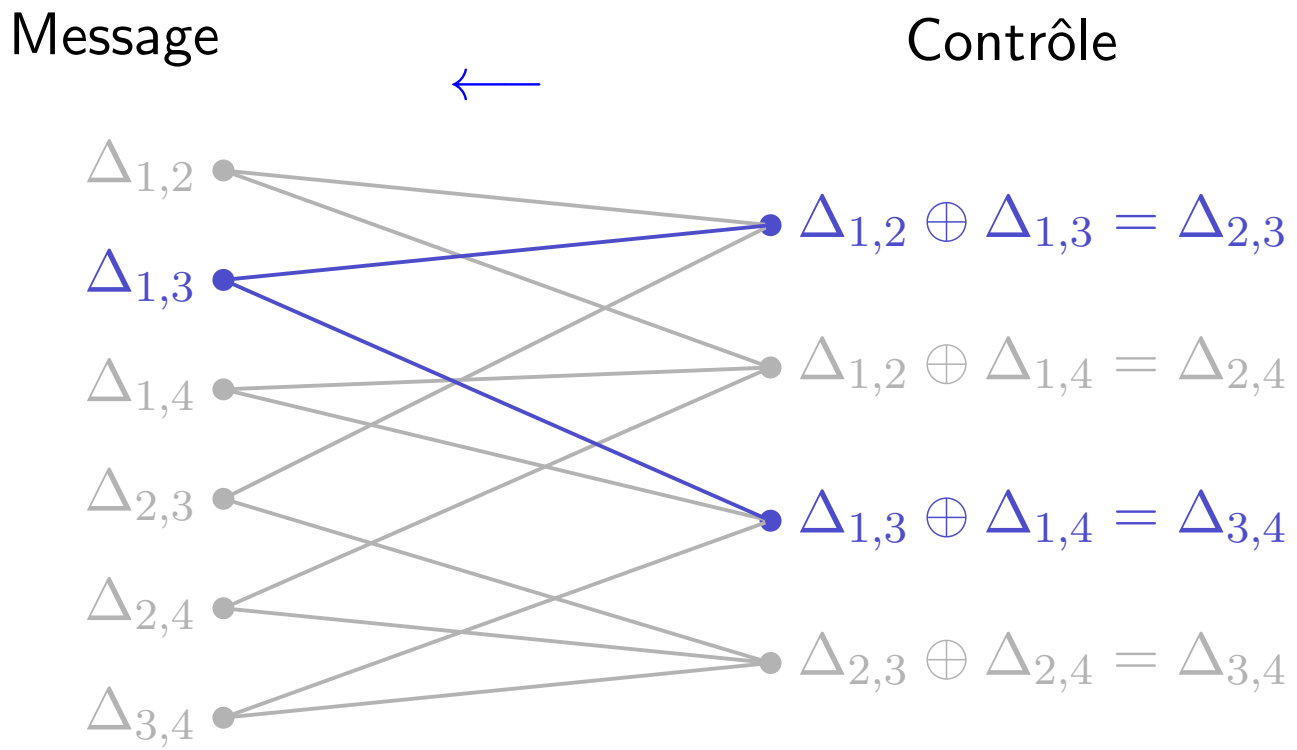
Décodage par propagation de croyance



Décodage par propagation de croyance

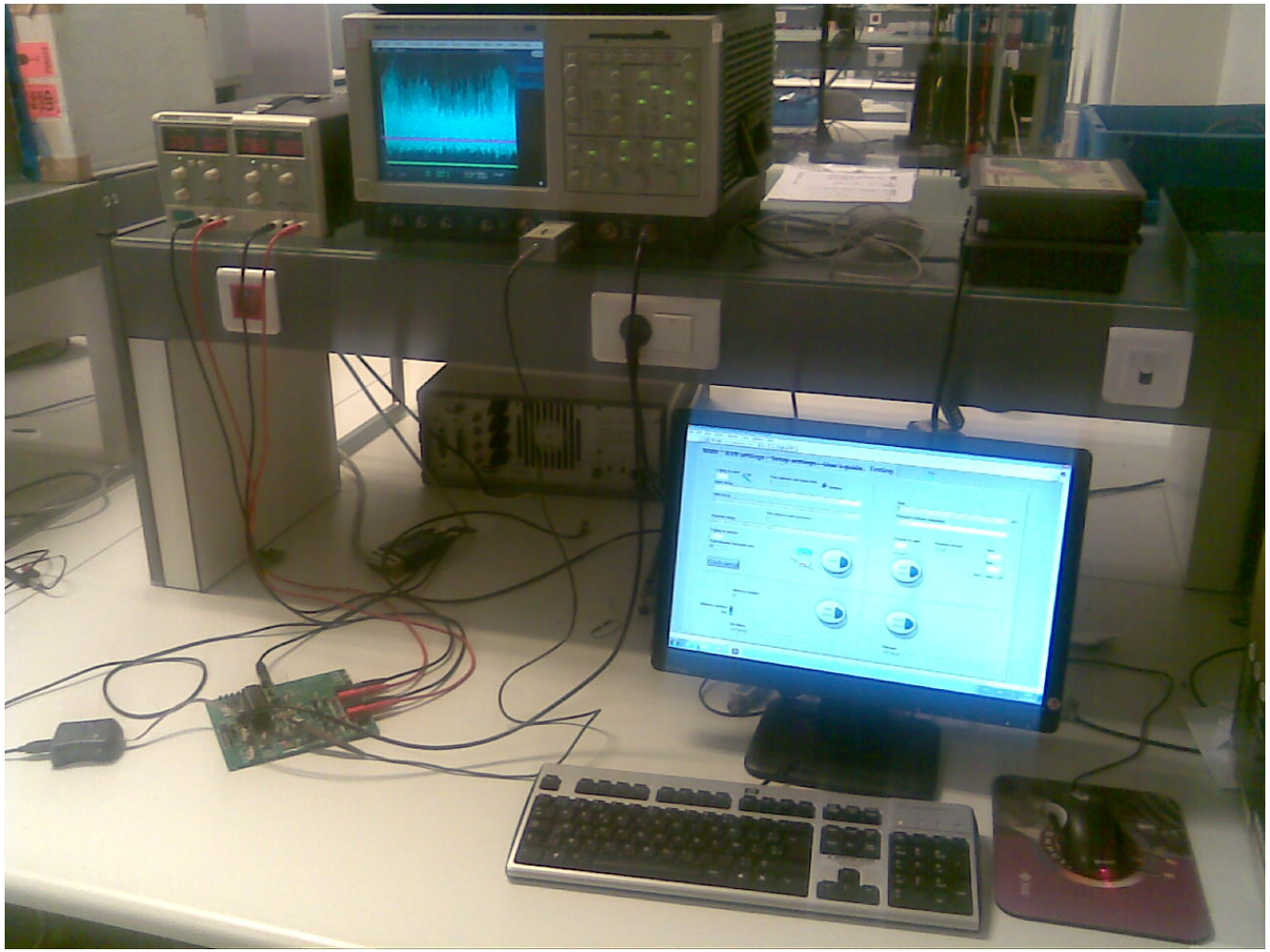


Décodage par propagation de croyance

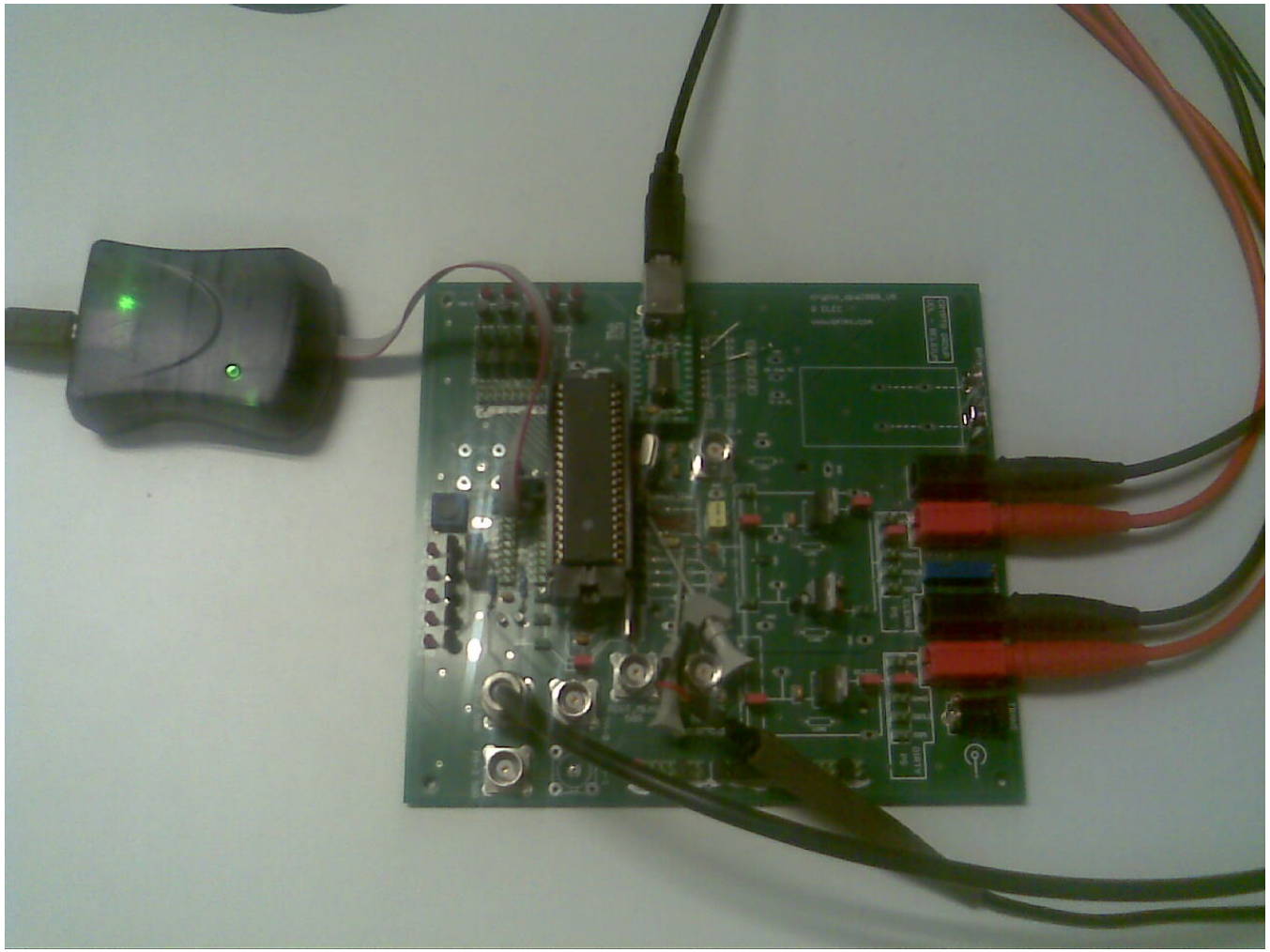


Plan

Conditions expérimentales



Conditions expérimentales



AES SubBytes

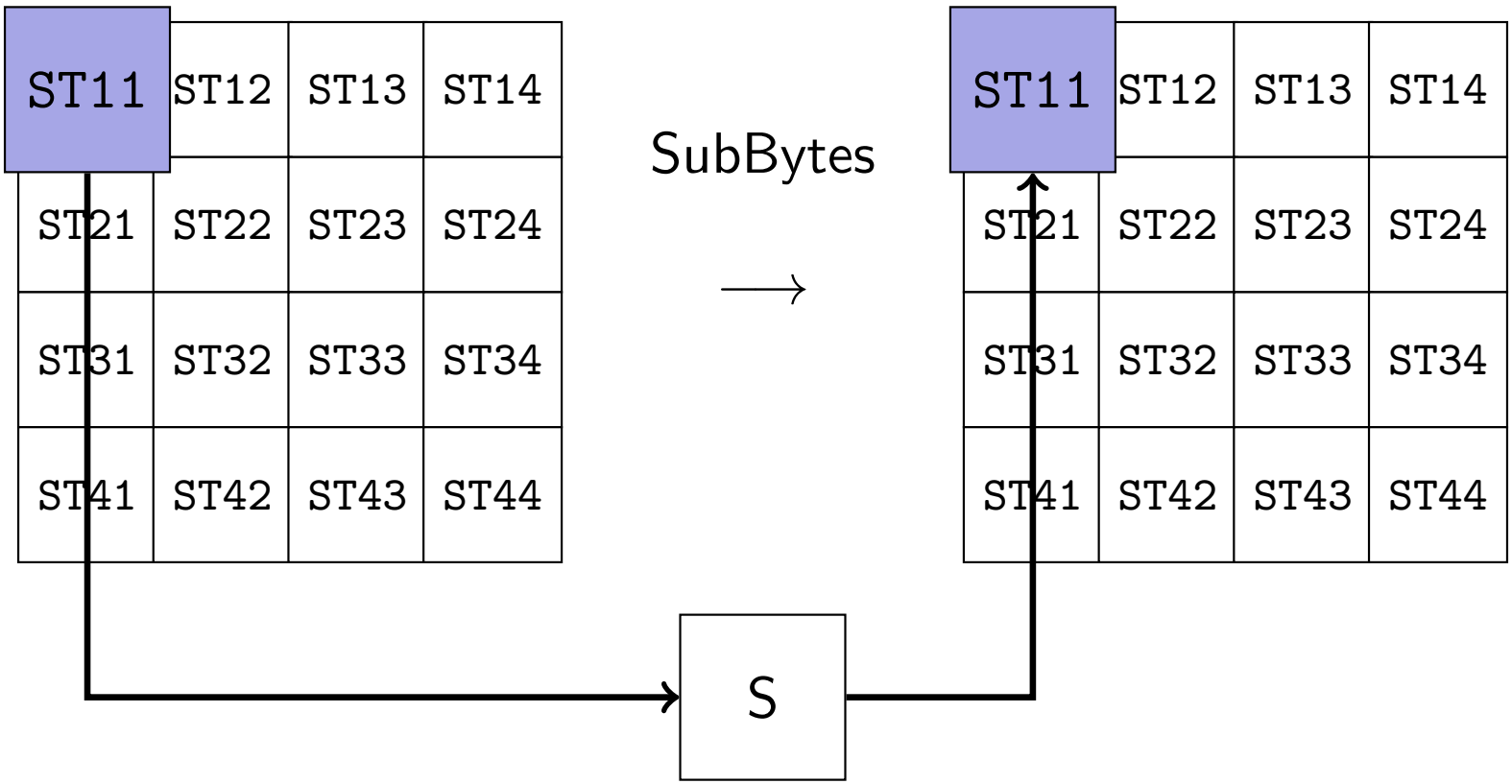
ST11	ST12	ST13	ST14
ST21	ST22	ST23	ST24
ST31	ST32	ST33	ST34
ST41	ST42	ST43	ST44

SubBytes

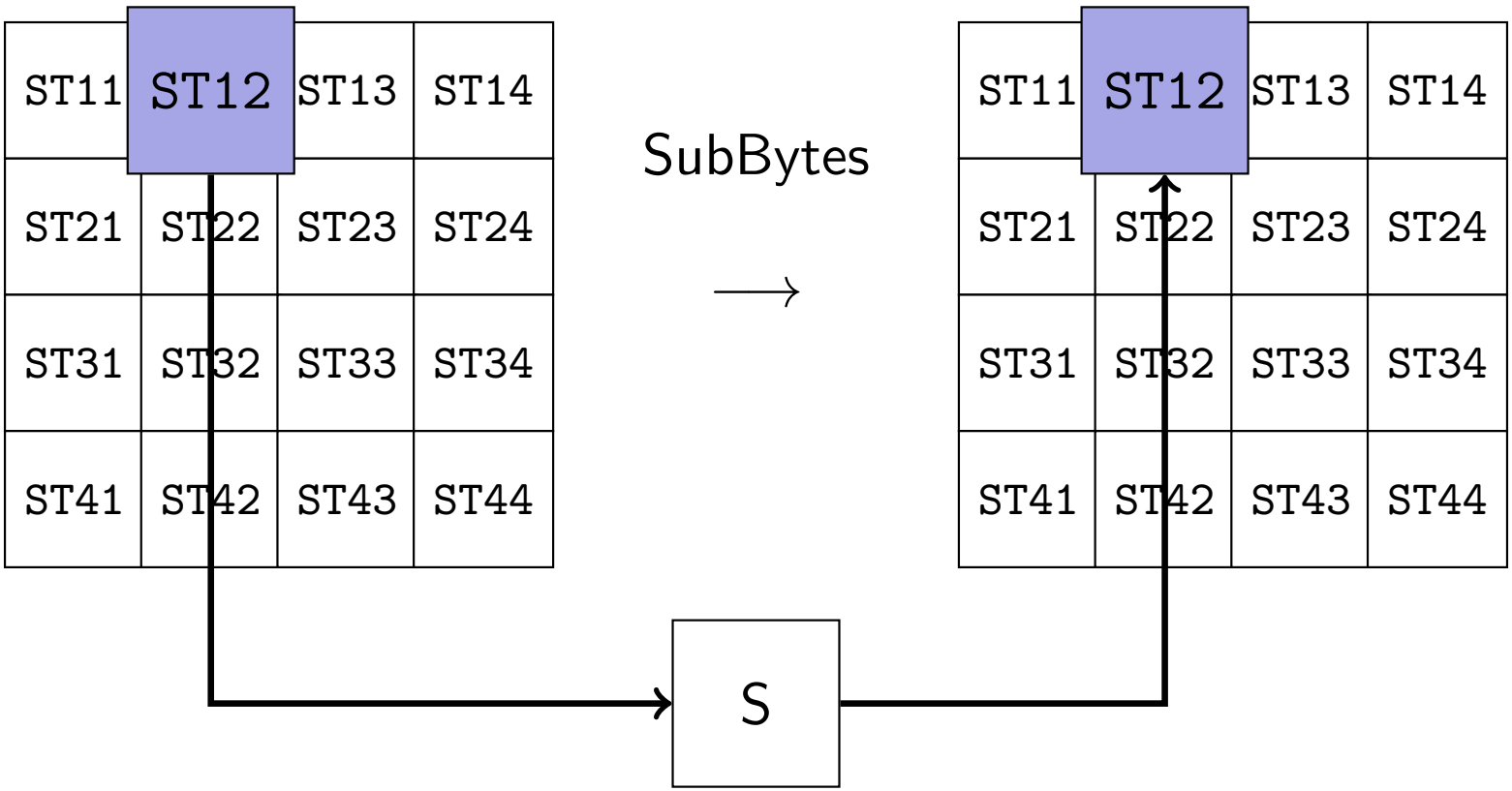
$\longrightarrow$

ST11	ST12	ST13	ST14
ST21	ST22	ST23	ST24
ST31	ST32	ST33	ST34
ST41	ST42	ST43	ST44

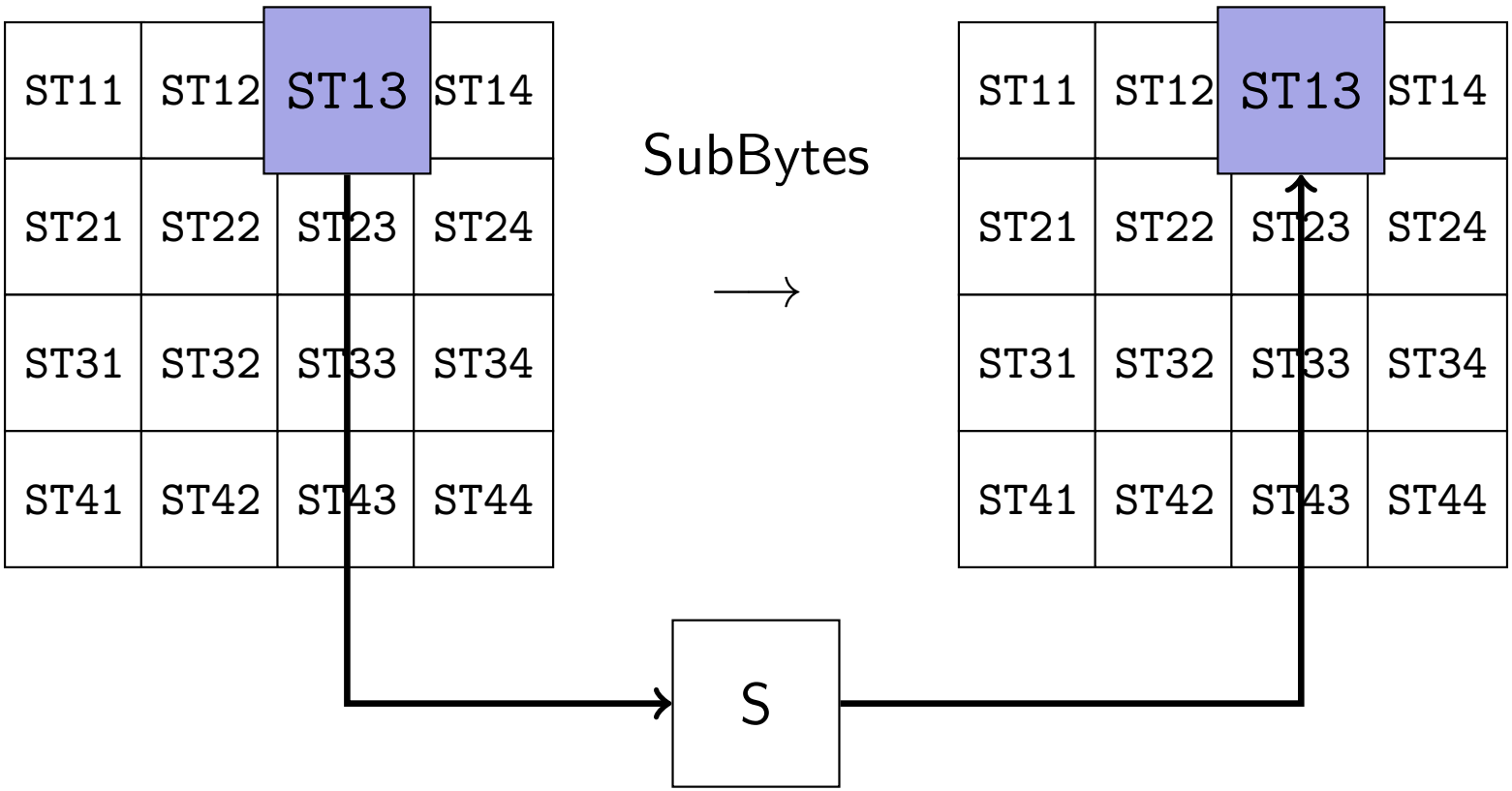
AES SubBytes



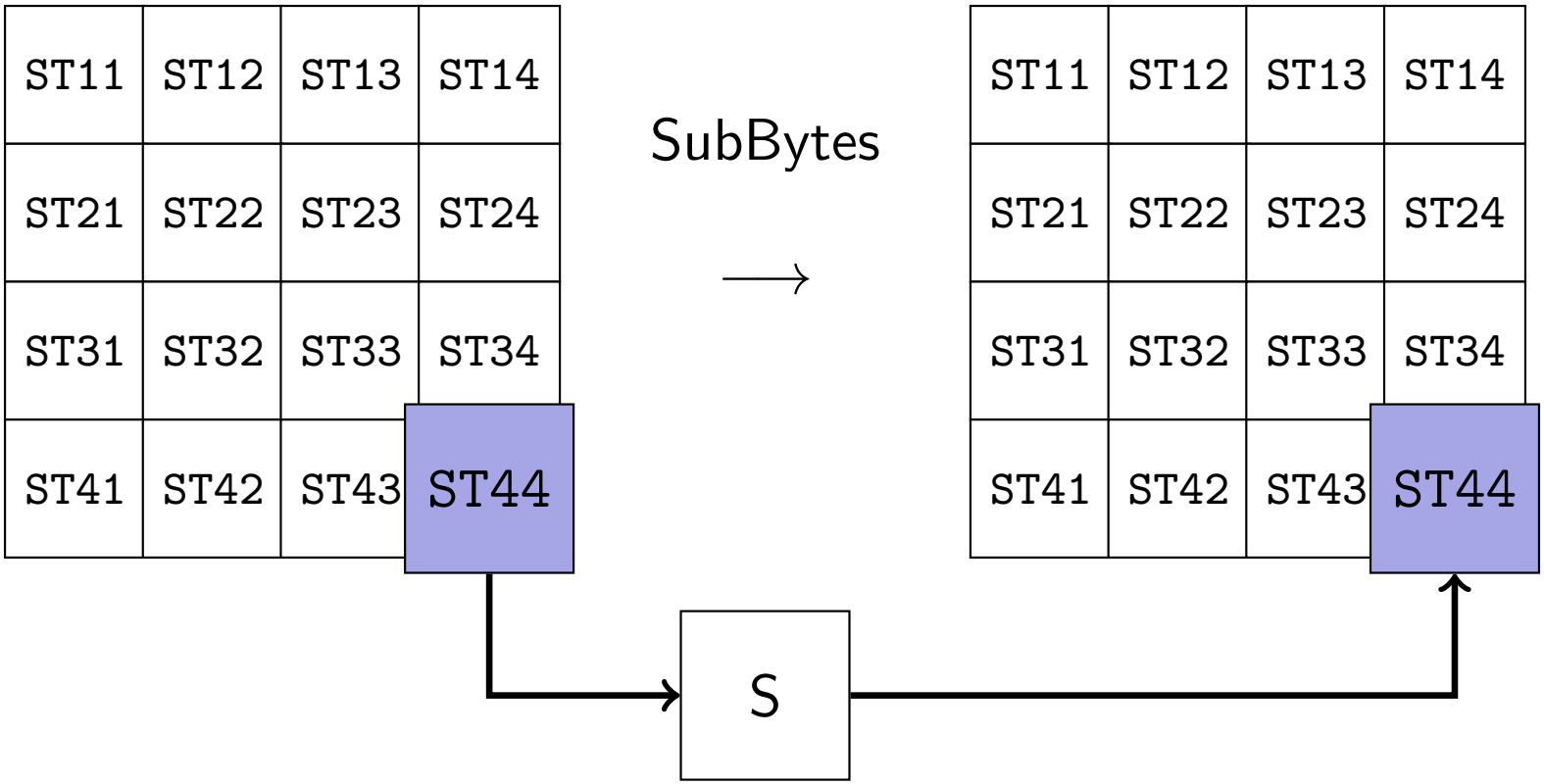
AES SubBytes



AES SubBytes



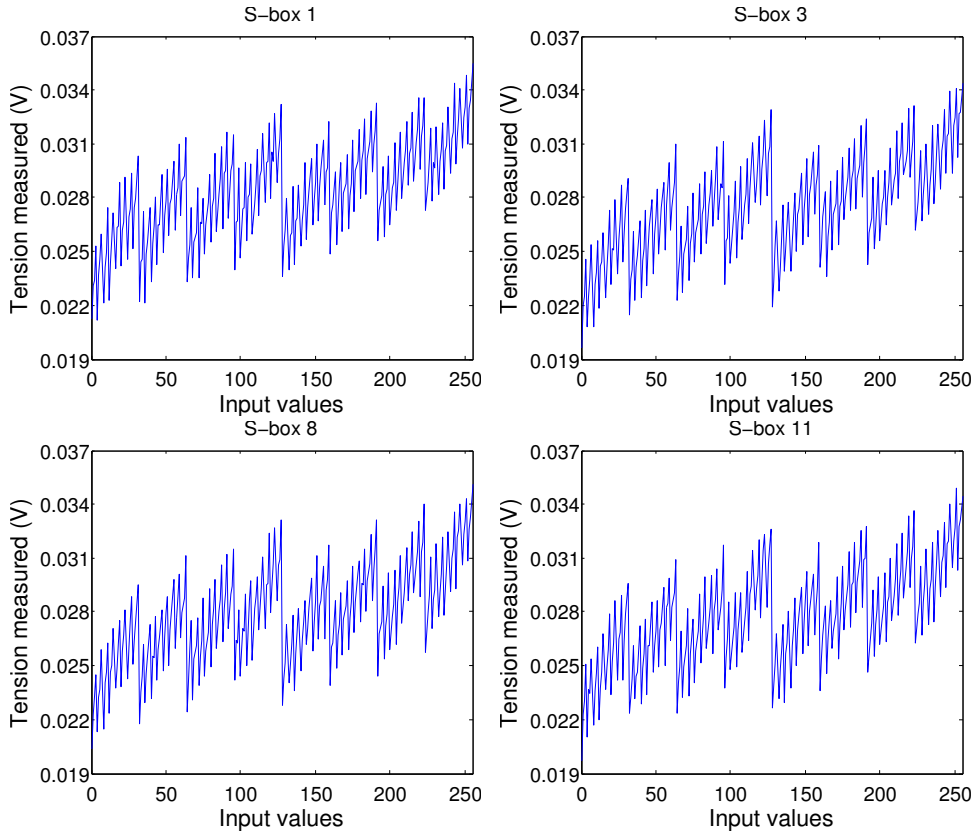
AES SubBytes



Implémentation de référence

Référence

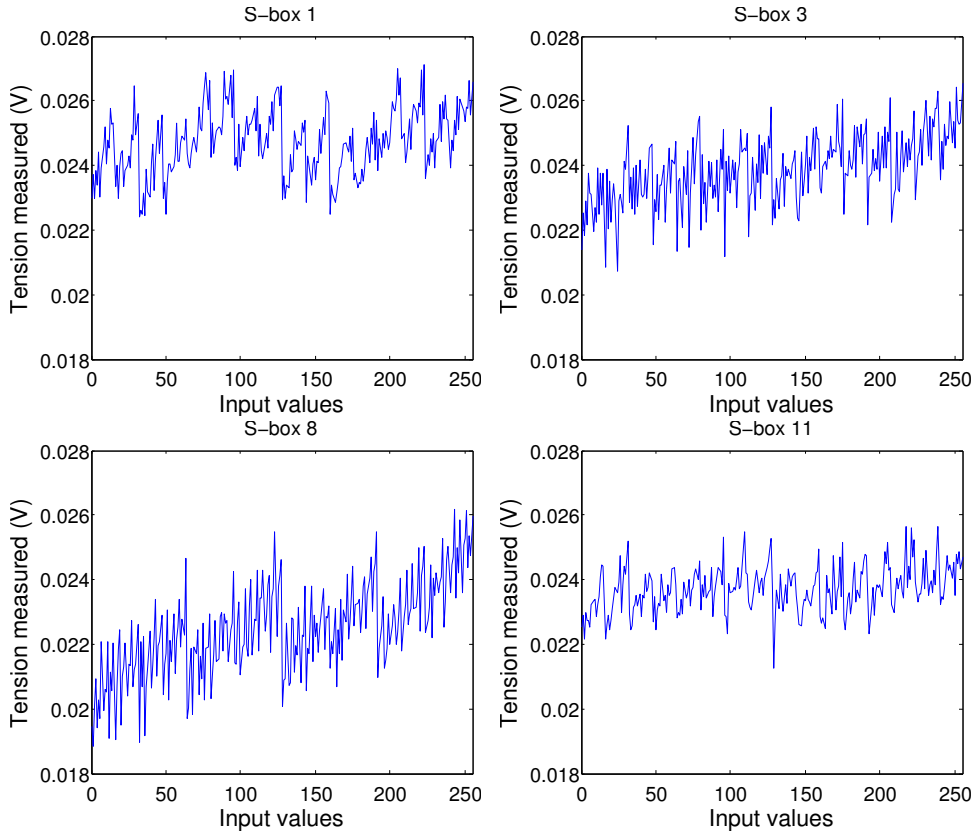
```
mov SR, STxy
mov ZL, SR
lpm SR, Z
mov STxy,SR
```



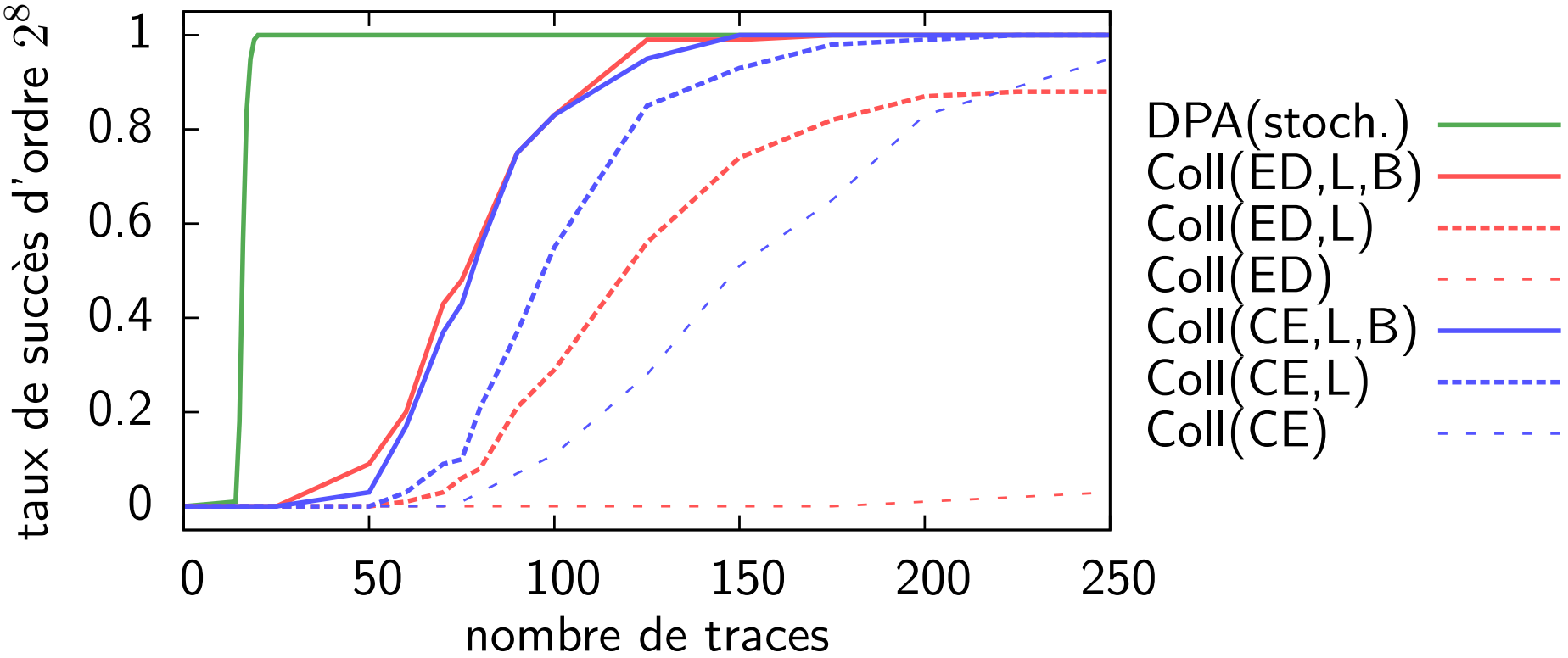
Implémentation furious

Furious

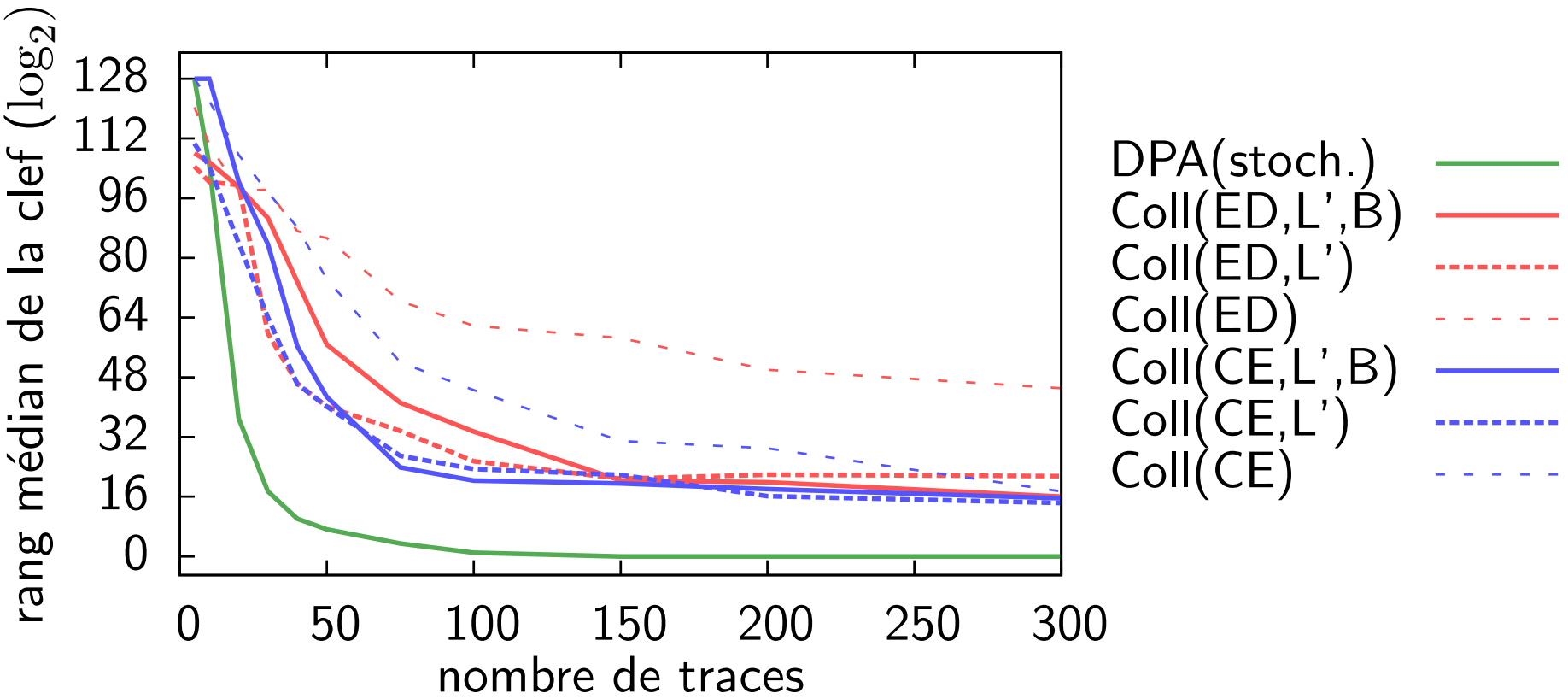
```
mov H1, ST21
mov ZL, ST22
lpm ST21, Z
mov ZL, ST23
lpm ST22, Z
```



Attaque sur l'implémentation de référence

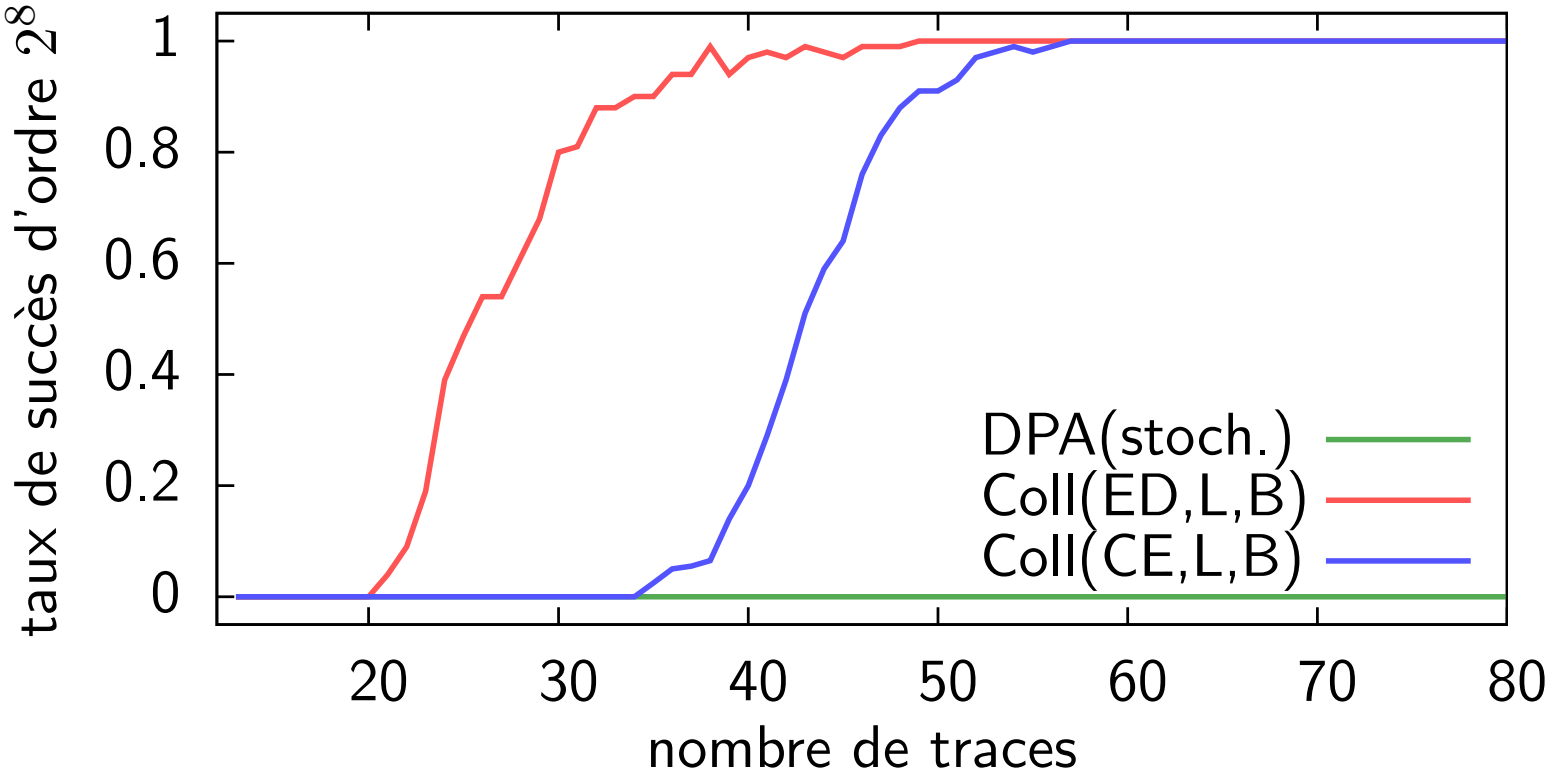


Attaque sur l'implémentation furious



Intérêt théorique des collisions

Cas hypothétique de fonctions de fuite non linéaires.



Plan

Attaque par collisions linéaires

Vers une attaque plus “propre”

Résultats expérimentaux

Perspectives

Conclusions

- Lien entre collisions linéaires et décodage LDPC.
- Exploitation “optimale” de l’information.
- Hypothèse de fuites identiques remise en question.
- Collisions moins performantes que DPA pour des microprocesseurs standards.
- Existence de contextes où seule les attaques par collisions fonctionnent?

Perspectives

- Proposer un algorithme de décodage en liste performant pour cette application.
- Extrapolation des performances de l'attaque à partir de l'information mutuelle observée entre fuite et variable intermédiaire.
- Adaptation à des implémentations protégées (masquage).
- Cas de collisions non-linéaires.