Analyse de primitives cryptographiques récentes

Soutenance de doctorat de Brice Minaud

Directeur de thèse :	Pierre-Alain Fouque	
Rapporteurs :	Henri Gilbert	Louis Goubin
Examinateurs :	Anne Canteaut Antoine Joux David Pointcheval	Jean- Sébastien Coron Reynald Lercier

Rennes, 7 octobre 2016

Plan

Introduction.

1. Cryptanalyse de Robin, iSCREAM et Zorro.

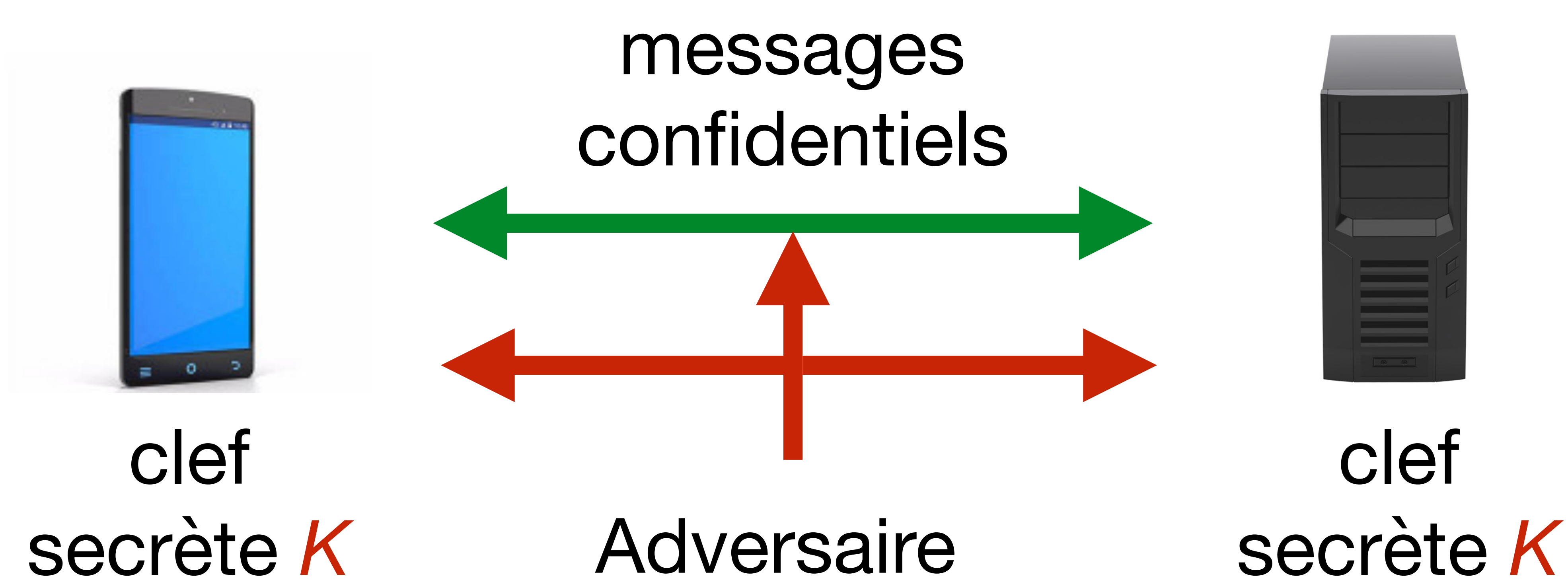
2. Cryptanalyse structurelle d'ASASA.

3. Cryptanalyse de l'application multilinéaire CLT15.

Introduction

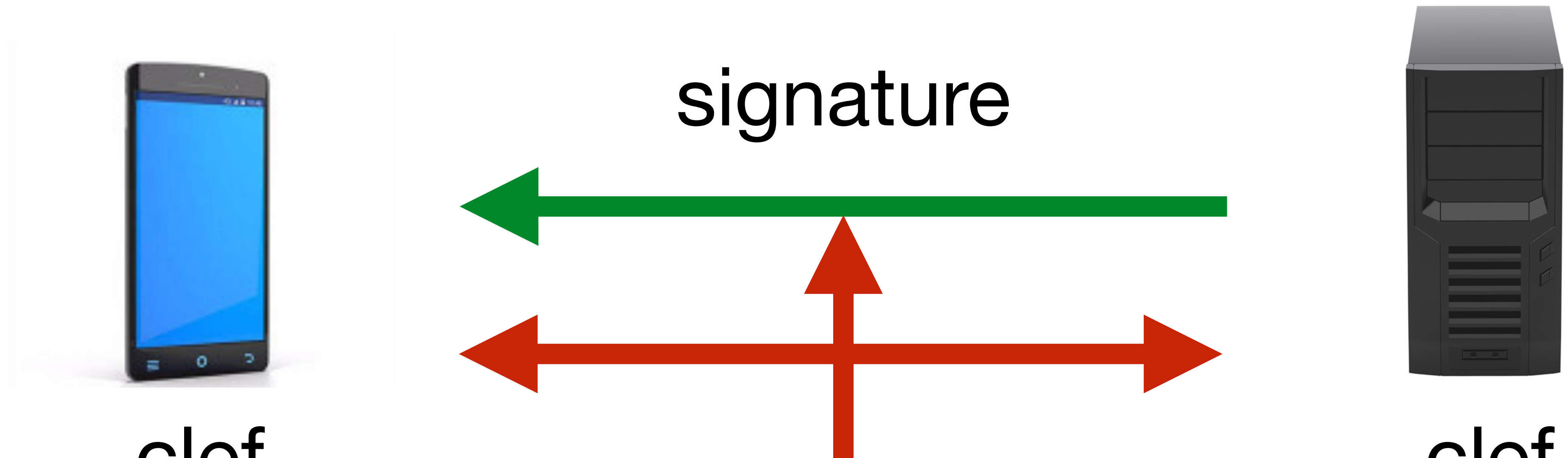
Cryptographie

Principalement : conception et analyse de communications (informatiques) sécurisées.



L'adversaire n'apprend rien du contenu des messages.

Cryptographie asymétrique



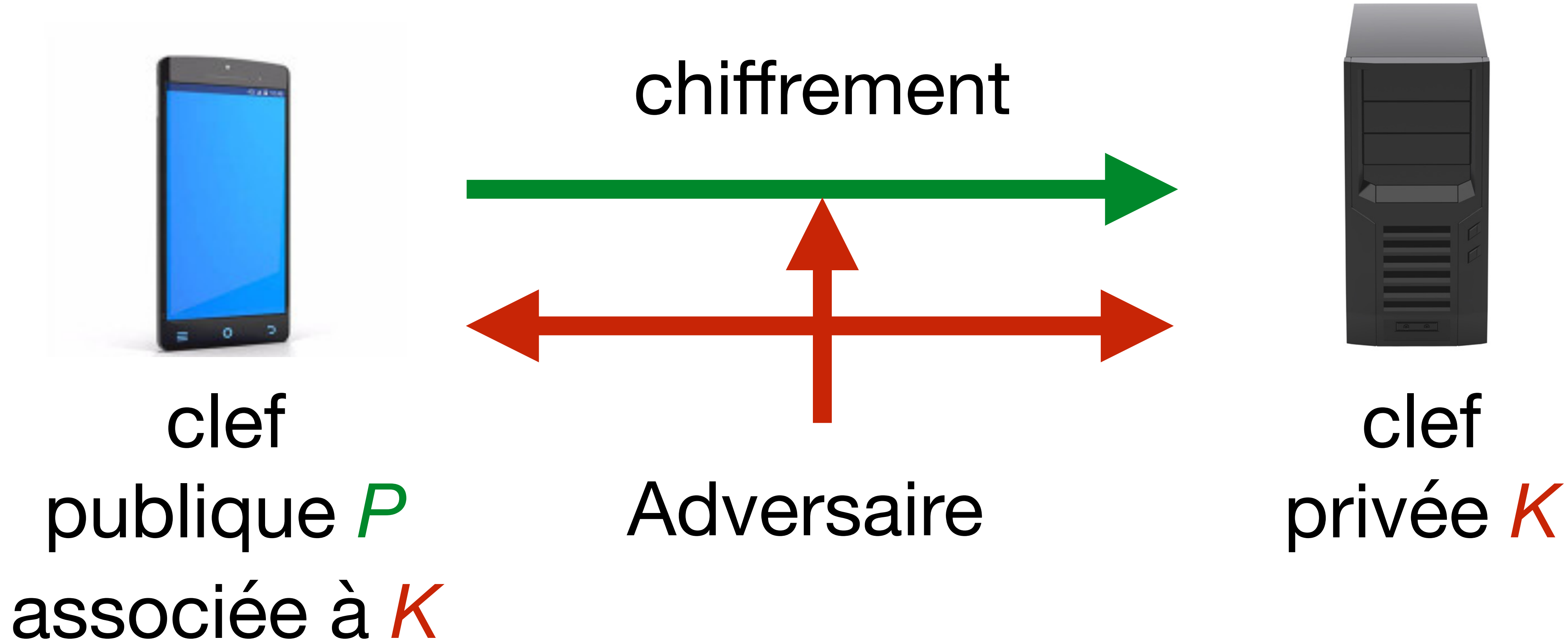
clef publique P associée à K

Adversaire

clef privée K

Notion asymétrique : signature électronique.

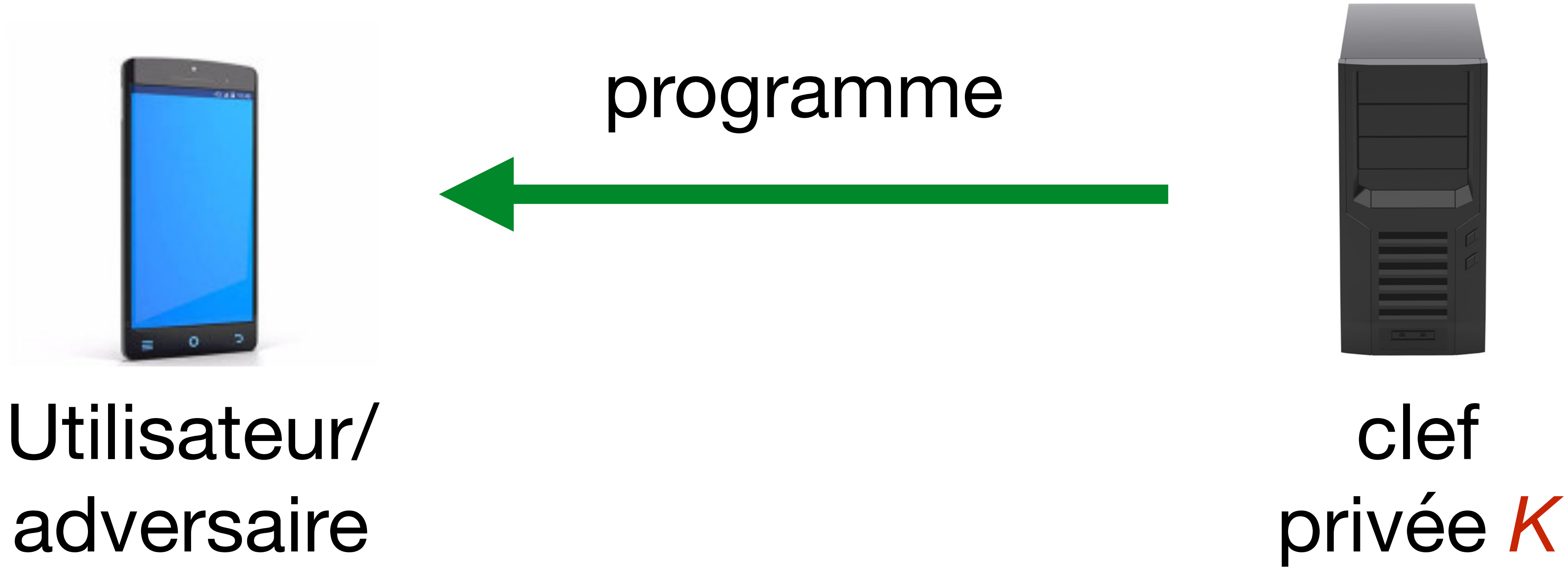
Cryptographie asymétrique



Notion asymétrique : signature électronique.

Notion «duale» : chiffrement à clef publique.

Obfuscation



Autre exemple de notion cryptographique : l'obfuscation.

L'utilisateur peut exécuter un programme sans rien apprendre de sont fonctionnement interne.

Cryptanalyse

La **cryptanalyse** consiste à étudier la sécurité des constructions précédentes.

Concrètement : chercher des attaques.

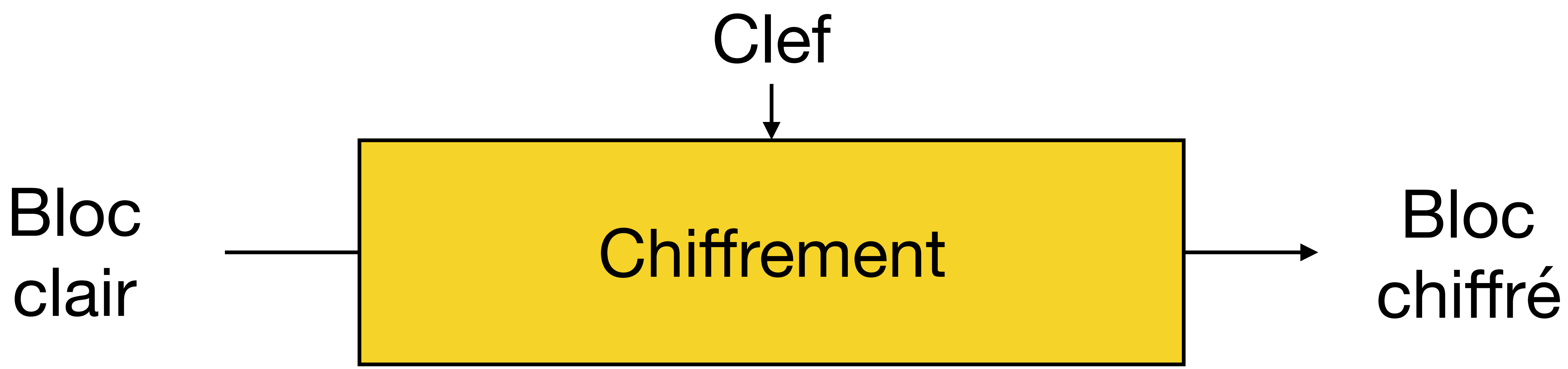
A plus long terme : améliore la compréhension des constructions.

1e partie

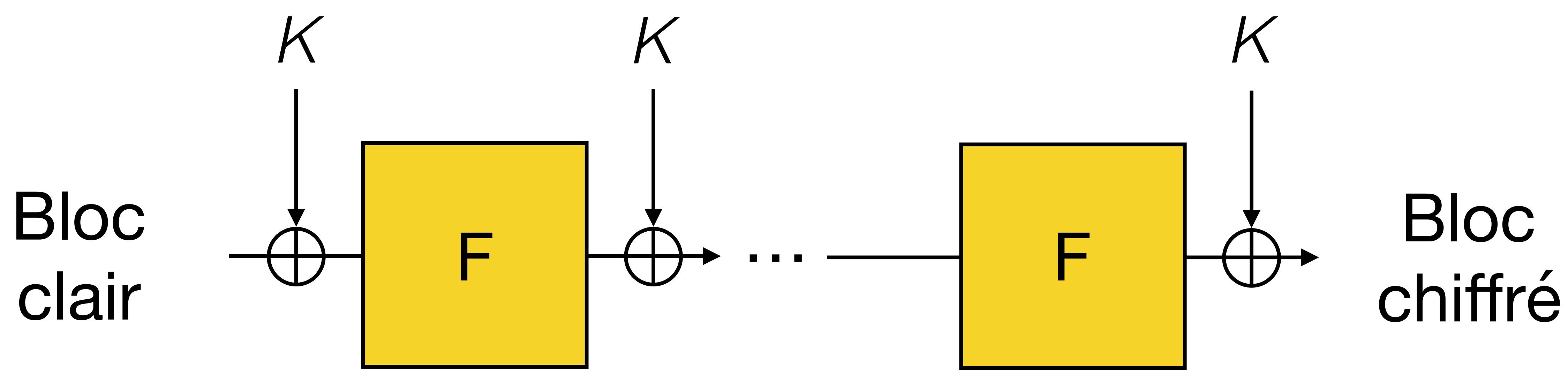
Cryptanalyse de Robin, iSCREAM et Zorro

Attaque par sous-espaces invariants

Chiffrements par bloc



Chiffrements par bloc



Certains chiffrements légers n'ont pas de cadencement de clef.

Noekeon (NESSIE 2000), LED (CHES 2011), Zorro (CHES 2013), Robin (FSE 2014)…

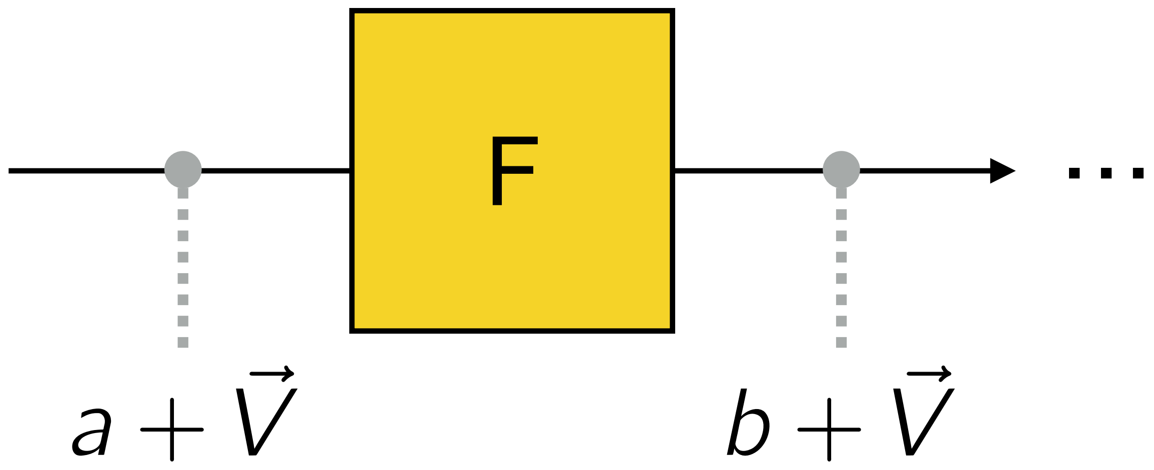
Attaques par sous-espace invariant

Les **Attaques par sous-espaces invariants** (Invariant Subspace Attacks) ont été introduites à CRYPTO 2011.

Utilisées pour casser PRINTCIPHER [LAKZ11].

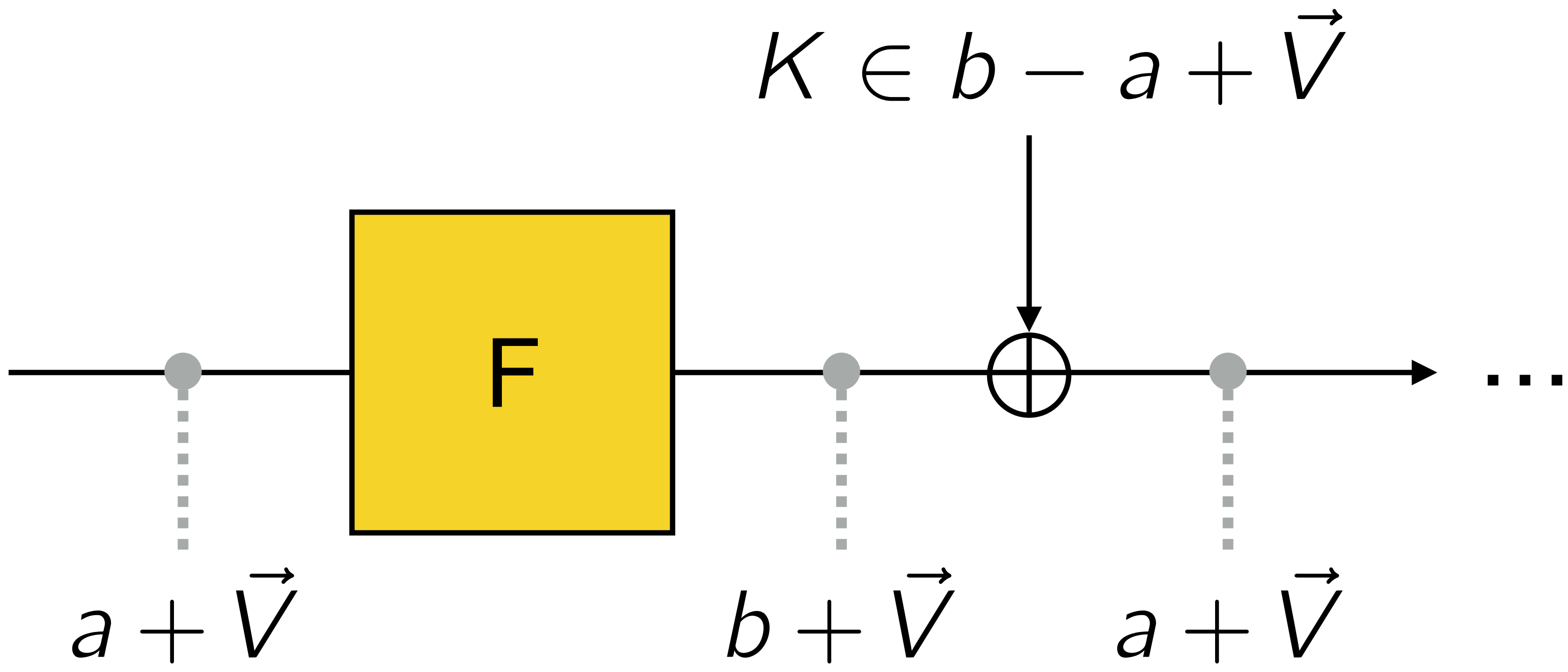
S'appuie sur l'absence de cadencement de clef.

Sous-espace invariant



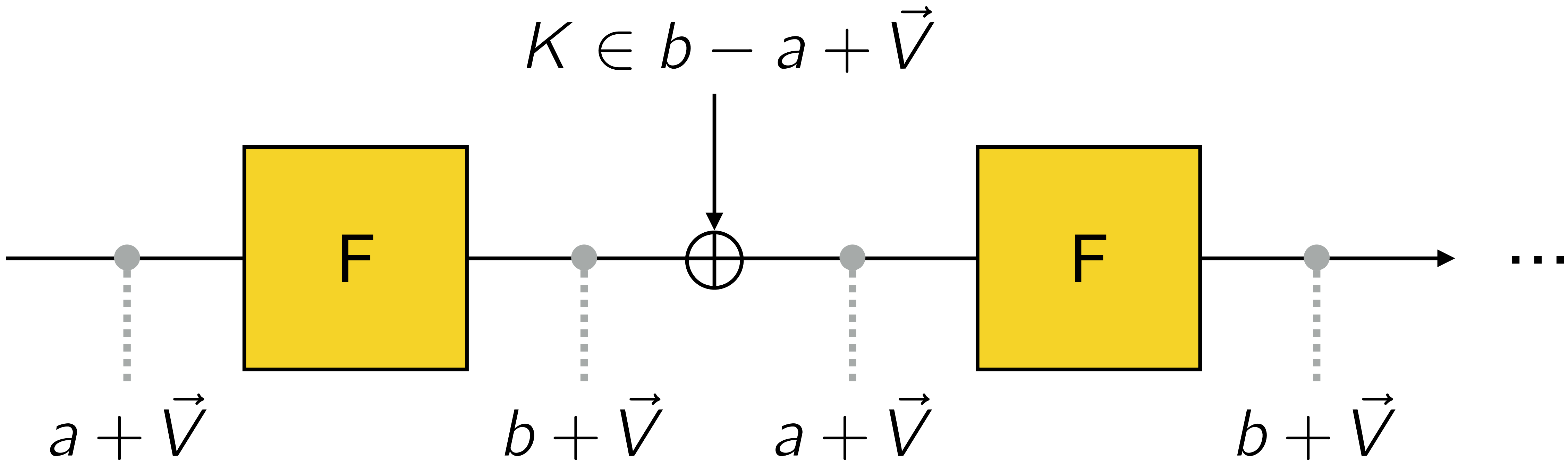
Supposons que la fonction de tour envoie un espace vectoriel sur un coset du même espace.

Sous-espace invariant



Supposons aussi $K \in b - a + \vec{V}$...

Sous-espace invariant

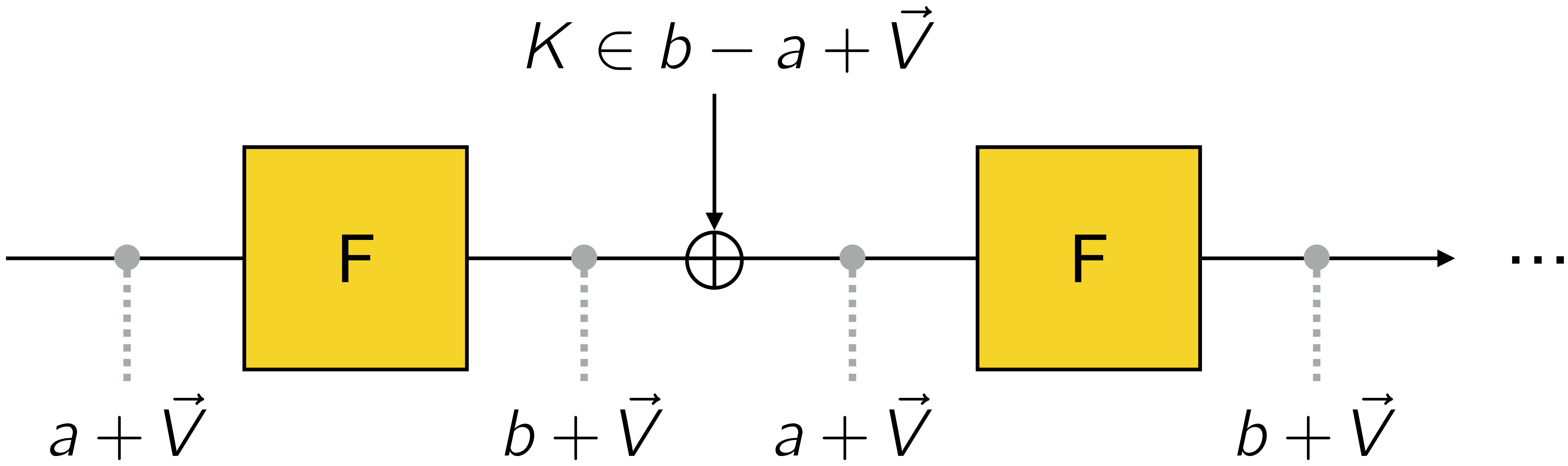


Supposons aussi $K \in b - a + \vec{V}$…

Alors le processus se répète.

Les clairs de $a + \vec{V}$ sont chiffrés dans $b + \vec{V}$.

Sous-espace invariant

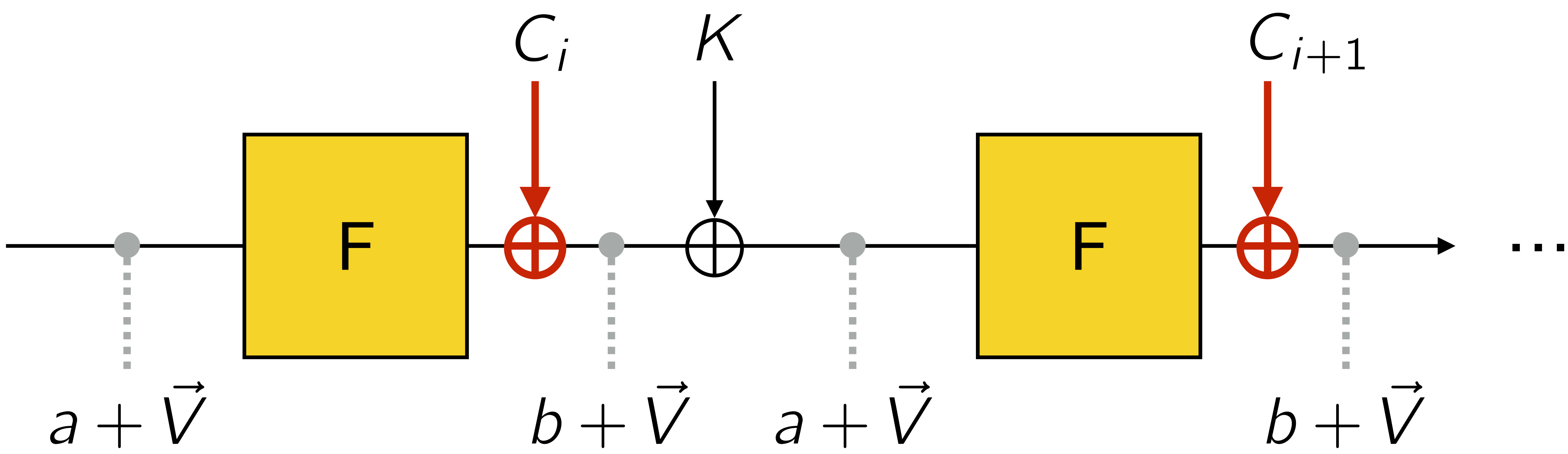


La confidentialité est perdue.

Densité des clefs faibles: $2^{-\operatorname{codim} \vec{V}}$

Algorithme générique de recherche de sous-espace invariant

Constantes de tour



En fait on veut $\forall i, C_i \in \vec{V}$

Cela fournit un “noyau” $\vec{W} = \text{span}\{C_i\} \subseteq \vec{V}$

Si on devine un offset $s \in a + \vec{V}$,
on connaît un sous-espace de $a + \vec{V}$.

Algorithme générique

<u>Algorithme générique</u>

1. $\vec{W} \leftarrow \mathrm{span}\,\{C_i\}$
2. Deviner offset s
3. Calculer $\mathrm{Closure}(s + \vec{W})$
4. Répéter jusqu'à ce que $\dim(\mathrm{Closure}) < n$

Si $a + \vec{V}$ est en fait linéaire : résultat instantané.

Sinon, en moyenne : $2^{-\mathrm{codim}\,\vec{V}}$ essais.

Résultats de l'algorithme générique

	Résultat	Temps
Robin	**Sous-espace trouvé!** codimension 32	22h
iSCREAM	**Sous-espace trouvé!** codimension 32	22h
Zorro	**Sous-espace trouvé!** codimension 32	<1h
Fantomas	Avec probabilité 99.9%: Pas d'espace invariant de codimension < 32	
NOEKEON		
LED		
Keccak		

➡ Clefs faibles de densité 2^{-32}, qui entraînent perte de confidentialité pour Robin, iSCREAM, Zorro.

Cryptanalyse de Robin et Fantomas

Robin

Robin and Fantomas [GLSV14], FSE 2014.

Chiffrements légers avec masquage facilité.

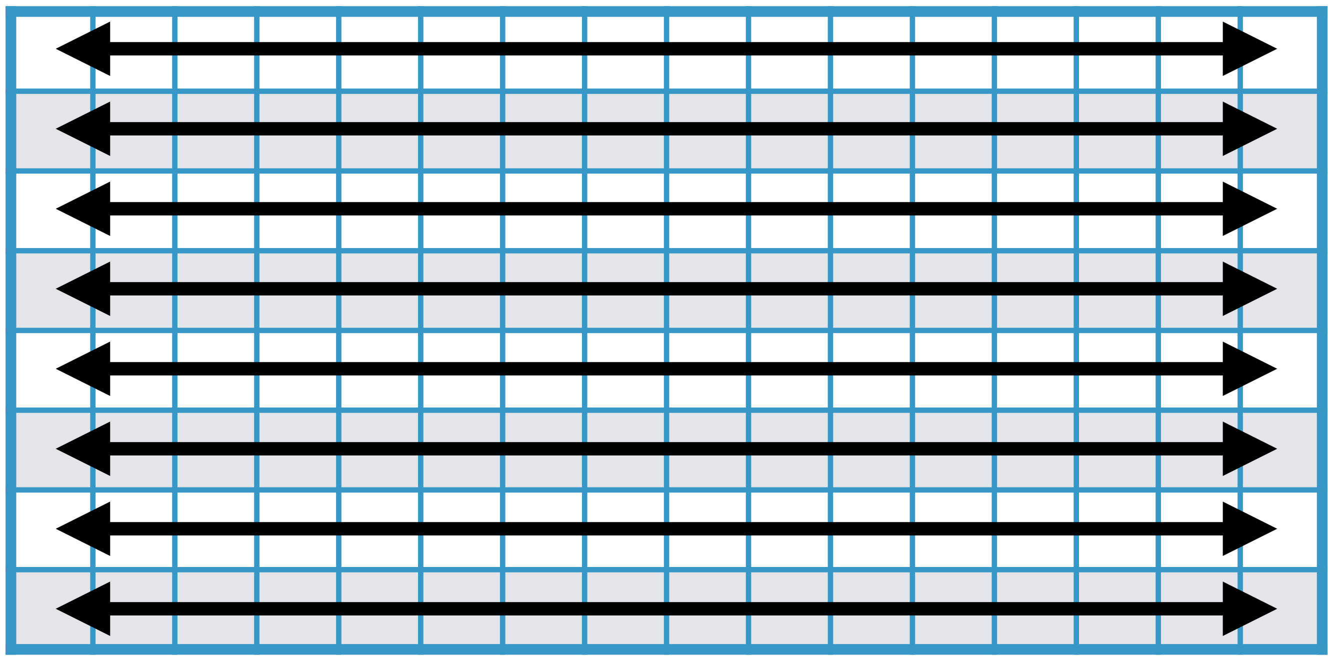
Bloc =128 bits — Sécurité = 128 bits

Robin = version involutive.

Design simple et élégant : “LS-design”.

Robin: couche **L**

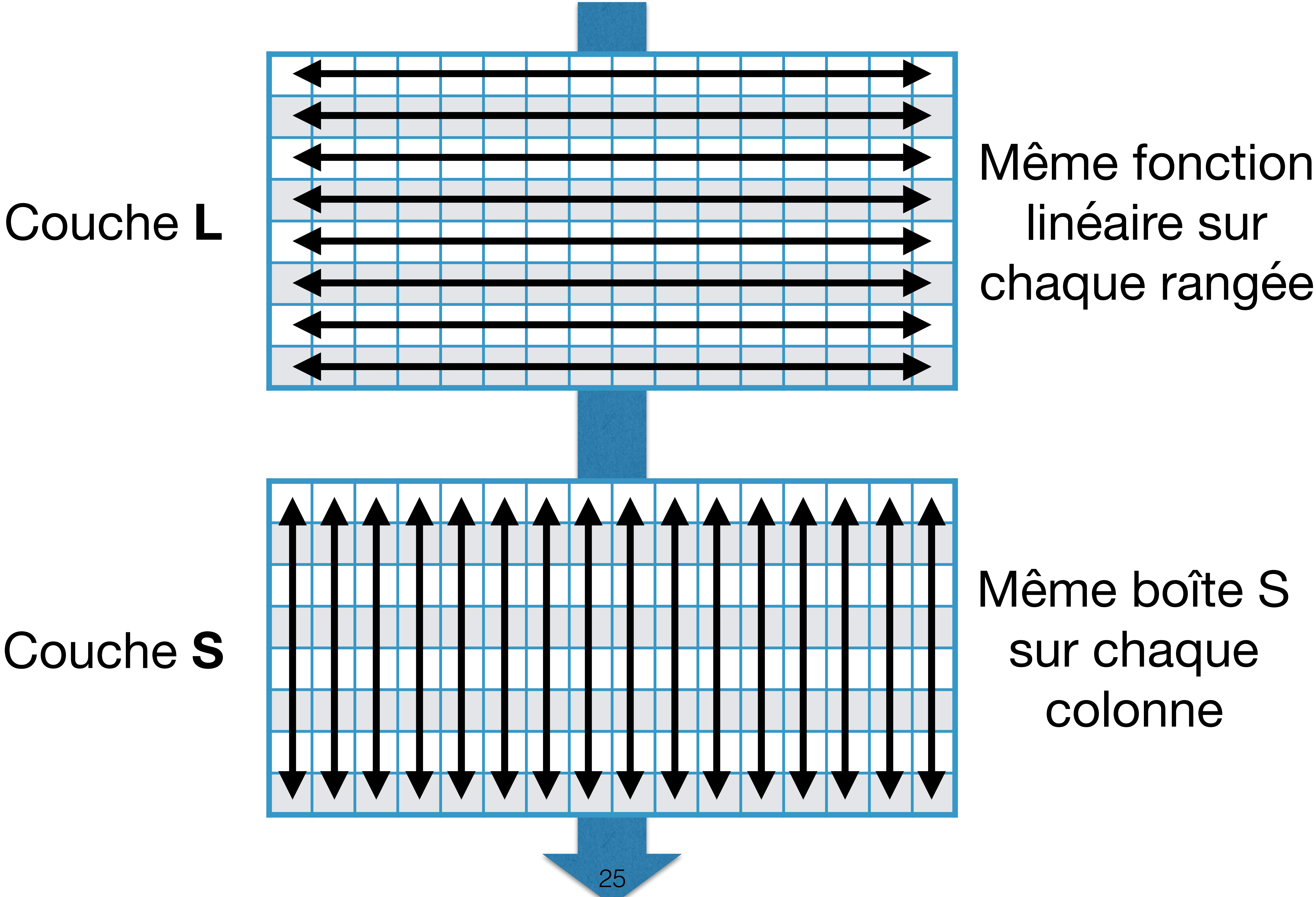
Couche **L**



État
16 x 8 bits

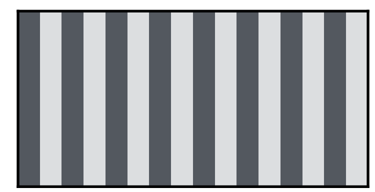
La même fonction linéaire *L* est appliquée à chaque rangée.

Robin: couches **LS**



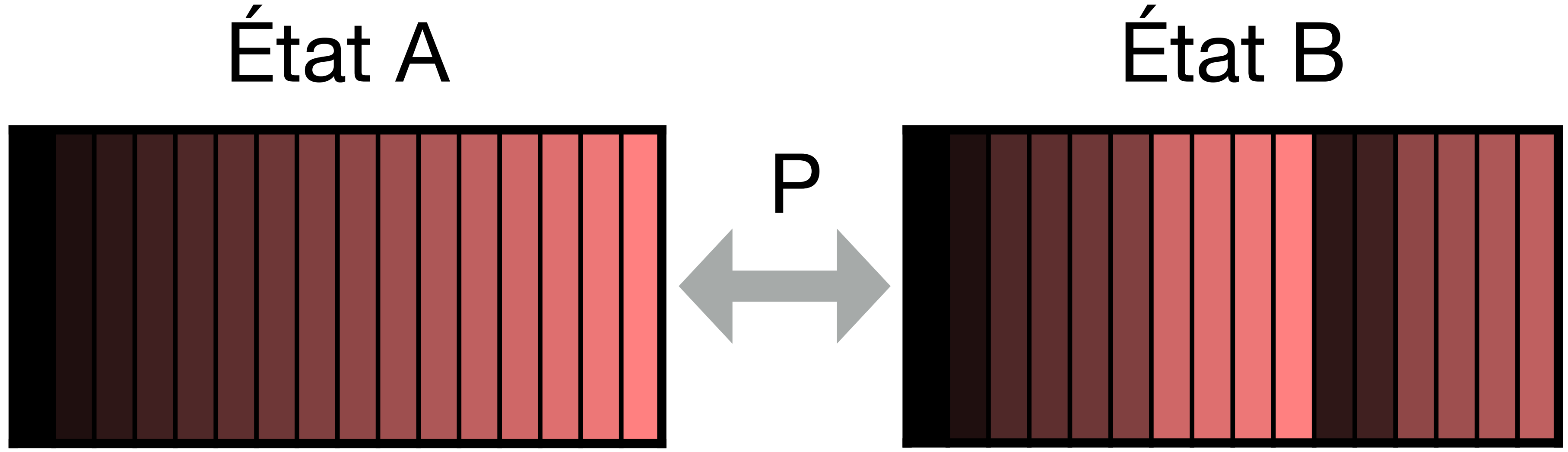
Fonction de tour de Robin

Un tour =

- Couche **L** 
- Couche **S** 
- Ajout de constante de tour
- Ajout de clef (pas de cadencement)

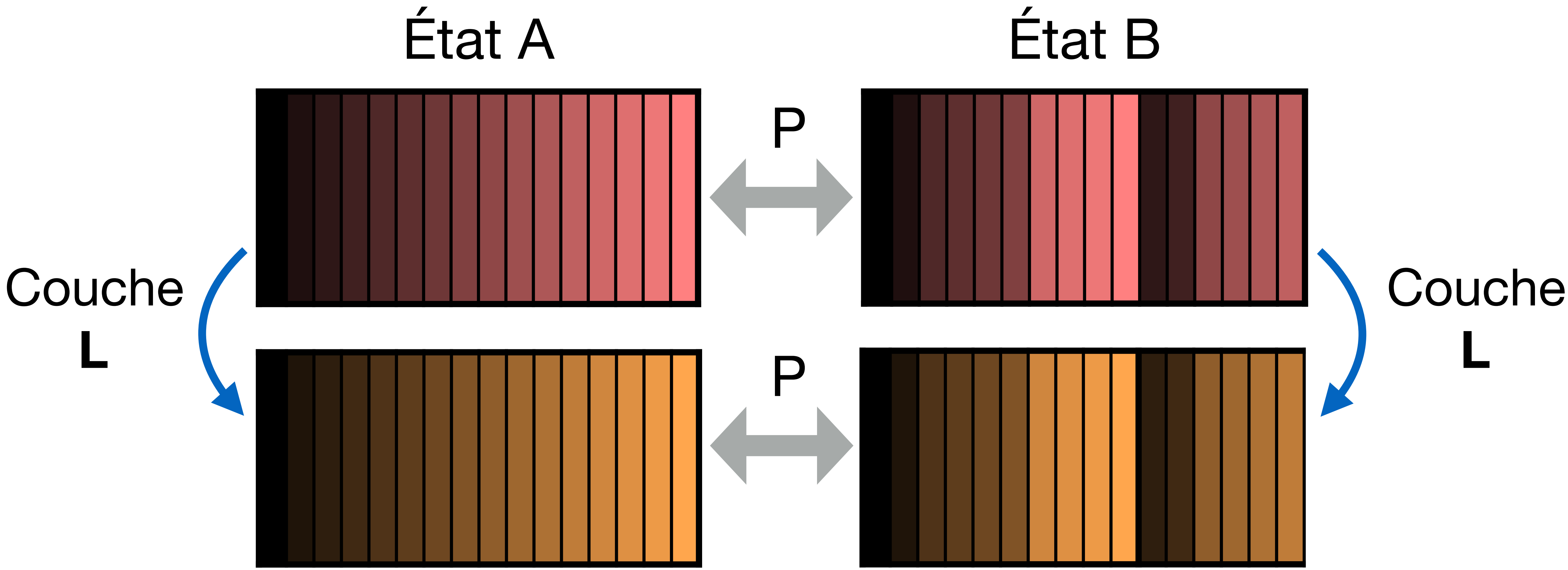
Un chiffrement = 16 tours.

Permutations invariantes



État B = permutation des colonnes de l'état A

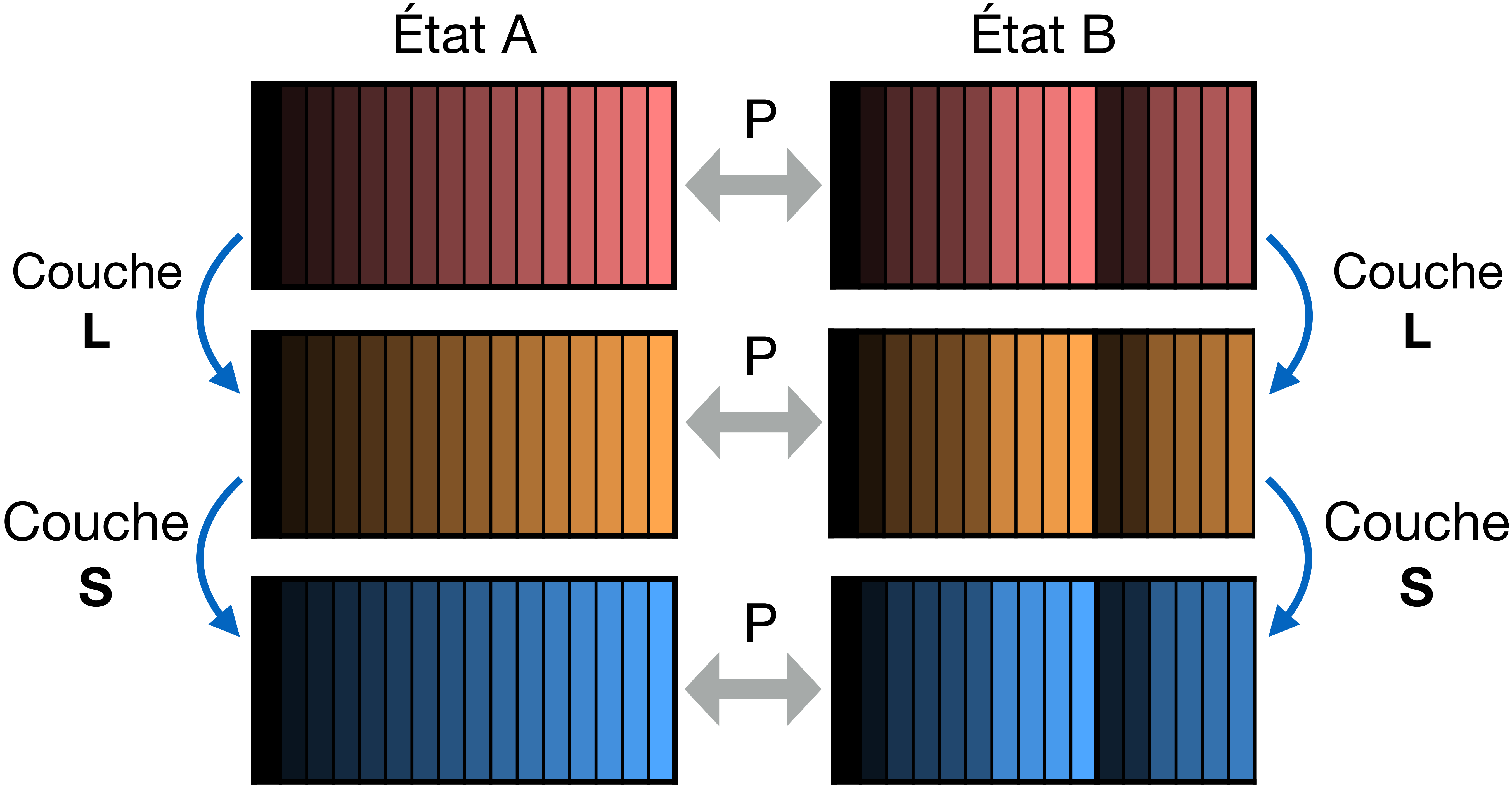
Permutations invariantes



Supposons **PL = LP**.

Alors l'état B reste une permutation de l'état A à travers la couche **L**.

Permutations invariantes



La couche **S** a la même propriété gratuitement !

Permutations invariantes

L'état B reste une permutation de l'état A à travers…

- La couche **L**: OK si $LP = PL$.
- La couche **S**: OK.
- L'ajout de constante : OK si $P(C_i) = C_i$.
- L'ajout de clef : OK si $P(K_A) = K_B$.

➡ P commute avec la fonction de tour entière !

Attaque par permutation invariante

Si $LP = PL$ et $\forall i, C_i \in \ker(P + \mathrm{Id})$:

alors pour deux *clefs reliées* $K_2 = P(K_1)$,
deux *clairs reliés* $P_2 = P(P_1)$ restent liés à travers le chiffrement et donnent des *chiffrés reliés* $C_2 = P(C_1)$.

Si $LP = PL$ et $\forall i, C_i \in \ker(P + \mathrm{Id})$:

alors pour une clef ***auto-reliée*** $K = P(K)$,
deux *clairs reliés* $P_2 = P(P_1)$ restent liés à travers le chiffrement et donnent des *chiffrés reliés* $C_2 = P(C_1)$.

Attaque par sous-espace invariant

Si $LP = PL$ et $\forall i, C_i \in \ker(P + \mathrm{Id})$:

alors pour une clef *auto-reliée* $K = P(K)$, des clairs *auto-reliés* $M = P(M)$ produisent des chiffrés *auto-reliés* $C = P(C)$.

Ceci est une attaque par sous-espace invariant !

Le sous-espace invariant est $\ker(P + \mathrm{Id})$.

Attaque sur Robin et iSCREAM

Robin et iSCREAM : une permutation adéquate P.

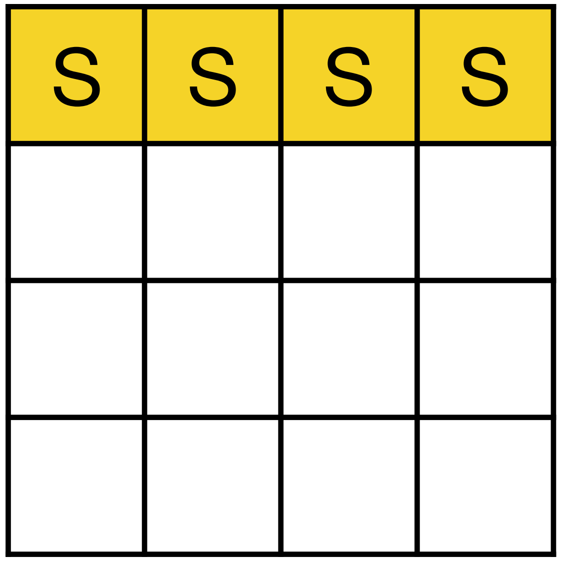
- Attaque à **clef faible**. Densité $2^{-\operatorname{codim}\ker(P+\mathrm{Id})} = 2^{-32}$
- Attaque à **clef reliée**.
- Données requises : 2 clairs choisis, coûts en temps et mémoire négligeables.

De plus, pour des clefs faibles:

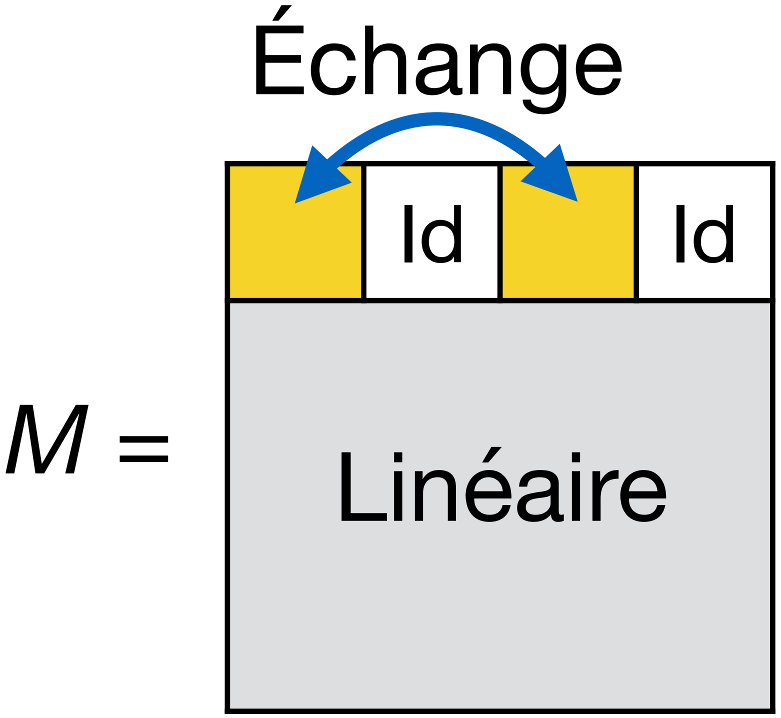
- Les points fixes de P forment un sous-chiffrement.
- Recouvrement de la clef en temps 2^{64}.

Robin vs Zorro

Zorro est un variante d'AES avec quelques différences notables :



- Pas de cadencement de clef.
- S-boîtes sur une seule rangée.



Cependant : il existe encore M qui commute avec la fonction de tour !

➡ **Toutes** les faiblesses de Robin.
En particulier, clefs faibles de densité 2^{-32}.

Conclusion

- Algorithme générique pour espaces invariants.

  Trouve automatiquement les attaques sur Robin, iSCREAM et Zorro.

- Cassage pratique de Robin, iSCREAM et Zorro.

  Clefs faibles de densité 2^{-32} dans les trois cas.

  Basé sur un nouveau type d'autosimilarité.

  iSCREAM éliminé de CAESAR.

- Espaces invariants sur Midori, Haraka, AES.

  Invariant non-linéaire sur SCREAM, Midori [TLS16].

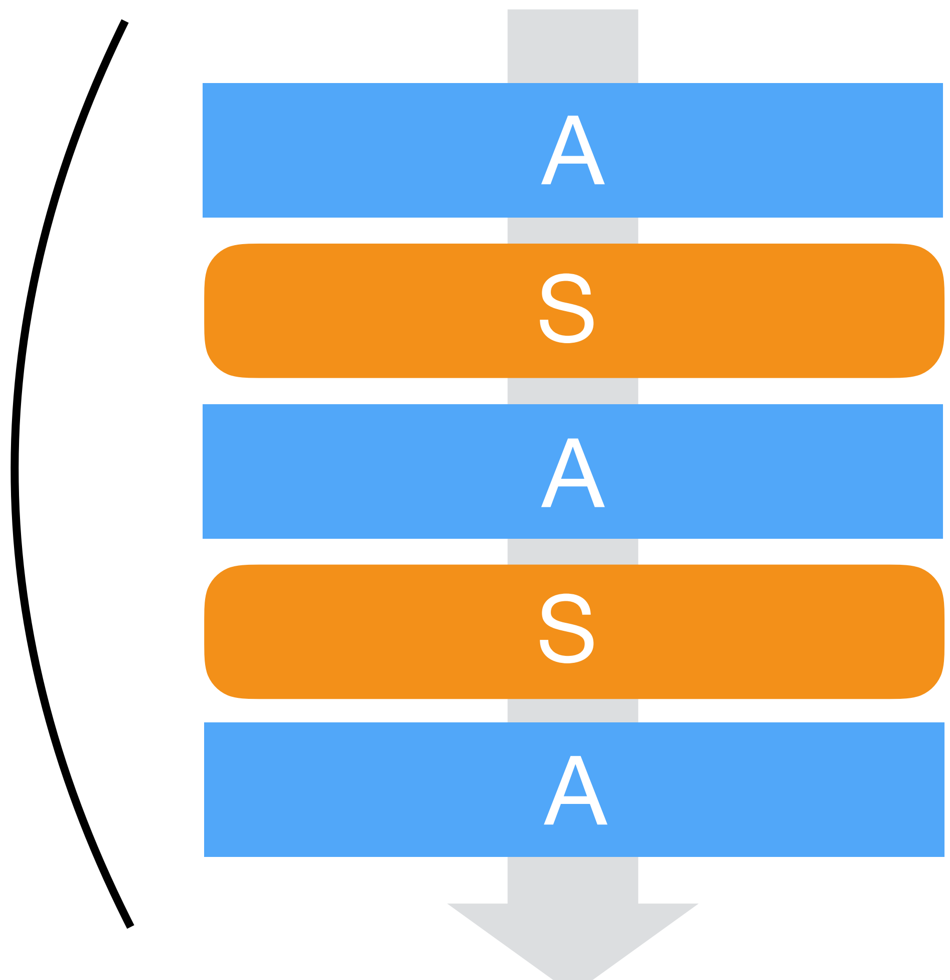
2^{e} partie

Cryptanalyse structurelle d'ASASA

Structure ASASA

À Asiacrypt 2014, Biryukov, Bouillaguet et Khovratovich ont proposé plusieurs applications de la structure ASASA.

$\mathbf{F} = \mathbf{A} \circ \mathbf{S} \circ \mathbf{A} \circ \mathbf{S} \circ \mathbf{A}$



Couche Affine

Couche non-linéaire
e.g. boîtes S

ASASA

Trois applications proposées dans [BBK14]:

- 1 schéma «boîte noire»

  ≈ chiffrement par bloc ✘ [MDFK15]

- 2 schémas «boîte blanche forte»

  ≈ chiffrements à clef publique

  - Variante «expanding S-box» ✘ [GPT15]
  - Variante «avec χ» ✘ [MDFK15]

- 1 schéma «boîte blanche faible» ✘ [MDFK15] & [DDKL15]

même attaque !

Plan

1. Chiffrement à clef publique **ASASA** «avec χ».
2. Cryptanalyse.
3. Chiffrement symétrique **ASASA**.
4. Cryptanalyse (identique).

Chiffrement ASASA «avec χ»

Cryptographie multivariée

Problème difficile: résoudre un système aléatoire d'équations quadratiques sur un corps fini.

→ comment déduire un chiffrement $\mathbb{F}_q^n \to \mathbb{F}_q^n$:

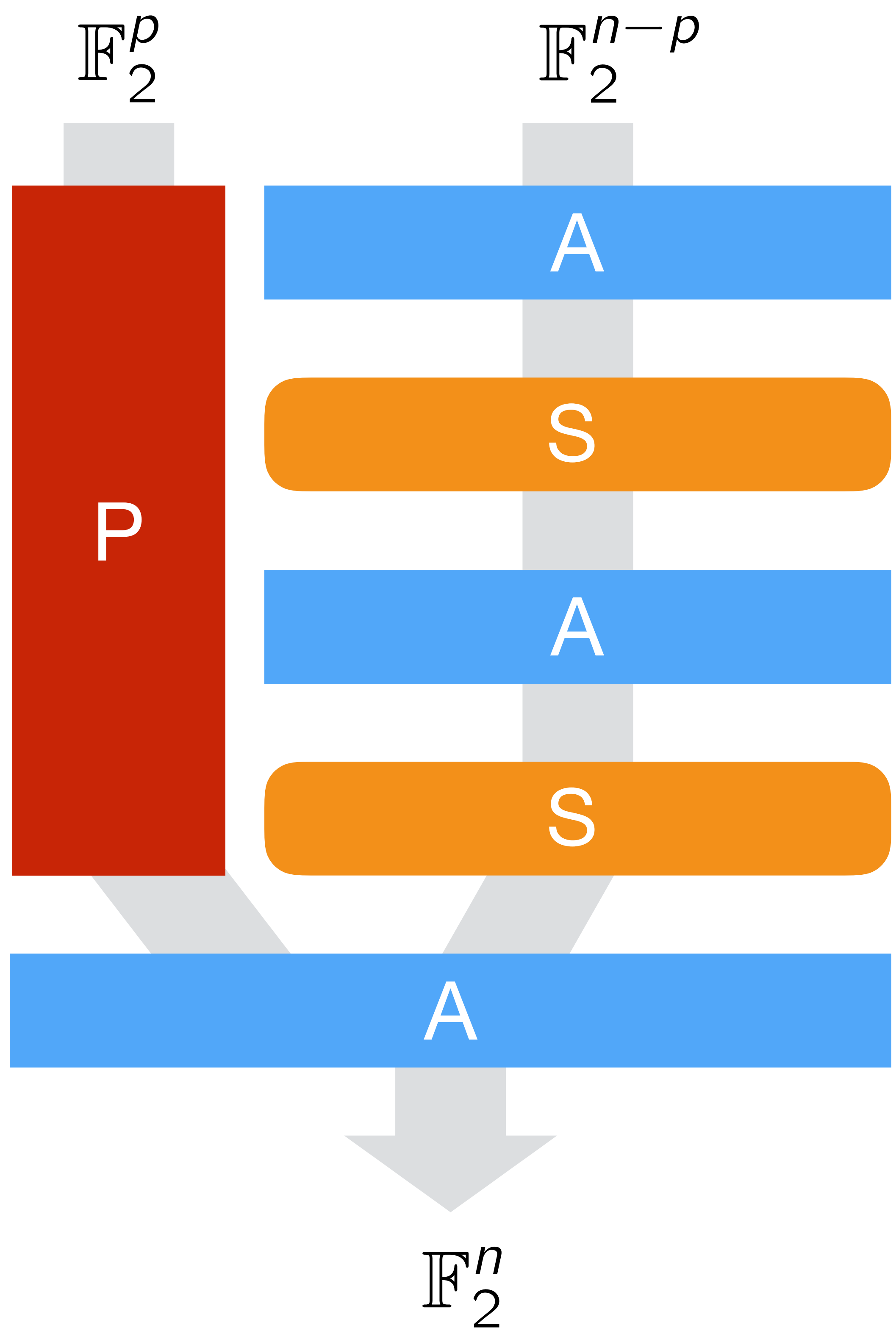
Clef publique: chiffrement **F** décrit comme une suite de *n* polynômes quadratiques en *n* variables.

Clef privée: structure cachée (décomposition) de **F** qui rend l'inversion facile.

+: messages de petite taille, rapide avec clef privée.
-: lent avec clef publique, grande clef, pas de réduction.

Structure ASASA + P [BBK14]

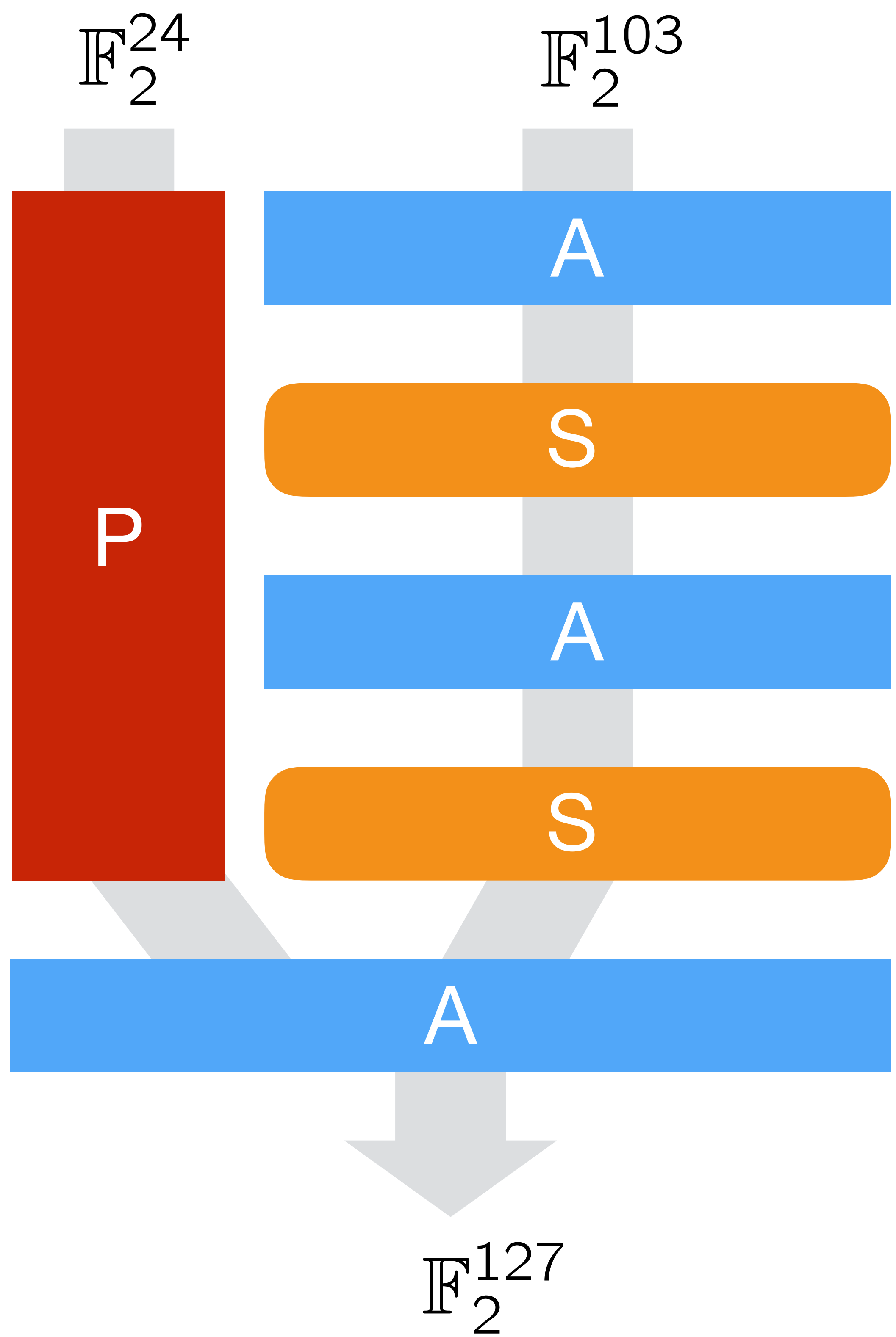
Perturbation : polynômes aléatoires de degré 4



Couche quadratique

Couche quadratique

Instance «avec χ»

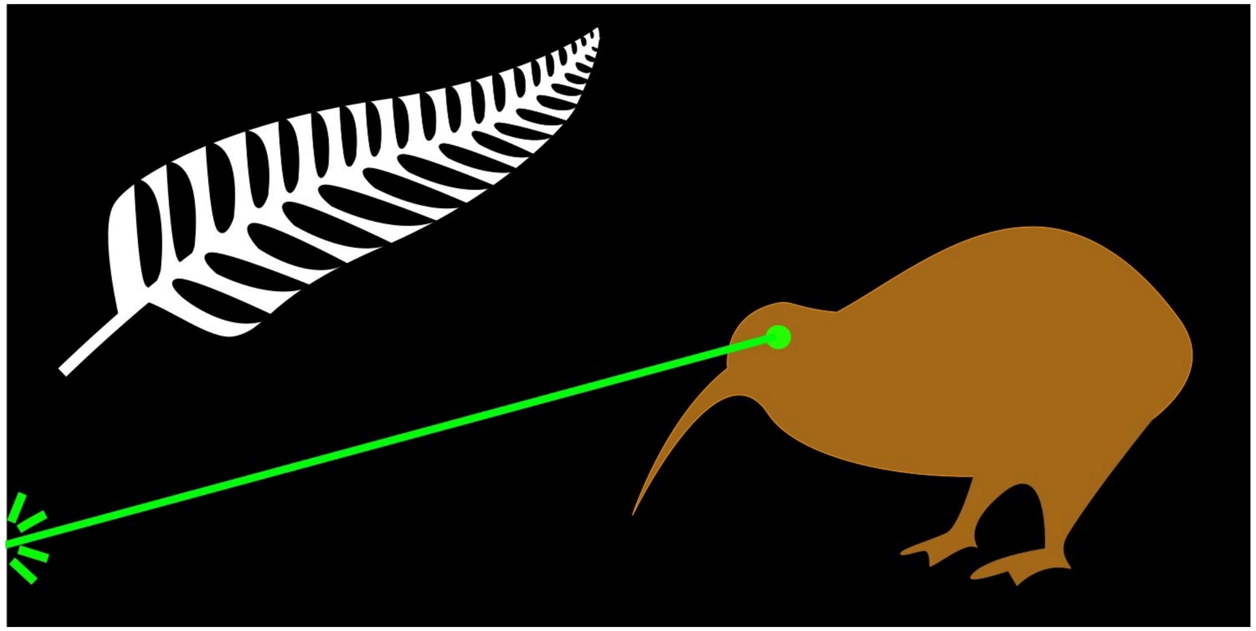


Perturbation : polynômes aléatoires de degré 4

χ

χ

Couche affine inversible aléatoire



Cryptanalyse

Cubes

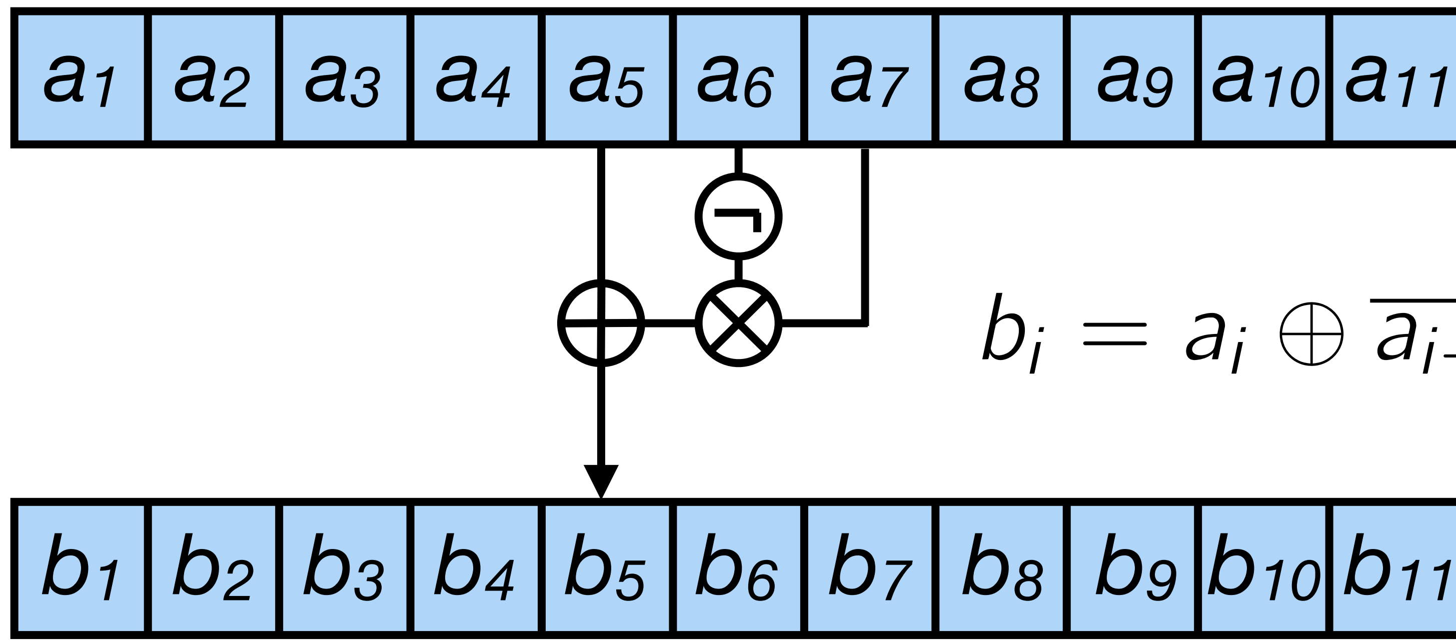
Un **cube** est un sous-espace affine [DS08].

Fait : Soit *f* un polynôme de degré *d* sur des variables binaires. Si *C* est un cube de dimension *d+1*, alors :

$$\sum_{c \in C} f(c) = 0$$

Fonction χ de Keccak

$a \in \mathbb{F}_2^n$



$b_i = a_i \oplus \overline{a_{i+1}} \cdot a_{i+2}$

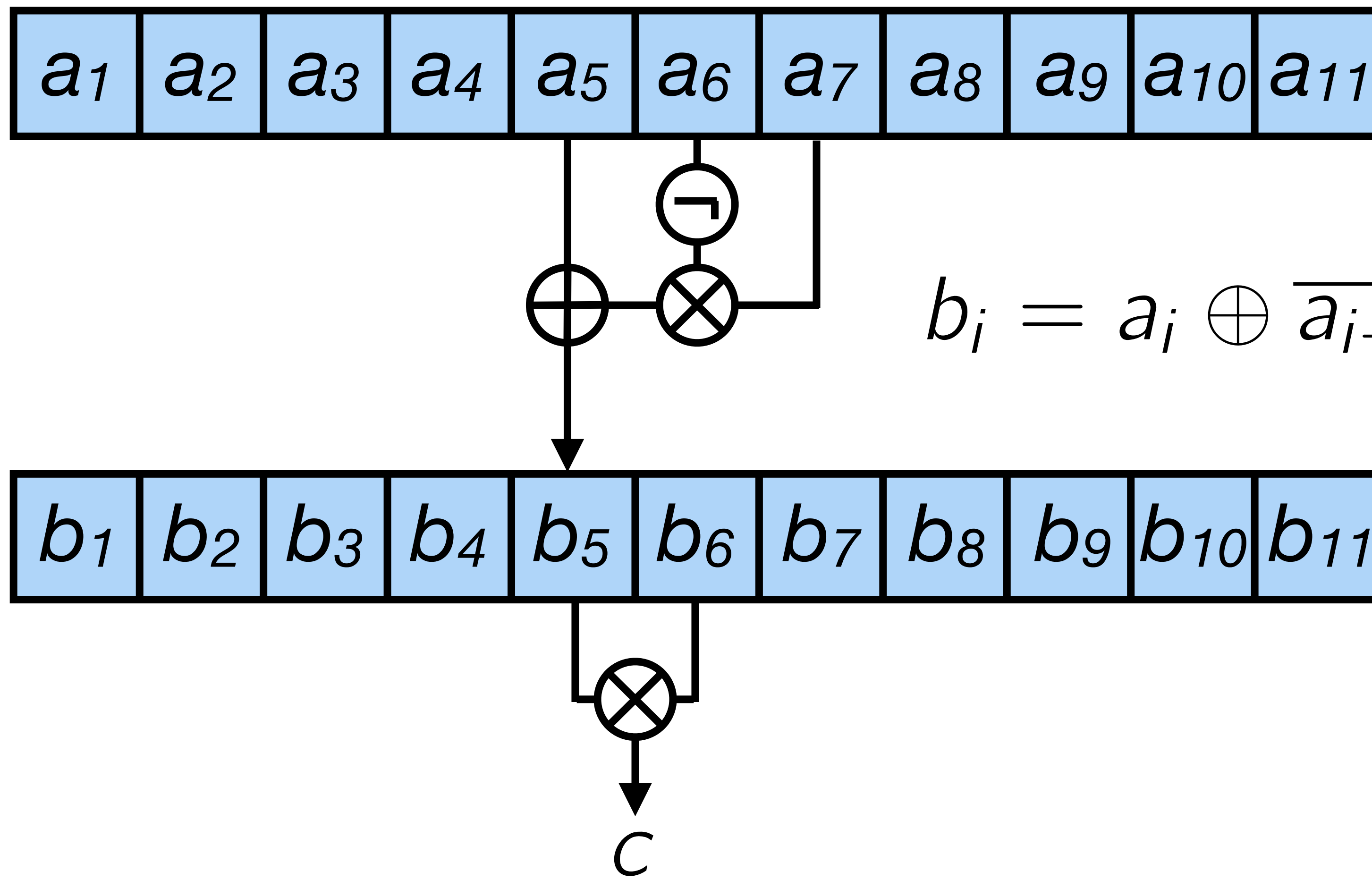
$b = \chi(a)$

Introduite par Daemen en 1995, rendue célèbre par son utilisation dans Keccak.

Déficience de degré

$a \in \mathbb{F}_2^n$

$b = \chi(a)$

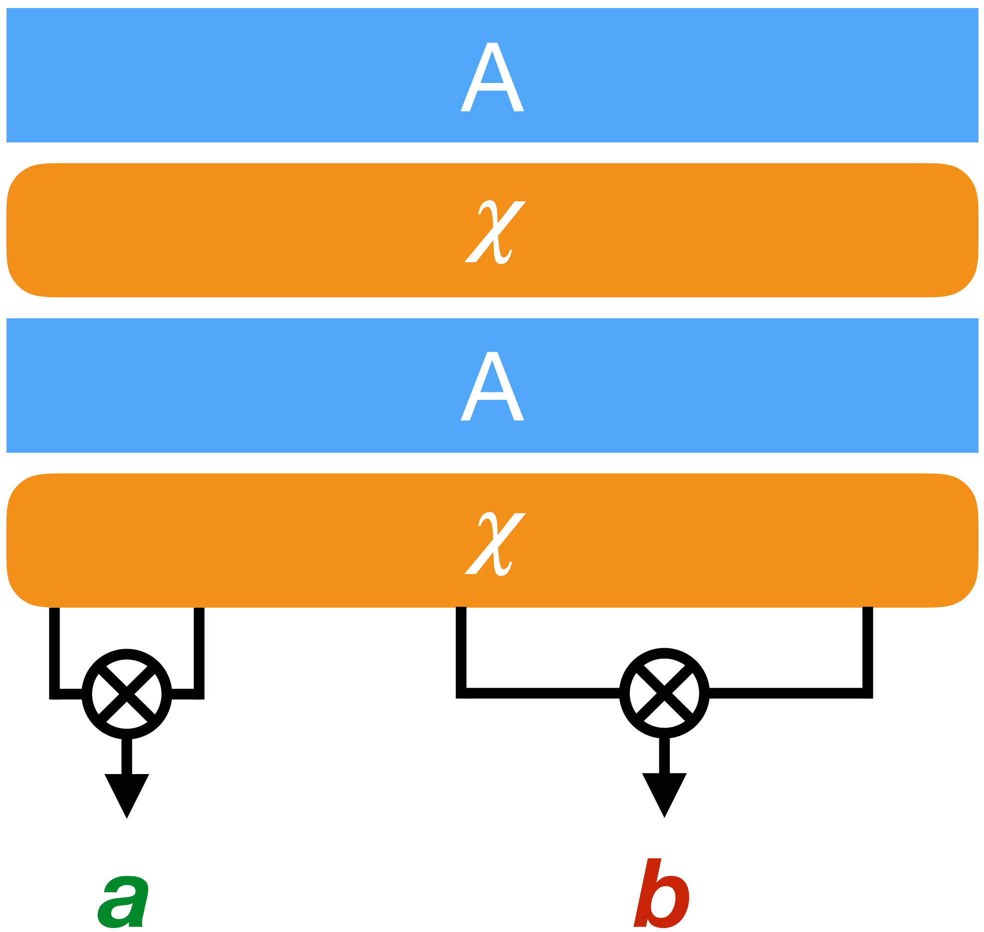


$$b_i = a_i \oplus \overline{a_{i+1}} \cdot a_{i+2}$$

$$\begin{aligned} c &= b_i \cdot b_{i+1} \\ &= (a_i \oplus \overline{a_{i+1}} \cdot {\color{red} a_{i+2}}) \cdot (a_{i+1} \oplus \overline{{\color{red} a_{i+2}}} \cdot a_{i+3}) \end{aligned}$$

→ c a degré 3. Somme à 0 sur un cube de dim 4.

Cryptanalyse d'ASASA



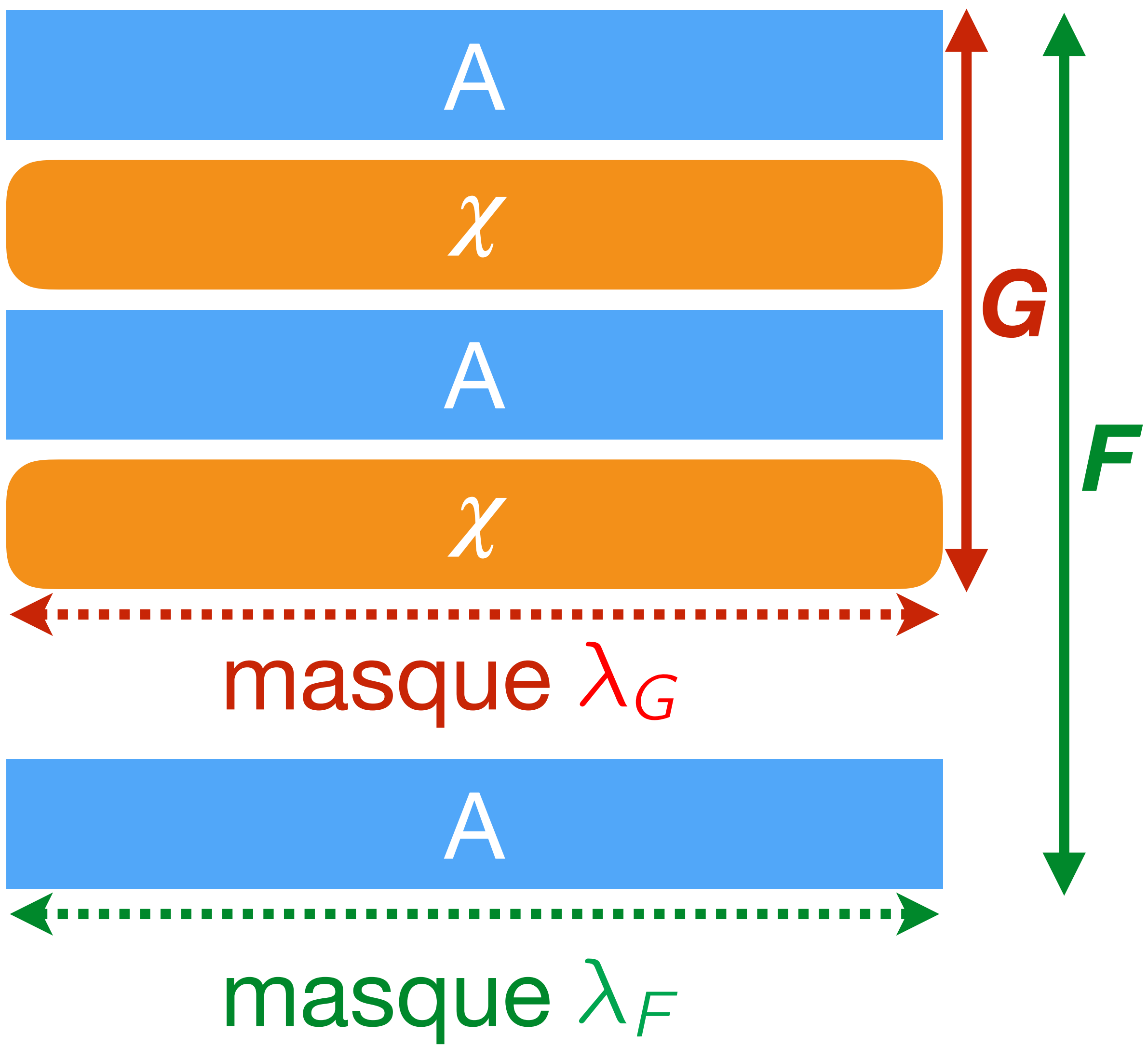
- Soit ***a*** = produit de 2 bits **adjacents** en sortie de χ.

  Alors ***a*** est de degré 6.

- Soit ***b*** = produit de 2 bits **non-adjacents** en sortie de χ.

  Alors ***b*** est de degré 8.

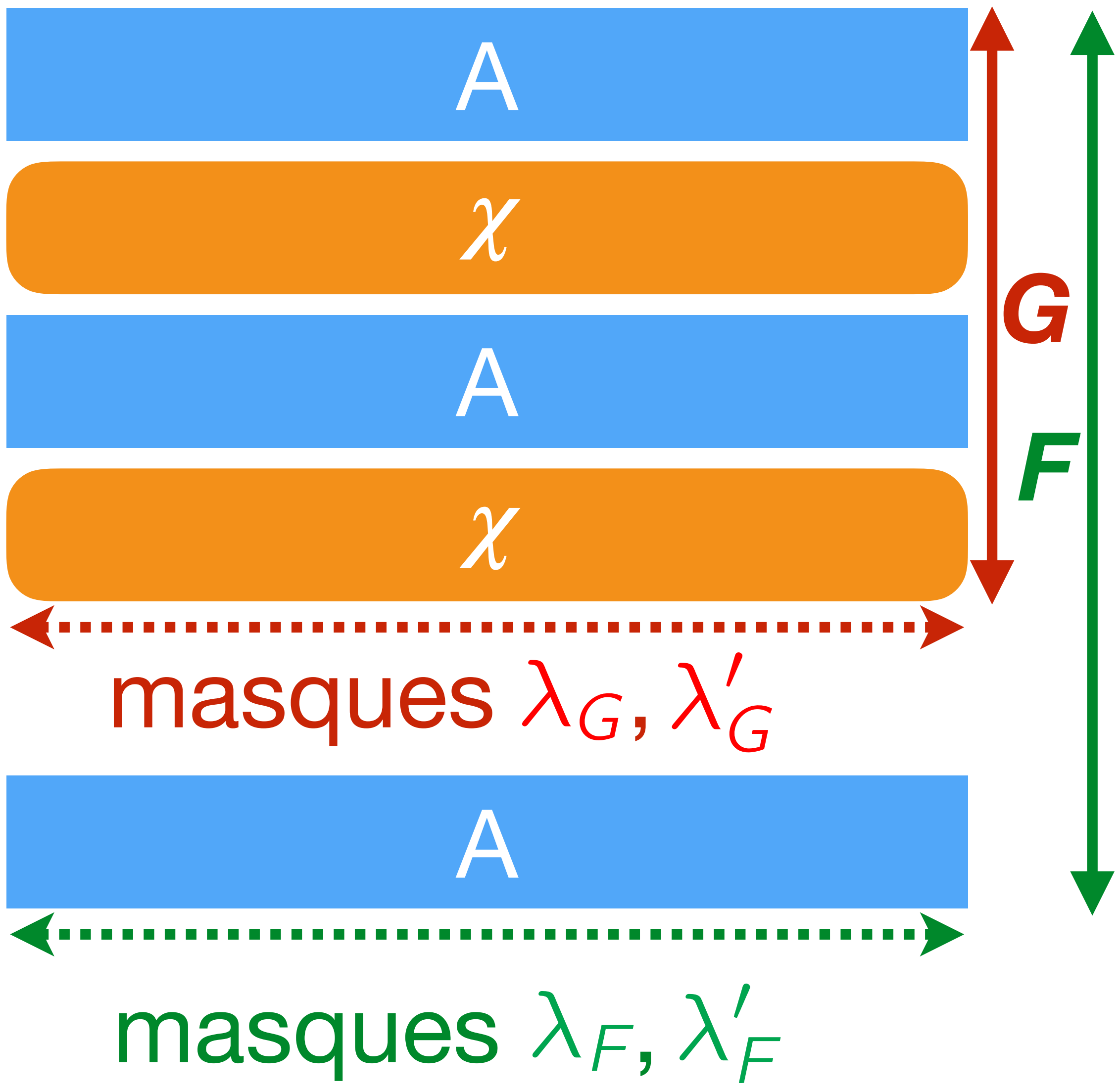
Cryptanalyse d'ASASA



Soit λ_F un masque en sortie, i.e. on regarde $\langle F|\lambda_F\rangle = x \mapsto \langle F(x)|\lambda_F\rangle$.

Alors il existe un masque λ_G tel que $\langle F|\lambda_F\rangle = \langle G|\lambda_G\rangle$.

Cryptanalyse d'ASASA

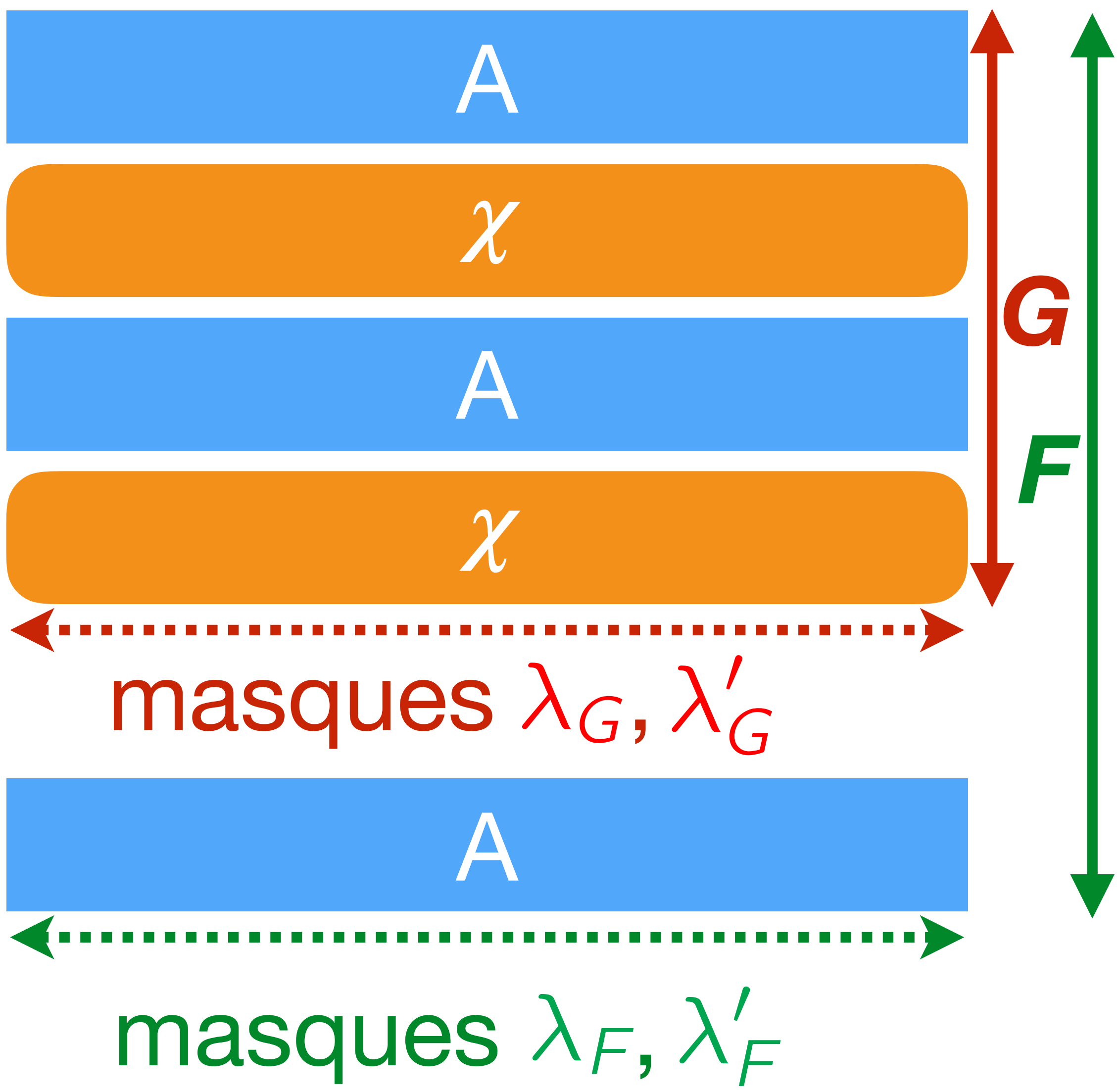


Soient λ_F, λ'_F deux masques en sortie, and λ_G, λ'_G avant **A**.

- Si λ_G et λ'_G activent deux bits **isolés** et **adjacents**, alors $\langle F|\lambda_F\rangle \cdot \langle F|\lambda'_F\rangle$ est de degré 6.

- Sinon $\langle F|\lambda_F\rangle \cdot \langle F|\lambda'_F\rangle$ est de degré 8.

Cryptanalyse d'ASASA



But : Trouver λ_F, λ'_F tels que

$$\deg(\langle F|\lambda_F\rangle \cdot \langle F|\lambda'_F\rangle) = 6$$

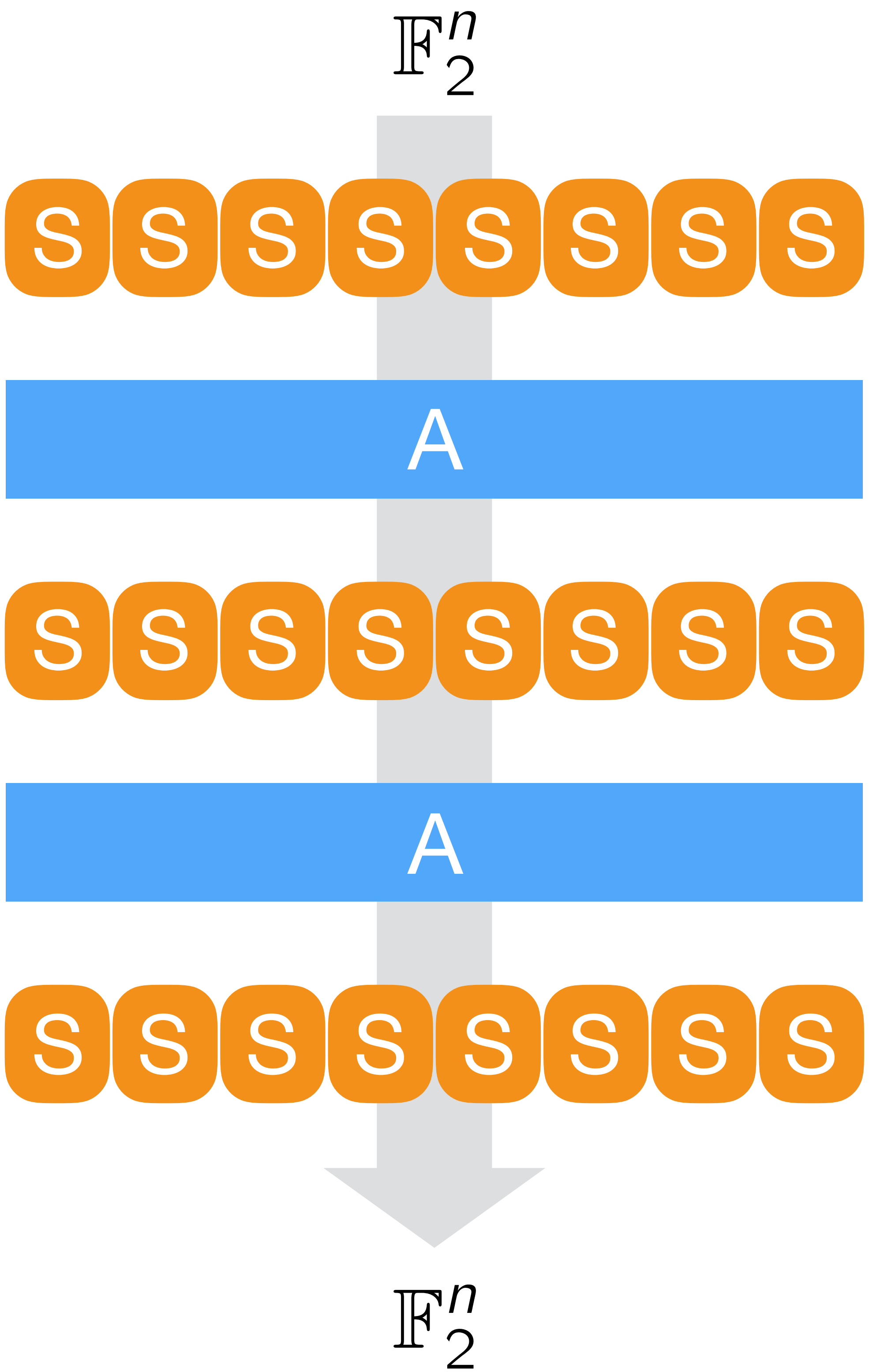
Soit C cube de dimension 7. Alors :

$$\Sigma_{c\in C}\langle F(c)|\lambda_F\rangle \cdot \langle F(c)|\lambda'_F\rangle = 0$$

→ donne une équation sur λ_F, λ'_F.

ASASA en «boîte noire»

Structure SASAS



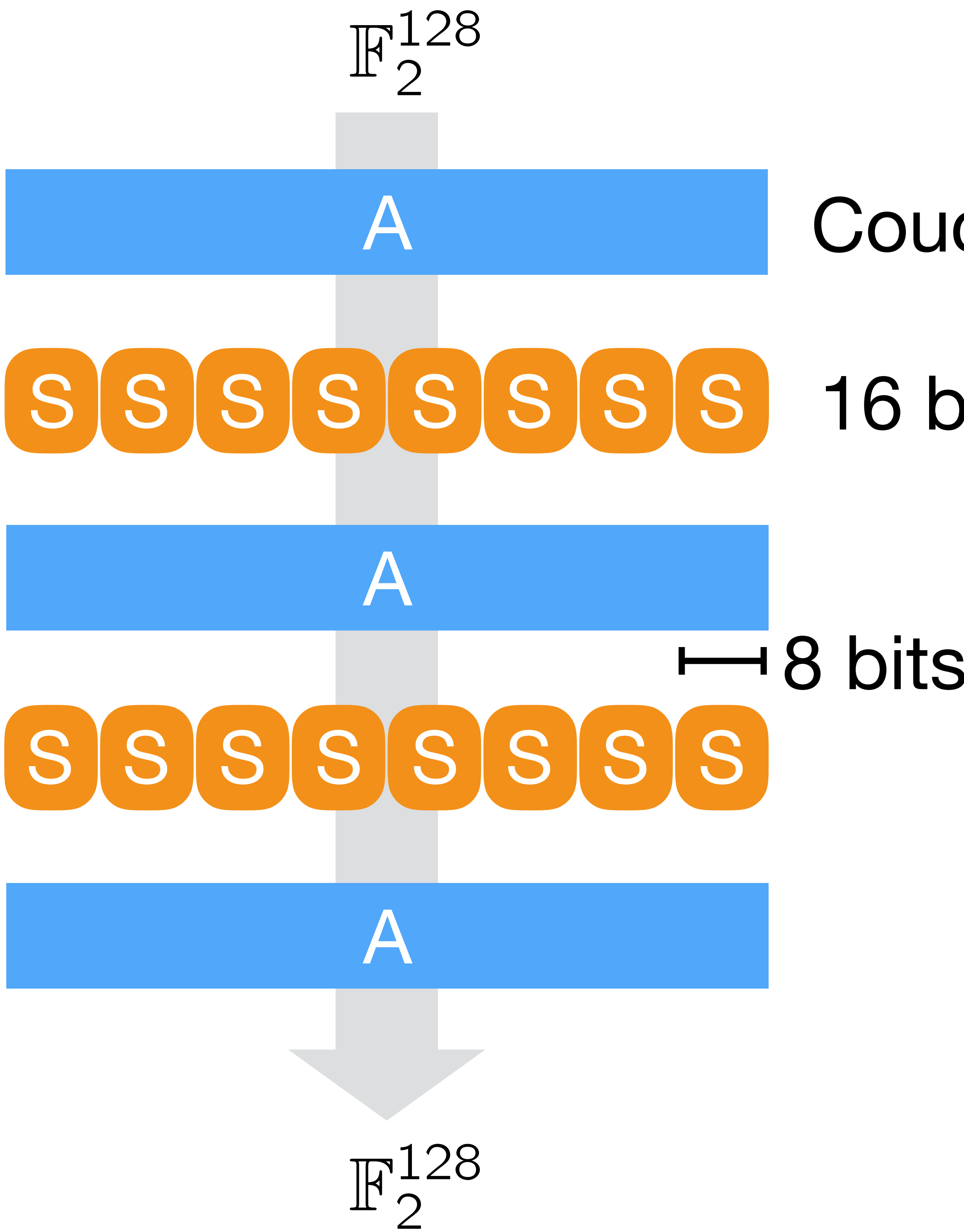
Analysée par Biryukov et Shamir à Eurocrypt 2001.

Couche **A**ffine aléatoire sur n bits.

Boîtes **S** aléatoires indépendantes chacune sur k bits.

→ **But**: recouvrer tous les composants internes (couches affines **A** et boîtes **S**) avec accès «boîte noire» (KP/CP/CC).

ASASA «boîte noire» [BBK14]



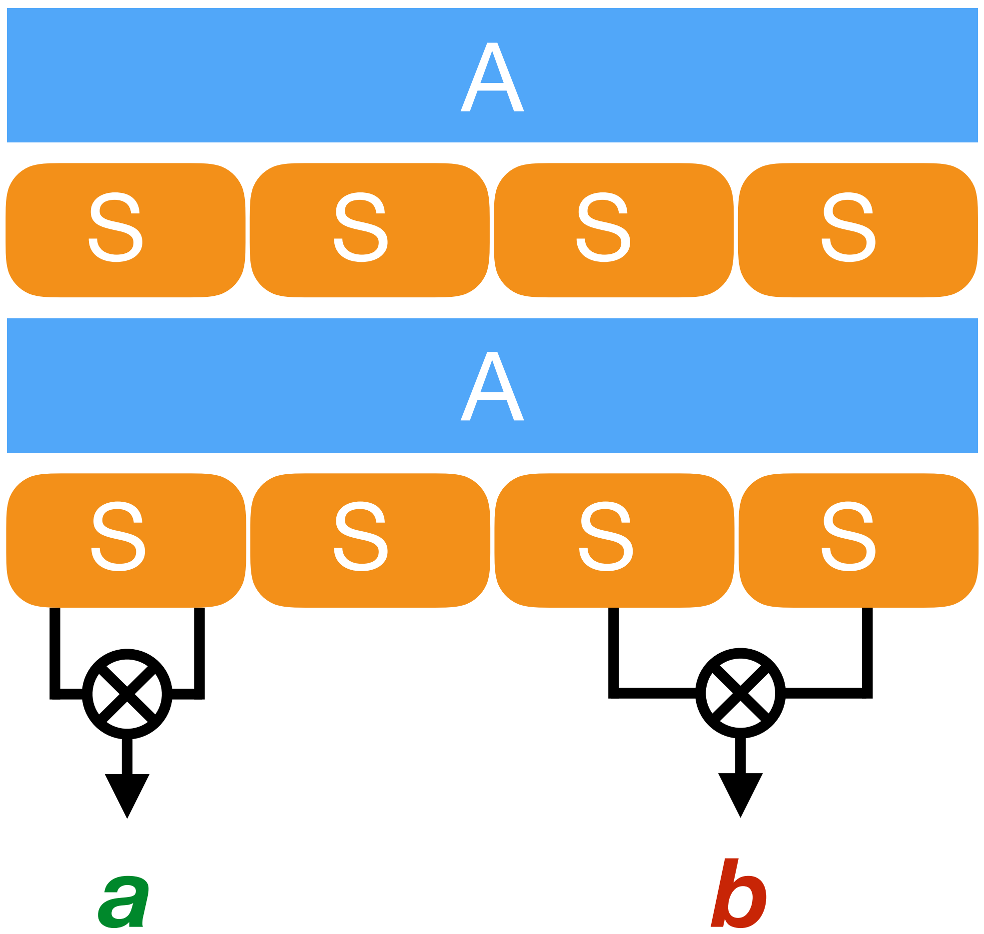
Couches **A** aléatoires sur 128 bits.

16 boîtes **S** indépendantes

But : recouvrer tous les composants internes.

Note: degré ≤ 49
$\Rightarrow$ distingueur en 2^{50} CP.

Cryptanalyse d'ASASA



Degré d'une boîte S = 7.

- Soit *a* = produit de 2 bits en sortie d'**une seule boîte S commune**.

  Alors *a* est de degré 7x7 = 49.

- Soit *b* = produit de 2 bits en sortie de deux boîtes S **distinctes**.

  Alors *b* est de degré max (127).

Variantes d'ASASA

- Article de Biryukov et Khovratovich :

  La même attaque s'étend à ASASASA et même SASASASAS (ePrint, june 2015).

- Observation de Dinur, Dunkelman, Kranz et Leander :

  La même attaque s'applique encore pour des petites tailles de blocs (variantes «boîtes blanches faibles» de [BBK14]).

En effet l'obstacle principal est que la fonction globale ne doit pas être de degré max (→ bornes de Boura, Canteaut et de Cannière sur le degré de fonctions booléennes composites).

Conclusion

- Nouvelle attaque sur structures ASASA.

- Non présenté : attaque à base de LPN sur le schéma «avec χ».

- Problèmes ouverts :

  Autres applications de l'attaque.

  Construction sûre en boîte blanche.

3e partie

Cryptanalyse de CLT15

Applications multilinéaires

- Nombreuses applications, «crypto-complet».

  Obfuscation générale de programme, échange de clef multiparti sans interaction, witness encryption...

- Peu de constructions.

GGH13 (réseaux) Garg, Gentry, Halevi, Eurocrypt'13	✘ [HJ15] (éch. de clef)
CLT13 (entiers) Coron, Lepoint, Tibouchi, Crypto'13	✘ [CHLRS15] (idem)
GGH15 (avec graphes) Gentry, Gorbunov, Halevi, TCC'15	✘ [CLLT16] (idem)
CLT15 (modification de CLT13) Coron, Lepoint, Tibouchi, Crypto'15	✘ [CFLMR16] (idem) ([CLR15] + [MF15])

Applications multilinéaires

Message : $c \in \mathbb{Z}/n\mathbb{Z}$

Encodage : $g^c \in \mathbb{G}$ $\qquad$ $\mathbb{G}$ groupe d'ordre n généré par g.

Homomorphisme additif :

Addition de messages = multiplication des encodages. ✔

Multiplication de messages = Diffie-Hellman. ✘

Applications multilinéaires

Application multilinéaire :

$$\begin{aligned} e : \mathbb{G}^{\kappa} &\to \mathbb{H} \\ (g^{x_1}, g^{x_2}, \ldots, g^{x_\kappa}) &\mapsto h^{x_1 \cdots x_\kappa} \end{aligned}$$

où g, h sont générateurs de $\mathbb{G}$, $\mathbb{H}$.

Application multilinéaire à niveaux (*leveled*) :

$$\begin{aligned} e_{i,j} : \mathbb{G}_i \times \mathbb{G}_j &\to \mathbb{G}_{i+j} \qquad \text{pour } i + j \leq \kappa. \\ (g_i^x, g_j^y) &\mapsto g_{i+j}^{xy} \end{aligned}$$

où g_i est générateur de $\mathbb{G}_i$, $i \leq \kappa$.

Schémas d'encodage gradués

Schéma d'encodage gradué (*graded encoding scheme*) :

Message : $c \in \mathcal{P}$

Encodage : $\mathrm{enc}_i(c) \in \mathcal{C}_i$ au niveau *i*. Non déterministe.

Les encodages sont munis de :

- Addition : $\mathrm{enc}_i(x) + \mathrm{enc}_i(y) = \mathrm{enc}_i(x + y)$
- Multiplication : $\mathrm{enc}_i(x) \cdot \mathrm{enc}_j(y) = \mathrm{enc}_{i+j}(xy)$

Les encodages sont bruités (comme FHE).

Zéro-test : procédure publique $\mathrm{z} : \mathcal{C}_\kappa \to \{0, 1\}$

$$\mathrm{enc}_\kappa(x) \mapsto 1 \text{ ssi } x = 0$$

Encodage dans CLT15

Soient n nombres premiers g_i et p_i avec $g_i \ll p_i$.
Soit $z < x_0 = \prod p_i$.

Espace des messages : $\prod_{i \leq n} \mathbb{Z}/g_i\mathbb{Z}$

Encodage de $(m_1, \ldots, m_n) \in \prod_{i \leq n} \mathbb{Z}/g_i\mathbb{Z}$ au niveau k :

entier e tel que $\forall i$, $e \bmod p_i = \dfrac{r_i g_i + m_i}{z^k} \bmod p_i$:

$$e = \mathrm{CRT}_{(p_i)}\left(\frac{r_i g_i + m_i}{z^k}\right) + a x_0$$

avec r_i, a, petits bruits (secrets).

Opérations dans CLT15

Encodage au niveau k : $e = \mathrm{CRT}_{(p_i)}\left(\frac{r_i g_i + m_i}{z^k}\right) + a x_0$

Addition et multiplication d'encodages
= addition et multiplication dans les entiers !

La multiplication double la taille des encodages...

- Une **échelle** d'encodages de zéro de tailles croissantes permet de réduire la taille des encodages obtenus.

Zéro-test dans CLT15

Encodage au niveau κ : $e = \mathrm{CRT}_{(p_i)}\left(\frac{r_i g_i + m_i}{z^\kappa}\right) + a x_0$

En développant : $e = \sum (r_i + m_i {g_i}^{-1}) u_i + a x_0$

Zéro-test : premier $N \gg x_0$ et entier $p_{zt} < N$ tels que :

$$|v_0| = |x_0 p_{zt} \bmod N| \ll N$$

$$|v_i| = |u_i p_{zt} \bmod N| \ll N$$

Par conséquent pour un encodage de zéro e :

$$|e p_{zt} \bmod N| = \left|\sum r_i v_i + a v_0\right| \ll N$$

- Procédure de zéro-test : z(e) renvoie 1 ssi $|e p_{zt} \bmod N| \ll N$

Cryptanalyse

Attaque : extraction entière

Encodage de zéro au niveau κ :

$$e = \sum r_i u_i + a x_0$$

$$e p_{zt} \bmod N = \sum r_i v_i + a v_0 \qquad \text{dans les entiers}$$

«Extraction entière» :

$$\phi : \sum r_i u_i + a x_0 \mapsto \sum r_i v_i + a v_0$$

- ϕ est bien définie (pour r_i dans $]-p_i/2, p_i/2]$).
- $\phi(e) = e p_{zt} \bmod N$ **pour e petit.**
- **Pour e grand**, on calcule ϕ en remarquant que ϕ est $\mathbb{Z}$-linéaire ! (tant que les r_i ne dépassent pas p_i)

Attaque : déterminant

Prenons : $n+1$ encodages a_i au niveau 1.
$n+1$ encodages b_i au niveau $\kappa-1$.

On peut écrire :
$$a_i b_j = \sum a_{i,k} b_{j,k} u_k + c_{i,j} x_0$$
$$\phi(a_i b_j) = \sum a_{i,k} b_{j,k} v_k + c_{i,j} v_0$$

Modulo v_0 c'est un produit matriciel !

$$\begin{bmatrix} & \vdots & \\ \dots & \phi(a_i b_j) & \dots \\ & \vdots & \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \dots & a_{i,k} & \dots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \ddots & & 0 \\ & v_k & \\ 0 & & \ddots \end{bmatrix} \begin{bmatrix} & \vdots & \\ \dots & b_{j,k} & \dots \\ & \vdots & \end{bmatrix}^T$$

- Le rang est ≤ n, donc $\det\left(\left[\phi(a_i b_j)\right]\right) = 0 \bmod v_0$.
- $v_0 = \operatorname{pgcd}\left(\det\left(\left[\phi(a_i b_j)\right]\right), \det\left(\left[\phi(a'_i b'_j)\right]\right)\right)$.

Conclusion de l'attaque

L'attaque retrouve v_0 en temps polynomial. On recouvre aussi directement $x_0 = v_0 / p_{zt} \bmod N$.

La connaissance de x_0 ramène essentiellement CLT15 à CLT13.

On remonte ensuite à tous les autres paramètres secrets comme dans [CHLRS15].

Bilan

	Échange de clef multiparti	Obfuscation v1	Obfuscation v2
GGH13	✘	✘	?
CLT13	✘	?	
GGH15	✘		
CLT15	✘	= CLT 13	

Obfuscation v1 : constructions qui supposent que les applications multilinéaires sont sûres.

Obfuscation v2 : constructions qui tiennent compte des attaques existantes et cherchent à les éviter structurellement.

Publications

- Thomas Fuhr and Brice Minaud. Match box meet-in-the-middle attack against KATAN. FSE 2014.
- Brice Minaud. Linear biases in AEGIS keystream. SAC 2014.
- Gregor Leander, Brice Minaud, and Sondre Rønjom. A generic approach to invariant subspace attacks: cryptanalysis of Robin, iSCREAM and Zorro. EUROCRYPT 2015.
- Brice Minaud and Yannick Seurin. The iterated random permutation problem with applications to cascade encryption. CRYPTO 2015.
- Brice Minaud, Patrick Derbez, Pierre-Alain Fouque, and Pierre Karpman. Key-recovery attacks on ASASA. ASIACRYPT 2015. Invited to the Journal of Cryptology.
- Jung Hee Cheon, Pierre-Alain Fouque, Changmin Lee, Brice Minaud, and Hansol Ryu. Cryptanalysis of the new CLT multilinear map over the integers. EUROCRYPT 2016.
- Pierre-Alain Fouque, Pierre Karpman, Paul Kirchner, and Brice Minaud. Efficient and provable white-box primitives. To appear in ASIACRYPT 2016.

Merci !