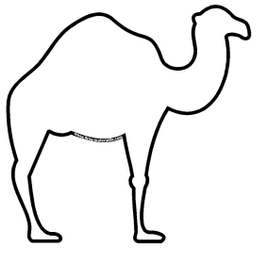
Attaques par sous-espaces invariants sur les chiffrements légers

Cryptanalyse de Robin, iSCREAM et Zorro

Gregor Leander[1], Brice Minaud[2], Sondre Rønjom[3]

[1] Ruhr-Universität Bochum, Germany
[2] ANSSI et Université Rennes 1, France
[3] Nasjonal Sikkerhetsmyndighet, Norway



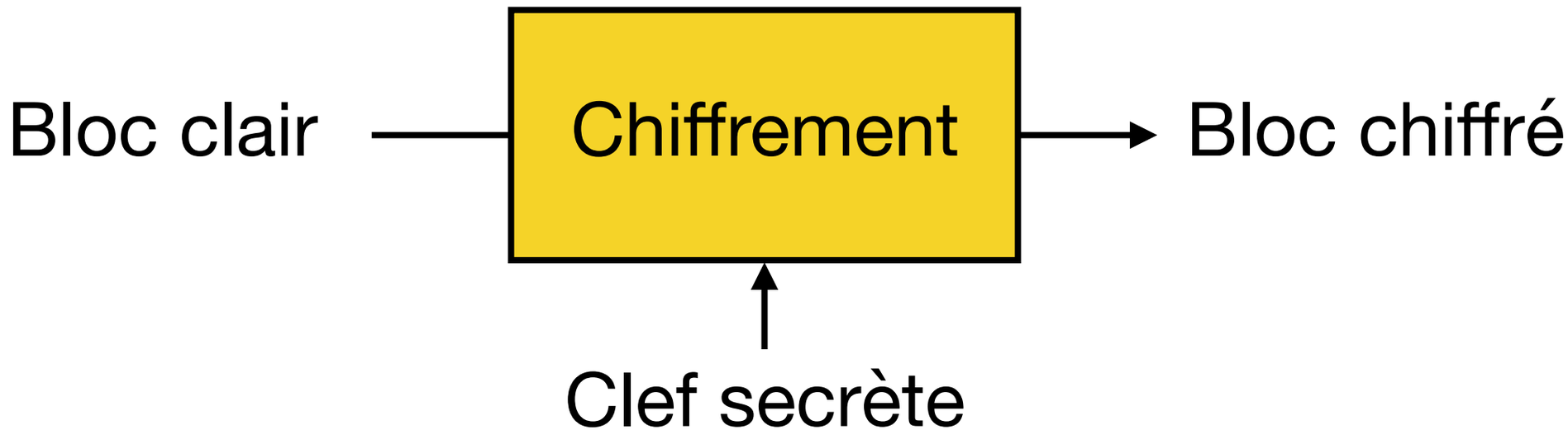
Séminaire Dromadaire EMSEC 2016

Plan

1. Cryptographie légère.
2. Attaques par sous-espaces invariants.
3. Algorithme générique pour chercher ces attaques.
4. Résultats sur Robin, iSCREAM et Zorro.
5. Fonctions linéaires commutant sur Robin et Zorro.
6. Conclusion.

Cryptographie légère : chiffrements par bloc légers

Chiffrements par bloc



Exemples : AES, DES.

On s’intéresse à la primitive cryptographique indépendamment du mode opératoire (CBC, CTR…) ou du protocole.

Chiffrements par bloc

On sait très bien concevoir ce type de chiffrement : AES résiste aux attaques depuis plus de 10 ans.

Algorithmes efficaces, beaucoup implémentés, bien étudiés.

Chiffrements par bloc légers

AES est parfois trop coûteux pour des systèmes embarqués ultralégers. Cependant ces systèmes ont besoin de cryptographie (authentification).

Par ailleurs le contexte de sécurité est différent : on peut se permettre moins de bits de clefs, de plus petits blocs.

AES : bloc 128, clef 128 à 256
PRESENT : bloc 64, clef 80,128
Simon : bloc 32,48,64,96,128, clef 64 à 256

Comparaisons de taille

Nom	Taille (clef/bloc)	Hardware		Software (8-bit MC)	
		Surface (GE)	Ko/s	Flash (oct)	Ko/s
AES 128	**128/128**	**2400**	56	**943**	445
Simon	**128/128**	1317	23	732	342
	32/64	**566**	22	**384**	-
	32/64	722	88		
PRESENT	**64/80**	1030	12	487	96

Source: NSA, document sur Simon et Speck.

Solutions industrielles historiques

À l'origine : algorithmes propriétaires. Lorsqu'ils deviennent public, ils sont généralement cassés assez rapidement.

- **KeeLoq** : pour clefs de voitures électroniques. Cassé par des attaques réalistes (216 challenges, soit 1h de challenge-réponse, + 3j de calculs). 1995-?.

- **DST** : clefs de voiture, systèmes de paiement. 40 bits de clefs, cassable en temps réaliste (1h). ?-2005.

- **CRYPTO1** : notamment transports publics (MIFARE Classic). Cassable en quelques secondes. 2008.

Solutions académiques

Grand nombre de propositions académiques :

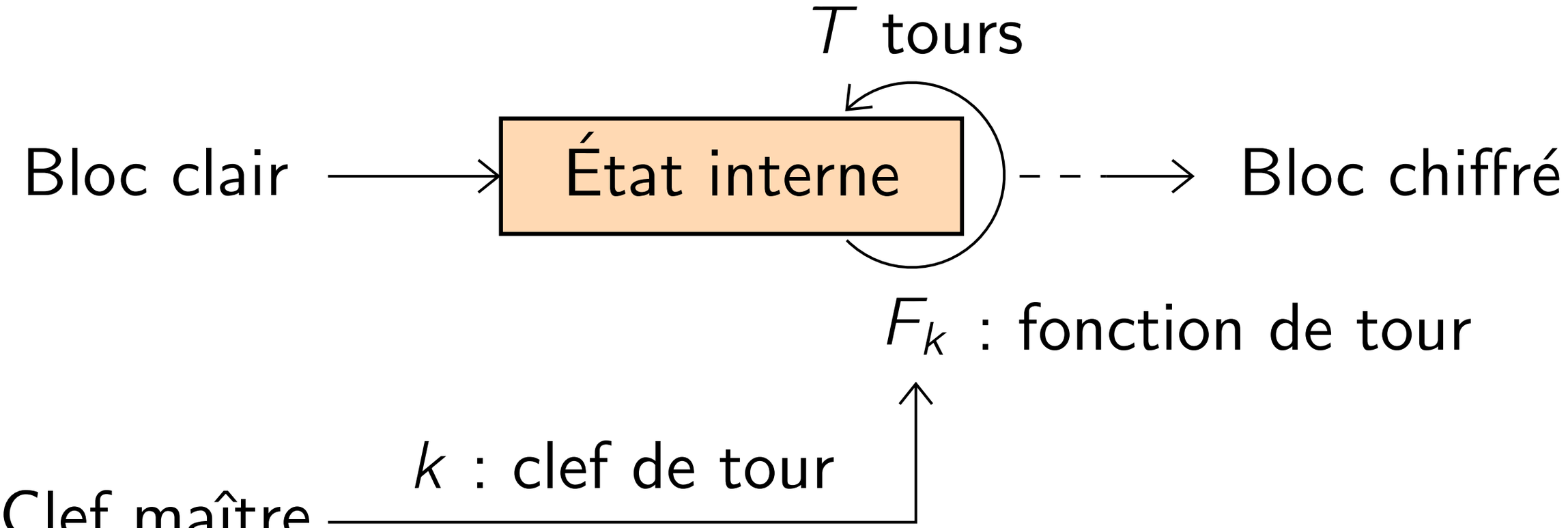
PRESENT (2007), KATAN (2009), PRINTCipher (2010), Piccolo (2011), LED (2011), SIMON (2013), . . .

Réalité : encore rarement utilisés.

Standardisation ISO/IEC : PRESENT, Chaskey.

Attaques par sous-espaces invariants

Chiffrement par bloc : détail



Cadencement de clef : dérivation des clefs de tours à partir de la clef maître.

Invariant Subspace Attacks

Chiffrements légers sur environnements contraints
→ économie maximale.

Cible potentielle : le cadencement de clef.

Certains chiffrements légers n’ont pas du tout de cadencement de clef :

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

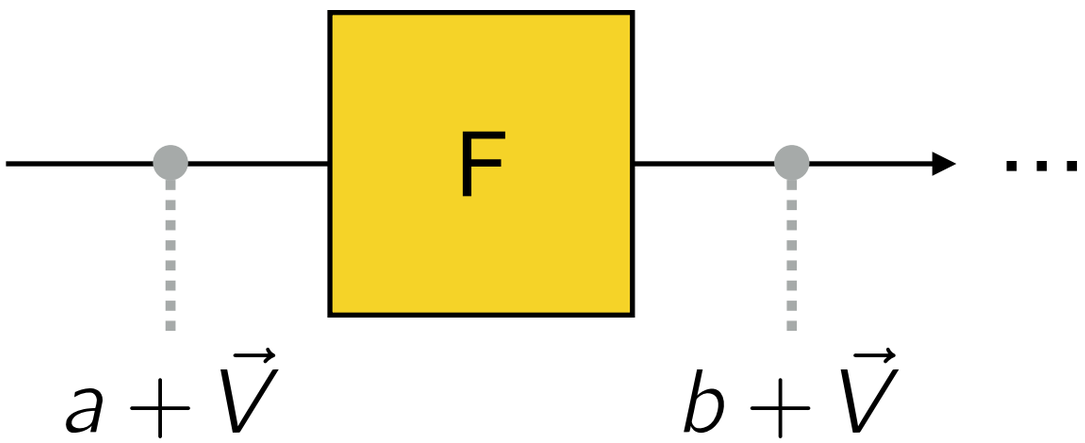
Attaques par sous-espaces invariants

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

Utilisées pour casser PRINTCIPHER en temps pratique [LAKZ11].

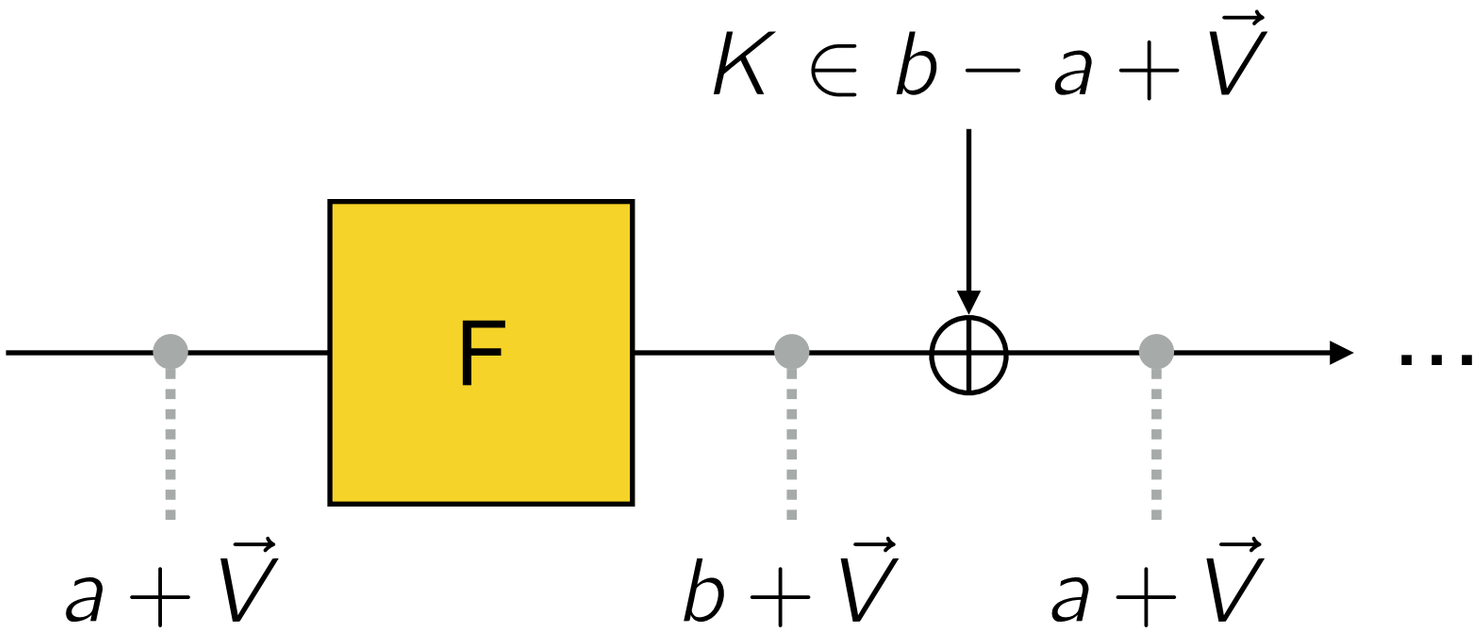
S’appuie sur l’absence de cadencement de clef.

Sous-espaces invariants



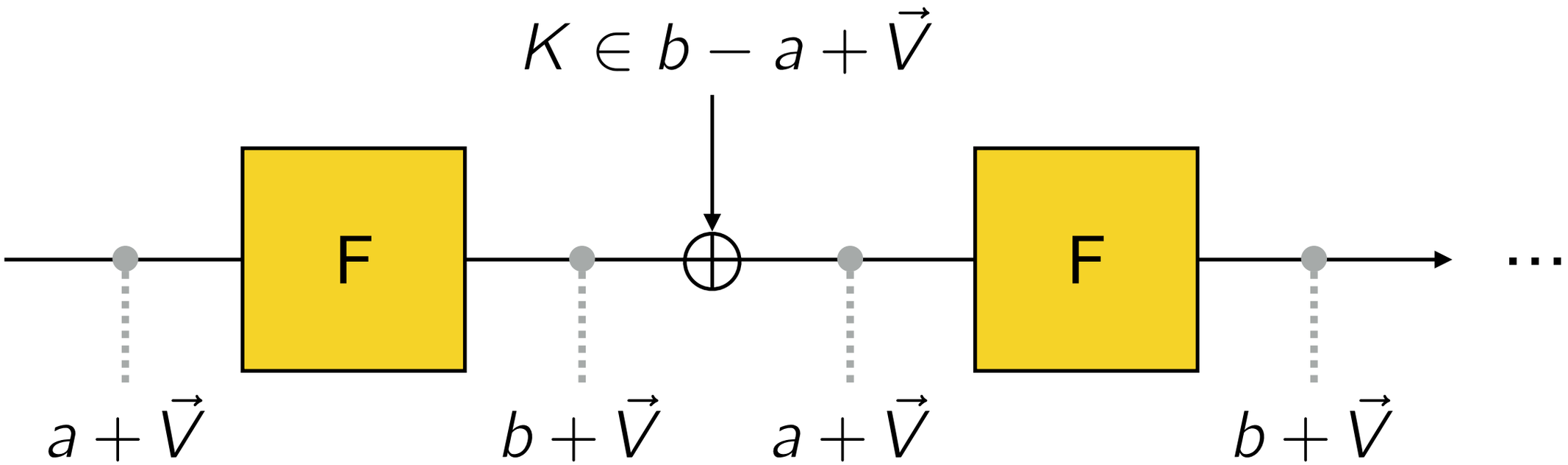
Supposons que la fonction de tour envoie un espace vectoriel sur un coset du même espace.

Sous-espaces invariants



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

Sous-espaces invariants

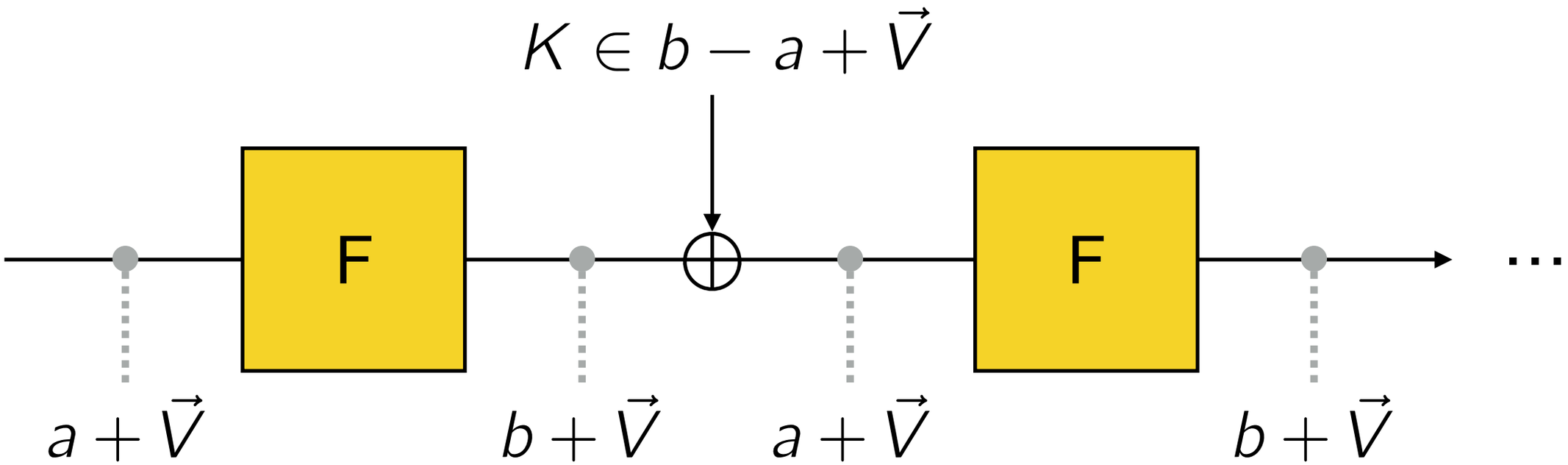


Supposons aussi $K \in b - a + \vec{V}$...

Alors le processus se répète.

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

Sous-espaces invariants

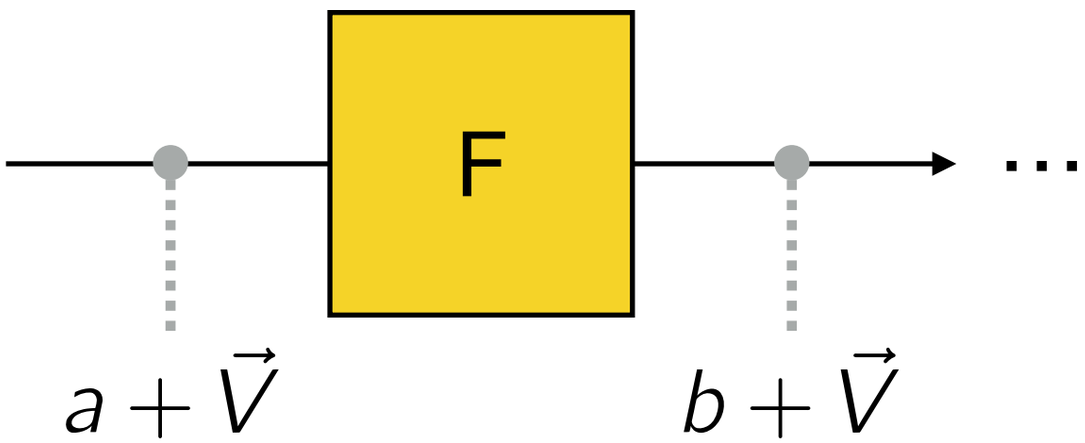


La confidentialité est perdue.

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

Trouver les sous-espaces invariants: un algorithme générique

Un algorithme générique

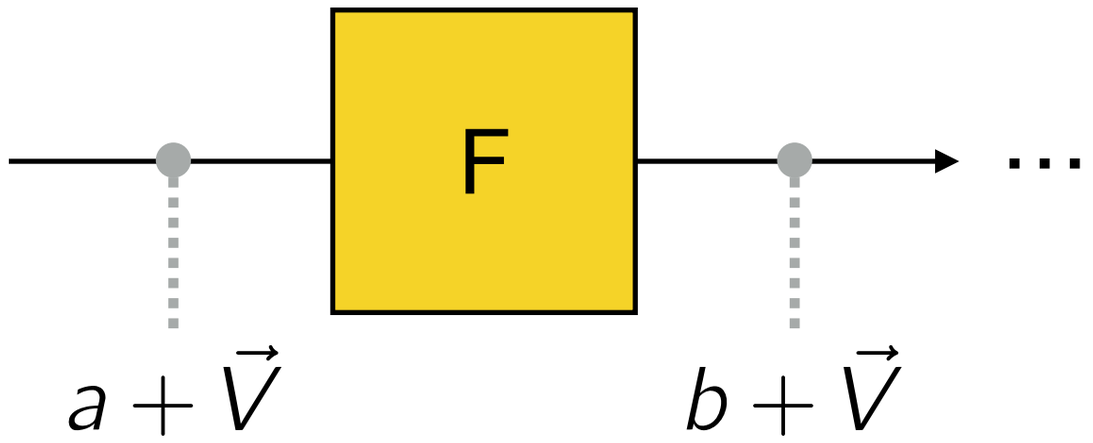


Bootstrap: supposons connus $s, t \in a + \vec{V}$

Alors $F(s), F(t) \in b + \vec{V}$ donc $F(s) - F(t) \in \vec{V}$

$\Rightarrow$ On connaît un nouveau vecteur de $\vec{V}$.

Un algorithme générique



Algorithme de “**Fermeture**”

Entrée: s, W tels que $s + W \subseteq a + V$

Sortie: $a + \vec{V}$

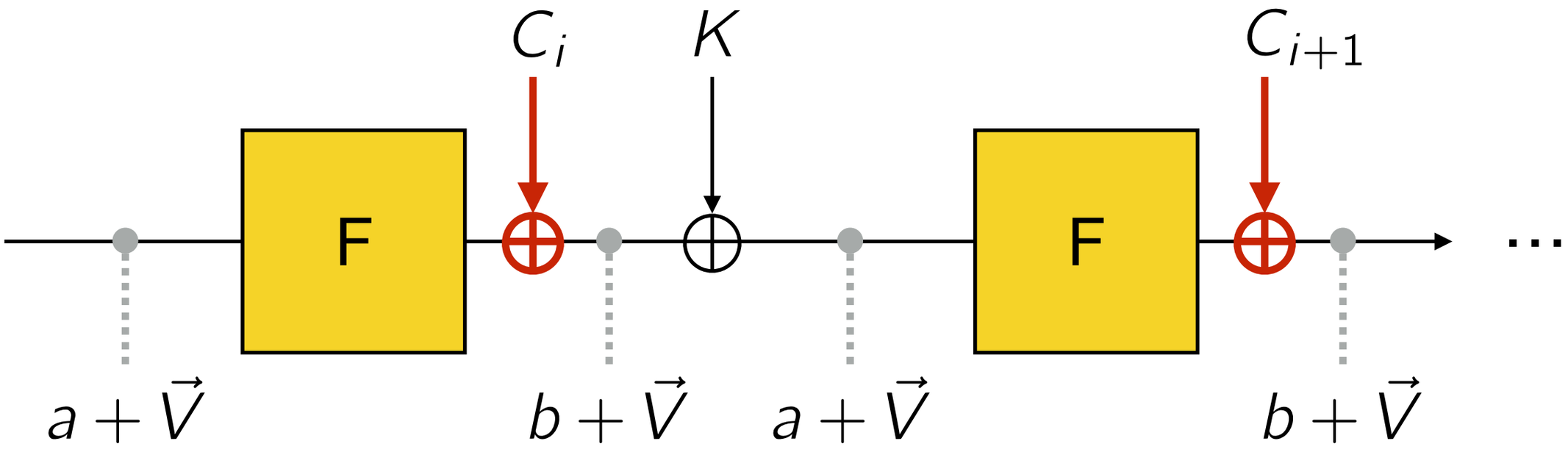
1. Tirer $w \leftarrow_{\$} \vec{W}$
2. Ajouter $F(s + w) - F(s)$ à $\vec{W}$
3. Réitérer les étapes 1 et 2 jusqu’à ce que $\vec{W}$ reste stable pendant N itérations.
4. Renvoyer $s + \vec{W}$

Un algorithme générique

Quelques remarques…

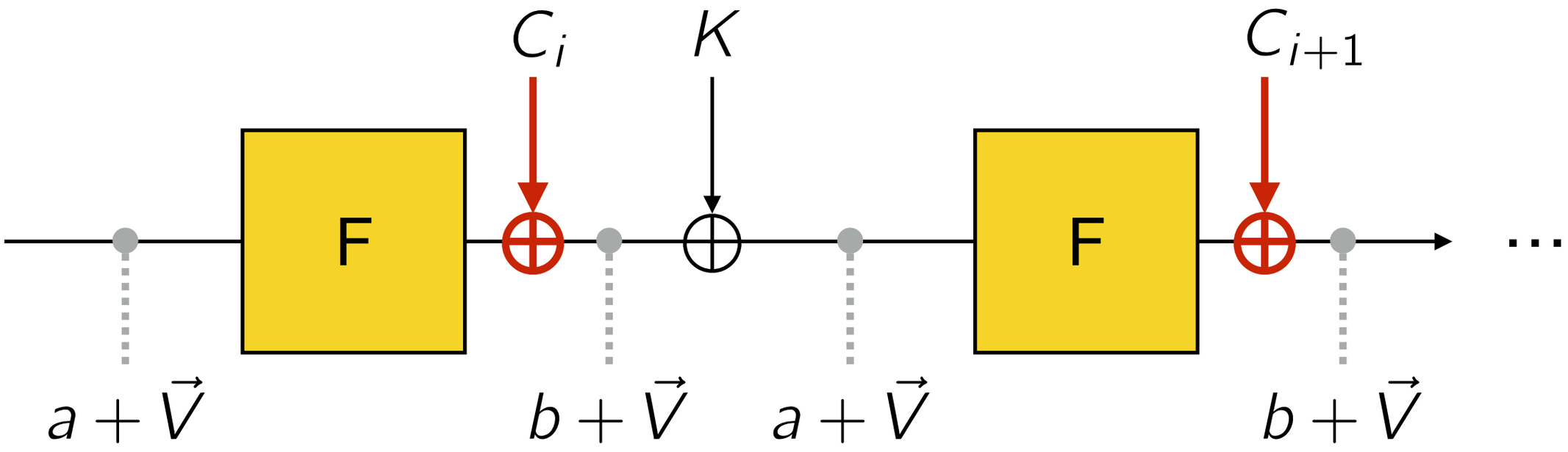
- L’algorithme renvoie seulement le plus petit espace invariant contenant l’entrée.
- … on doit encore bootstraper.

Bootstrap de l'algorithme



On a un peu menti. Il y a des constantes de tour.

Bootstrap de l'algorithm



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 $a \neq 0$, il reste à trouver un offset $s \in a + \vec{V}$.
On essaie simplement des offsets aléatoires.

Complexité

Algorithm générique

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

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

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

Propriétés de l'algorithme

- Générique: fonctions de tour utilisées en boîtes noires.
- Ne réfute pas l'existence de "petites" espaces.
- Implémentation publique : `http://invariant-space.gforge.inria.fr`

Résultats sur Robin, iSCREAM et Zorro

Robin, iSCREAM et Zorro

Robin et Fantomas: chiffrements légers, créés pour illustrer les LS-designs, FSE 2014 [GLSV14].

SCREAM et **iSCREAM**: variantes authentifiées de Robin et Fantomas, pour la compétition CAESAR.

Zorro: chiffrement léger avec couche non-linéaire partielle [GGNS13]. Cassé par attaques linéaires et différentielles. Meilleure attaque : 2^{40} données et 2^{40} complexité [BDDLKT14].

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 **NOEKEON** **LED** **Keccak**	Avec probabilité 99.9%: Pas d'espace invariant de codimension < 32	

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

Fonctions linéaires qui commutent

Robin

Robin and Fantomas [GLSV14], FSE 2014.

Chiffrements légers avec masquage facilité.

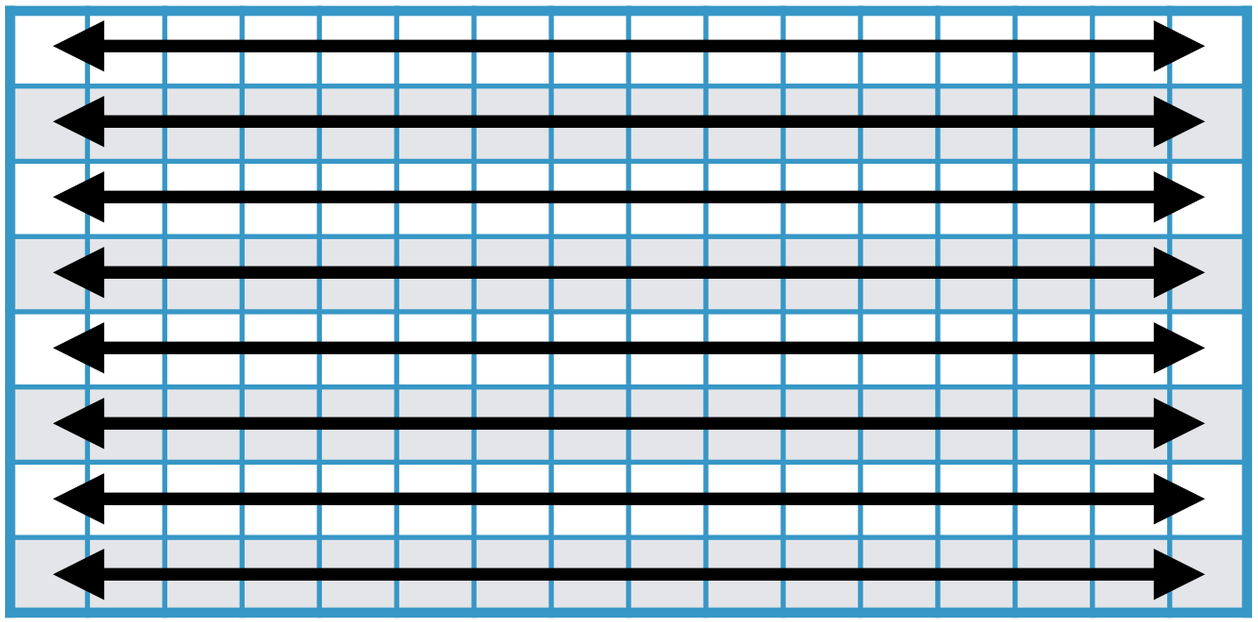
Bloc =128 bits — Sécurité = 128 bits

Robin = version involutive.

Design simple et élégant : “LS-design”.

Robin: couche **L**

Couche **L**

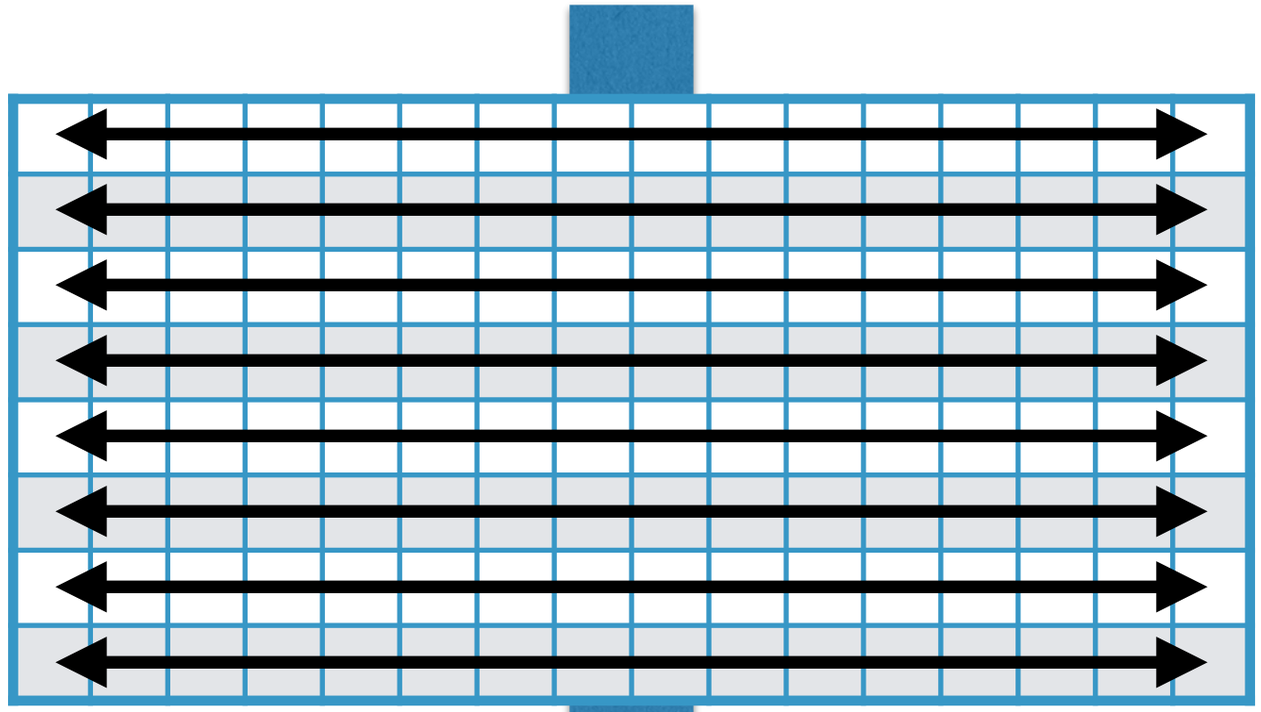


État
16 x 8 bits

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

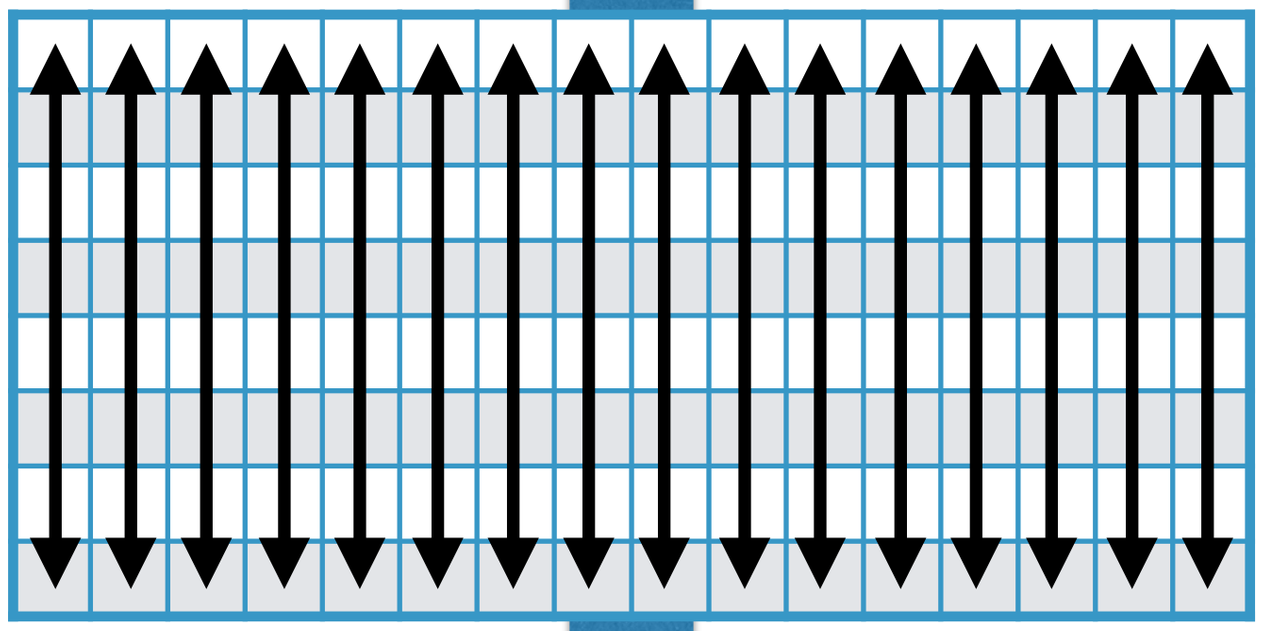
Robin: couches **LS**

Couche **L**



Même fonction linéaire sur chaque rangée

Couche **S**



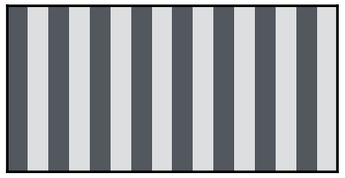
Même boîte S sur chaque colonne

Fonction de tour de Robin

Un tour =

- Couche **L**

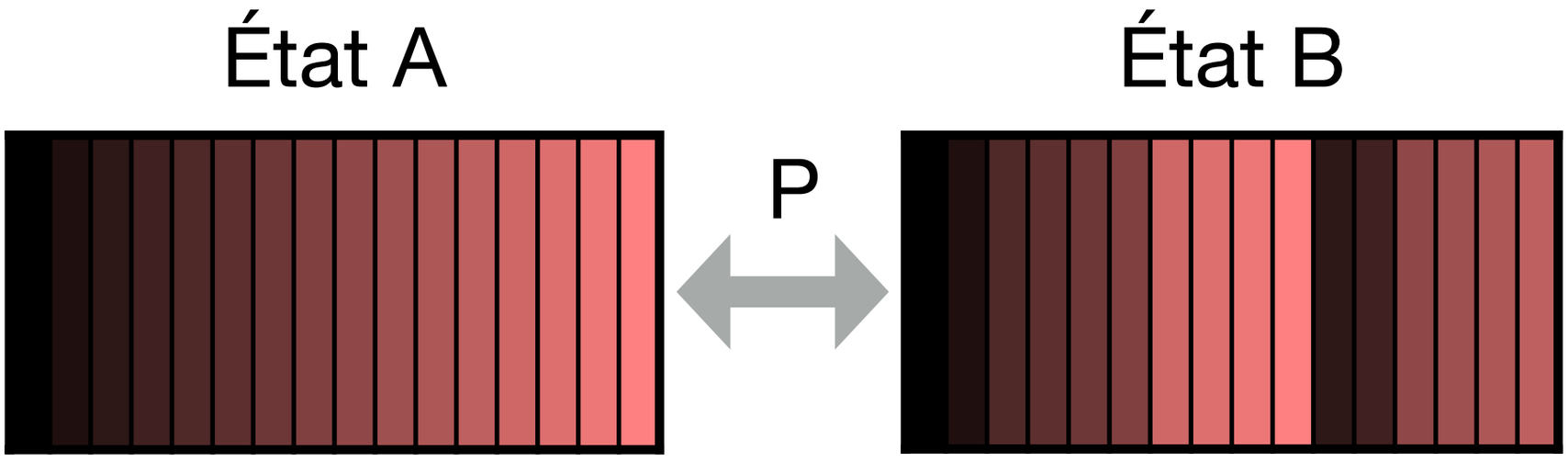
- Couche **S**



- Ajout de constante de tour
- Ajout de clef

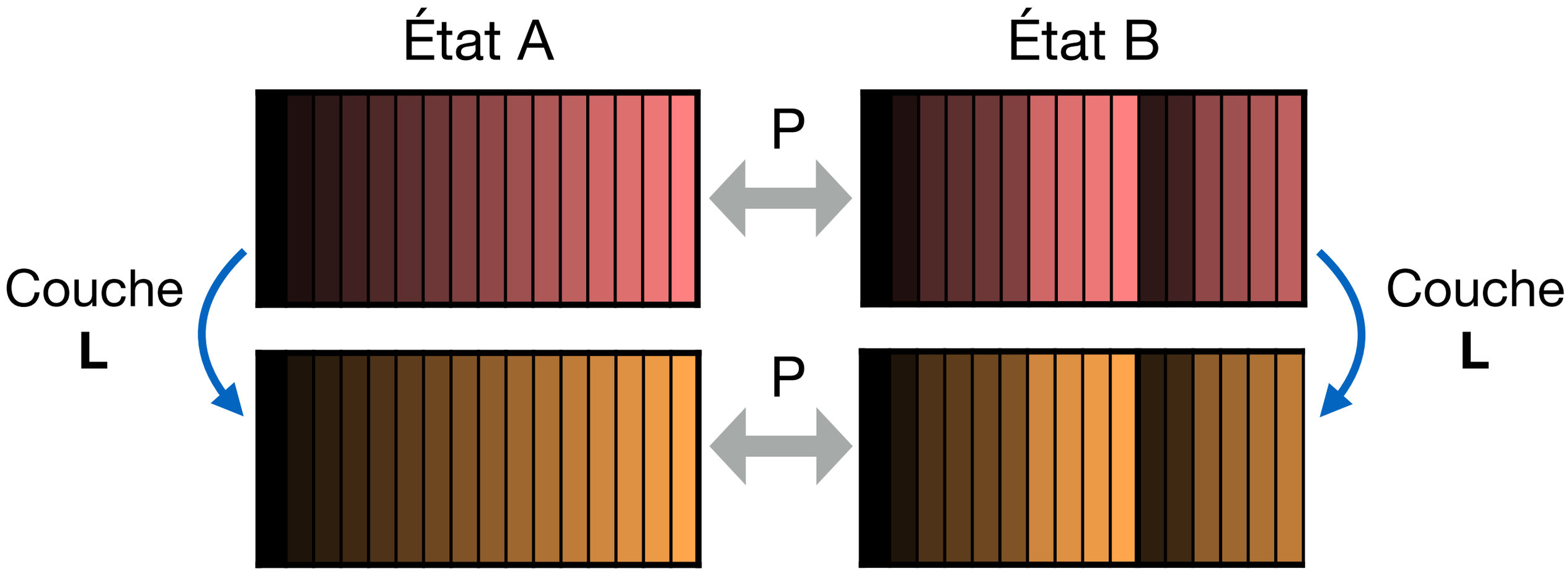
Un chiffrement = 16 tours.

Permutations invariantes



État B = permutation des colonnes de l’état A

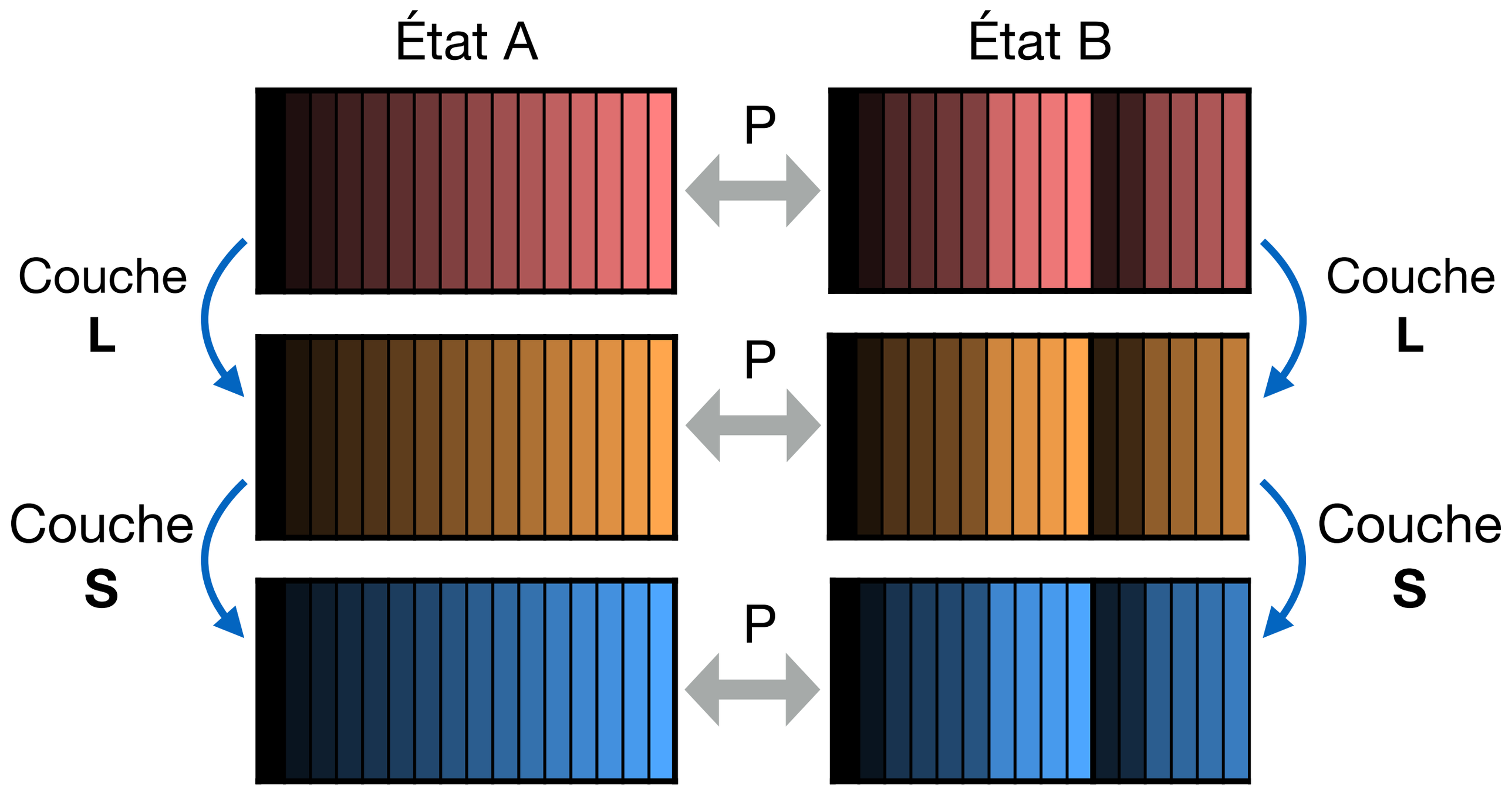
Permutations invariantes



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

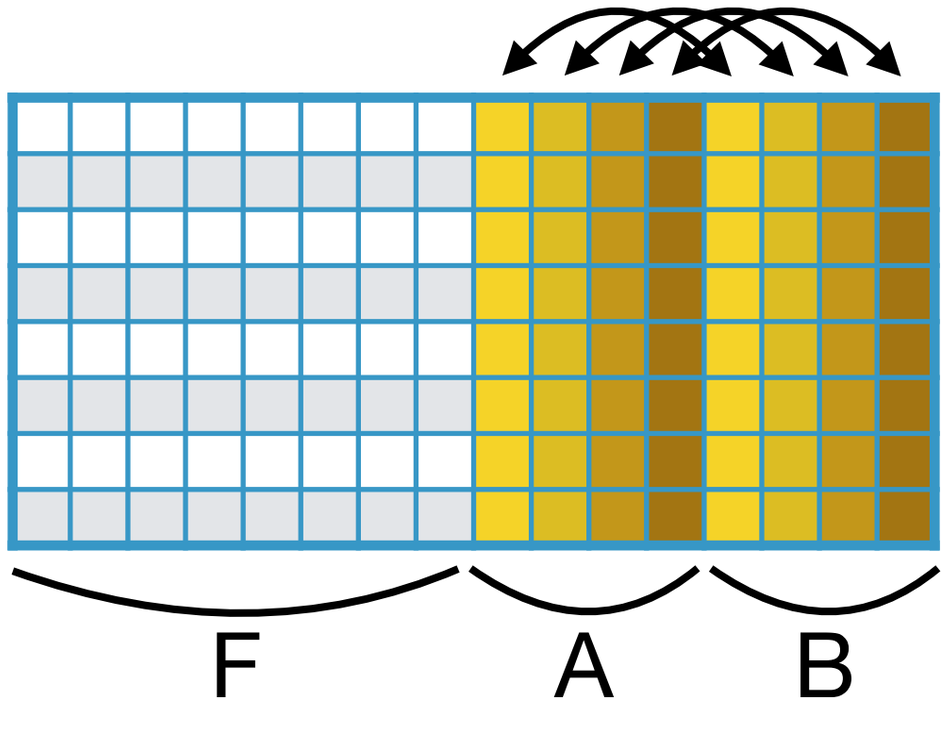
chiffrés *auto-reliés* $C = P(C)$.

Ceci est une attaque par sous-espace invariant !

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

Sous-chiffrement

Permutation P : échange 8 colonnes



Soient *m*, *m'* **auto-reliés**, π_F, π_A, π_B projections sur F,A,B.

Prop 1. Si $\pi_F(m) = 0$ alors $\pi_F(L(m)) = 0$.

$$P(\pi_A(m)) = \pi_B(m) \Rightarrow P \circ L(\pi_A(m)) = L(\pi_B(m)) \Rightarrow \pi_F(L(\pi_A(m)+\pi_B(m))) = 0.$$

Prop 2. Si $\pi_F(m) = \pi_F(m')$ alors $\pi_F(E_K(m)) = \pi_F(E_K(m'))$.

De plus cette valeur ne dépend que de $\pi_F(K)$.

Attaque sur Robin et iSCREAM

Robin et iSCREAM : une permutation adéquate P.

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

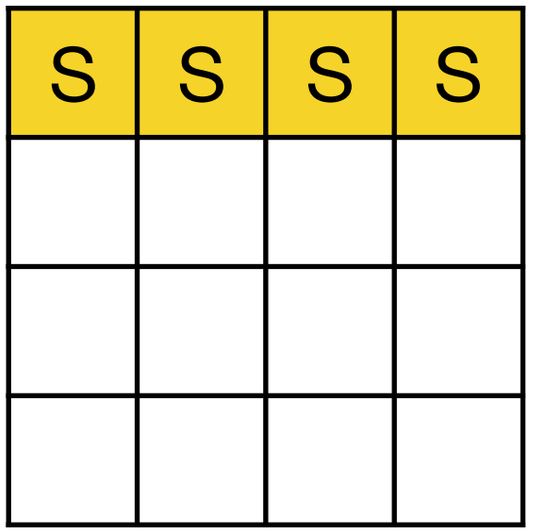
De plus, pour des clefs faibles:

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

Robin vs Zorro

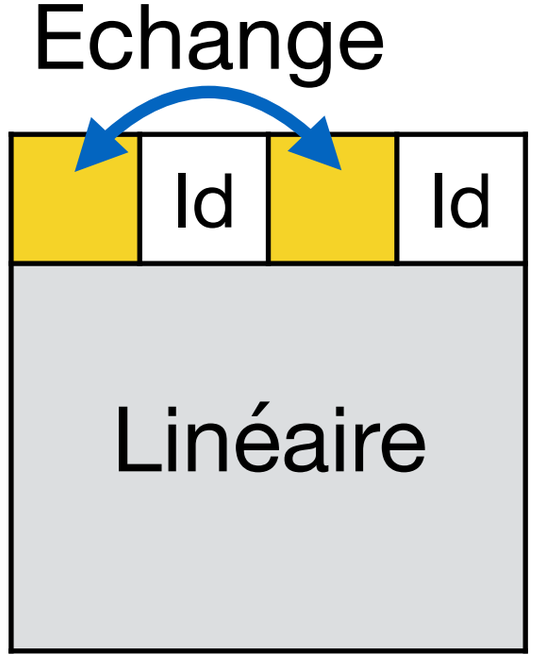
Zorro est un variante d'AES avec quelques différences notables :

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



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

$M =$



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

Comparaison des attaques

	Type	Data	Time	Reference
Robin, iSCREAM	Clefs faibles, densité 2^{-32}	2 CP	négligeable	**ce papier**
Zorro	Clefs faibles, densité 2^{-32}	2 CP	négligeable	**ce papier**
	Différentiel	$2^{41.5}$ CP	2^{45}	[BDDLKT14]
	Linéaire	2^{45} KP	2^{45}	[BDDLKT14]

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.

  Propriétés liées : fonction linéaire qui commute, sous-chiffrement…

- Épilogue : iSCREAM éliminé de CAESAR.

  Petit sous-espace invariant sur Midori (2015).

Conclusion

Merci pour votre attention!

Questions ?