Keccak and permutation-based symmetric cryptography

Gilles Van Assche[1]

En collaboration avec
Guido Bertoni[1], Joan Daemen[1], et Michaël Peeters[2]

[1]STMicroelectronics [2]NXP Semiconductors

Séminaire CCA, INRIA, Paris, 11 janvier 2013

Plan

Plan

Ce que disent les livres sur la crypto symétrique

Primitives en cryptographie symétrique :

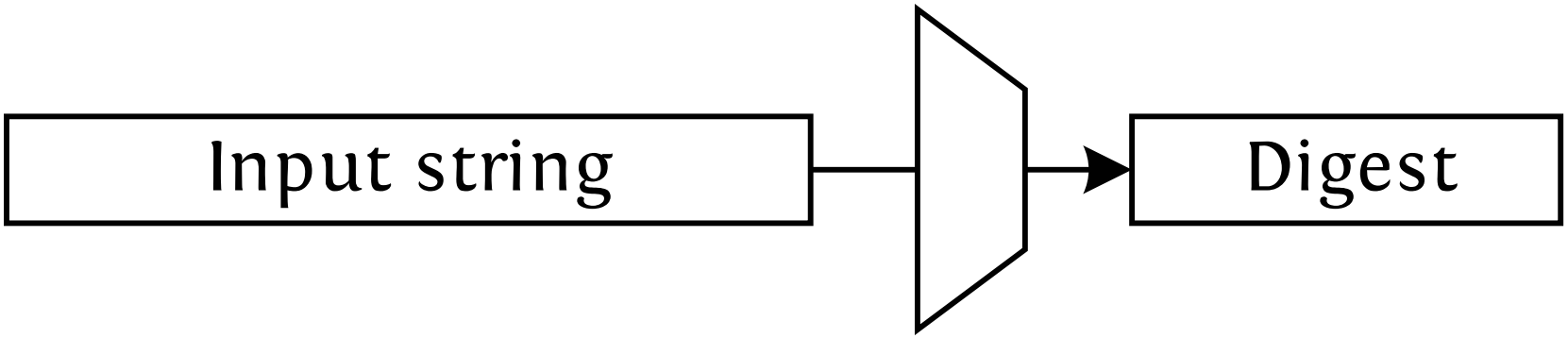
- Chiffrement de bloc (block ciphers)
- Chiffrement de flux (stream ciphers)
- Fonctions de hachage (hash functions)
  - Sans clé
  - Avec clé : authentification (MAC)

Et leurs mode d'utilisation...

Fonction de hachage = couteau suisse de la crypto ?

Fonction de hachage

$$h : \{0,1\}^* \rightarrow \{0,1\}^n$$

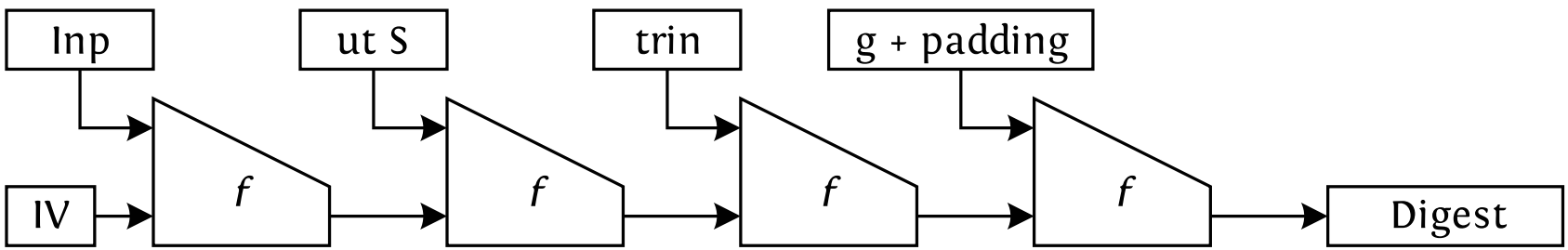


- MD5 : $n = 128$ (Ron Rivest, 1992)
- SHA-1 : $n = 160$ (NSA, NIST, 1995)
- SHA-2 : $n \in \{224, 256, 384, 512\}$ (NSA, NIST, 2001)

Hachage en deux couches

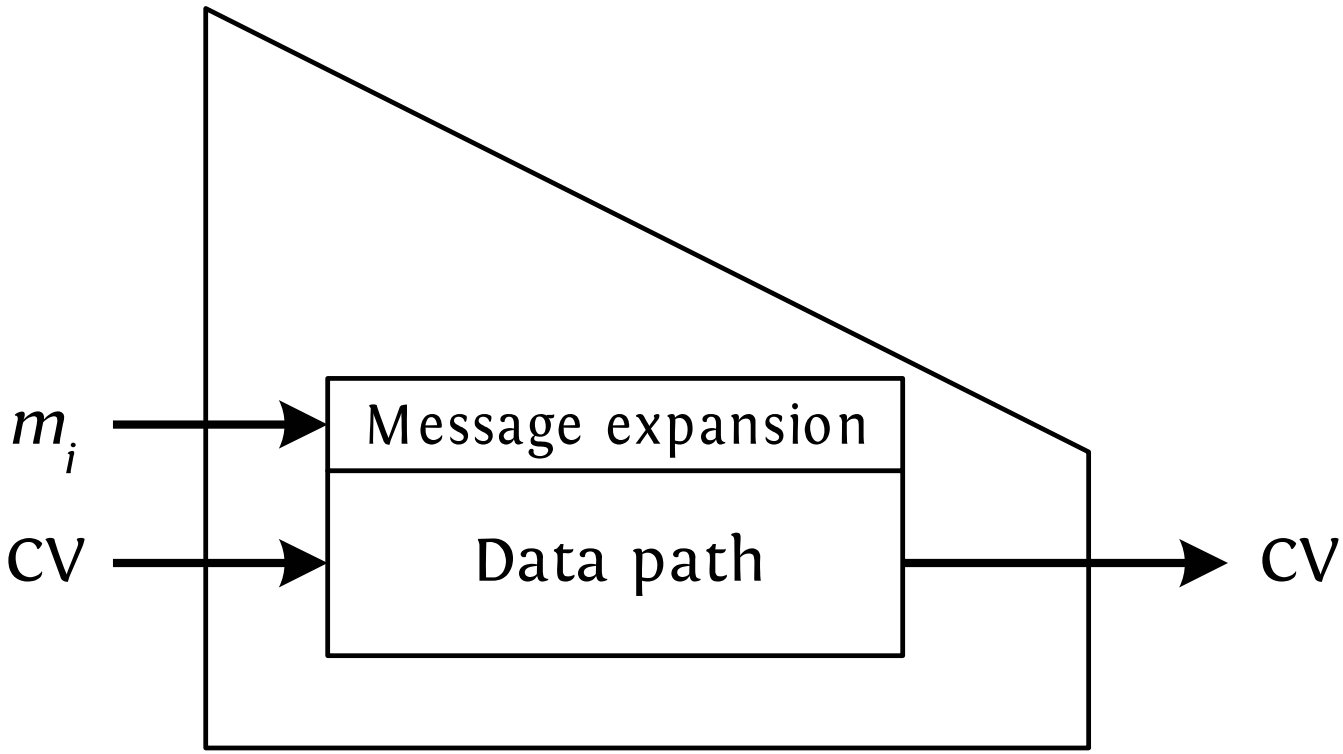
1. Mode d'itération (entrée infinie $\{0,1\}^*$)
2. Fonction de compression (primitive à entrée finie)

Un mode d'itération : Merkle-Damgård



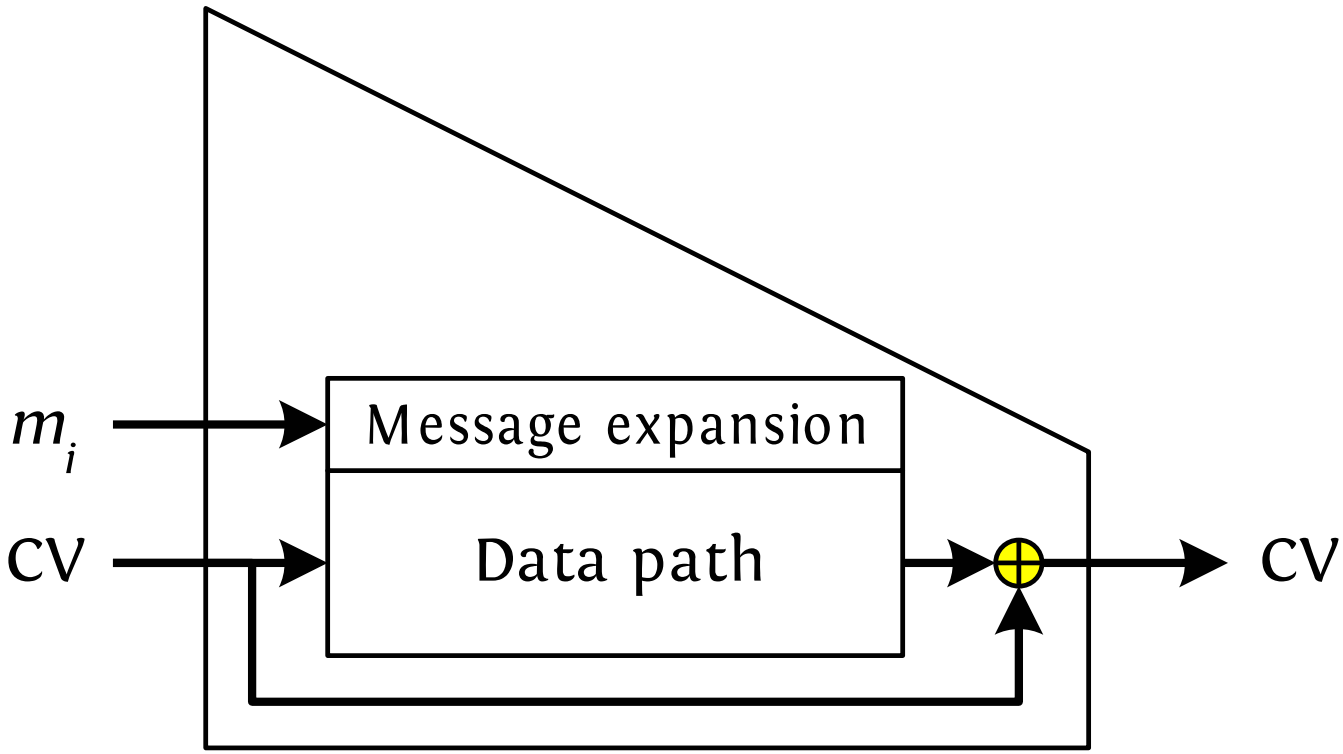
MD5, SHA-1, SHA-2

Du chiffrement pour le hachage : Davies-Meyer (±)



Pas entièrement à sens unique !

Du chiffrement pour le hachage : Davies-Meyer



Sens unique pour la valeur de chaînage aussi

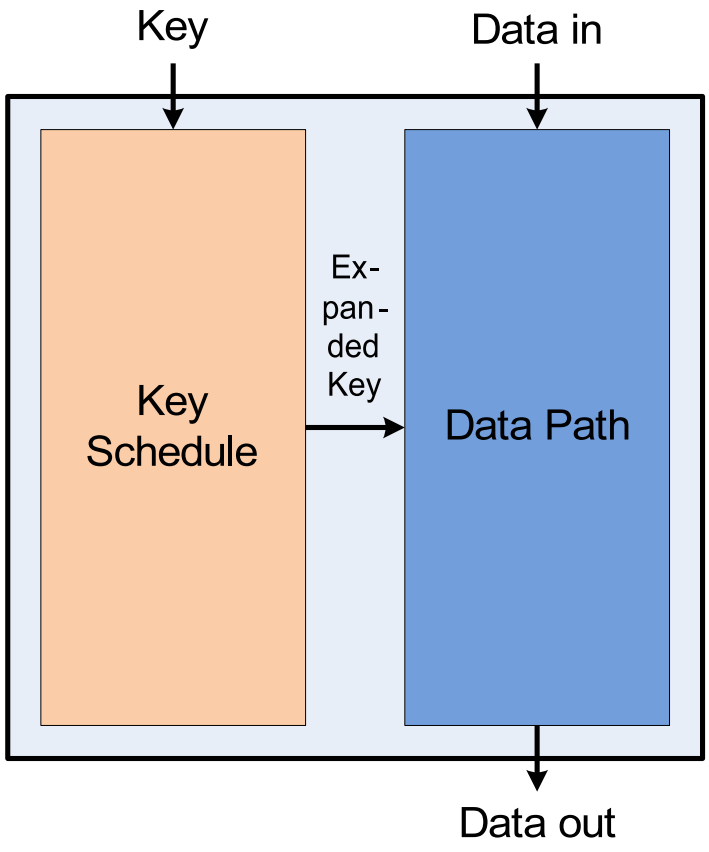
Autres modes pour le chiffrement de bloc

- Chiffrement : ECB, CBC, ...
- Chiffrement de flux :
  - synchrone : CTR, OFB, ...
  - auto-synchrone : CFB
- Authentification : CBC-MAC, C-MAC, ...
- Chiffrement authentifié : OCB, GCM, CCM, ...

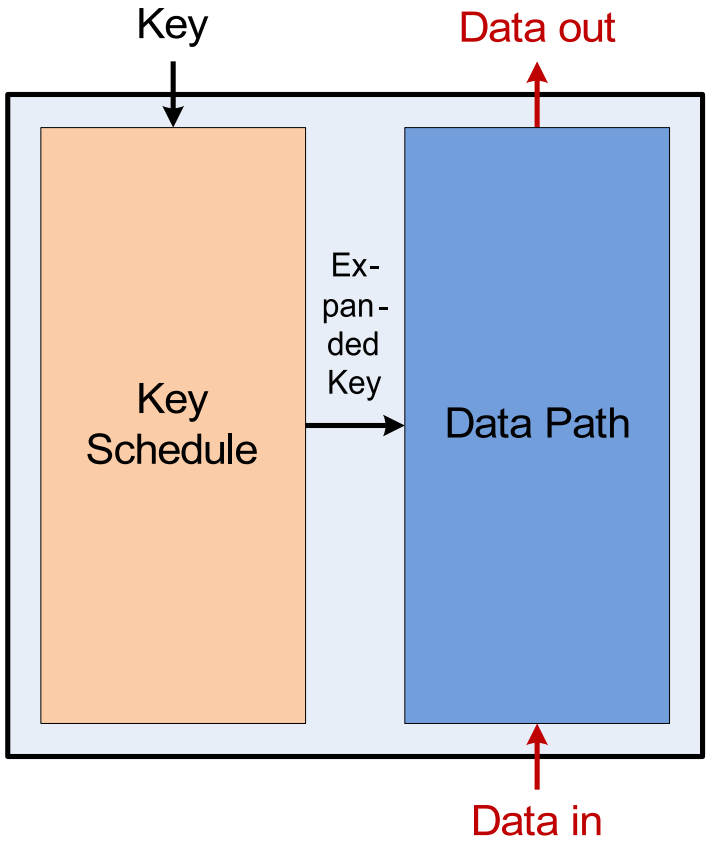
Chiffrement de bloc = couteau suisse de la crypto !

Plan

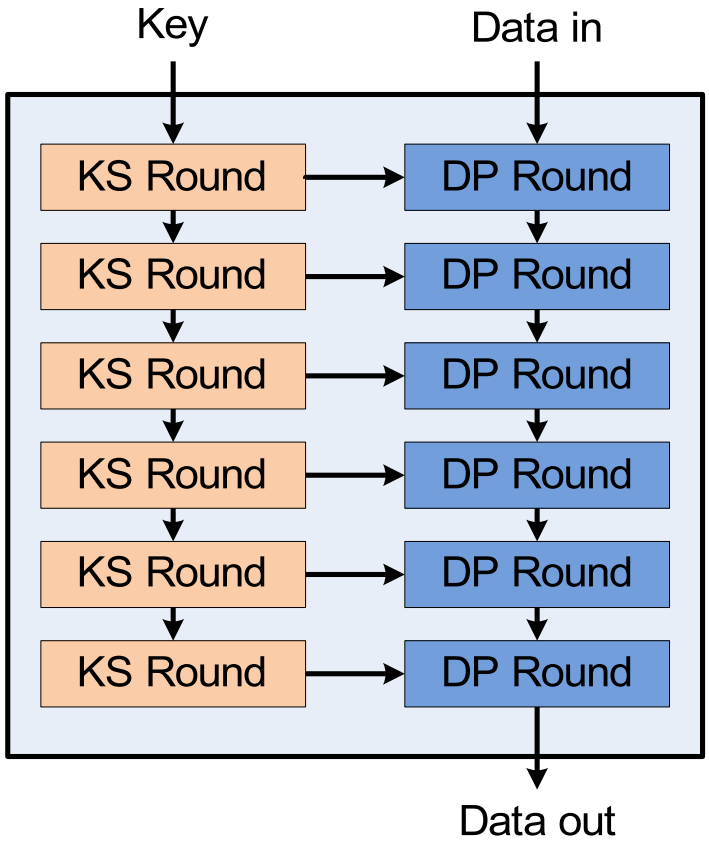
Structure d'un algorithme de chiffrement de bloc



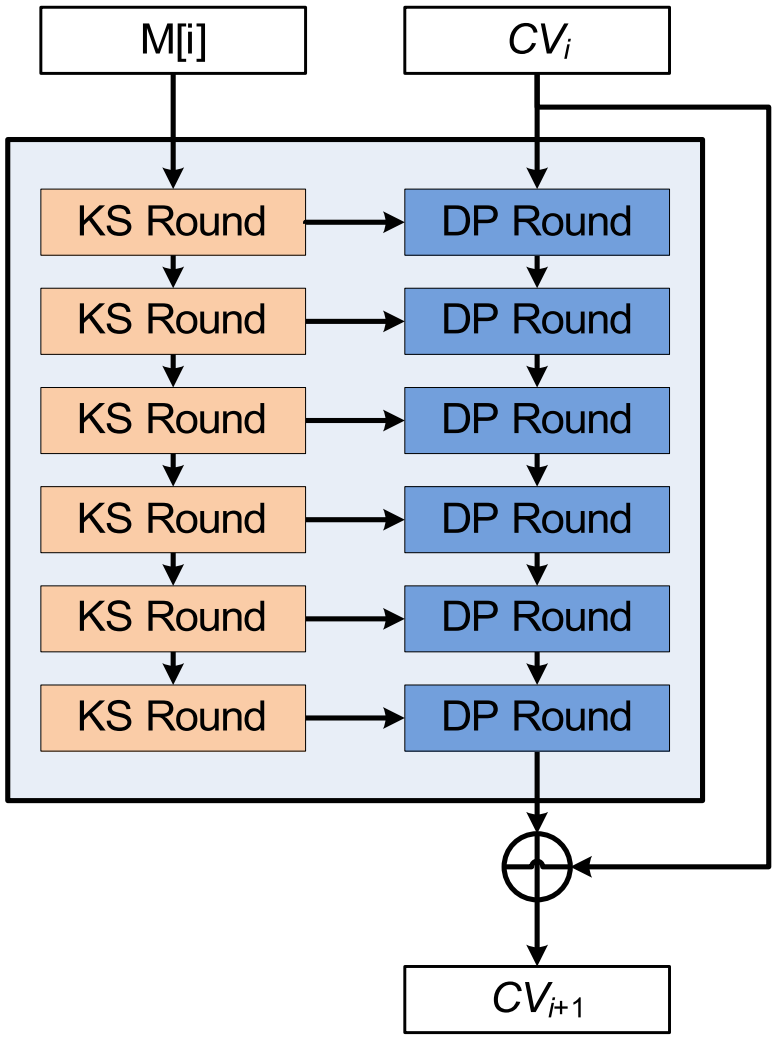
Structure d'un algorithme de chiffrement de bloc



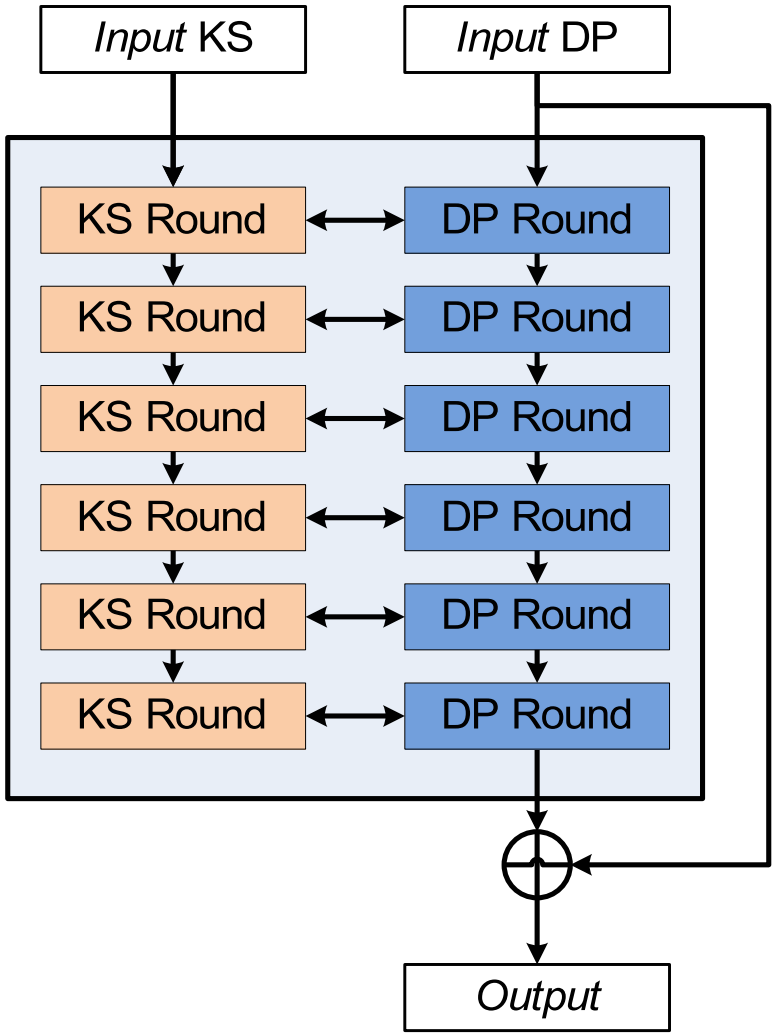
Structure d'un algorithme de chiffrement de bloc



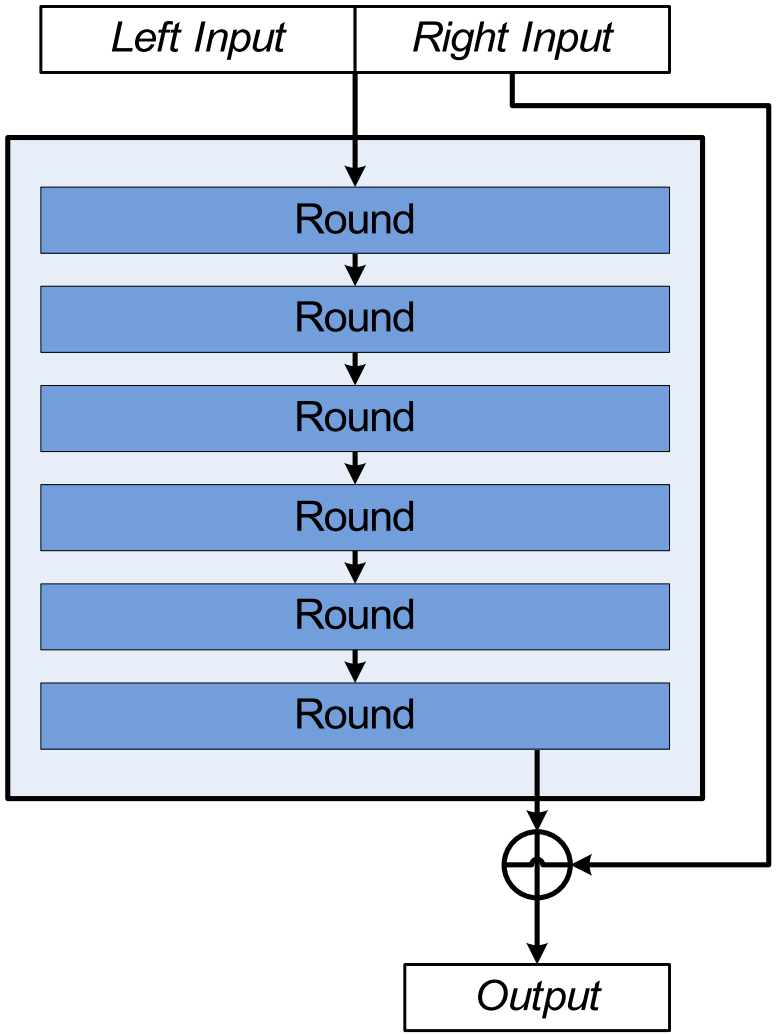
Avec Davies-Meyer



Pourquoi restreindre la diffusion ?



Simplifions !

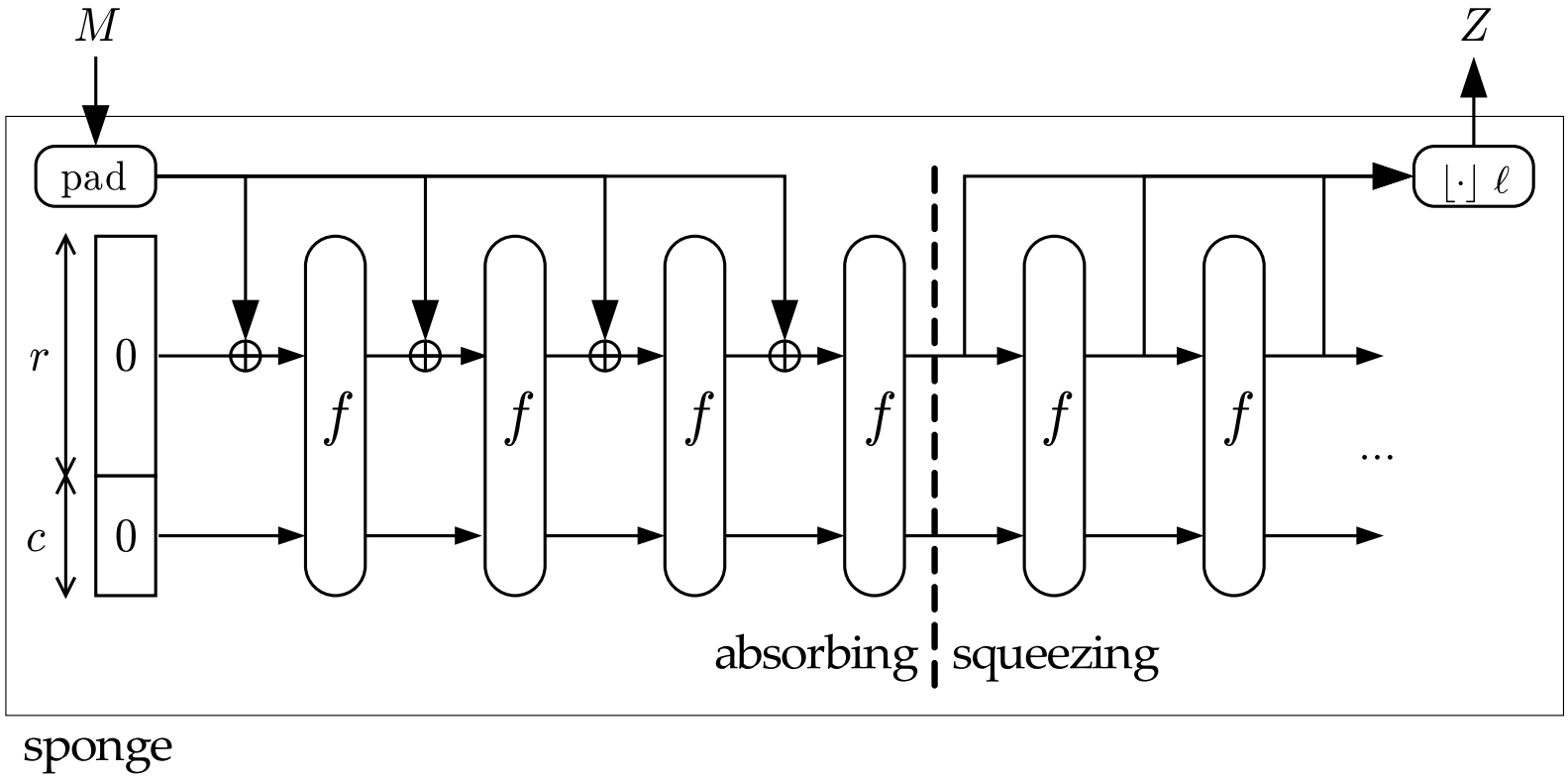


Utilisation d'une permutation

- Hachage : Snefru [Merkle '90], FFT-Hash [Schnorr '90], Grindahl [Knudsen, Rechberger, Thomsen '07], MD6 [Rivest et al. '07], Grøstl [Gauravaram, Knudsen, Matusiewicz, Mendel, Rechberger, Schläffer, Thomsen '08], JH [Wu '08], ...
- Hachage et chiffrement de flux : Subterranean [Daemen '91], Panama [Daemen et Clapp '98], RadioGatún, ...
- Authentification : Pelican-MAC [Daemen, Rijmen '05]
- Chiffrement de flux : Salsa et ChaCha [Bernstein '07]
- Fonctions-éponges...

Plan

La construction « éponge »



sponge

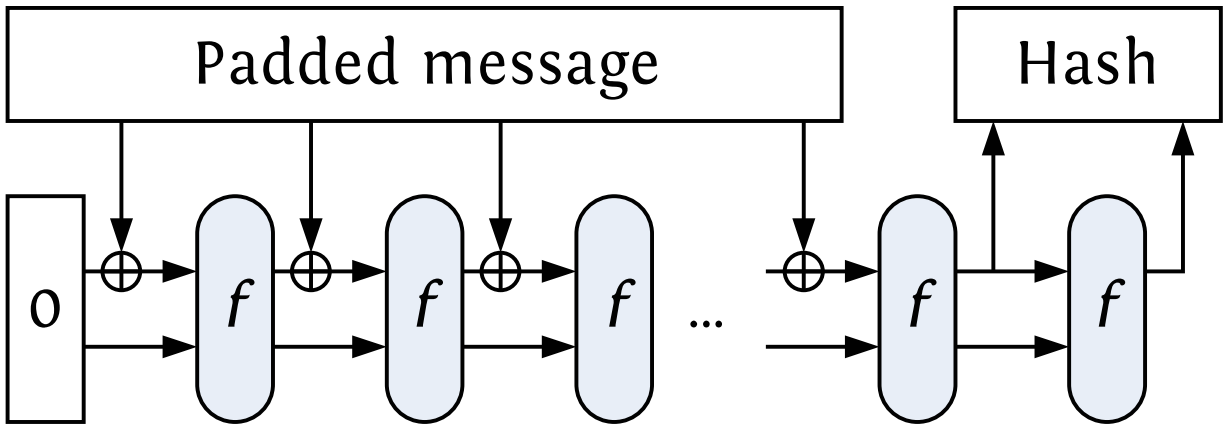
- r : taille des blocs (rate)
- c : capacité

Fonctions-éponges aléatoires

Comme un oracle aléatoire en dessous de $2^{c/2}$

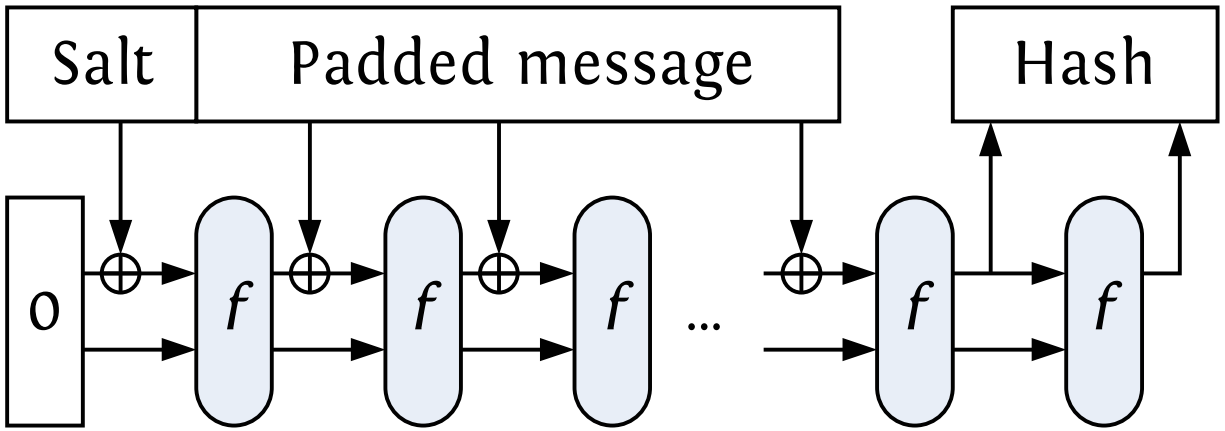
Absence d'attaques génériques en moins de $2^{c/2}$ appels à f

Comment utiliser une fonction-éponge ?



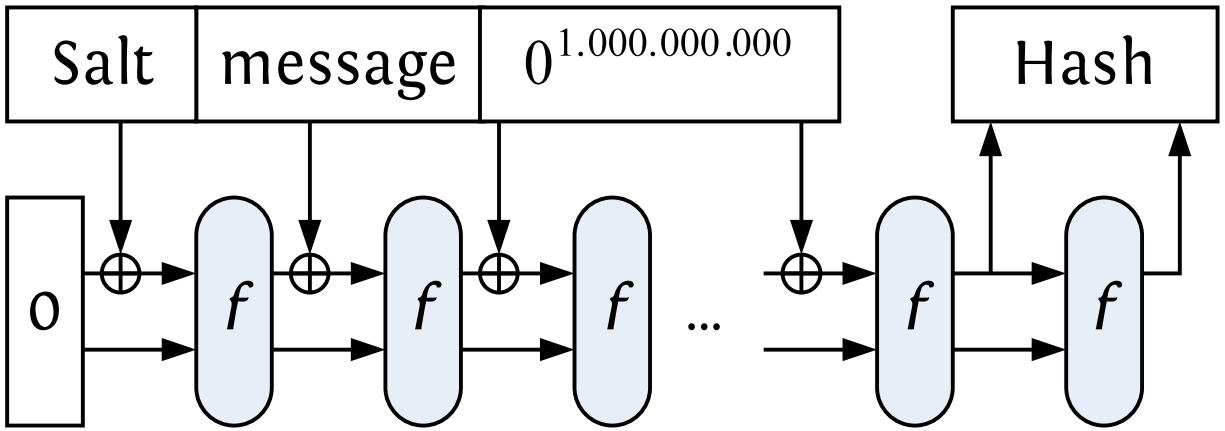
- Pour du hachage

Comment utiliser une fonction-éponge ?



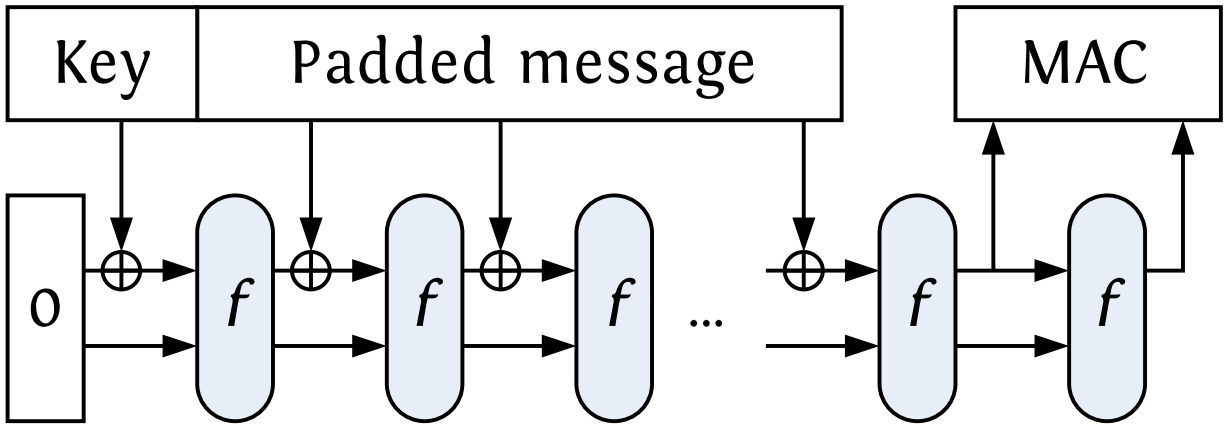
- Pour du hachage (avec sel)

Comment utiliser une fonction-éponge ?



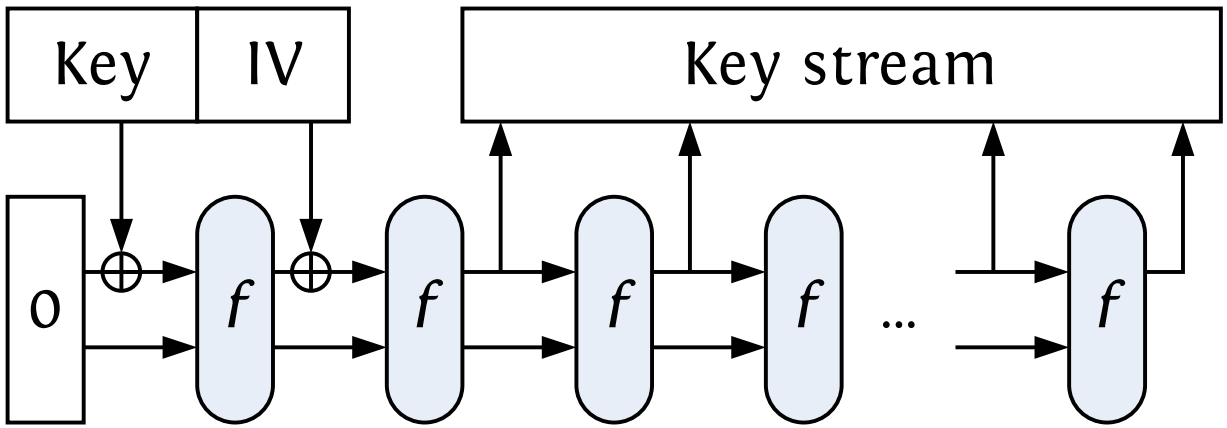
- Pour du hachage lent (mots de passe)

Comment utiliser une fonction-éponge ?



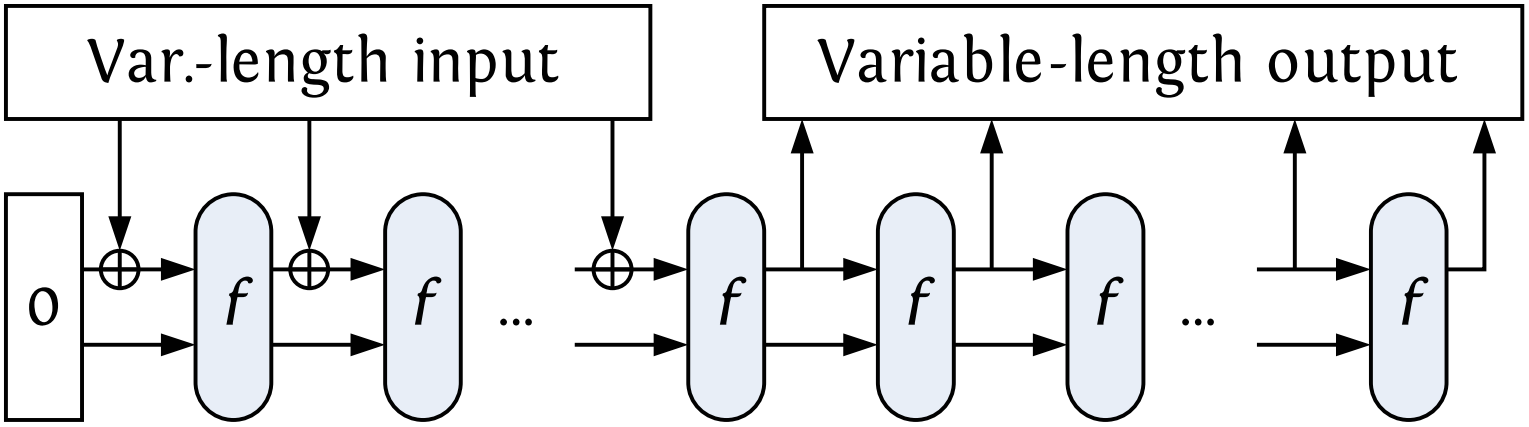
- Pour un code d'authentification (MAC)

Comment utiliser une fonction-éponge ?



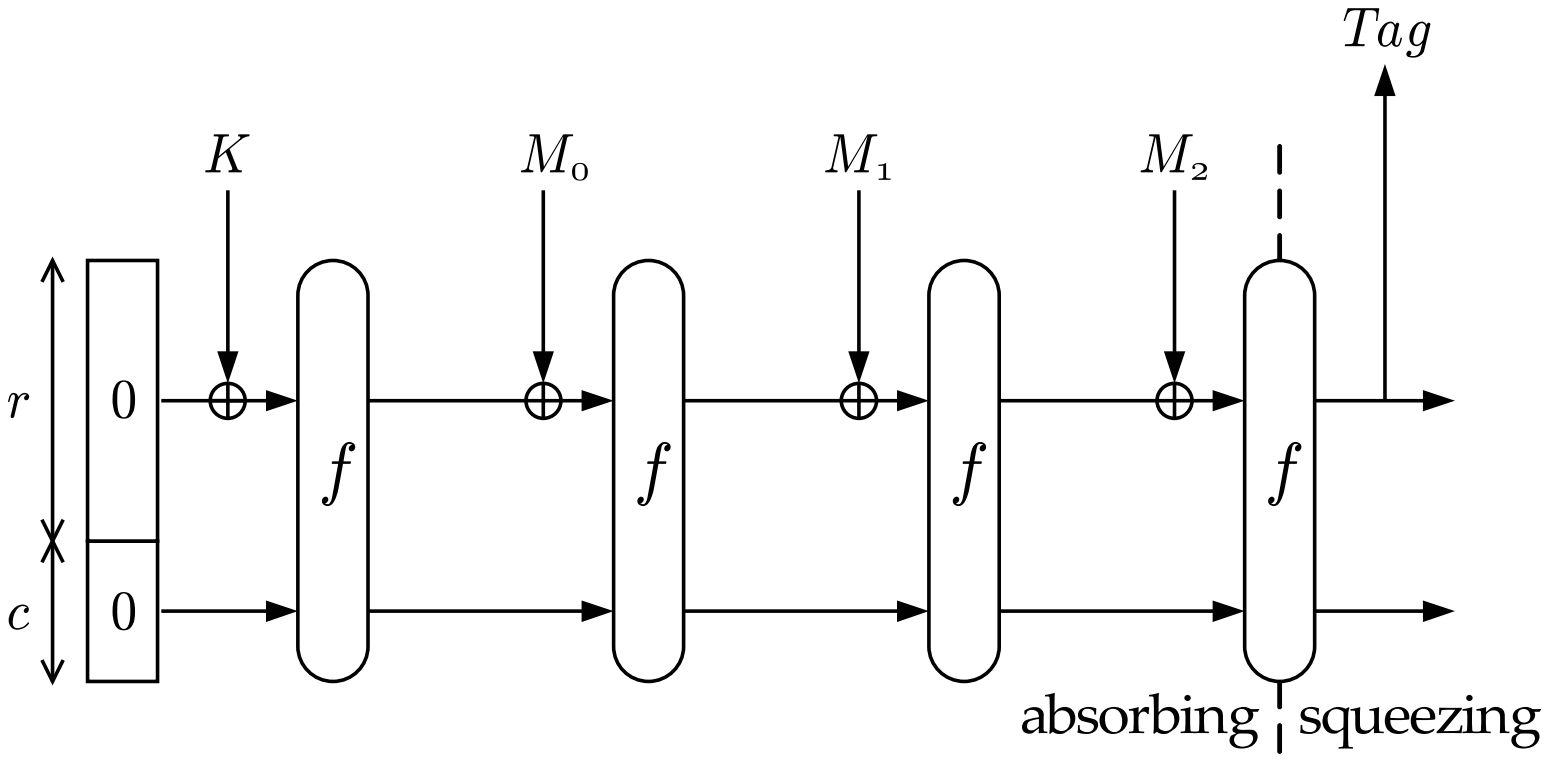
- Pour du chiffrement de flux

Comment utiliser une fonction-éponge ?

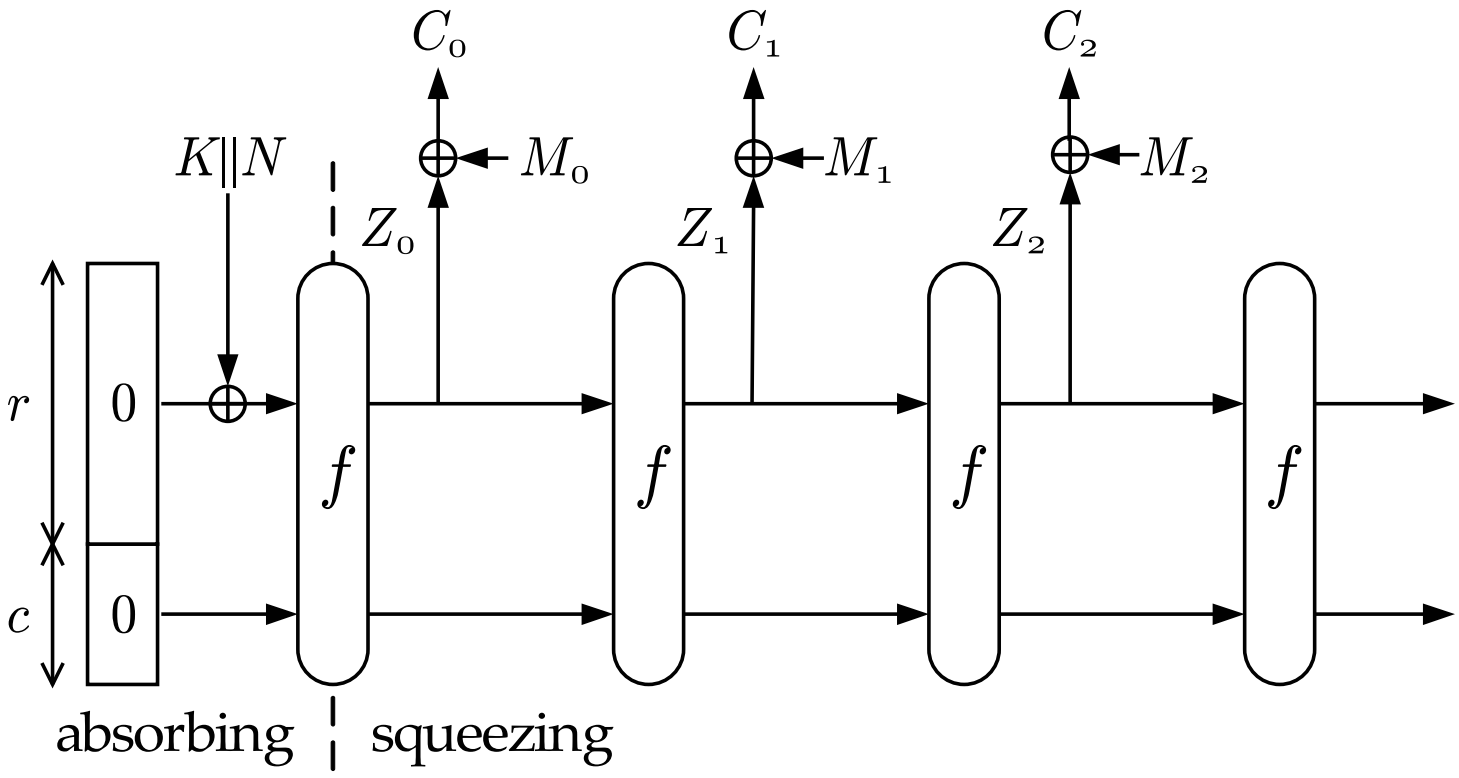


- Comme « *mask generating function* » [PKCS#1, IEEE Std 1363a]

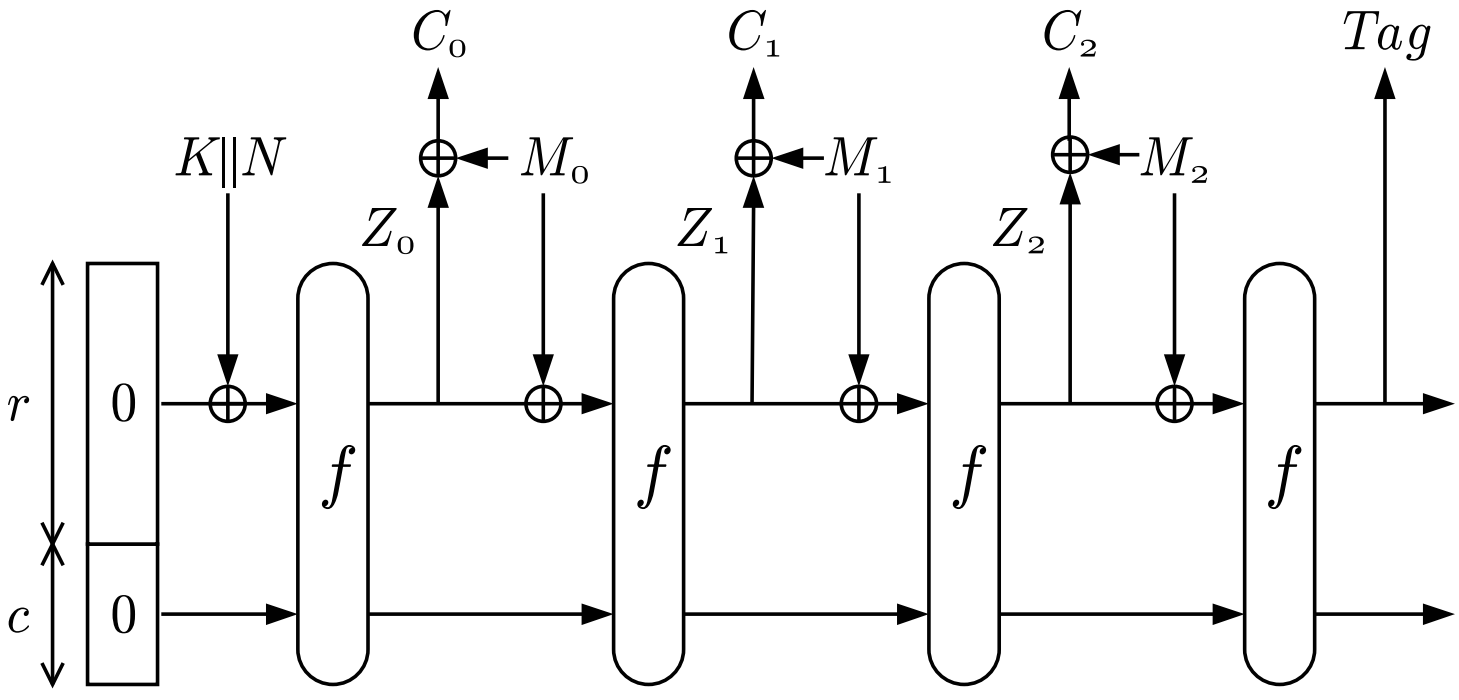
Générer un MAC avec une éponge



Chiffrement avec une éponge



Les deux en même temps ?



Absorbing and squeezing

Quelques exemples...

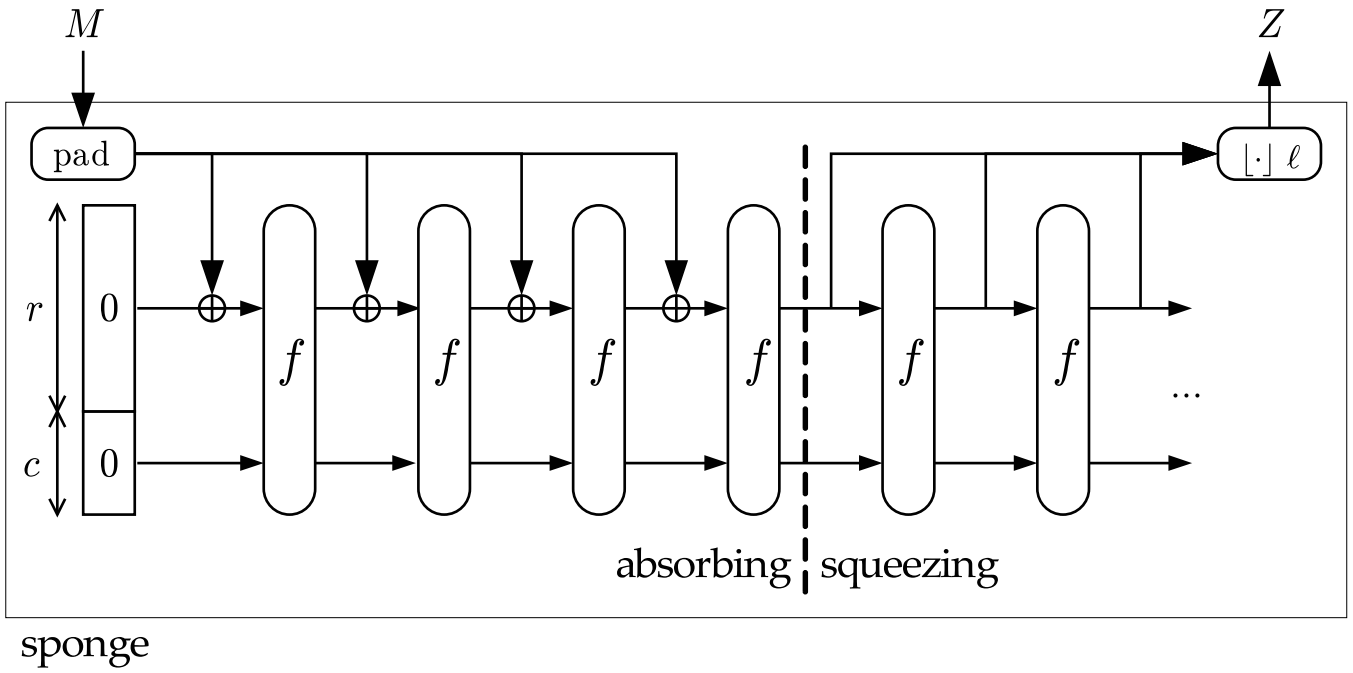
Nom			$r + c$
KECCAK	Bertoni, Daemen, Peeters, Van Assche	SHA-3 2008	25, 50, 100, 200 400, 800, 1600
Quark	Aumasson, Henzen, Meier, Naya-Plasencia	CHES 2010	136, 176 256, 384
Photon	Guo, Peyrin, Poschmann	Crypto 2011	100, 144, 196, 256, 288
Spongent	Bogdanov, Knezevic, Leander, Toz, Varici, Verbauwhede	CHES 2011	88, 136, 176 248, 320
Gluon	Berger, D'Hayer, Marquet, Minier, Thomas	Africacrypt 2012	136, 176, 256

Plan

Ligne du temps

- Subterranean : Daemen (1991)
- StepRightUp : Daemen (1994)
- Panama : Daemen et Clapp (1998)
- RadioGatún : Bertoni, Daemen, Peeters et Van Assche (2006)
- Keccak : idem (2008)

Utilisons la construction « éponge »



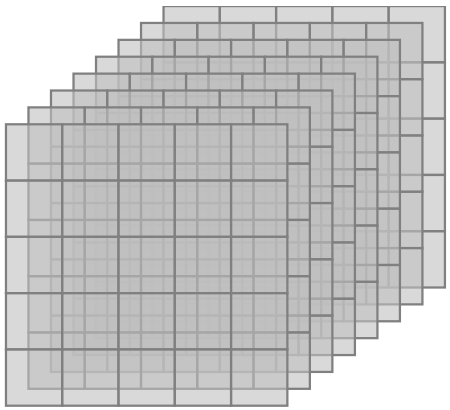
Notre mission :

Créer une permutation Keccak-f qui ne peut pas être distinguée d'une permutation tirée au hasard.

Keccak

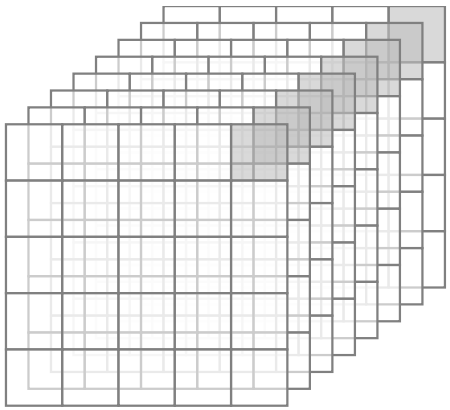
- Fonction-éponge avec une permutation Keccak-f
  - 7 permutations : $r + c \in \{25, 50, 100, 200, 400, 800, 1600\}$
- Exemples
  - Défaut SHA-3 : $r = 1024$ et $c = 576$ pour sécurité $2^{c/2} = 2^{288}$
  - Petite emprunte : $r = 40$ et $c = 160$ pour sécurité $2^{c/2} = 2^{80}$

L'état : un parallélépipède de $5 \times 5 \times 2^\ell$ bits



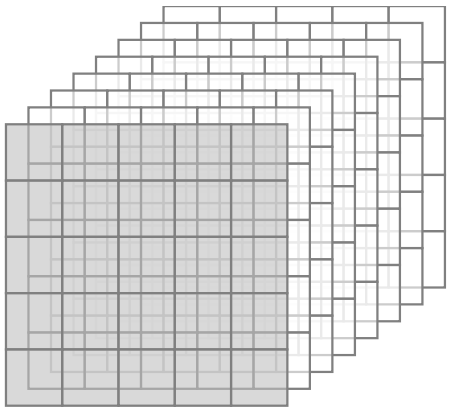
- 5×5 "lanes", chacune de 2^ℓ bits (1, 2, 4, 8, 16, 32 ou 64)
- 2^ℓ tranches de 5×5 bits

L'état : un parallélépipède de $5 \times 5 \times 2^\ell$ bits



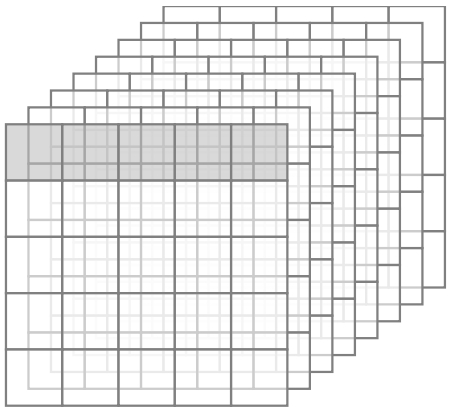
- 5×5 "lanes", chacune de 2^ℓ bits (1, 2, 4, 8, 16, 32 ou 64)
- 2^ℓ tranches de 5×5 bits

L'état : un parallélépipède de $5 \times 5 \times 2^\ell$ bits



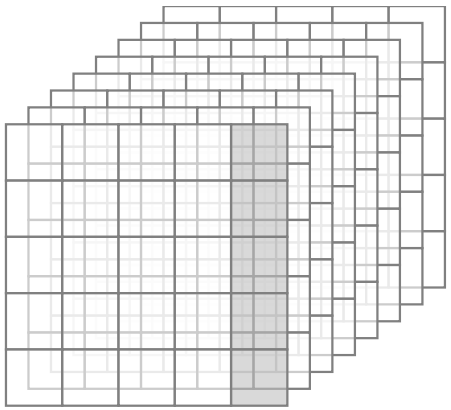
- 5×5 "lanes", chacune de 2^ℓ bits (1, 2, 4, 8, 16, 32 ou 64)
- 2^ℓ tranches de 5×5 bits

L'état : un parallélépipède de $5 \times 5 \times 2^\ell$ bits



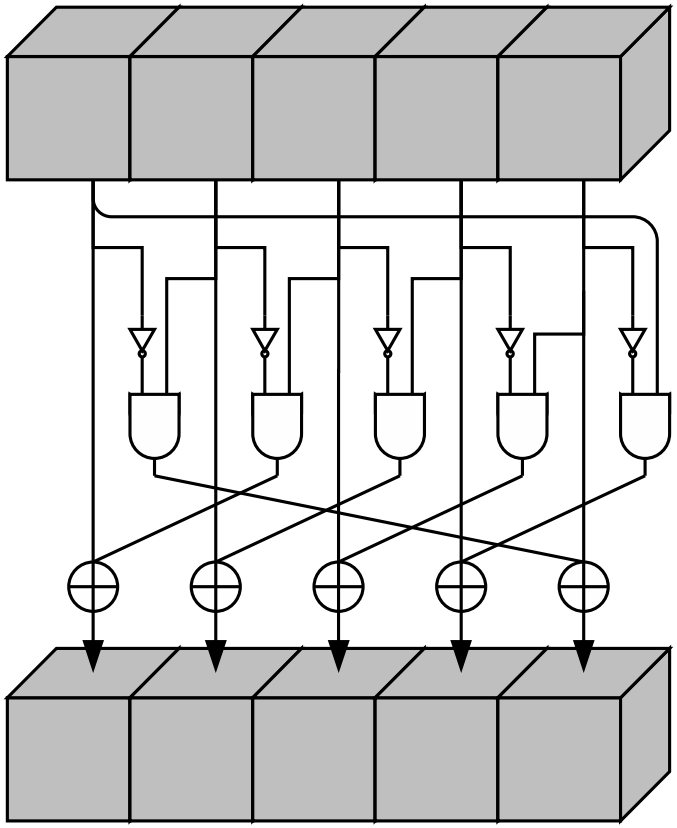
- 5×5 "lanes", chacune de 2^ℓ bits (1, 2, 4, 8, 16, 32 ou 64)
- 2^ℓ tranches de 5×5 bits

L'état : un parallélépipède de $5 \times 5 \times 2^\ell$ bits

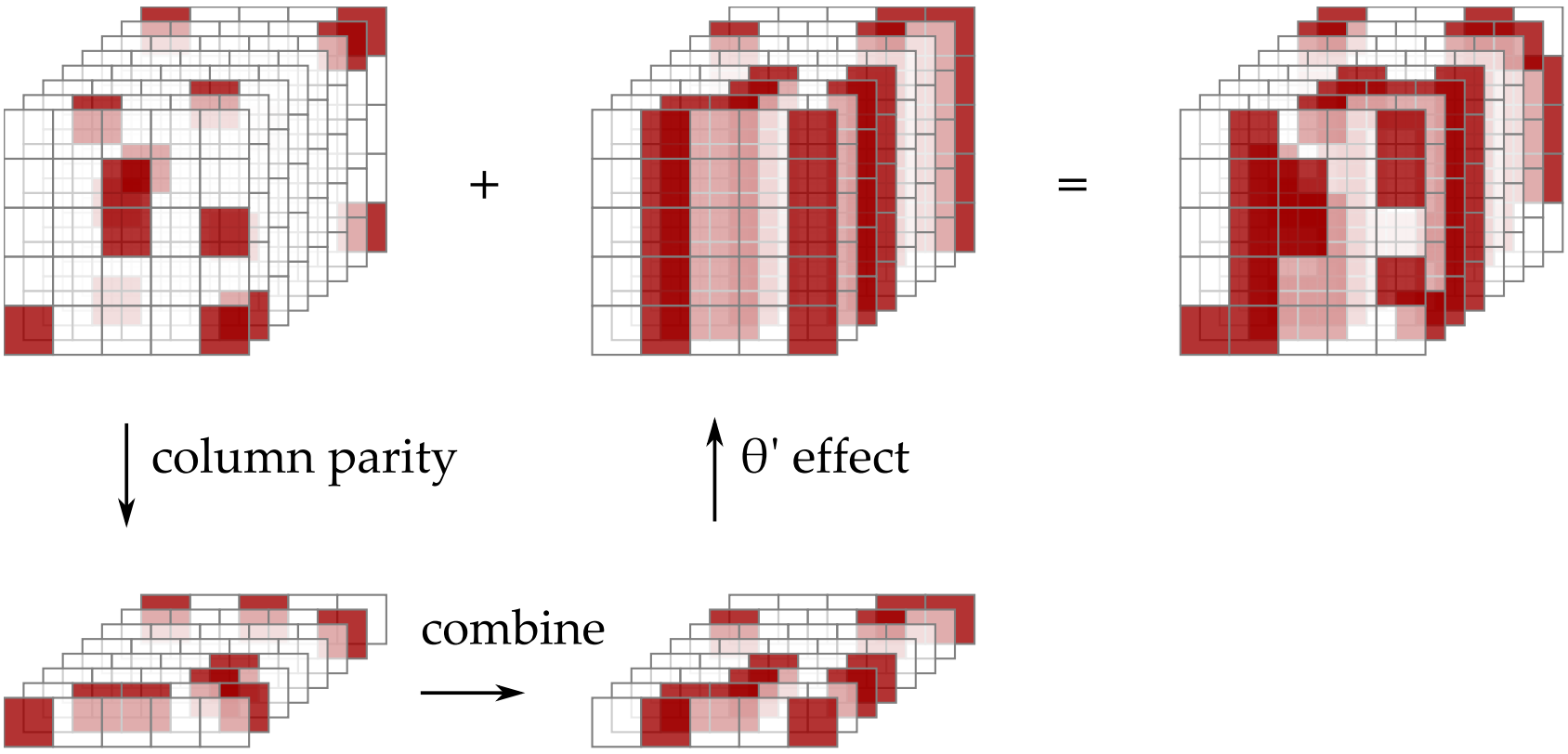


- 5×5 "lanes", chacune de 2^ℓ bits (1, 2, 4, 8, 16, 32 ou 64)
- 2^ℓ tranches de 5×5 bits

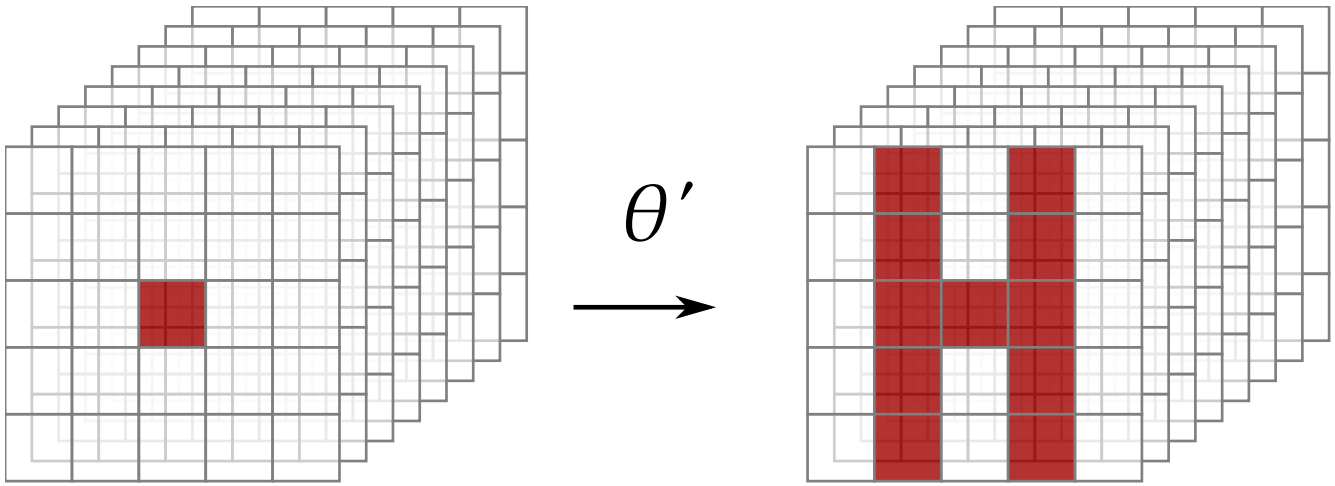
χ, l'opération non-linéaire de Keccak-f



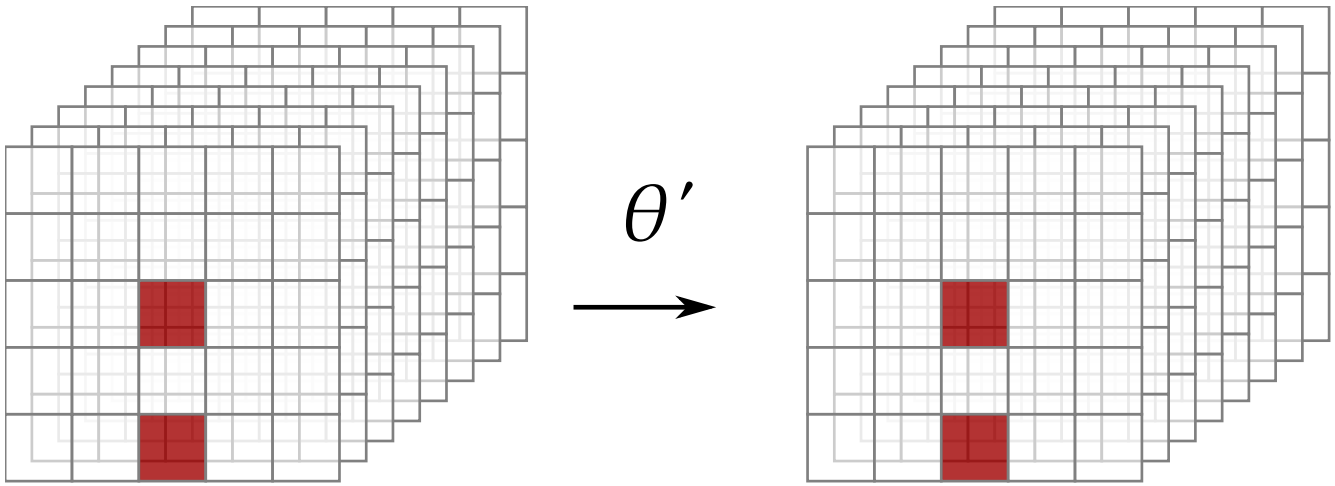
θ', première tentative pour mixer les bits



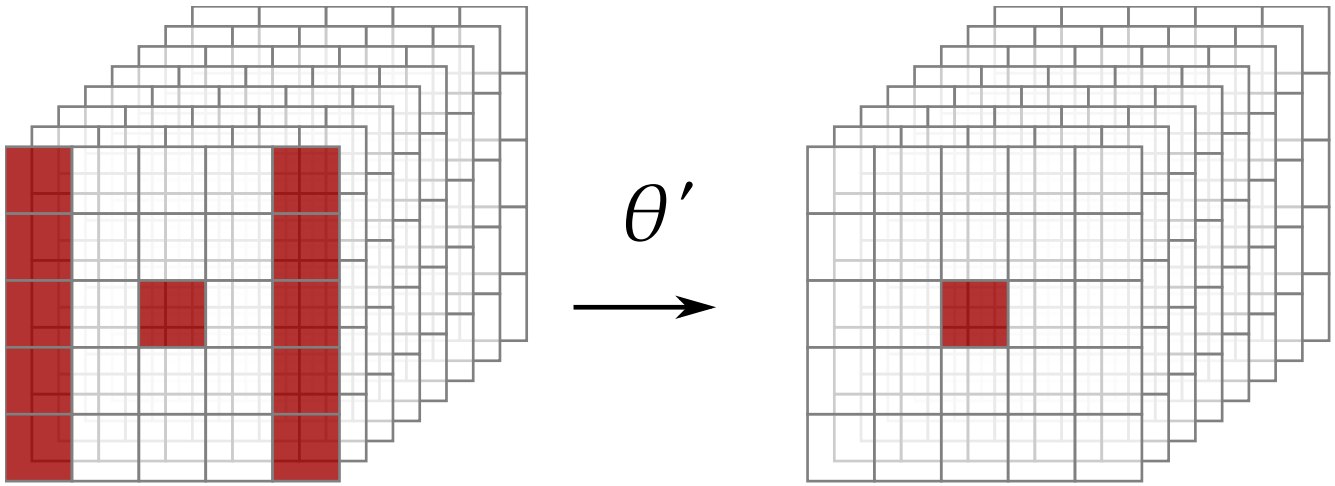
Diffusion de θ'



Diffusion de θ' (kernel)

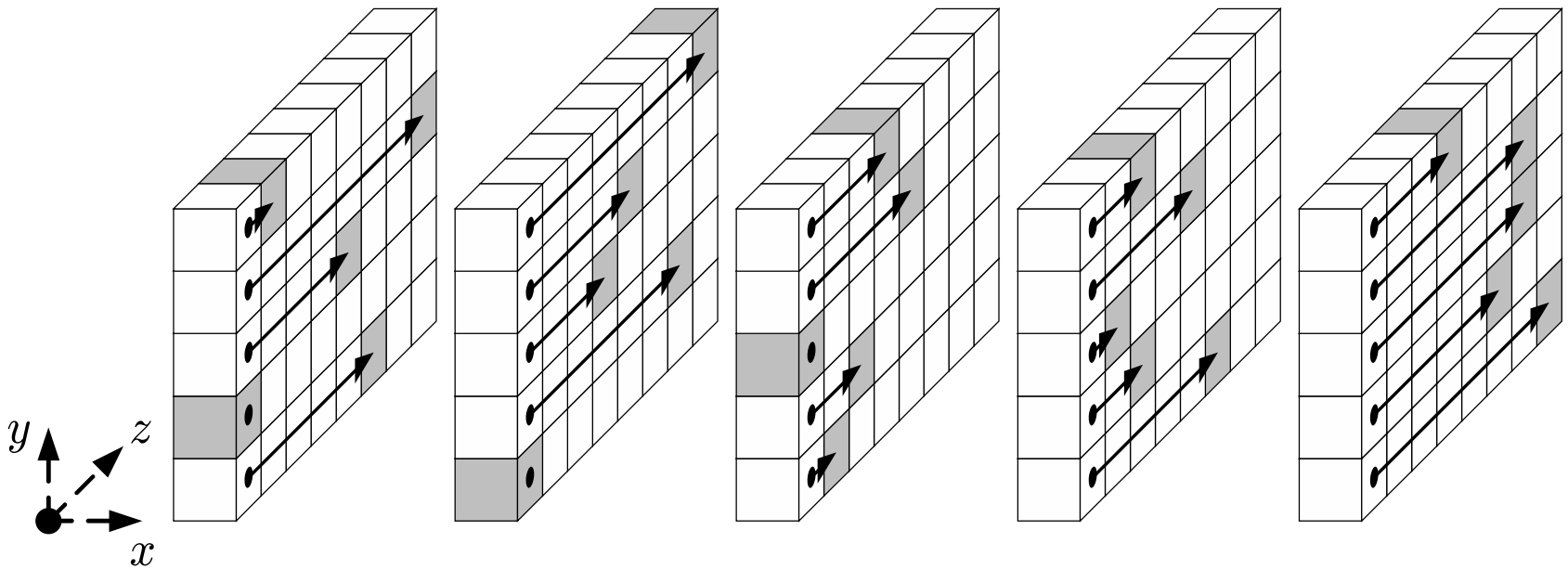


Diffusion de l'inverse de θ'



ρ, dispersion inter-tranches

- ρ : translation des lignes de $i(i+1)/2 \bmod 2^\ell$ positions

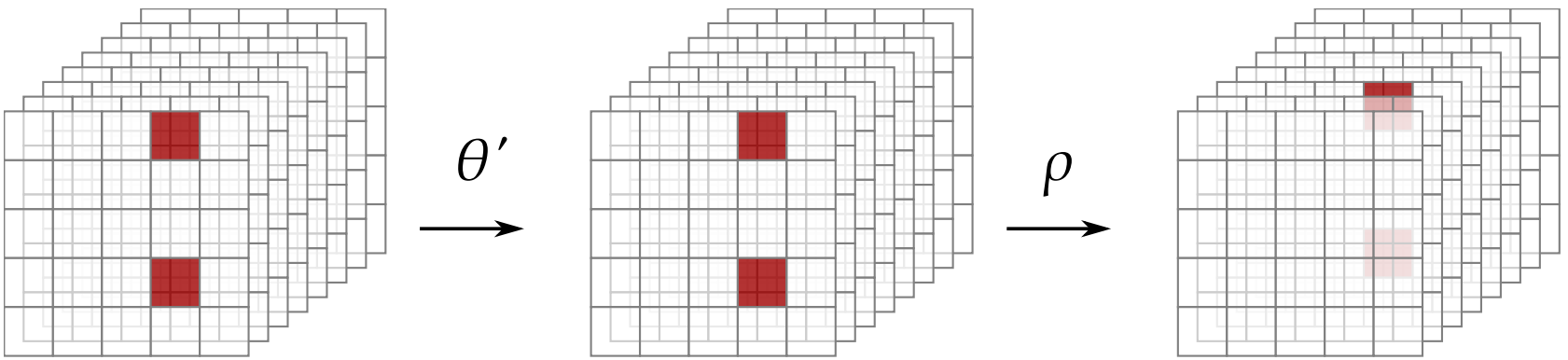


ι, pour casser la symétrie

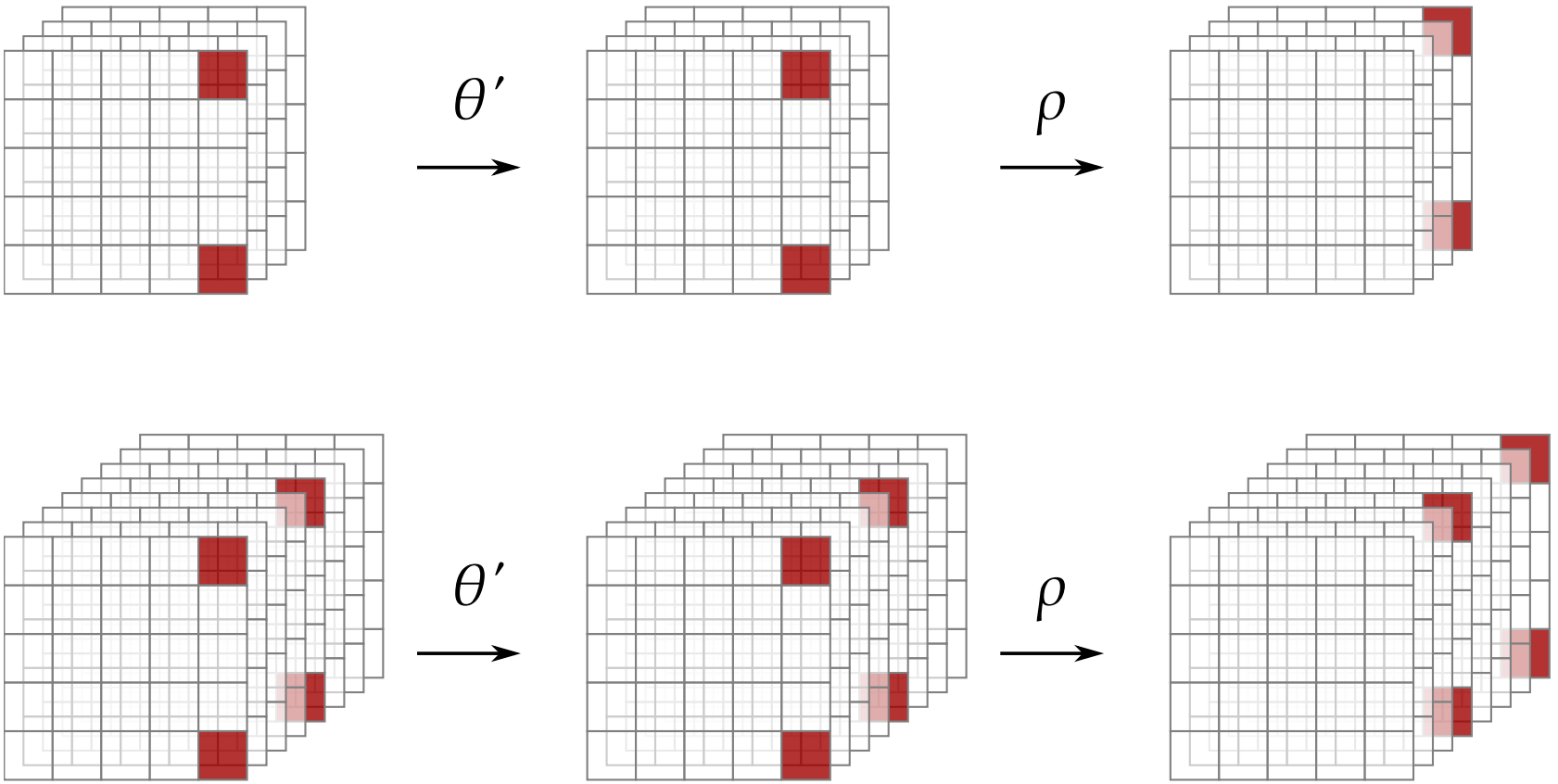
- Constante de tour ajoutée à la ligne $(0, 0)$
- Sans ι, tout serait invariant selon z
- Sans ι, tous les tours seraient identiques
- Sans ι, il y aurait des points fixes simples (000 and 111)

Keccak-f : premier essai

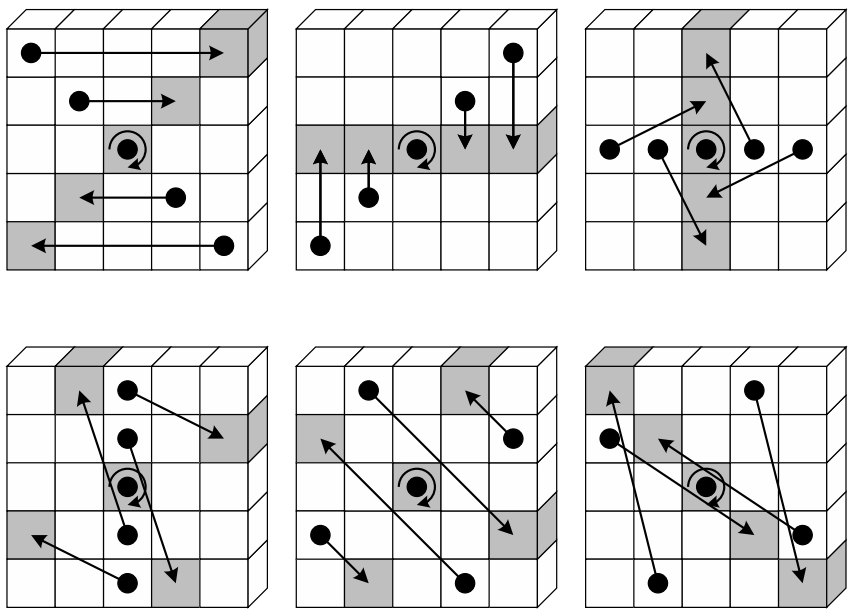
- Fonction de tour : $\mathsf{R} = \iota \circ \chi \circ \rho \circ \theta'$
- Problème : chemin simple périodique



Motifs périodiques (Matryoshka)



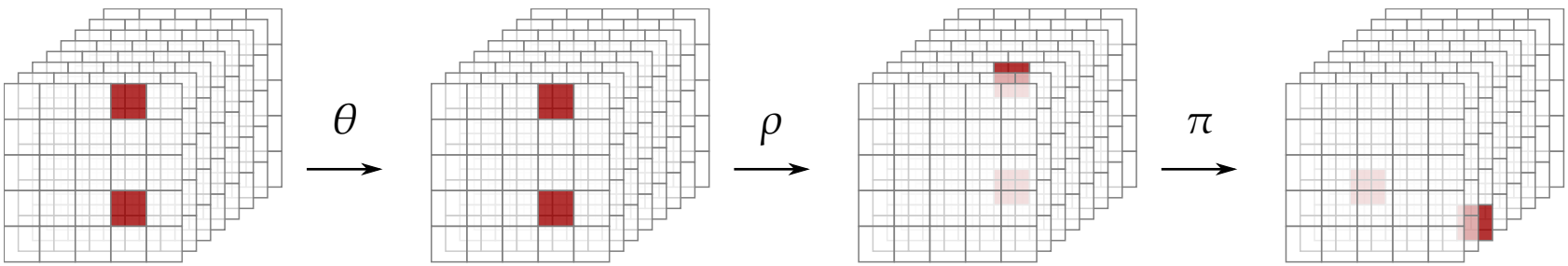
π pour perturber l'alignement horizontal/vertical



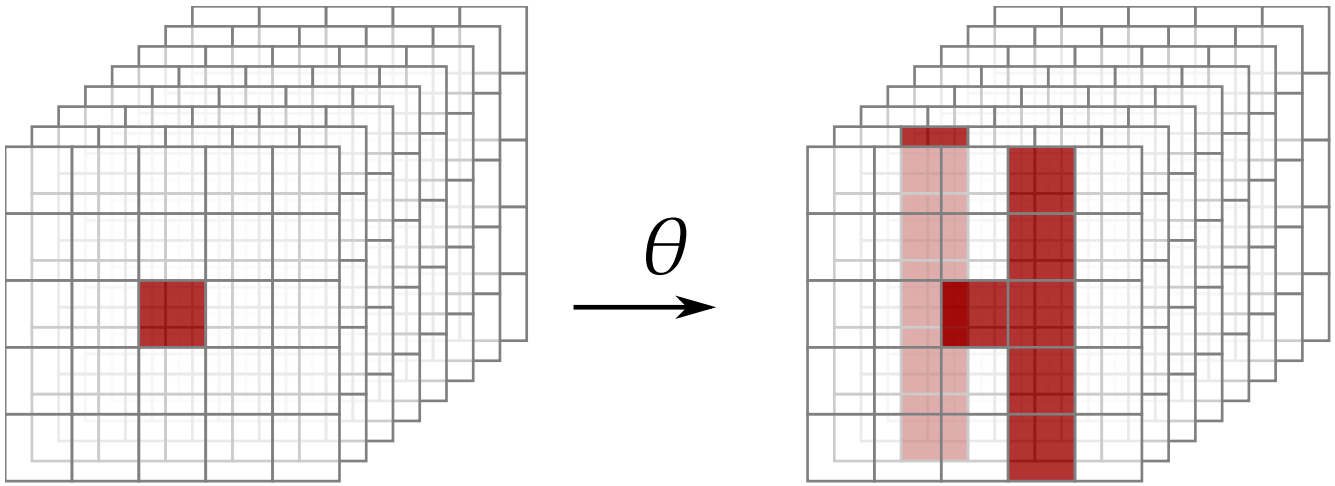
$$a_{x,y} \leftarrow a_{x',y'} \text{ avec } \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Keccak-f : deuxième essai

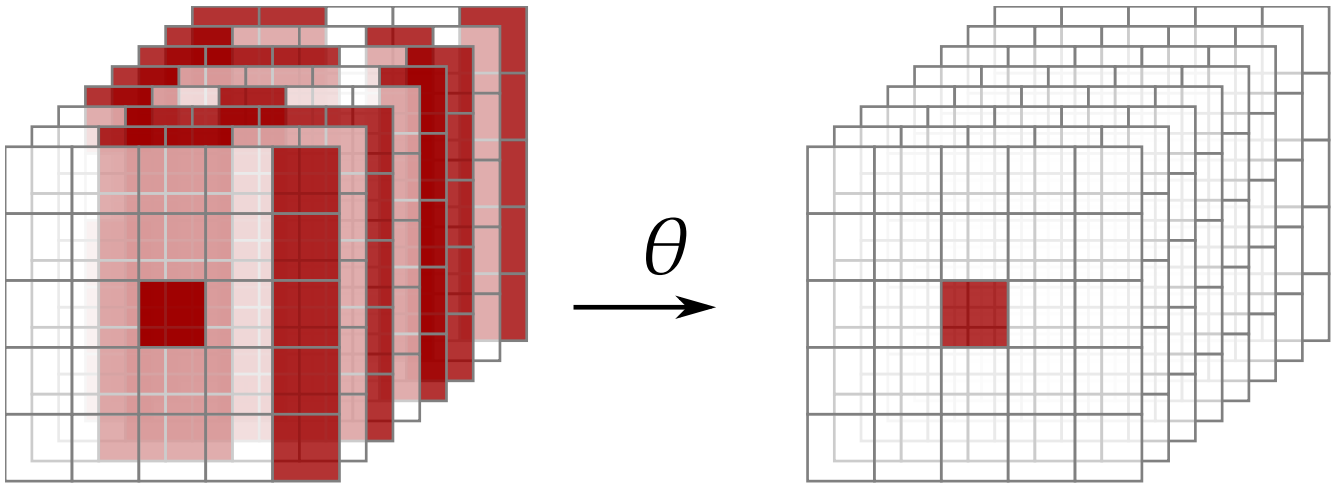
- Fonction de tour : $\mathsf{R} = \iota \circ \chi \circ \pi \circ \rho \circ \theta'$



Changeons θ' en θ



Inverse de θ



Keccak-f en résumé

- Fonction de tour :

$$\mathsf{R} = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$

- Nombre de tours : $12 + 2\ell$
  - Keccak-$f[25]$ comporte 12 tours
  - Keccak-$f[1600]$ comporte 24 tours
- Efficacité
  - parallélisme
  - flexibilité : “bit-interleaving”
  - logiciel : compétitif
  - matériel : très rapide
  - protection contre les attaques par canaux auxiliaires

Ce que les livres devraient dire sur la crypto symétrique

Primitives en cryptographie symétrique :

- **Permutations**
- Chiffrement de bloc (block ciphers)
- Chiffrement de flux (stream ciphers)
- Fonctions de hachage (hash functions)
  - Sans clé
  - Avec clé : authentification (MAC)

Et leurs mode d'utilisation...

Questions ?

Merci pour votre attention !

http://sponge.noekeon.org/
http://keccak.noekeon.org/