Analyse de PRESENT avec peu de données

(Un tour de plus dans les attaques meet-in-the-middle)

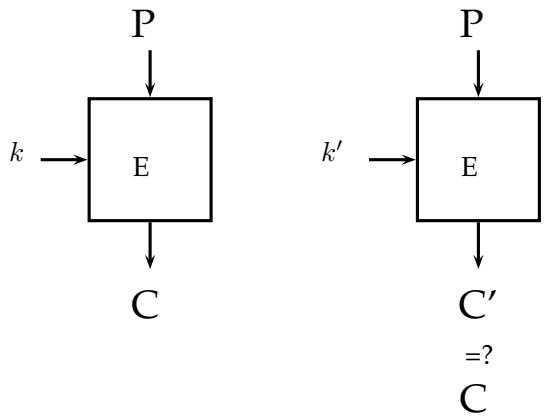
María Naya-Plasencia [1], Bastien Vayssière [2]

1:INRIA Rocquencourt
2:PRISM, Université de Versailles

12 octobre 2012

Plan

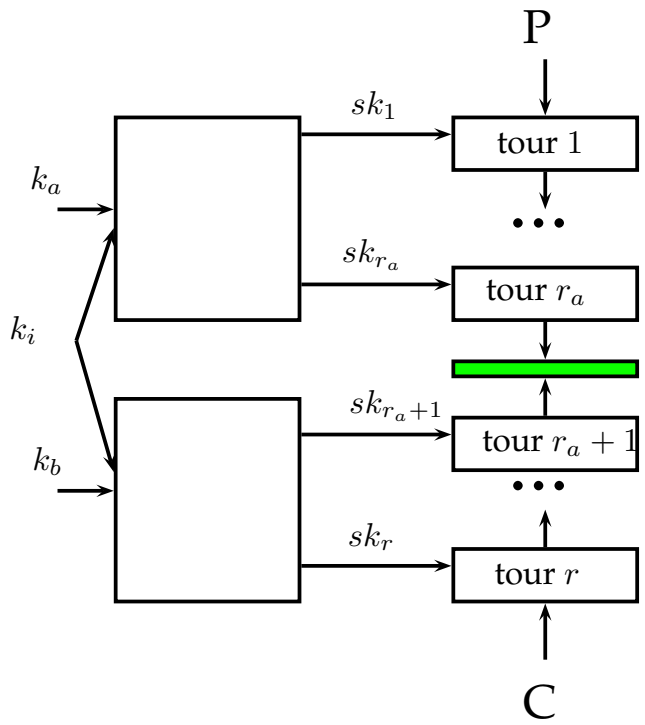
- Une chiffrement par bloc sur n bits, avec une clé k. On cherche à retrouver la clé.



- Une attaque générique : la recherche exhaustive de clé
- Donnée : une paire clair-chiffré (P, C) par la clé inconnue.
- Calcul des chiffrements de P par toutes les clés possibles. On garde celle(s) qui produisent C.
- On compare la complexité des attaques aux $2^{|k|}$ chiffrements de la recherche exhaustive.

Meet-in-the-middle classique sur un chiffrement itératif à r tours

Principe : séparer les bits de clé en deux ensemble distincts, intervenant sur deux parties de l'algorithme.



- La première fois en 1977, par Diffie et Hellmann, sur le DES appliqué en cascade avec 2 clés, puis en 1985 par Chaum et Evertse sur des versions réduites du DES.
- Variantes utilisées sur l'AES, KTANTAN, XTEA, etc. et sur des fonctions de hachage pour la recherche de préimage.

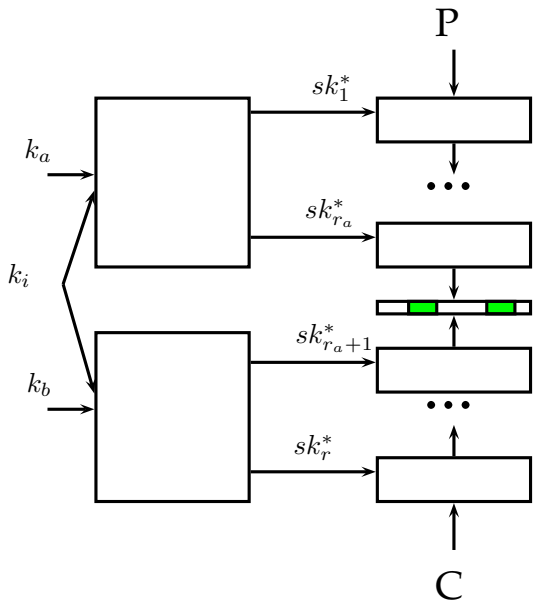
Complexité

- Temps : $2^{|k_i|}(2^{|k_a|} + 2^{|k_b|})$ chiffrements partiels.
- Mémoire :$|k_a|2^{|k_a|}$ bits

Rapport complexité en temps recherche exhaustive / MITM :

$$\frac{2^{|k_a|+|k_b|}}{2^{|k_a|} + 2^{|k_b|}}$$

Calcul d'une partie v du chiffrement/déchiffrement au milieu.

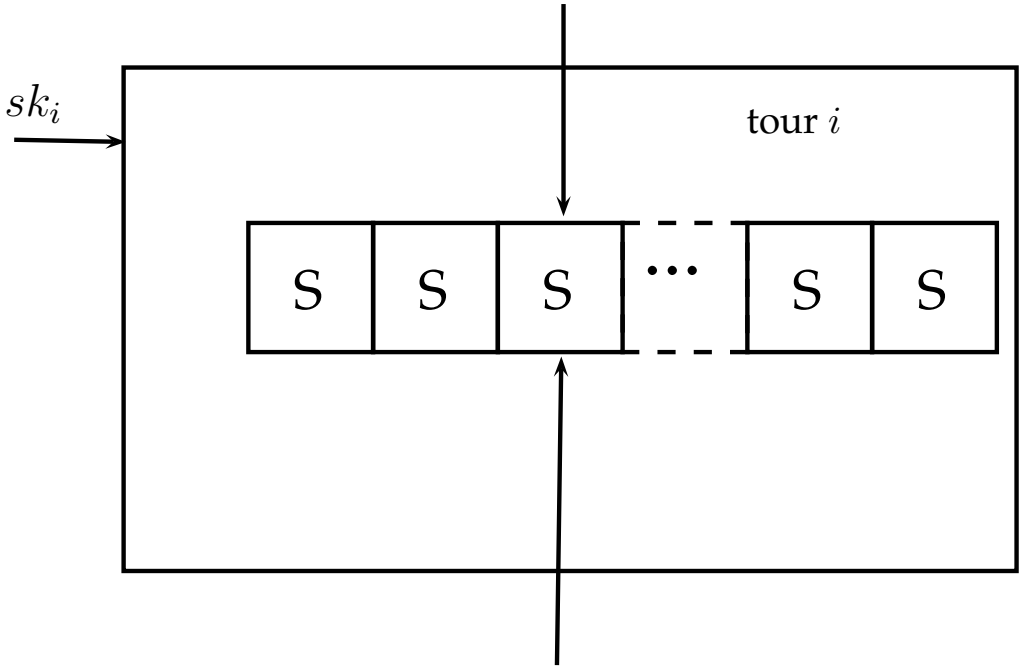


Complexité

On récupère la clé en :

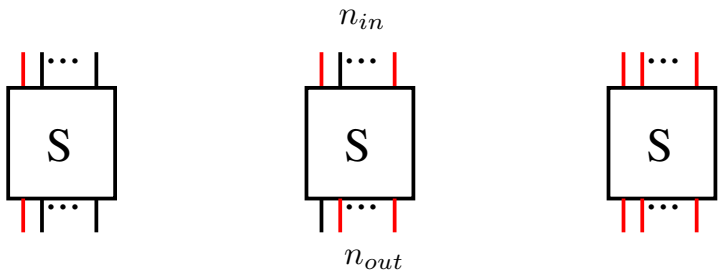
- $2^{|k_i|}(2^{|k_a|} + 2^{|k_b|})$ chiffrements partiels
- $+ 2^{|k|-|v|}$ chiffrements de clés restantes, en moyenne.

Considérons un tour de chiffrement contenant des boîtes S de taille m.



- Jusqu'ici : calcul d'une partie des bits au milieu
- Amélioration : calcul d'une partie de l'entrée et la sortie de S

On calcule n_{in} bits en entrée, n_{out} bits en sortie d'une boîte S.



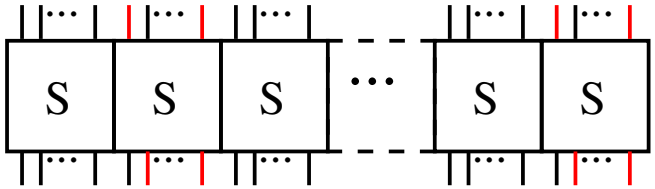
Recherche de transition possible

- Probabilité p qu'il existe une transition à travers S avec les valeurs des bits calculés ?
- Pour n_{in} bits fixés, il y a au plus $2^{m-n_{in}}$ valeurs distinctes des n_{out} bits possibles. D'où :

$$p \leq \frac{2^{m-n_{in}}}{2^{n_{out}}} = \frac{1}{2^{n_{in}+n_{out}-m}}$$

- Pour éliminer des mauvaises clés, il suffit d'avoir $p < 1$, c'est à dire :

$$n_{in} + n_{out} > m$$



- On fait le même raisonnement sur un sous-ensemble de ℓ boîtes S.
- Avec P, pour chaque valeur de (k_a, k_i) on calcule N_{in} bits en entrée parmi ℓm.
- Avec C, pour chaque valeur de (k_b, k_i) on calcule N_{out} bits en sortie.

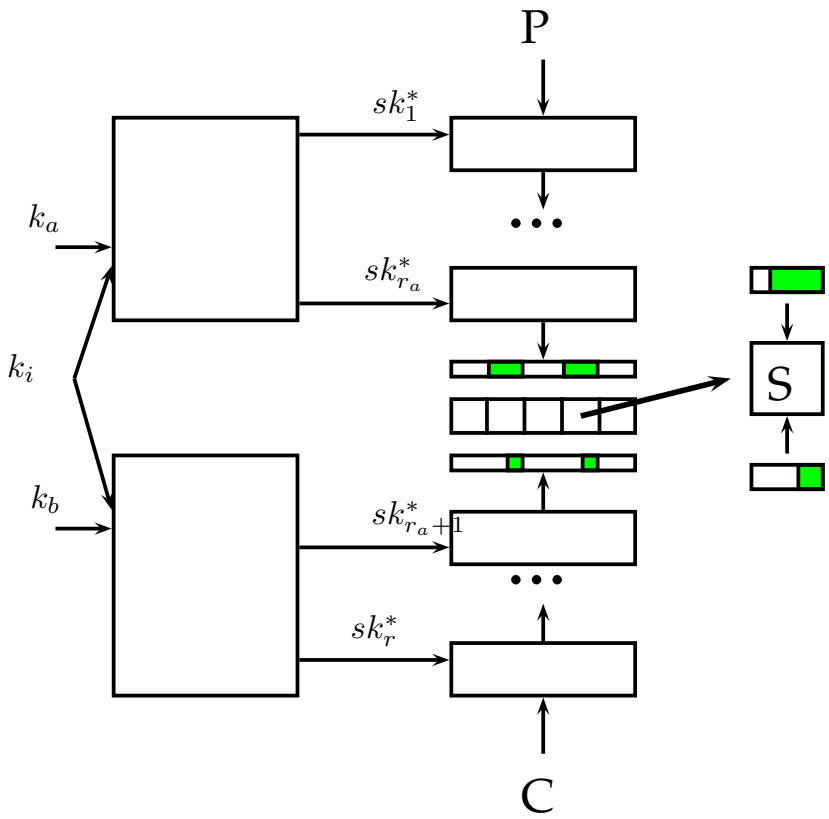
Probabilité du crible

La probabilité qu'une transition soit possible au milieu est majorée par :

$$\frac{1}{2^{N_{in}+N_{out}-\ell m}}$$

Pour éliminer des mauvaises clés, il suffit d'avoir :

$$N_{in} + N_{out} > \ell m$$



On applique le crible sur ℓ boîtes S d'un tour au milieu, à la place de la recherche de collisions.
On sélectionne en moyenne au plus $\frac{2^{|k|}}{2^{N_{in}+N_{out}-\ell m}}$ clés.

Complexité

- $2^{|k_i|}(2^{|k_a|} + 2^{|k_b|})$ chiffrements partiels
- + en moyenne $\frac{2^{|k|}}{2^{N_{in}+N_{out}-\ell m}}$ chiffrements (dans le pire cas de boîtes S)

Ce qui change par rapport aux meet-in-the-middle précédents :

1. la "fusion" au milieu
2. sélection d'une fraction $\frac{1}{2^{N_{in}+N_{out}-\ell m}}$ des clés au lieu de $\frac{1}{2^{|v|}}$.

Mais cela peut nous permettre de gagner un tour ...
... exemple d'application sur le chiffrement par blocs PRESENT.

Chiffrement par blocs sur $n = 64$ bits, avec une clé de 80 ou 128 bits. Proposé par Bogdanov et al. en 2007.

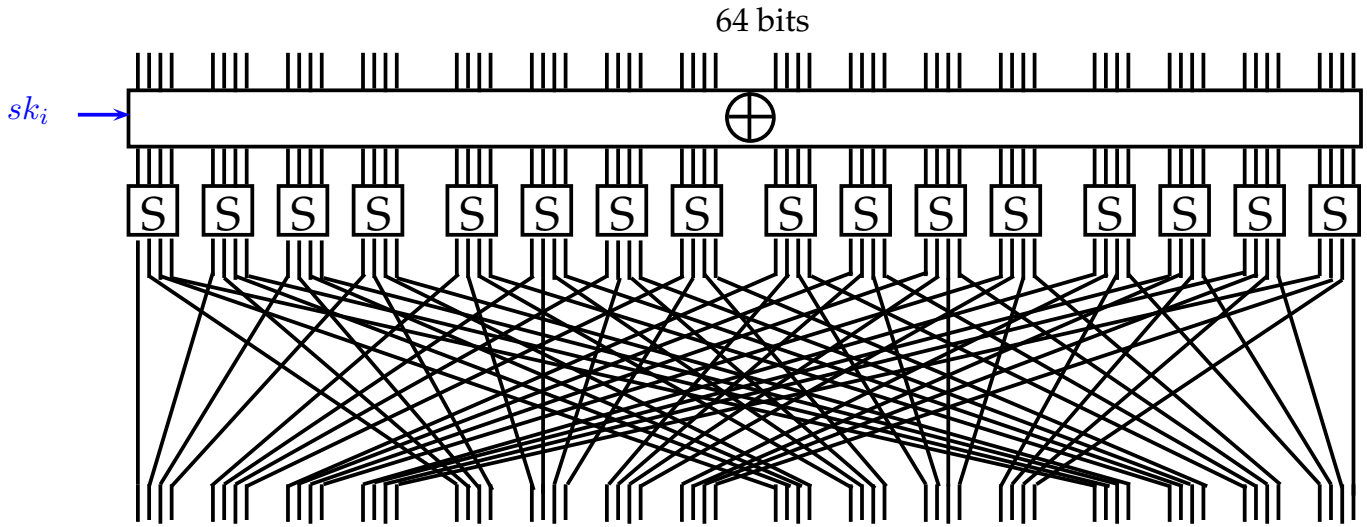


Figure: Représentation du tour i de PRESENT-80

31 itérations d'un réseau de substitution-permutation, et un xor final avec une sous-clé.

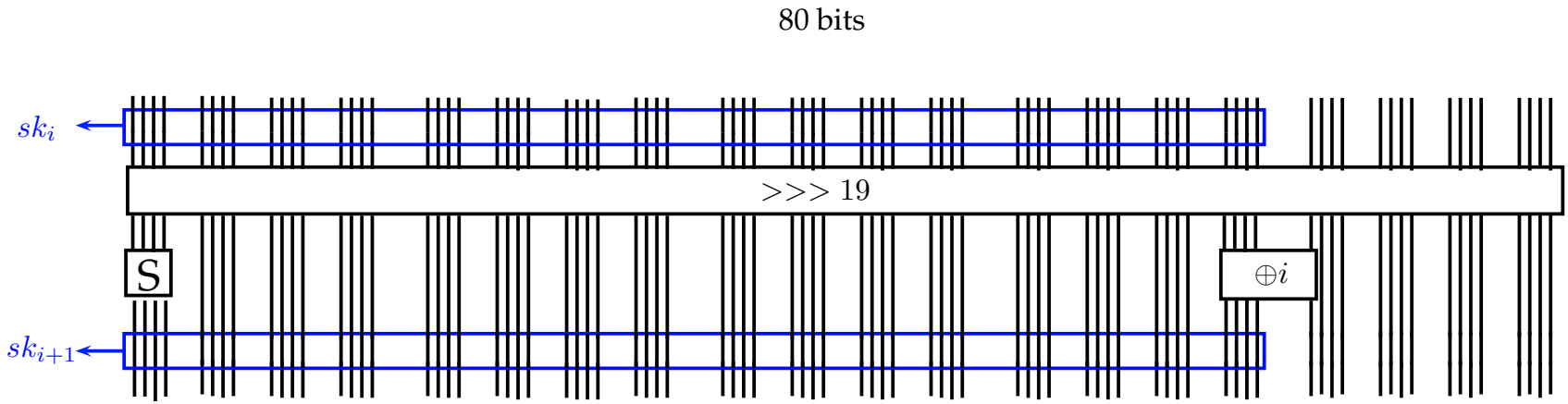


Figure: Représentation du tour i du key scheduling de PRESENT-80

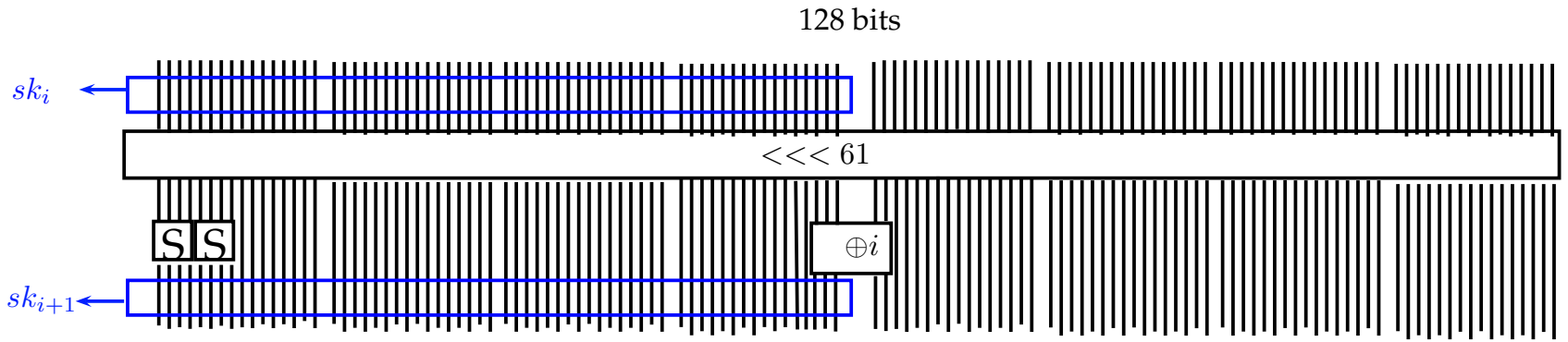


Figure: Représentation du tour i du key scheduling de PRESENT-128

tours	taille de clé	technique	données	auteurs
8	128	integral	$2^{24.3}$	[Reza Z'aba et al. 2008]
16	80	differential	2^{64}	[Wang 2008]
17	128	related keys	2^{63}	[Özen et al. 2009]
18	80	multiple differential	2^{64}	[Blondeau, Gérard 2011]
18	128	multiple differential	2^{64}	[Blondeau, Gérard 2011]
19	128	algebraic differential	2^{62}	[Albrecht, Cid 2009]
24	80	linear	$2^{63.5}$	[Ohkuma 2009]
24	80	statistical saturation	2^{57}	[Collard, Standaert 2010]
25	128	linear	2^{64}	[Nakahara et al. 2009]
26	80	multiple linear	2^{64}	[Cho 2010]

Table: Complexité en données des cryptanalyses de versions réduites de PRESENT

- Sur PRESENT-80, $> 2^{57}$ paires clair-chiffré.
- Sur PRESENT-128, $> 2^{24.3}$ paires clair-chiffré.

Question

Comme Bouillaguet, Derbez l'ont fait pour l'AES, on peut se demander :
Avec peu de données, combien de tours de PRESENT peut-on casser ?

Appliquons l'amélioration du meet-in-the-middle sur PRESENT.
PRESENT-80 et PRESENT-128 comportent une couche de boîtes S identiques de taille $m = 4$.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S[x]	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Table: Representation hexadécimale de la fonction non-linéaire de la boîte S

On va utiliser la probabilité d'une transition à travers S entre :

- 3 bits de poids faibles en entrée
- 2 bits de poids faibles en sortie d'une boîte S de PRESENT.
- Sur ℓ boîtes S on a $N_{in} + N_{out} - \ell \cdot m = 3\ell + 2\ell - 4\ell = \ell$, donc la probabilité qu'une transition soit possible est :

$$p \leq \frac{1}{2^{\ell}}$$

- Le crible permet de sélectionner en moyenne $p \cdot 2^{|k|} \leq 2^{80-\ell}$ clés potentielles, dont la bonne.

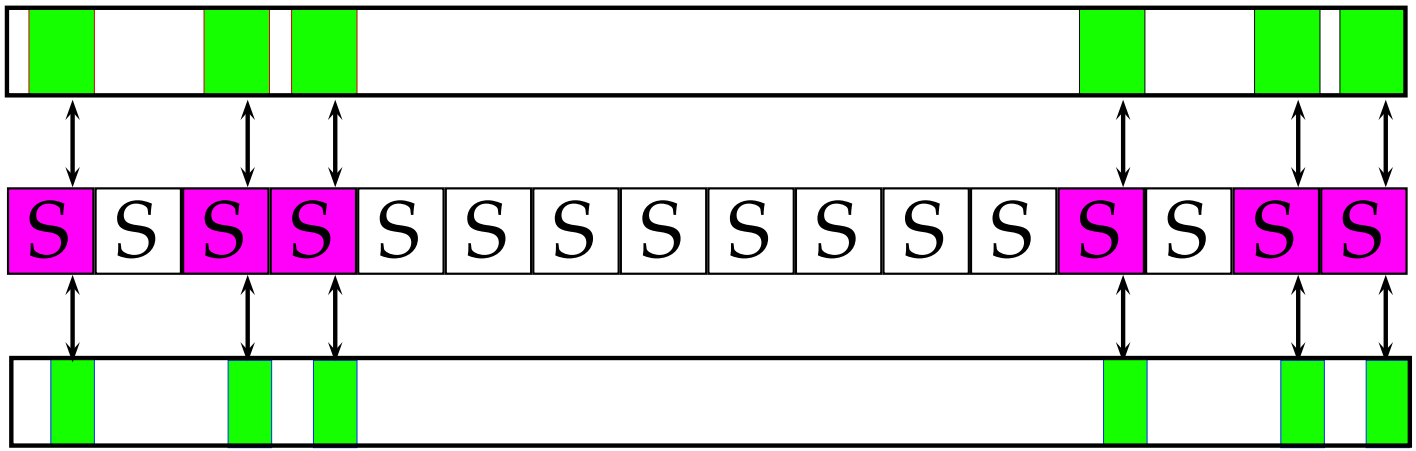
Correspondance à travers des boîtes S de PRESENT

$x_3x_2x_1x_0$	$S(x)_3S(x)_2S(x)_1S(x)_0$
0000	1100
0001	0101
0010	0110
0011	1011
0100	1001
0101	0000
0110	1010
0111	1101
1000	0011
1001	1110
1010	1111
1011	1000
1100	0100
1101	0111
1110	0001
1111	0010

$x_2x_1x_0 \rightarrow s\,y_1y_0$
000→00
000→11
001→01
001→10
010→10
010→11
011→00
011→11
100→00
100→01
101→00
101→11
110→01
110→10
111→01
111→10

16 valeurs des bits x_2, x_1, x_0, y_1, y_0, parmi 32 au total, ont une transition possible par la boîte S de PRESENT

Correspondance à travers des boîtes S de PRESENT



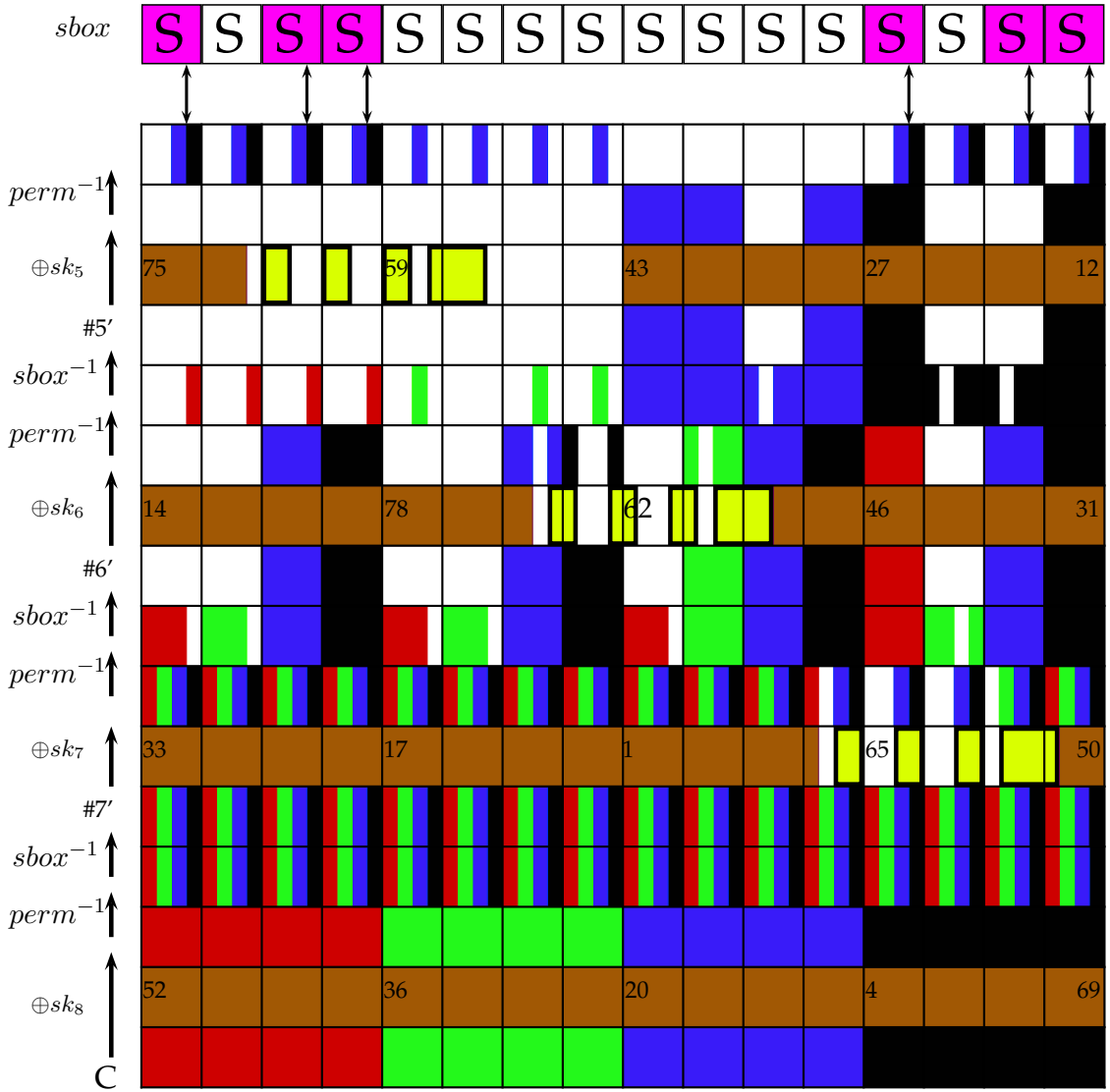
- On calcule les mêmes 5 bits en entrée/sortie de ℓ boîtes S parmi les 16.
- 16^ℓ transitions possibles tandis que l'espace des bits calculés est de taille $(2^5)^\ell$.
- La probabilité qu'une transition soit possible au milieu est alors :

$$p = \frac{16^\ell}{2^{5\ell}} = \frac{1}{2^\ell}$$

- Pour ce choix de bits, la boîte S de PRESENT atteint la majoration générique donnée précédemment.

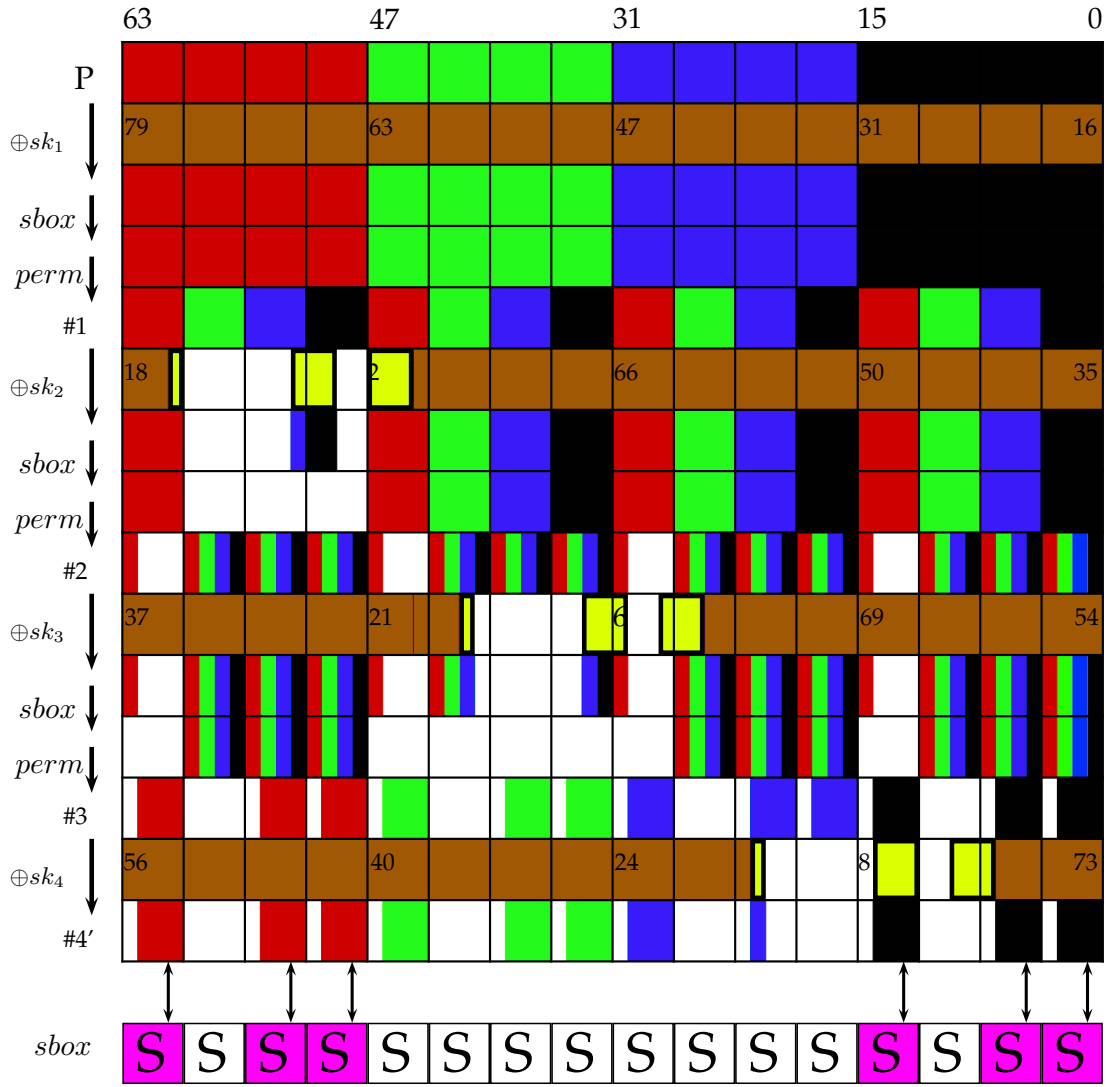
Application à PRESENT-80 : description

Calcul de 3 bits en sortie de 6 boîtes S à partir du chiffré de la paire donnée, et d'une partie des bits de clé testés.



Application à PRESENT-80 : description

Calcul de 2 bits en entrée de 6 boîtes S à partir du clair de la paire donnée, et d'une autre partie des bits de clé testés.



Application à PRESENT : résultats

Répartition des bits de clé : 65 bits communs, 6 bits pour la partie haute, 9 bits pour la partie basse.

Complexité

- Mémoire : table de 2^9 valeurs de $6 \times 2 = 12$ bits $< 2^{13}$ bits.
- Temps :
  - $2^{65}(2^9 + 2^6)$ chiffrements partiels
  - $+$ en moyenne 2^{74} chiffrements de 7 tours
  - Au total : l'équivalent de moins de 2^{75} chiffrements.

Implémentation

- Implémentation de la partie essentielle de l'attaque en fixant 48 bits communs et en retrouvant les 32 bits restants.
- On a obtenu autour de $2^{75-48} = 2^{27}$ chiffrements.
- Résultats cohérents avec les complexités données.

Pour PRESENT-128

Avec la même méthode, on a obtenu une attaque en 2^{127} chiffrements sur 9 tours de PRESENT-128, avec un crible sur la première boîte S.

Application à PRESENT : jusqu'où peut-on aller ?

nombre de tours	longueur de clé	paires	chiffrements	mémoire
7	80	2	2^{75}	2^{9}
9	128	2	2^{127}	2^{3}

Table: Complexité des attaques obtenues sur 7 et 9 tours

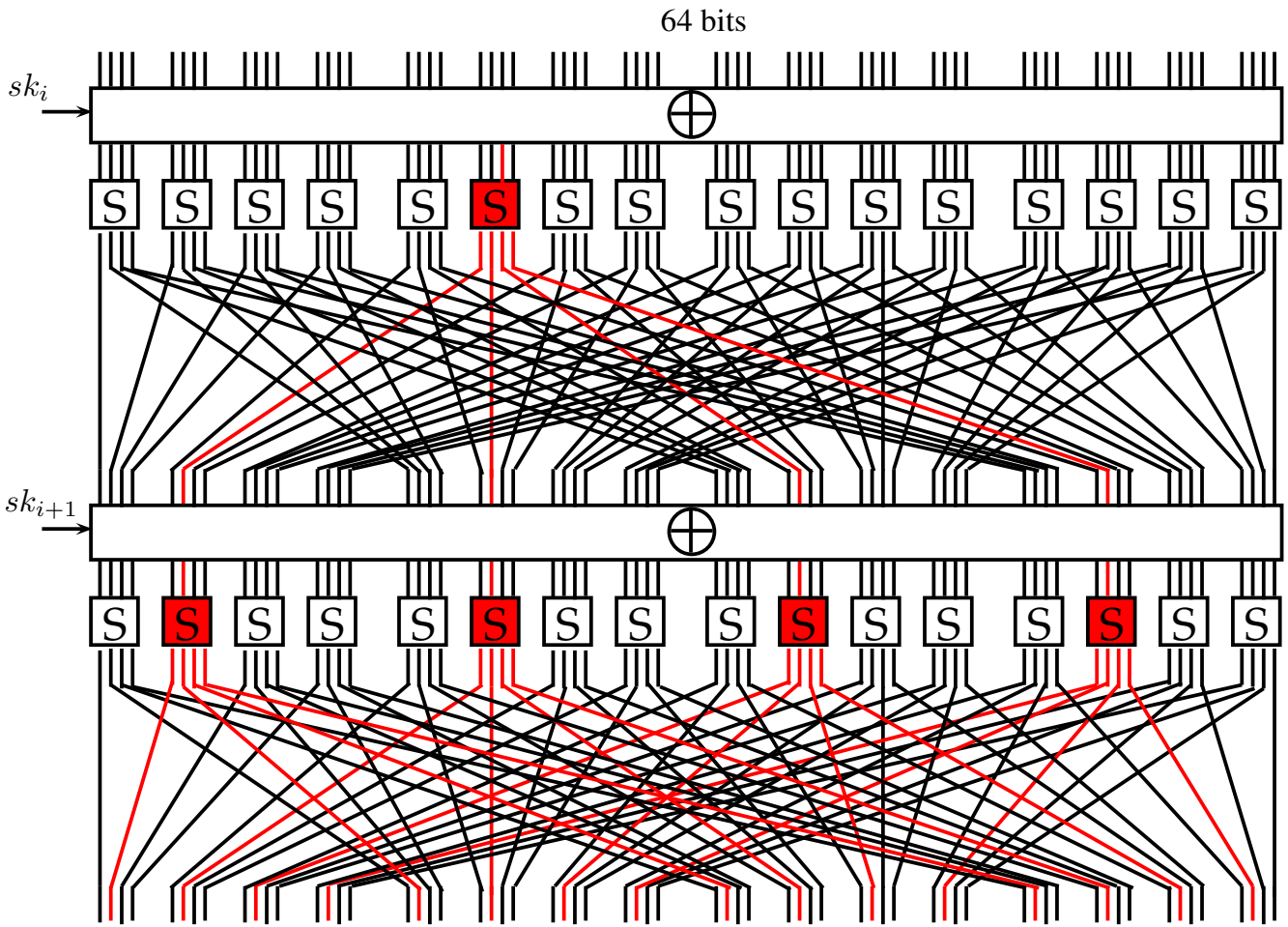
Nombre de tours maximal ?

Il faut pouvoir partiellement chiffrer à partir de P / déchiffrer à partir de C jusqu'au milieu sans utiliser tous les bits de clé.

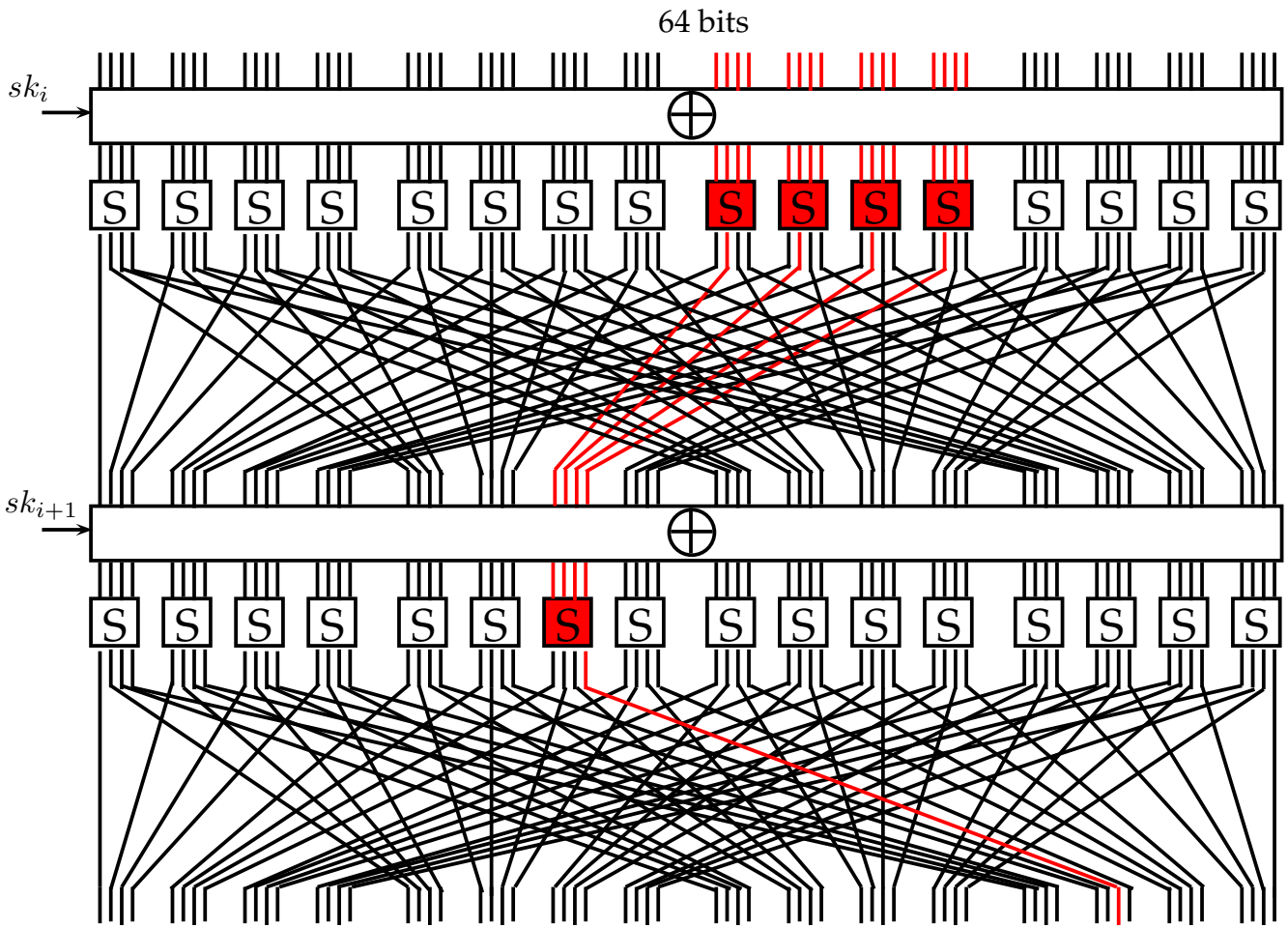
1. Sans fixer tous les bits de clé, pour combien de tours peut-on chiffrer/déchiffrer en choisissant tous les bits de toutes les sous-clés ?
2. Une fois qu'il manque un bit de sous-clé, de combien de tours peut-on encore connaître des bits en sortie ?

Application à PRESENT : jusqu'où peut-on aller ?

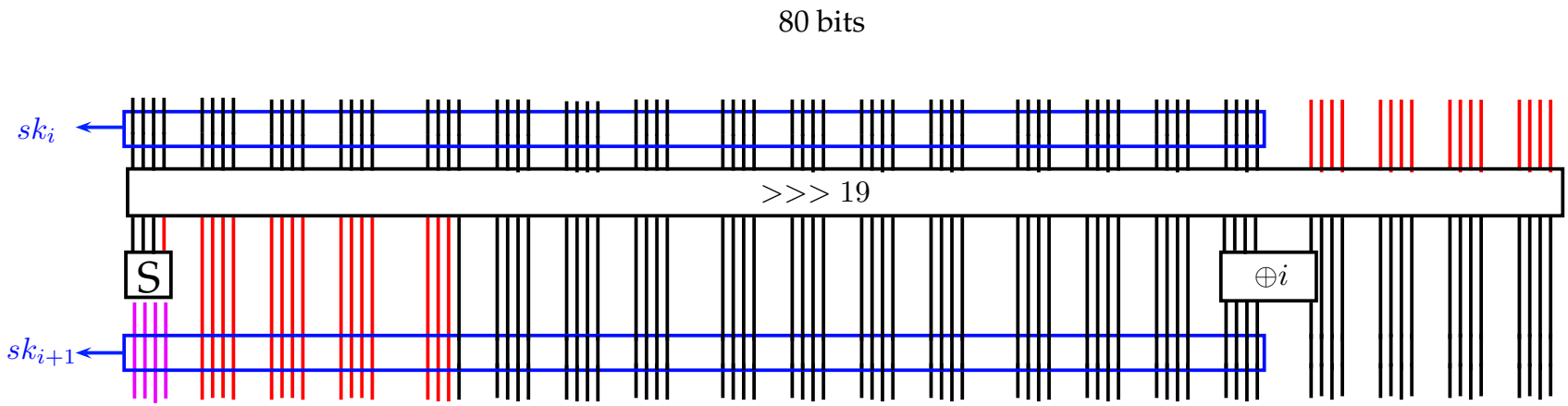
Nombre de tours de chiffrements partiellement faisables malgré un bit de sous-clé inconnu : $x_a = 2$



Nombre de tours de déchiffrement partiellement faisables malgré un bit de sous-clé inconnu : $x_b = 2$



Une borne supérieure sur les sous-clé successives calculables sans tous les bits de clé : $N_{80} = 1$.



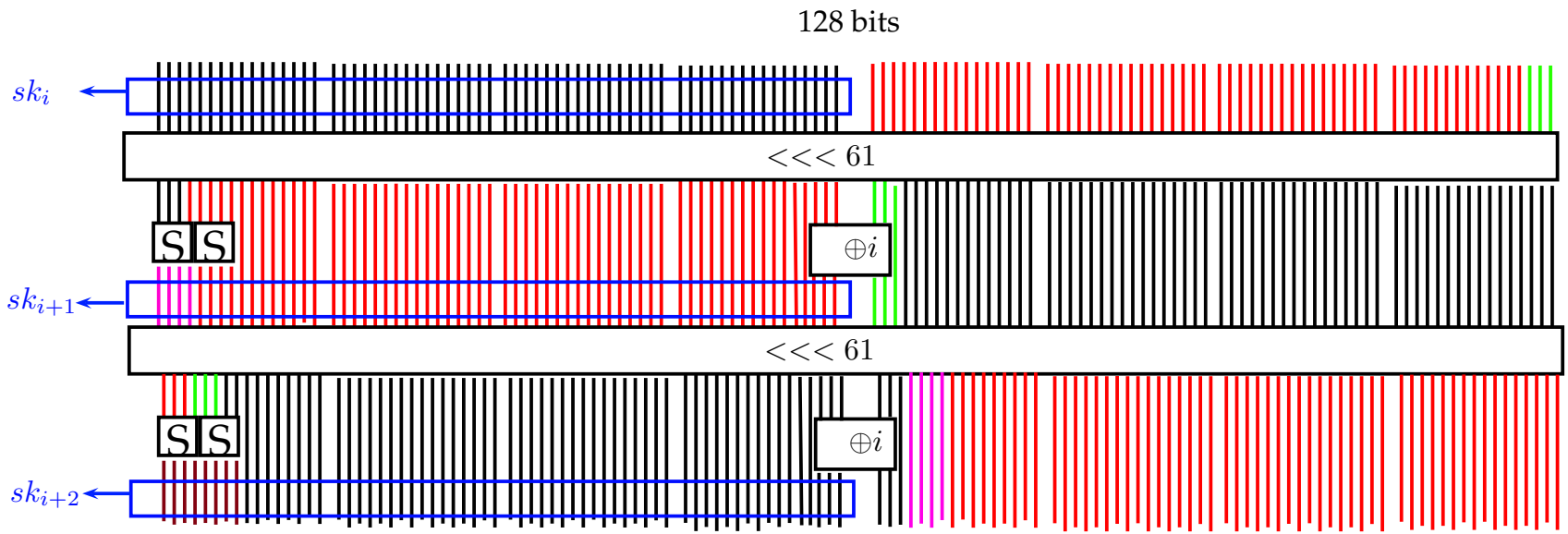
On a donc une borne supérieure sur le nombre de tours sur PRESENT-80 :

$$2N_{80} + x_a + x_b + 1 = 2 + 2 + 2 + 1 = 7$$

On a atteint cette borne supérieure avec notre attaque.

Application à PRESENT : jusqu'où peut-on aller ?

Une borne supérieure sur les sous-clé successives calculables sans tous les bits de clé : $N_{128} = 2$.



Borne supérieure sur le nombre de tours attaquable de PRESENT-128 :

$$2N_{128} + x_a + x_b + 1 = 2 \cdot 2 + 2 + 2 + 1 = 9$$

On a atteint cette borne supérieure.

Application à PRESENT : jusqu'où peut-on aller ?

- On a atteint des bornes supérieures sur le nombre de tours de PRESENT-80/128 attaquable avec un meet-in-the-middle, par notre amélioration.
- En la combinant avec la méthode des bicliques proposée par Khovratovitch, Rechberger, Savelieva en 2011, on gagne encore encore un tour.

Bilan et perspectives

Ce qu'on a fait

- Amélioration générique des meet-in-the-middle : "meet-through-an-Sbox"

Travail en cours, avec Anne Canteaut

- Que se passe-t-il lorsque le nombre de bits contrôlés est plus petit que la taille des entrées ? Avec des boîtes S non-bijectives ?
- Autres exemples d'application ?

Merci de votre attention !