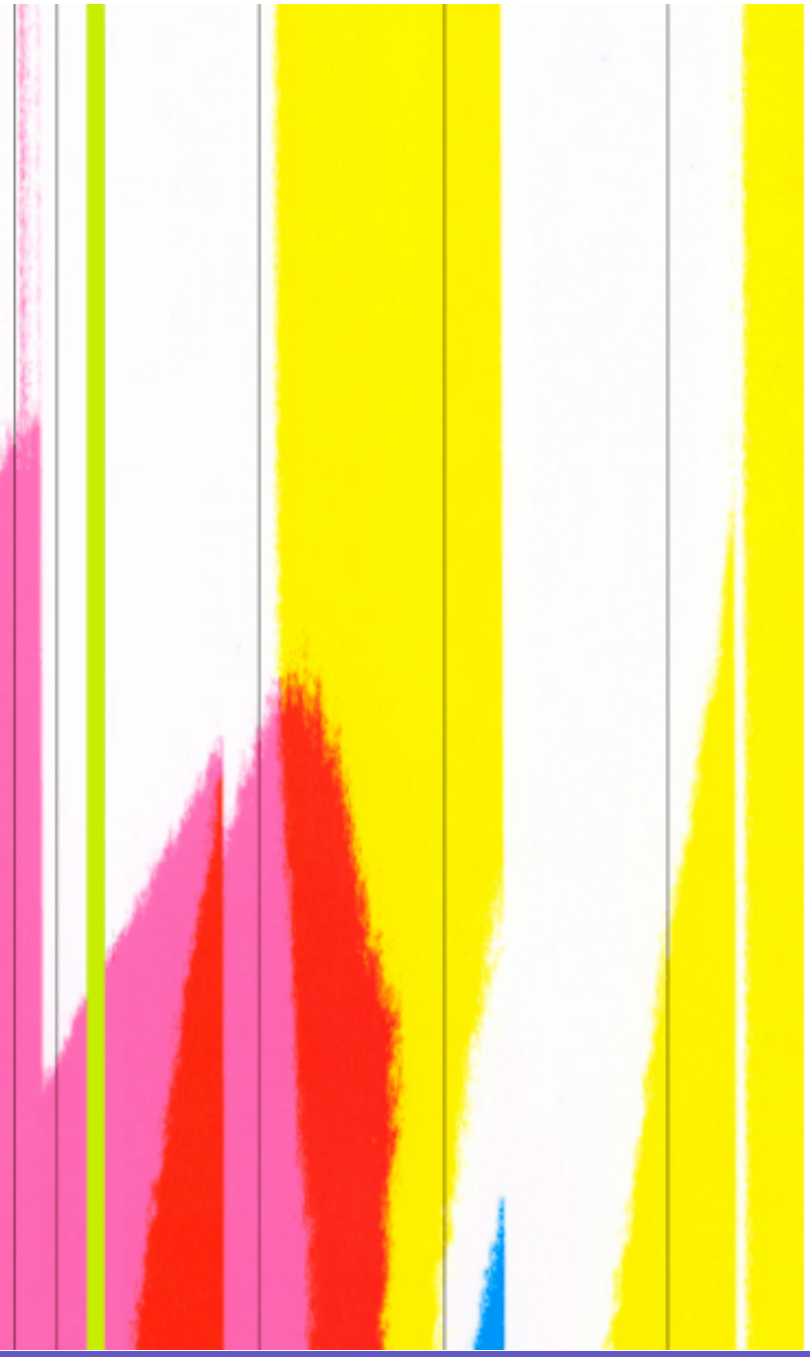


Produire une collision pour SHA-0 en une heure

Stéphane Manuel
Thomas Peyrin

INRIA Rocquencourt, Équipe SECRET
Orange Labs - AIST

Journées Codage et Cryptographie
17-21 mars 2008
Carcans Maubuisson

Outline

1. Introduction
2. Précédentes attaques par collision
3. Nouveaux résultats
4. Conclusion

Outline

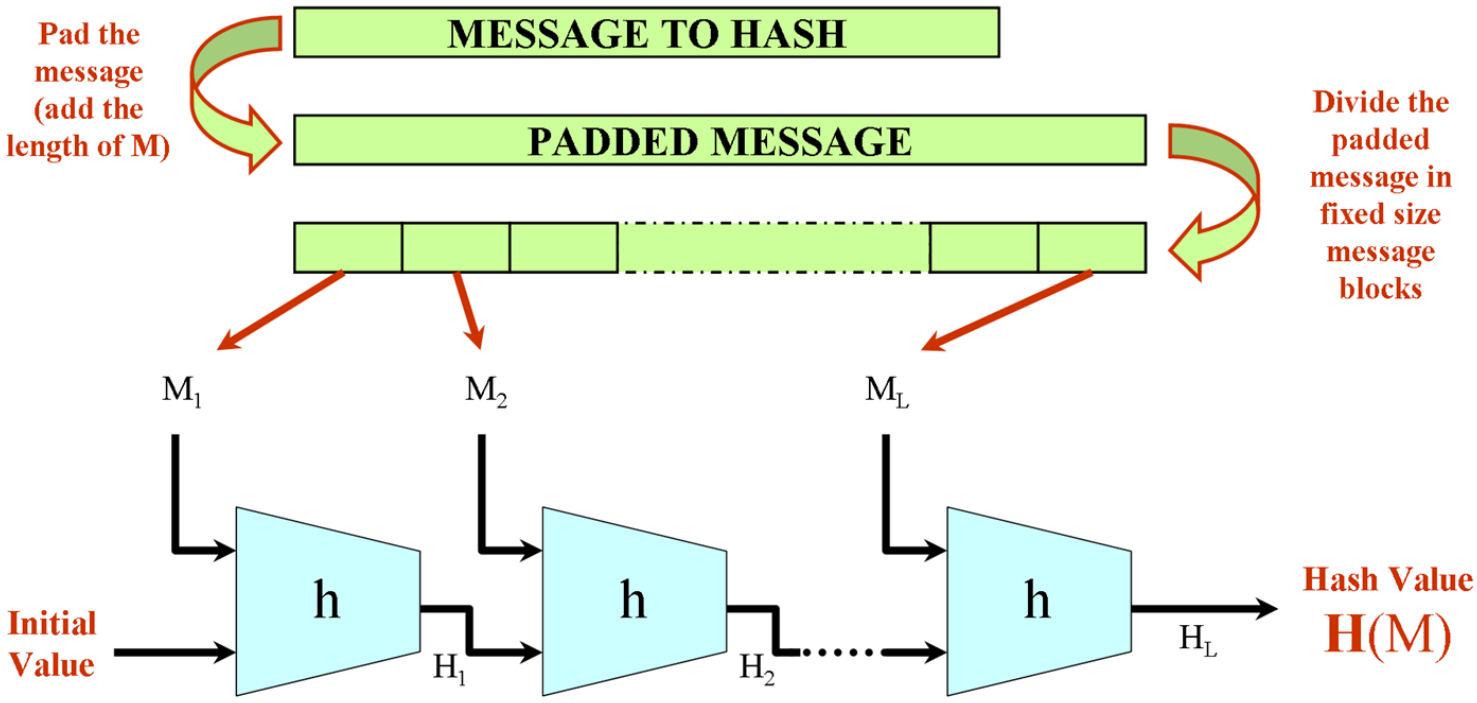
1. Introduction
2. Précédentes attaques par collision
3. Nouveaux résultats
4. Conclusion

Fonction de hachage cryptographique

- Une fonction de hachage est un algorithme prenant en entrée une chaîne de bits de longueur arbitraire finie (le message) et restituant en sortie une chaîne de bits de longueur fixée (le haché).
- Propriétés cryptographiques classiques :
  - Résistance aux attaques par pre-image : étant donné un haché quelconque fixé, il est impossible de construire un message correspondant à ce haché.
  - Résistance aux attaques de seconde pre-image : étant donné un message quelconque fixé, il est impossible de trouver un autre message possédant le même haché.
  - Résistance aux attaques par collision : il est impossible de trouver deux messages distincts possédant un même haché.

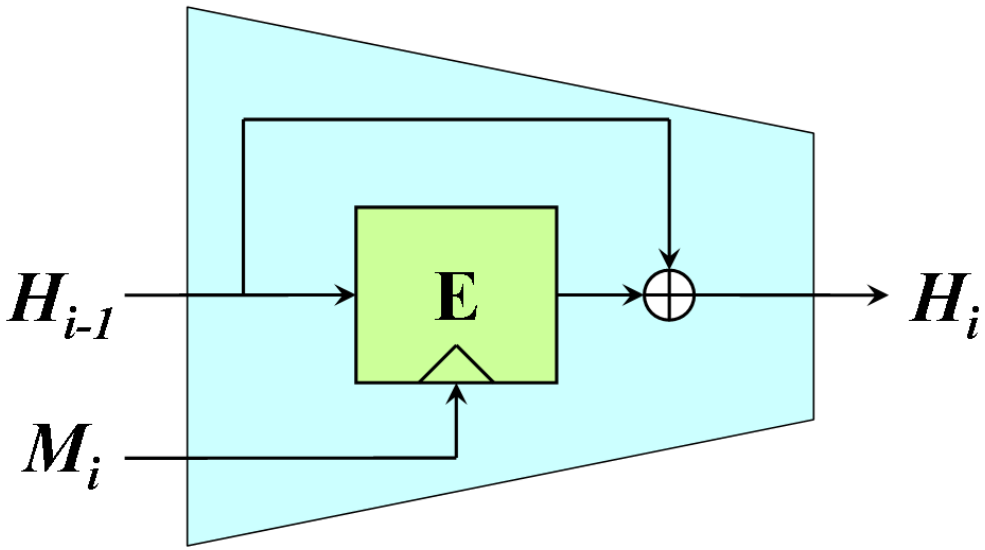
Extenseur de domaine

- L'algorithme de Merkle-Damgård renforcé :



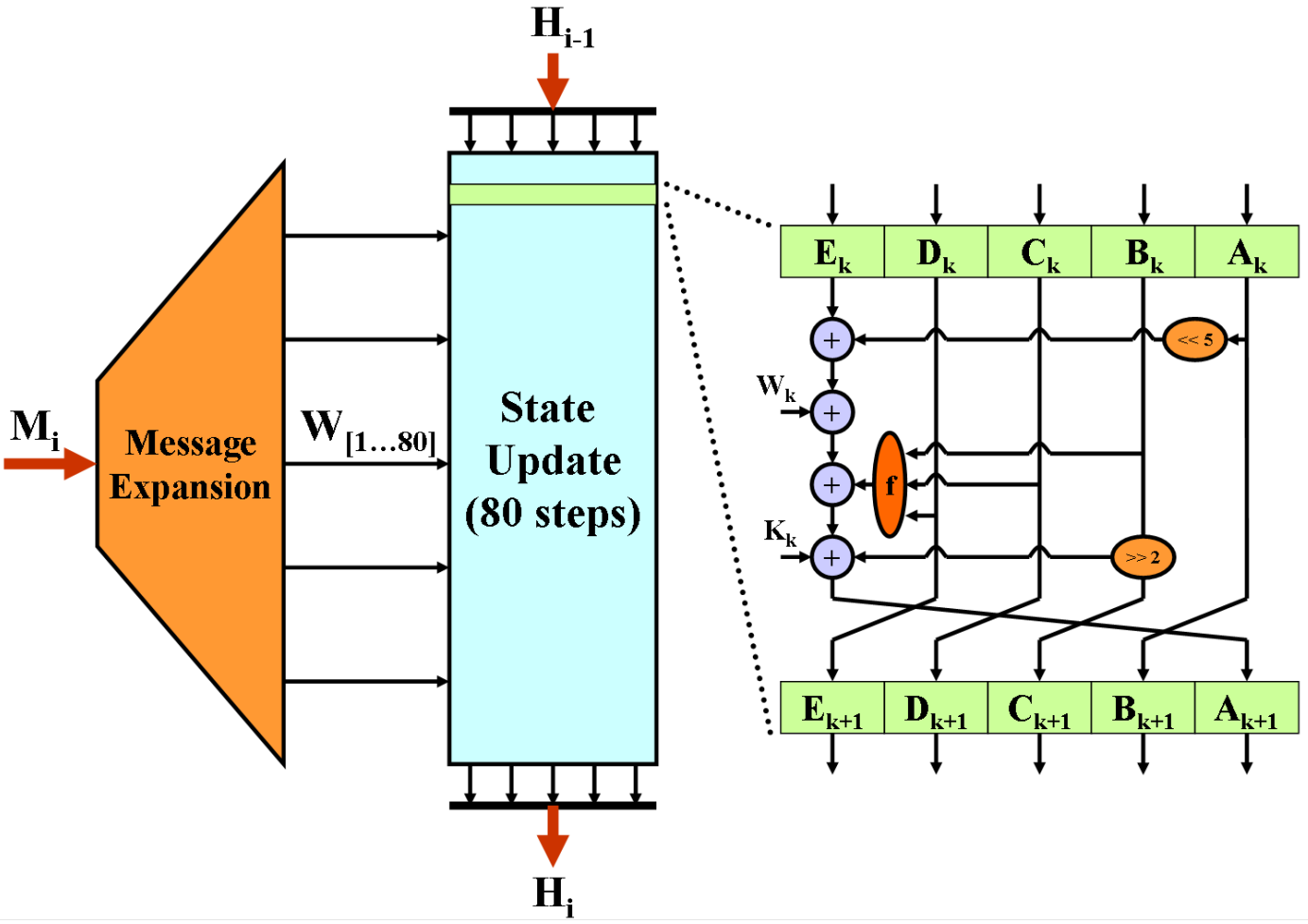
Fonction de compression

- La construction de Davies-Meyer :



La fonction SHA-0

- Construite en 1993, 160 bits en sortie.



La fonction SHA-0

- Expansion de message :

$$W_k = \begin{cases} M_k, & \text{pour } 0 \leq k \leq 15 \\ W_{k-16} \oplus W_{k-14} \oplus W_{k-8} \oplus W_{k-3}, & \text{pour } 16 \leq k \leq 79 \end{cases}$$

- Fonctions booléennes :

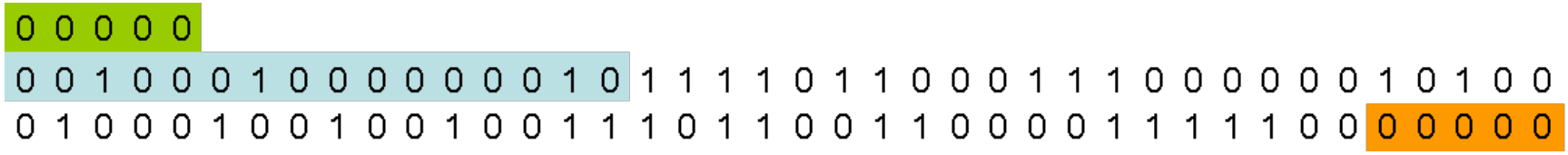
pas k	$f_k(B, C, D)$
$1 \leq k \leq 20$	$f_{IF} = (B \wedge C) \oplus (\overline{B} \wedge D)$
$21 \leq k \leq 40$	$f_{XOR} = B \oplus C \oplus D$
$41 \leq k \leq 60$	$f_{MAJ} = (B \wedge C) \oplus (B \wedge D) \oplus (C \wedge D)$
$61 \leq k \leq 80$	$f_{XOR} = B \oplus C \oplus D$

Outline

1. Introduction
2. Précédentes attaques par collision
3. Nouveaux résultats
4. Conclusion

Chabaud et Joux [CRYPTO 98]

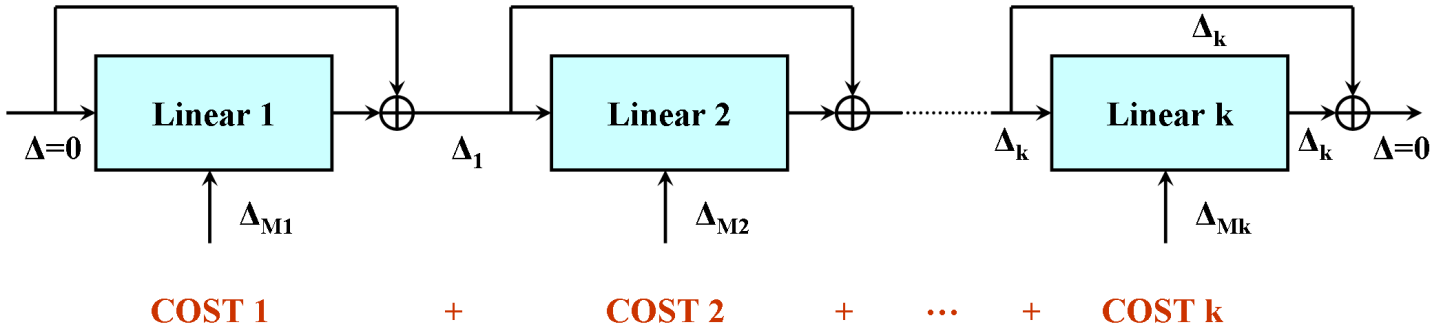
- Collisions locales : insertion d'une perturbation puis de corrections dans les 5 pas suivants.
- Trouver des chemins différentiels linéaires constitués de collisions locales entrelacées : caractéristique linéaire.
- Trois contraintes sur le vecteur de perturbations :
  - pas de collision locale tronquée,
  - pas de perturbations consécutives dans les 16 premiers pas,
  - pas de collision locale commençant après le pas 74.



La complexité de l'attaque est évaluée par la probabilité que toutes les collisions locales soient conformes.

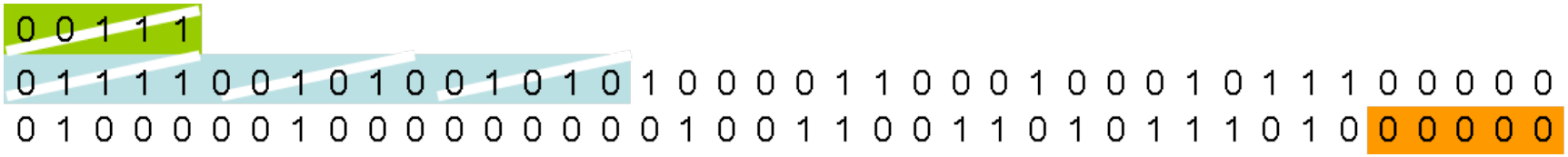
Biham *et al.*

- Biham et Chen [CRYPTO 04] : bits neutres
  - Technique d'accélération durant la phase de recherche de collision.
  - Permet de générer efficacement un ensemble de messages qui se conforment au chemin différentiel jusqu'à un certain pas.
- Biham *et al.* [EUROCRYPT 2005] : blocs multiples
  - Utilisation de plusieurs blocs de message pour construire la collision.
  - Composition de pseudo-collisions et de quasi-collisions.



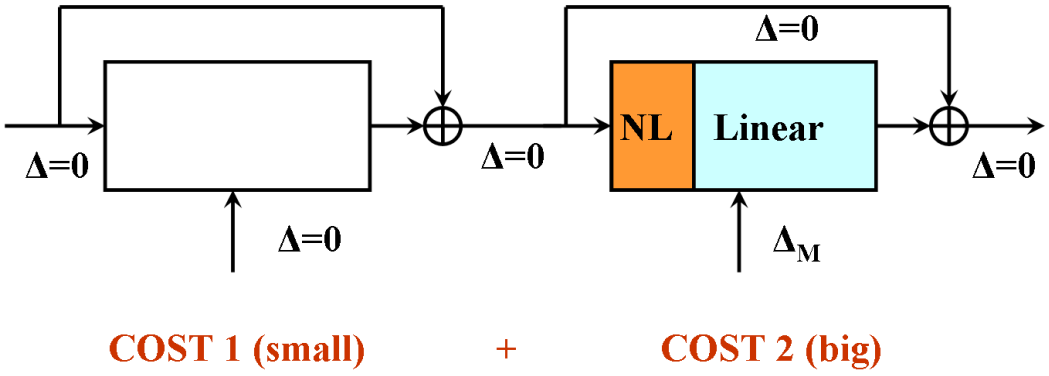
Wang *et al.* [CRYPTO 05]

- Relâcher les deux premières contraintes sur les vecteurs de perturbations.
- Modifier (à la main) les premiers pas du chemin différentiel pour compenser les collisions locales tronquées et les perturbations consécutives : caractéristique non-linéaire.
- Utiliser de “nouveaux” outils :
  - soustraction modulaire,
  - la propagation de retenue,
  - la non-linéarité de la fonction f_{IF}.



Wang *et al.* [CRYPTO 05]

- Modifications de message : nouvelle technique d'accélération.
- Construction d'un premier bloc de message aléatoire dont la variable de chaînage vérifie des conditions spécifiques (16 conditions).



La complexité est donnée en terme de nombre de conditions à vérifier (avec un décompte commençant au pas 20).

Naito *et al.* [ASIACRYPT 06]

- Utilise les caractéristiques, linéaire et non-linéaire, de Wang *et al.*
- Modifications sous-marines : décompte des conditions à partir du pas 24.
- Complexité :
  - théoriquement 2^{36} appels à la fonction ...
  - ... mais en moyenne 100 heures de calcul.
  - Notre estimation : $2^{40,5}$ appels à la fonction en pratique.

De Cannière *et al.* [Hash Workshop 2007] ont proposé que la complexité soit donnée en nombre d'appels à la fonction relativement à une implémentation efficace (*i.e.* OpenSSL) sur une plate-forme standard.

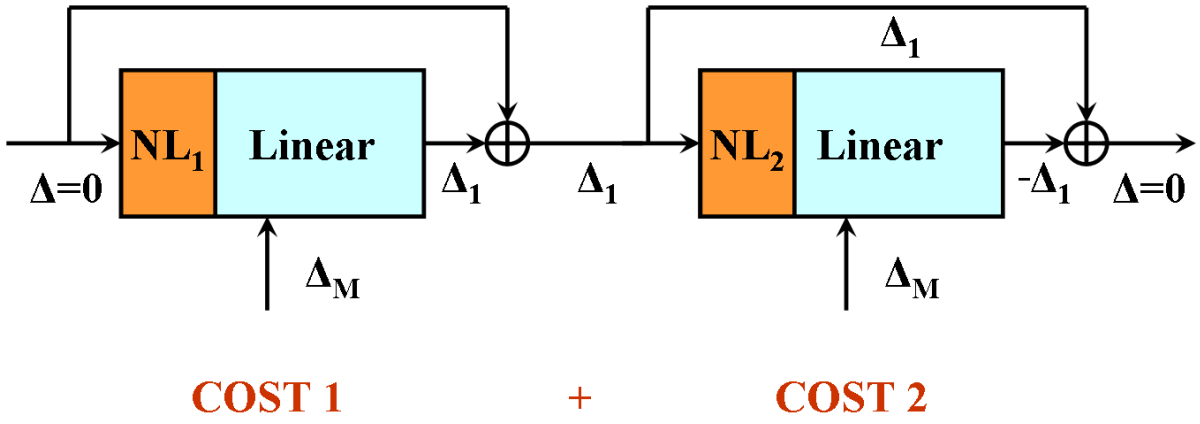
Outline

1. Introduction
2. Précédentes attaques par collision
3. **Nouveaux résultats**
4. Conclusion

Améliorations possibles

- Relâcher la dernière contrainte afin de trouver les “meilleurs” vecteurs de perturbations :
  - pas de collision locale commençant après le pas 74.

- Ce qui implique :
  - d'utiliser la technique des blocs multiples,
  - de trouver des caractéristiques non-linéaires adaptées,
  - d'utiliser une technique d'accélération générique.



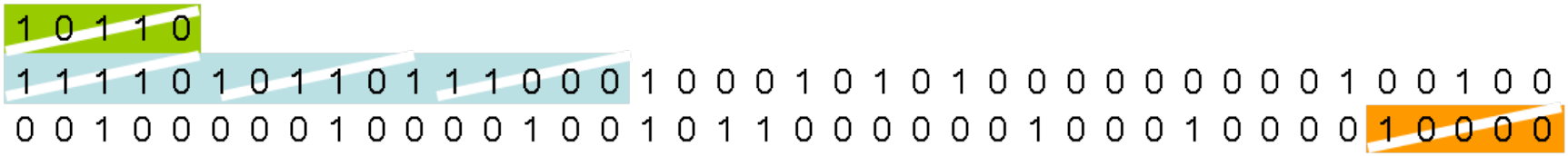
Améliorations possibles

Adapter les outils développés pour les dernières attaques contre SHA-1.

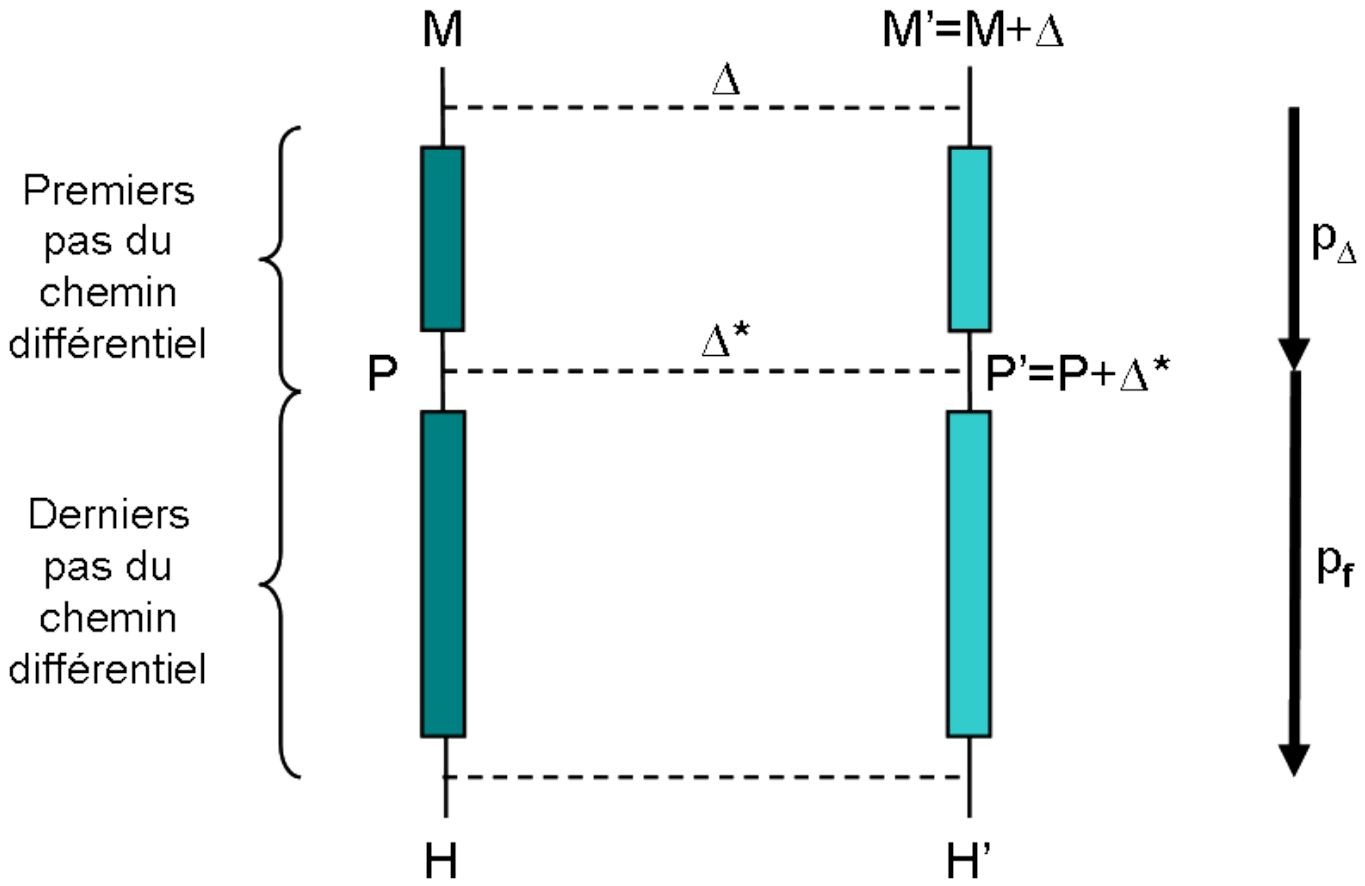
- Caractéristique non-linéaire :
  - générateur de caractéristiques non-linéaires de De Cannière et Rechberger (2006).
- Technique d'accélération :
  - attaques par boomerang de Joux et Peyrin (2007).

Nouveau vecteur de perturbations

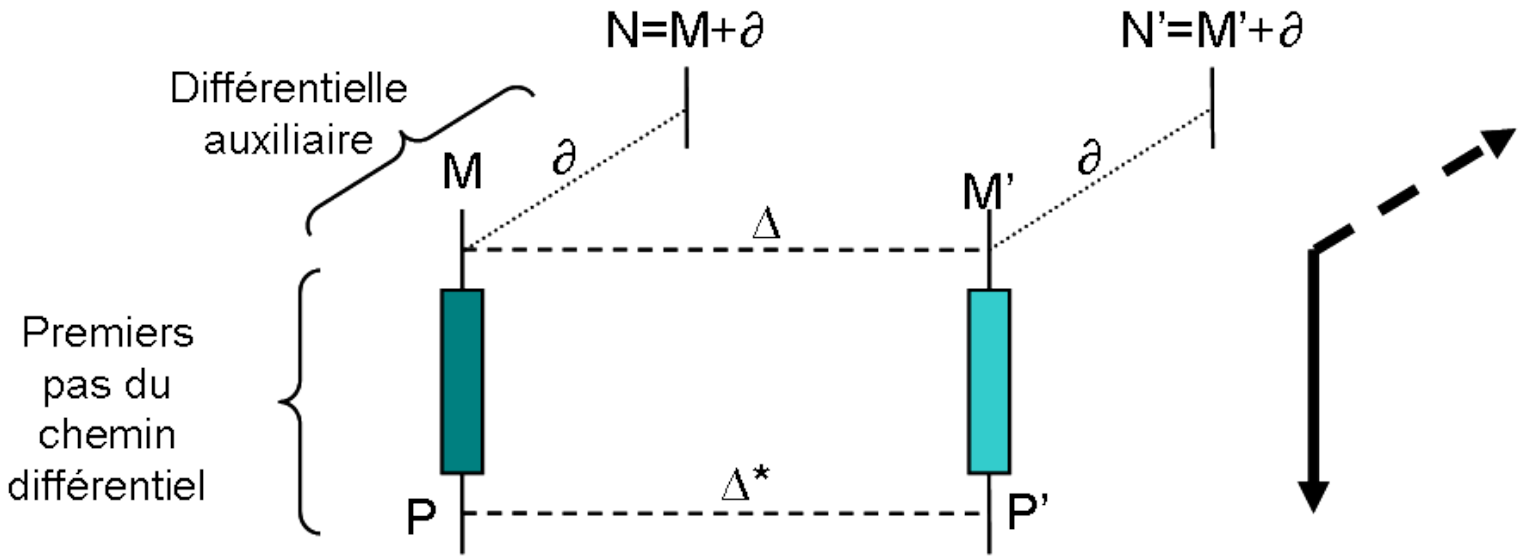
- Critères de recherche :
  - minimiser le nombre de conditions entre les pas 16 et 80,
  - le pas de départ pour le décompte des conditions dépend de la technique d'accélération,
  - adéquation avec le générateur de caractéristiques non-linéaires.
- Plusieurs bons candidats possible.
  - Notre vecteur :



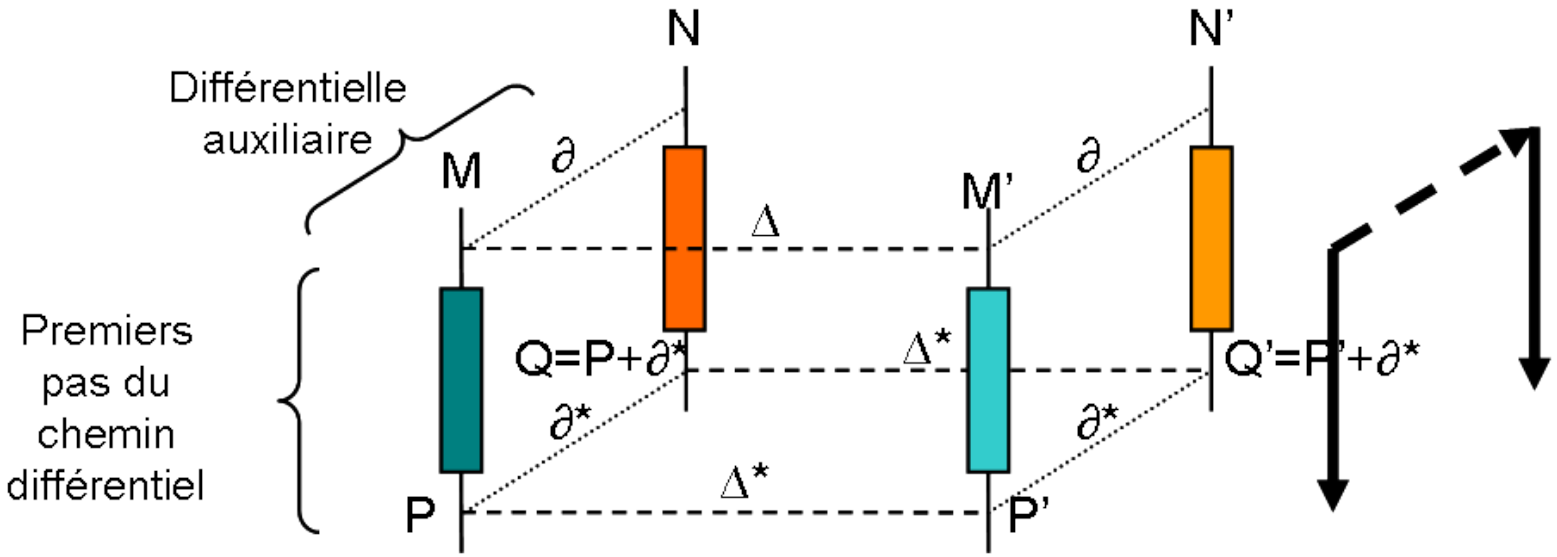
Attaque par boomerangs



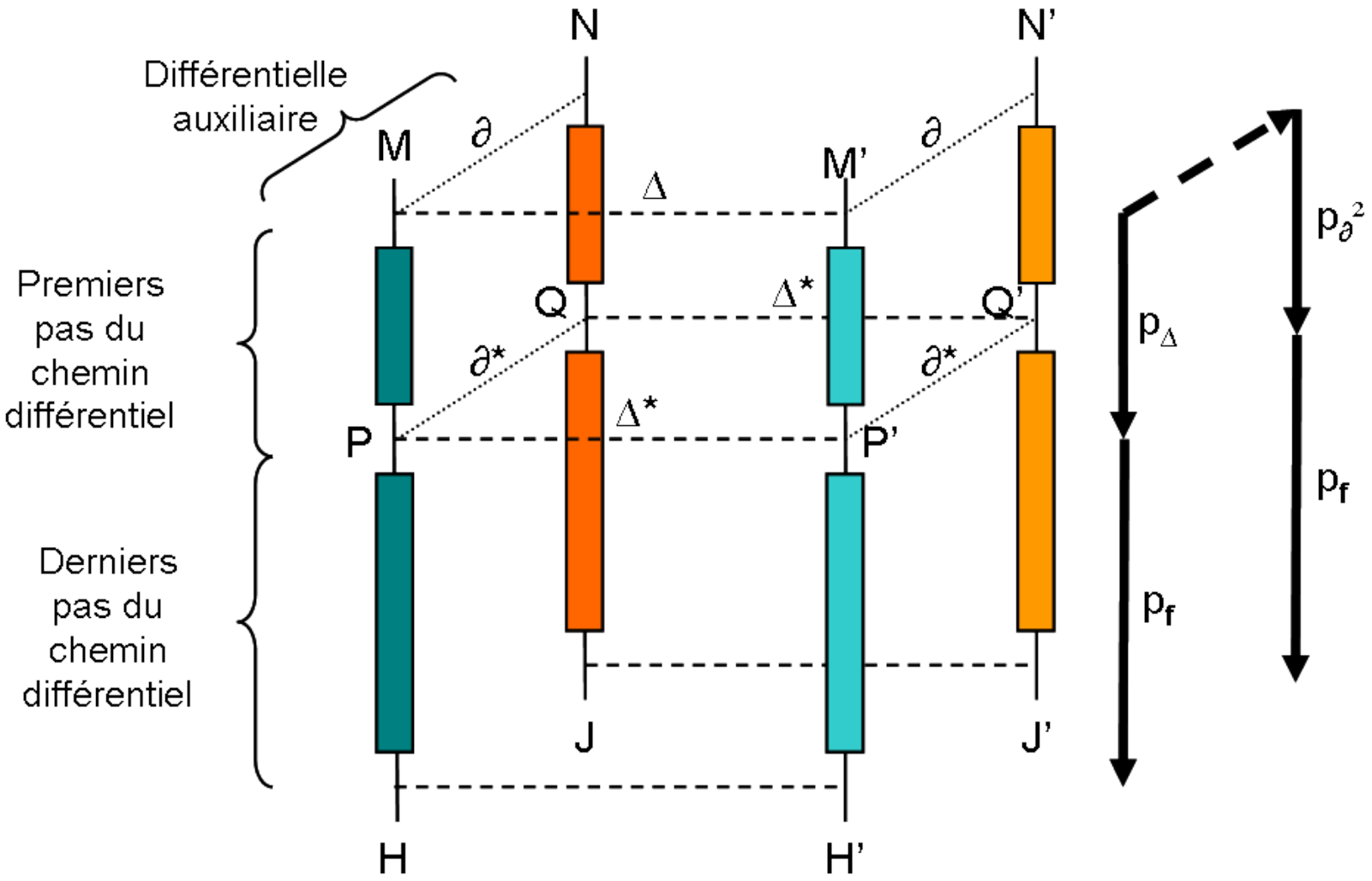
Attaque par boomerangs



Attaque par boomerangs



Attaque par boomerangs



Attaque par boomerangs

- Les boomerangs sont une trame :
  - L'attaquant construit des différentielles auxiliaires utilisables dans les contextes de bits neutres ou de modifications de message.
  - Dans le contexte des bits neutres elles constituent un outil facile à utiliser durant la phase d'accélération.
  - Les contraintes sont choisies et imposées au chemin différentiel afin de donner de bons bits neutres.

- Notre approche :
  1. Trouver de bonnes différentielles auxiliaires génériques.
  2. Les placer de façon à ce qu'elles n'interfèrent pas avec la caractéristique linéaire.
  3. Faire tourner le générateur de caractéristiques non-linéaires en prenant en compte ces différentielles auxiliaires.

Différentielles auxiliaires

- Deux types de différentielles auxiliaires :
  - légère mais courte (peu de contraintes mais peu étendue),
  - lourde mais longue (plus étendue mais de nombreuses contraintes).
- Ces différentielles auxiliaires sont utilisées comme bits neutres pour les pas 23 et 28 respectivement.
- Nous avons placé 5 différentielles auxiliaires (7 dans le cas du premier bloc) :
  - amélioration d'un facteur 2^5 sur l'attaque brute.

Première différentielle auxiliaire

	W_0 to W_{15}	W_{16} to W_{31}
perturbation mask	0000001000000000	
differences on W^j	0000001000000000	0000101101100111
differences on W^{j+5}	0000000100000000	0000010110110011
differences on W^{j-2}	0000000000010000	000$\underline{1}$001000000010

i	A_i	W_i
-1 :	----------	
00 :	----------	----------
01 :	----------	----------
02 :	----------	----------
03 :	----------	----------
04 :	----------	----------
05 :	----------b--	----------
06 :	----------b--	----------a-
07 :	----------a-	----------$\bar{a}$----
08 :	----------0	----------
09 :	----------1	----------
10 :	----------	----------
11 :	----------	----------$\bar{a}$
12 :	----------	----------
13 :	----------	----------
14 :	----------	----------
15 :	----------	----------

Seconde différentielle auxiliaire

	W_0 to W_{15}	W_{16} to W_{31}
perturbation mask	1010000000100000	
differences on W^j	1010000000100000	0000000$\underline{1}$0110110
differences on W^{j+5}	0101000000010000	0000000001011011
differences on W^{j-2}	0001111100000011	0000000000001110

i	A_i	W_i
-1 :	----------d--	
00 :	----------d--	----------a-
01 :	----------e-a-	----------$\bar{a}$----
02 :	----------e--1	----------b-
03 :	----------b-0	----------$\bar{b}$---$\bar{a}$
04 :	----------0	----------$\bar{a}$
05 :	----------0	----------$\bar{a}$
06 :	----------	----------$\bar{b}$
07 :	----------	----------$\bar{b}$
08 :	----------	----------
09 :	----------f--	----------
10 :	----------f--	----------c-
11 :	----------c-	----------$\bar{c}$----
12 :	----------0	----------
13 :	----------0	----------
14 :	----------	----------$\bar{c}$
15 :	----------	----------$\bar{c}$

Exemple de collision

	1^{st} block		2^{nd} block	
	M_1	M'_1	M_2	M'_2
W_0	0x4643450b	0x46434549	0x9a74cf70	0x9a74cf32
W_1	0x41d35081	0x41d350c1	0x04f9957d	0x04f9953d
W_2	0xfe16dd9b	0xfe16dddb	0xee26223d	0xee26227d
W_3	0x3ba36244	0x3ba36204	0x9a06e4b5	0x9a06e4f5
W_4	0xe6424055	0x66424017	0xb8408af6	0x38408ab4
W_5	0x16ca44a0	0x96ca44a0	0xb8608612	0x38608612
W_6	0x20f62444	0xa0f62404	0x8b7e0fea	0x0b7e0faa
W_7	0x10f7465a	0x10f7465a	0xe17e363c	0xe17e363c
W_8	0x5a711887	0x5a7118c5	0xa2f1b8e5	0xa2f1b8a7
W_9	0x51479678	0xd147963a	0xca079936	0x4a079974
W_{10}	0x726a0718	0x726a0718	0x02f2a7cb	0x02f2a7cb
W_{11}	0x703f5bfb	0x703f5bb9	0xf724e838	0xf724e87a
W_{12}	0xb7d61841	0xb7d61801	0x37ffc03a	0x37ffc07a
W_{13}	0xa5280003	0xa5280041	0x53aa8c43	0x53aa8c01
W_{14}	0x6b08d26e	0x6b08d26c	0x90811819	0x9081181b
W_{15}	0x2e4df0d8	0xae4df0d8	0x312d423e	0xb12d423e

A_2	B_2	C_2	D_2	E_2
0x6f84b892	0x1f9f2aae	0x0dbab75c	0x0afe56f5	0xa7974c90

Outline

1. Introduction
2. Précédentes attaques par collision
3. Nouveaux résultats
4. Conclusion

Comparaison des complexités

Équipe	Théorique	Pratique	Temps sur PC
Chabaud et Joux (1998)	2^{61}		
Biham *et al.* (2004)	2^{51}	2^{51}	20 ans
Wang *et al.* (2005)	2^{39}		
Naito *et al.* (2006)	2^{36}	$2^{40.3}$	100 heures
Manuel et Peyrin (2008)	2^{33}	$2^{33.6}$	1 heure

Références

- ***Differential Collisions in SHA-0***, F. Chabaud and A. Joux. CRYPTO'98, LNCS 1462, pages 56–71.
- ***Near-Collisions of SHA-0***, E. Biham and R. Chen. CRYPTO'04, LNCS 3152, pages 290–305.
- ***Collisions of SHA-0 and Reduced SHA-1***, E. Biham, R. Chen, A. Joux, P. Carribault, C. Lemuet and W. Jalby. EUROCRYPT 2005, LNCS 3494, pages 36–57.
- ***Efficient Collision Search Attacks on SHA-0***, X. Wang, H. Yu and Y.L. Yin. CRYPTO'05, LNCS 3621, pages 1–16.
- ***Improved Collision Search for SHA-0*** Y. Naito, Y. Sasaki, T. Shimoyama, J. Yajima, N. Kunihiro and K. Ohta. ASIACRYPT'06, LNCS 4284, pages 21–36.
- ***Collisions on SHA-0 in one hour***, S. Manuel and T. Peyrin. FSE 2008.