

Mémoire de master  
Étude et automatisation de l'attaque de Wang sur MD4

Gaëtan Leurent

sous la direction de:  
Pierre-Alain Fouque et Phong Q. Nguyễn

Année universitaire 2005–2006

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Fonction de hachage . . . . .	4
1.2	Schéma de Merkle-Damgård . . . . .	6
1.3	MD4 . . . . .	7
<b>2</b>	<b>Description de l'attaque de Wang et al.</b>	<b>9</b>
2.1	Chemin différentiel . . . . .	10
2.1.1	Notation pour la différentielle . . . . .	10
2.1.2	Rapport entre $\delta$ et $\partial$ . . . . .	10
2.2	Conditions suffisantes . . . . .	11
2.3	Modifications de message . . . . .	12
2.3.1	Modifications simples . . . . .	12
2.3.2	Modifications multi-messages . . . . .	12
<b>3</b>	<b>Recherche de chemin différentiel</b>	<b>12</b>
3.1	Outils mathématiques . . . . .	13
3.1.1	Interactions entre $\delta$ et $\lll$ . . . . .	13
3.1.2	Interactions entre $\sum$ et $\lll$ . . . . .	14
3.2	Choix de la différentielle $\Delta$ sur les messages . . . . .	16
3.2.1	La différentielle de [YWZW05] . . . . .	16
3.2.2	La différentielle de [WLF <sup>+</sup> 05] . . . . .	17
3.3	Calcul des conditions suffisantes . . . . .	18
3.3.1	Fixer le bon $\partial_i$ . . . . .	18
3.3.2	Choix à effectuer . . . . .	20
3.4	Recherche du chemin . . . . .	21
3.4.1	Idée générale . . . . .	21
3.4.2	Structures de données . . . . .	21
3.4.3	Corriger les erreurs . . . . .	22
3.4.4	Corrections indirectes . . . . .	24
3.4.5	Diriger la recherche . . . . .	26
<b>4</b>	<b>Conclusion</b>	<b>27</b>
4.1	Résultats . . . . .	27
<b>A</b>	<b>Chemins différentiels</b>	<b>29</b>

## Table des figures

1	La méthode $\rho$ . . . . .	6
2	Schéma de Merkle-Damgård . . . . .	7
3	Fonction de compression de MD4 . . . . .	8

## Liste des tableaux

1	Le dernier tour du chemin de [YWZW05]. . . . .	17
2	Une collision locale dans le dernier tour . . . . .	17
3	Le dernier tour du chemin de [WLF <sup>+</sup> 05] . . . . .	18
4	Conditions sur les fonctions $\Phi_i$ . . . . .	19
5	Un exemple de morceau de chemin, avec ses conditions suffisantes . . . . .	20
6	Exécution de l'algorithme sur le chemin de [WLF <sup>+</sup> 05] . . . . .	23
7	Une modification indirecte . . . . .	25
8	Comparaison des différents chemins avec la différentielle de Wang . . . . .	27
13	Détail d'une collision . . . . .	40
14	Un chemin pour la première partie de l'attaque sur HMAC-MD4 . . . . .	41
15	Un chemin pour la deuxième partie de l'attaque sur HMAC-MD4 . . . . .	42

## Liste des algorithmes

1	Recherche de collisions avec la méthode $\rho$ . . . . .	5
2	Recherche de chemin différentiel . . . . .	22
3	Calcul du bit à modifier . . . . .	24
4	Calcul du bit à modifier avec modifications indirectes . . . . .	26

## Liste des chemins

1	Le chemin de Wang [WLF <sup>+</sup> 05] . . . . .	30
2	Le chemin de [SO06] . . . . .	32
3	Un meilleur chemin avec la même différentielle . . . . .	34
4	Un chemin avec une autre collision locale dans le dernier tour . . . . .	36
5	Le chemin de [YWZW05] . . . . .	38
6	Un meilleur chemin en introduisant la différence sur le bit 25 . . . . .	39

## 1 Introduction

J'ai effectué mon stage au Laboratoire d'Informatique de l'ENS, sous la direction de Pierre-Alain Fouque et Phong Q. Nguyen. Le but du stage était de comprendre et d'automatiser les attaques récentes sur les fonctions de hachage de la famille de MD4, par l'équipe de Xiaoyun Wang. Son équipe a attaqué MD4 [WLF<sup>+</sup>05], MD5 [WY05], SHA-0 [WYY05b], et SHA-1 [WYY05a]. Je me suis concentré sur MD4, et après avoir implémenté son attaque j'ai automatisé la phase de recherche du chemin différentiel, qui est encore mal comprise.

Dans ce rapport, je vais présenter les fonctions de hachage de façon générale et la famille de MD4, puis décrire l'attaque de Wang, et enfin comment j'ai automatisé la recherche de chemin différentiel.

### 1.1 Fonction de hachage

Les fonctions de hachage sont une primitive importante en cryptographie. Une fonction de hachage idéale serait un oracle aléatoire, c'est à dire une fonction (déterministe)  $H : \{0, 1\}^* \rightarrow \{0, 1\}^n$  qui prend en entrée une suite de bits quelconque, et donne en sortie une suite de bits de taille fixe, qui est choisie aléatoirement la première fois qu'on demande  $H(x)$ . En pratique, cela signifie qu'on ne doit avoir aucune information sur  $x$  à partir de  $H(x)$ , et que la seule façon d'apprendre quelque chose sur  $H(x)$  à partir de  $x$  est de faire le calcul. En particulier, la probabilité que deux entrées différentes aient le même haché est de  $2^{-n}$ , et on choisira  $n$  suffisamment grand pour que ce soit négligeable ( $n \geq 128$ ). Une telle fonction peut donc servir pour vérifier l'intégrité d'un objet, pour faire de la mise en gage, ou comme brique de base dans de nombreuses constructions cryptographique, par exemple pour faire de l'authentification.

Le niveau de sécurité qu'on demande à une fonction de hachage cryptographique est beaucoup plus élevé que ce qu'on demande pour d'autres usages, par exemple pour construire une table de hachage. On demande notamment les trois propriétés suivantes :

**Définition 1.** *Résistance en préimage.* Étant donné  $H(m)$  pour un message aléatoire  $m$ , il doit être difficile de trouver  $m'$  tel que  $H(m) = H(m')$ .

Pour une fonction idéale il faut en moyenne  $2^n$  appels à  $H$  pour trouver une préimage ; si on a un algorithme plus efficace sur une fonction de hachage donnée, on considèrera qu'elle est cassée.

**Définition 2.** *Resistance en seconde préimage.* Étant donné un message aléatoire  $m$ , il doit être difficile de trouver  $m' \neq m$  tel que  $H(m) = H(m')$ .

Pour une fonction idéale il faut en moyenne  $2^n$  appels à  $H$  pour trouver une seconde préimage ; si on a un algorithme plus efficace la fonction de hachage est cassée.

**Définition 3.** *Resistance en collision.* Il doit être difficile de trouver deux messages  $m' \neq m$  tel que  $H(m) = H(m')$ . On considère aussi parfois des near-collisions, c'est à dire des paires de messages dont les hachés sont proches, pour une certaine mesure.

Pour une fonction idéale il faut en moyenne  $2^{n/2}$  appels à  $H$  pour trouver une collision, à cause du paradoxe des anniversaires (cf proposition 1). En effet, si on calcule  $2^{n/2}$  hachés de messages aléatoire, on a  $2^{n/2}$  valeurs aléatoires dans  $\{0, 1\}^n$ , et on a une bonne chance d'avoir une collision. Un algorithme naïf requiert un espace mémoire en  $O(2^{n/2})$  mais on peut trouver des collision en espace mémoire constant en utilisant la méthode  $\rho$ , décrite par l'algorithme 1 page ci-contre, et illustrée par la figure 1.

**Proposition 1.** *Paradoxe des anniversaires : on considère un ensemble fini  $\mathcal{X}$  de  $N$  éléments. La probabilité d'avoir une collision en tirant aléatoirement  $k$  éléments de façon uniforme est :*

$$\begin{aligned} P(k) &= 1 - \left( \frac{N-1}{N} \right)^{\frac{k(k-1)}{2}} \\ &= 1 - \exp \left( \frac{k(k-1)}{2} \ln \left( 1 - \frac{1}{N} \right) \right) \\ &\geq 1 - \exp \left( -\frac{k(k-1)}{2N} \right) \end{aligned}$$

Ainsi, on a  $P(\sqrt{N} + 1) > 1 - e^{1/2} > 0.39$ , et la probabilité augmente très rapidement avec  $k$  :  $P(1.2\sqrt{N} + 1) > 0.5$ ,  $P(2\sqrt{N} + 1) > 0.86$ ,  $P(4\sqrt{N} + 1) > 0.999$ .

---

**Algorithme 1** Recherche de collisions avec la méthode  $\rho$

---

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

**Output:**  $x, y \in \{0, 1\}^*$  tel que  $H(x) = H(y)$

```

1:  $x \leftarrow \epsilon$  //  $\epsilon \notin \text{Im}H$ 
2:  $y \leftarrow \epsilon$ 
3: repeat
4:    $x \leftarrow H(x)$ 
5:    $y \leftarrow H(H(y))$ 
6: until  $x = y$ 
7:  $y \leftarrow \epsilon$ 
8: loop
9:    $x' \leftarrow H(x)$ 
10:   $y' \leftarrow H(y)$ 
11:  if  $x' = y'$  then
12:    return  $x, y$ 
13:  else
14:     $x \leftarrow x'$ 
15:     $y \leftarrow y'$ 
16:  end if
17: end loop

```

---

*Preuve de l'algorithme 1.* Il est évident que si l'algorithme termine il trouve bien une collision.

On note  $\epsilon_i = f^n(\epsilon)$  l'itérée  $i$ -ème de  $f$  appliquée à  $\epsilon$ , et  $x_i$  (respectivement  $y_i$ ) la valeurs de  $x$  (respectivement  $y$ ) au début de la  $i$ -ème itération de la première boucle (ligne 3). On a  $x_i = \epsilon_i$  et  $y_i = \epsilon_{2i} = x_{2i}$ . Si la fonction  $H$  est idéale, les valeur de la suite  $(\epsilon_i)_i$  sont aléatoirement réparties dans  $\{0, 1\}^n$ , on aura donc une collision au bout de  $O(2^{n/2})$  itération :  $\epsilon_{i_1} = \epsilon_{i_2}$  avec  $i_1 < i_2$ . Alors pour tout  $k \geq 0$ , on a aussi  $\epsilon_{i_1+k} = \epsilon_{i_2+k}$ , et par récurrence, pour tout  $\alpha \in \mathbb{N}$  et  $i \geq i_1$ ,  $\epsilon_i = \epsilon_{i+\alpha(i_2-i_1)}$ . En particulier, avec  $a = \left\lceil \frac{i_1}{i_2-i_1} \right\rceil$ , on a :

$$x_{a(i_2-i_1)} = \epsilon_{a(i_2-i_1)} = \epsilon_{a(i_2-i_1)+a(i_2-i_1)} = x_{2a(i_2-i_1)} = y_{a(i_2-i_1)}$$

avec  $i_1 \leq a(i_2 - i_1) < i_1 + (i_2 - i_1)$  : la collision sera repérée après  $i_2$  itérations de la boucle ligne 3 au maximum.



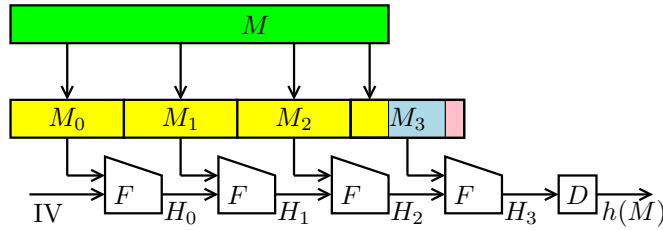


FIG. 2: Schéma de Merkle-Damgård

de hachage obtenue en appliquant le schéma de Merkle-Damgård avec MD-Strengthening est résistante en collision.

*Démonstration.* Soit  $M \neq M'$  deux messages de longueur  $n$  et  $n'$  (qui font  $l$  et  $l'$  blocs après padding) tels que  $H(M) = H(M')$ .

Si  $n \neq n'$ , alors le dernier bloc (avec le padding) de  $M$  et  $M'$  est différent<sup>1</sup>, et on a une pseudo-collision  $F(H_{l-1}, M_l) = F(H'_{l'-1}, M'_{l'})$ .

Sinon, soit  $k_1 = \max \{i : M_i \neq M'_i\}$  (cet ensemble est non-vidé car les messages sont différents et de même longueur) et  $k_2 = \min \{i \geq k_1 : H_i = H'_i\}$  (cet ensemble est non-vidé car  $H_l = H'_l$ ). Si  $k_1 = k_2$ , alors  $F(H_{k_1-1}, M_{k_1}) = H_{k_1} = H'_{k_1} = F(H'_{k_1-1}, M'_{k_1})$  fournit une pseudo-collision dans  $F$  car  $M_{k_1} \neq M'_{k_1}$ . Sinon, on a  $F(H_{k_2-1}, M_{k_2}) = H_{k_2} = H'_{k_2} = F(H'_{k_2-1}, M'_{k_2})$ , d'où une pseudo-collision car  $H_{k_2-1} \neq H'_{k_2-1}$ .  $\square$

Cependant ce mode de construction a aussi des inconvénients : si on donne tout l'état interne en sortie on peut calculer le haché d'un message  $M||M'$  en connaissant seulement  $H(M)$  et pas  $M$ , et si on connaît une collision entre deux messages de même taille on peut en générer un nombre arbitraire en ajoutant un suffixe aux messages. De même le schéma de Merkle-Damgård permet de construire efficacement des multi-collisions qui peuvent être exploitées de nombreuses façons [Jou04].

### 1.3 MD4

MD4 est une fonction de hachage construite en suivant le schéma de Merkle-Damgård avec *MD-Strengthening*. Son état interne et sa sortie sont sur 128 bits.

La fonction de compression de MD4 peut être vue comme construite à partir d'un système de chiffrement par bloc : si on note  $C_k$  la fonction de chiffrement avec la clef  $k$ , on a  $F(H, M) = C_M(H) \boxplus H$ . C'est la construction de Davies-Meyer, et si on suppose que la fonction de chiffrement est indistinguable d'une fonction aléatoire, alors la fonction de compression (donc la fonction de hachage) est sûre. Les autres fonctions de hachage de la famille de MD4 (MD4, MD5 et les SHA) sont construites de la même façon, le point central du design est donc la construction du système de chiffrement. Comme les propriétés qu'on souhaite avoir pour cette utilisation ne sont pas les mêmes que pour faire du chiffrement, on utilise une fonction ad-hoc, qui est une sorte de schéma de Feistel généralisé.

Elle est conçue pour s'exécuter rapidement sur les processeurs 32 bits courants, et utilise donc des opérations qui sont implémentées en matériel sur les processeurs courants :

- L'addition modulo  $2^{32}$  que l'on note  $\boxplus$

<sup>1</sup>on exclut le cas où les messages font plus de  $2^{64}$  bits

- La rotation d'un mot de 32 bits. On note  $x \lll s$  le mot obtenu en effectuant une rotation de  $s$  bits au mot  $x$ . Si  $x = \sum_{i=0}^{31} e_i 2^i$ , avec  $e_i \in \{0, 1\}$ , on a  $x \lll s = \sum_{i=0}^{31} e_i 2^{i \boxplus s}$ , où l'opération  $\boxplus$  sur les indices représente l'addition modulo 32. De même, on note  $x \ggg s = x \lll (32 - s)$ .
- Des fonctions booléennes bit-à-bit  $\Phi_i$ . Ces fonctions sont :
  - $\text{MUX}(x, y, z) = (x \wedge y) \vee (\neg x \wedge z)$ , qui sélectionne  $y$  ou  $z$  selon la valeur de  $x$ .
  - $\text{MAJ}(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$ , la fonction de majorité
  - $\text{XOR}(x, y, z) = x \oplus y \oplus z$ , le où exclusif

Les bits de sortie sont aléatoire et indépendants si les bits d'entrées le sont. Dans les autres fonctions de hachage de la famille de MD4, on utilise aussi  $y \oplus (x \vee \neg z)$ .

La sécurité de la fonction de hachage vient du mélange entre ces différentes opérations incompatibles.

Quand on a besoin d'un représentant d'une classe d'équivalence modulo  $2^{32}$ , on le choisira entre 0 et  $2^{32} - 1$ . Si  $x$  est un tel mot de 32 bits, on note  $x^{[j]}$  le  $j$ -ème bit de  $x$  avec  $0 \leq j < 32$ . Ainsi,  $x^{[j]} = (x \ggg j) \bmod 2$ .

Le message, qui sert de clef pour la fonction de chiffrement, subit une expansion pour avoir autant de mots de 32 bits disponibles que d'étapes du schéma de Feistel ; on notera  $m_i$  le mot utilisé pour la  $i$ -ème étape. Pour MD4, cette expansion est très simple, il s'agit juste de lire plusieurs fois les mots du bloc de message, dans des ordres différents.

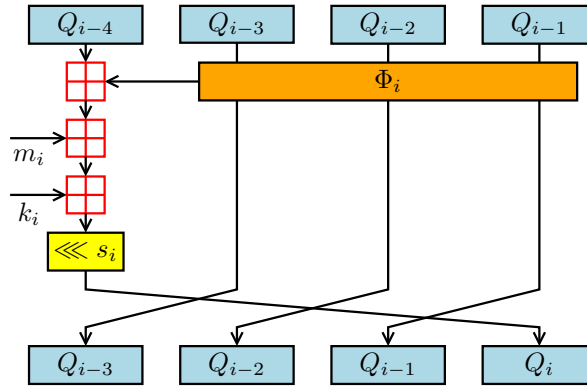


FIG. 3: Fonction de compression de MD4

La description originale de MD4 [Riv90] utilise 4 registres internes ( $A, B, C, D$ ) qui sont mis à jour un par un, mais pour décrire l'attaque de Wang on a besoin de considérer les différentes valeurs que prennent les registres au cours du calcul. Ainsi, pour décrire l'étape  $i$  de la fonction de compression ( $0 \leq i < 48$ ), on notera  $Q_i$  la nouvelle valeur du registre qui est mis à jour,  $m_i$  le mot du message utilisé,  $\Phi_i$  la fonction booléenne correspondante,  $s_i$  la valeur du décalage, et  $k_i$  la constante utilisée à cette étape. Une étape de calcul de la fonction de compression de MD4 s'écrit donc :

$$Q_i = (Q_{i-4} \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}) \boxplus m_i \boxplus k_i) \lll s_i.$$

La figure 3 illustre une telle étape de calcul.

Par rapport aux notations de Wang, on a

$$a_i = Q_{4i-4} \quad b_i = Q_{4i-1} \quad c_i = Q_{4i-2} \quad d_i = Q_{4i-3}$$

L'entrée de la fonction de compression est ainsi  $Q_{-4}||Q_{-1}||Q_{-2}||Q_{-3}$ , et la sortie  $Q_{-4} \boxplus Q_{44}||Q_{-1} \boxplus Q_{47}||Q_{-2} \boxplus Q_{46}||Q_{-3} \boxplus Q_{45}$ .

## 2 Description de l'attaque de Wang et al.

Les attaques de Wang sur les fonctions de compression de MD4, MD5, SHA-0 et SHA-1 sont des attaques de type différentiel. Le principe général d'une attaque différentielle sur une fonction  $F$  est de trouver deux opérations de groupe  $\diamond$  et  $\star$ , et deux différences  $\alpha$  (sur les entrées) et  $\beta$  (sur les sorties) telle que :  $F(x \diamond \alpha) = F(x) \star \beta$ , avec une bonne probabilité (sur  $x$ ). On appelle  $(\diamond, \star, \alpha, \beta)$  une différentielle, avec  $(\alpha, \beta) \neq (0, 0)$ . On choisit en général  $\alpha$  et  $\beta$  de poids faible, et les opérations de groupe sont soit le *ou exclusif* bit-à-bit  $\oplus$ , soit l'addition mot-à-mot modulo une puissance de deux  $\boxplus$ .

Quand on cherche une *collision* dans une fonction de compression, cela s'écrit :  $F(H, M) = F(H, M \diamond \alpha)$ , c'est-à-dire qu'on a  $\beta = 0$ , et la composante de  $\alpha$  sur  $H$  est nulle. Pour une *near-collision*, on cherche une différentielle  $F(H, M) = F(H, M \diamond_M \alpha) \star \beta$  avec  $\beta$  de poids faible, et pour une *pseudo-collision* on veut  $F(H, M) = F(H \diamond_H \alpha_H, M \diamond_M \alpha_M)$ , i.e.  $\beta$  est nulle, et les deux composantes de  $\alpha$  sont non-nulles. Ici, les probabilités seront donc sur  $(H, M)$ , et pour trouver des collisions on a besoin que la probabilité sur  $M$  qu'on obtient en fixant  $H$  (pour des collisions sur un bloc,  $H = IV$ ) reste bonne. On peut alors trouver un message  $M$  qui vérifie la différentielle. On utilisera des techniques plus intelligentes, mais l'idée de base est simplement de tester suffisamment de messages pour en trouver un qui marche.

Pour MD4 et SHA-0, ces attaques donnent des collisions dans la fonction de compression :

- pour MD4,  $H$  est complètement libre et on trouve donc des collisions dans la fonction de hachage avec des messages d'un bloc en prenant  $H = IV$ .
- pour SHA-0, on a 14 conditions sur  $H$ , on peut donc construire des collisions dans la fonction de hachage avec des messages de deux blocs, en prenant le premier bloc  $M_0$  identique (en testant  $O(2^{14})$  blocs, on en trouve un qui marche), et  $H = F(IV, M_0)$ .

Pour MD5 et SHA-1, on n'a pas de différentielle qui mène à une collision dans la fonction de compression, mais on a deux différentielles telles que :

$$\begin{aligned} F(H, M) &= F(H, M \boxplus \alpha) \boxplus \beta \\ F(H, M) &= F(H \boxplus \beta, M \boxplus \gamma). \end{aligned}$$

La première différentielle  $(\boxplus, \boxplus, (0, \alpha), \beta)$  donne des near-collisions, et la deuxième différentielle  $(\boxplus, \boxplus, (\beta, \gamma), 0)$  donne des pseudo-collisions qui utilisent la même différence  $\beta$  sur  $H$  que celle qu'on a en sortie de la première différentielle. Ainsi, on peut trouver des messages  $M_0$  et  $M_1$  qui satisfont chaque différentielle, et on a alors  $F(F(IV, M_0), M_1) = F(F(IV, M_0 \boxplus \alpha), M_1 \boxplus \gamma)$  : c'est une collision dans la fonction de hachage avec des messages de deux blocs.

En fait, il y a à chaque fois un bloc de padding en plus, mais c'est le même pour chaque paire de message car les messages ont la même taille ; il n'intervient donc pas dans l'attaque.

L'attaque de Wang peut se décomposer en deux grandes étapes qui seront décrites dans la suite :

1. Le choix des paramètres de l'attaque :
  - une différentielle  $\Delta$  sur le message
  - un chemin différentiel qui correspond à la différentielle
  - un jeu de conditions suffisantes pour ce chemin
2. La recherche d'un message qui remplit ces conditions, notamment grâce aux techniques de modification de message



Si on fixe  $\partial(x, y) = \langle \varepsilon_{31}, \varepsilon_{30}, \dots, \varepsilon_0 \rangle$ , alors  $\delta(x, y)$  est fixé, et il existe  $2^{\#\{i:\varepsilon_i=0\}}$  couples  $(x, y)$  possibles. En particulier, il en existe toujours au moins un.

Par contre, si on fixe  $\delta(x, y)$ , il existe en général plusieurs  $\partial(x, y) = \langle \varepsilon_{31}, \varepsilon_{30}, \dots, \varepsilon_0 \rangle$  possibles, la seule condition à remplir étant  $\sum_{j=0}^{31} \varepsilon_j 2^j = \delta(x, y)$ . Soit  $\langle \varepsilon_{31}, \varepsilon_{30}, \dots, \varepsilon_0 \rangle$  un  $\partial(x, y)$  particulier. On peut en construire des nouveaux en utilisant une *extension de retenue* sur un bit  $i_0$  tel que  $\varepsilon_{i_0} \neq 0$ . On note  $\varepsilon_+ = \sum_{j:\varepsilon_j=+1} \varepsilon_j 2^j$  et  $\varepsilon_- = \sum_{j:\varepsilon_j=-1} \varepsilon_j 2^j$ . On considère  $\varepsilon'_+ = \varepsilon_+ + 2^{i_0}$  et  $\varepsilon'_- = \varepsilon_- + 2^{i_0}$  (ce sont des éléments de  $\mathbb{Z}_{2^{32}}$ ), puis on construit

$$\varepsilon'_i = \begin{cases} 0 & \text{si } \varepsilon'_+ = \varepsilon'_- \\ 1 & \text{si } \varepsilon'_+ = 1 \text{ et } \varepsilon'_- = 0 \\ -1 & \text{si } \varepsilon'_+ = 0 \text{ et } \varepsilon'_- = 1 \end{cases}$$

On obtient ainsi une deuxième possibilité pour  $\partial(x, y)$ , puisque  $\varepsilon_{i_0} \neq \varepsilon'_{i_0}$  et

$$\sum_{j=0}^{31} \varepsilon'_j 2^j = \varepsilon'_+ - \varepsilon'_- = \varepsilon_+ - \varepsilon_- = \sum_{j=0}^{31} \varepsilon_j 2^j = \delta(x, y)$$

**Proposition 4.** Soit  $x, y, x', y' \in \mathbb{Z}_{2^{32}}$  avec  $\delta(x, y) = \delta(x', y')$ . Alors on peut passer de  $\partial(x, y) = \langle \varepsilon_{31}, \varepsilon_{30}, \dots, \varepsilon_0 \rangle$  à  $\partial(x', y') = \langle \varepsilon'_{31}, \varepsilon'_{30}, \dots, \varepsilon'_0 \rangle$  avec une ou plusieurs extension de retenue.

*Démonstration.* Soit  $i_0$  le plus petit indice tel que  $\varepsilon_{i_0} \neq \varepsilon'_{i_0}$ . En regardant modulo  $2^{i_0}$ , on voit qu'on a nécessairement  $\varepsilon_{i_0}$  et  $\varepsilon'_{i_0}$  non nuls. On peut donc appliquer une extension de retenue sur le bit  $i_0$  pour avoir  $\varepsilon_{i_0} \neq \varepsilon'_{i_0}$ , puis on répète cette opération autant que nécessaire.  $\square$

Ainsi, toutes les valeurs possibles pour  $\partial(x, y)$  peuvent être obtenues les unes à partir des autres en appliquant une ou plusieurs extension de retenues. En particulier, si  $\delta(x, y) = 2^k$ ,  $0 \leq k < 32$ , il y a  $33 - k$  valeurs possibles pour  $\partial(x, y)$  :

$$\begin{aligned} \langle \blacktriangle[k] \rangle &\rightarrow 2^k \\ \langle \blacktriangledown \blacktriangle[k, k+1] \rangle &\rightarrow 2^{k+1} - 2^k \\ \langle \blacktriangledown \blacktriangledown \blacktriangle[k, k+1, k+2] \rangle &\rightarrow 2^{k+2} - 2^{k+1} - 2^k \\ &\dots \\ \langle \blacktriangledown \dots \blacktriangledown \blacktriangle[k, k+1, \dots, 30, 31] \rangle &\rightarrow 2^{31} - 2^{30} - \dots - 2^k \\ \langle \blacktriangledown \dots \blacktriangledown \blacktriangledown \blacktriangle[k, k+1, \dots, 30, 31] \rangle &\rightarrow 2^{32} - 2^{31} - \dots - 2^k \end{aligned}$$

## 2.2 Conditions suffisantes

Dans l'attaque de Wang, on impose un chemin  $\partial$  en plus du chemin  $\delta$ , car cela va permettre de dériver un jeu de conditions suffisantes sur les  $Q_i$ . En effet, si on connaît les variations des entrées des fonctions  $\Phi_i$ , on peut assurer que la sortie variera comme prévu en ajoutant des conditions sur les entrées, comme le montre le tableau 4 page 19. Ces conditions vont fixer la valeur de certains bits de  $Q_i$  à une valeur constante, ou qui dépend des  $Q_i$  précédents. Il est inutile de poser des conditions sur les  $Q'_i$  car si un message suit le chemin différentiel, les bits qui diffèrent sont connus, et les autres sont identiques... La section 3.3 détaille comment calculer un jeu de conditions suffisantes associé à un chemin.

Pour trouver un message qui passe par le chemin différentiel choisi il suffit donc que les états internes remplissent certaines conditions. Cela nous permet d'avoir une bonne estimation de la probabilité qu'un couple  $(H, M)$  vérifie la différentielle : les conditions sur  $Q_{-4} \dots Q_{-1}$  sont des conditions sur  $H$  et les autres conditions dépendront de  $H$  et de  $M$ , mais on peut considérer que si  $H$  est fixé, un message aléatoire remplit une condition donnée avec une probabilité  $1/2$ . On fera aussi l'hypothèse que les conditions sont indépendantes.

## 2.3 Modifications de message

Si on prend d'un message aléatoire, la probabilité qu'il suive un chemin donné est faible, mais on peut modifier le message pour remplir certaines conditions de façon déterministe.

### 2.3.1 Modifications simples

Les conditions sur les étapes du premier tour ( $0 \leq i < 16$ ) sont très faciles à remplir car on a une liberté totale sur le message quand il est utilisé pour la première fois. Soit  $Q_i$  le résultat d'une étape de calcul du premier tour de la fonction de compression de MD4, et soit  $\tilde{Q}_i$  une valeur qui remplit les conditions souhaitées (il suffit de changer les bits qui ne conviennent pas). On va chercher un mot de message  $\tilde{m}_i$  qui permet d'obtenir  $\tilde{Q}_i$  quand on effectue le calcul :

$$\begin{aligned} Q_i &= (Q_{i-4} \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}) \boxplus m_i \boxplus k_i) \lll s_i \\ \tilde{Q}_i &= (Q_{i-4} \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}) \boxplus \tilde{m}_i \boxplus k_i) \lll s_i. \end{aligned}$$

On a  $(\tilde{Q}_i \ggg s_i) \boxminus (Q_i \ggg s_i) = \tilde{m}_i \boxminus m_i$ , il suffit donc de choisir

$$\tilde{m}_i = m_i \boxplus (\tilde{Q}_i \ggg s_i) \boxminus (Q_i \ggg s_i).$$

Dans la plupart des cas, un bit de différence sur  $Q_i$  donnera un bit de différence sur  $m_i$ , mais il faut prendre garde au fait que  $\boxplus$  et  $\lll$  ne commutent pas...

En fait, on peut aussi commencer par choisir un  $Q_i$  qui remplit les conditions, et en déduire  $m_i$  :

$$m_i = (Q_i \ggg s_i) \boxminus Q_{i-4} \boxminus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}) \boxminus k_i$$

On peut ainsi corriger facilement toutes les conditions sur le premier tour. Dans le cas de MD4, en utilisant le chemin différentiel de [WLF<sup>+</sup>05], il reste 27 conditions, on peut donc trouver des collisions en  $2^{27}$  calculs de hachés en implémentant seulement ces modifications simples.

### 2.3.2 Modifications multi-messages

Pour corriger les conditions du deuxième tour il faut assurer que les modifications qu'on applique au message ne modifient pas les conditions déjà vérifiées par les  $Q_i$ . Pour cela, il faut modifier plusieurs mots du message.

## 3 Recherche de chemin différentiel

Dans cette partie nous allons étudier la recherche de chemin différentiel. C'est une des parties les plus difficiles à automatiser de l'attaque. Une méthode est proposée dans [SO06], et le chemin trouvé est meilleur que celui de Wang [WLF<sup>+</sup>05] mais une partie du chemin (les *disturbance differences*) est encore trouvée à la main. Avec la méthode décrite ici, tout est automatisé, et le chemin trouvé est encore meilleur.

L'idée générale de la recherche de chemin est d'étudier comment les différences se propagent à travers la fonction de mise à jour de l'état interne :

$$Q_i = (Q_{i-4} \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}) \boxplus m_i \boxplus k_i) \lll s_i.$$

Comme on travaille avec une différentielle pour l'opération  $\boxplus$ , les opérations non-linéaires sont la rotation  $\lll$  et les fonctions booléennes  $\Phi_i$ .

### 3.1 Outils mathématiques

#### 3.1.1 Interactions entre $\delta$ et $\lll$

Pour construire l'attaque différentielle, on a besoin de savoir exprimer  $\delta(a \lll s, b \lll s)$  en fonction de  $\delta(a, b)$ , et éventuellement de  $a$ .

On note  $+$  l'addition dans  $\mathbb{Z}$ , et  $\boxplus$  l'addition mod  $2^{32}$ . On peut alors exprimer algébriquement  $x \lll s$  : c'est la somme :

1. des bits de poids faible de  $x$ , décalés vers la gauche pour devenir les bits de poids fort  $x \lll s$
2. des bits de poids fort de  $x$ , décalés vers la droite, pour devenir les bits de poids faible de  $x \lll s$

$$x \lll s = \underbrace{(2^s x \bmod 2^{32})}_1 + \underbrace{\left\lfloor \frac{x \bmod 2^{32}}{2^{32-s}} \right\rfloor}_2.$$

On a les inégalités suivantes avec les parties entières :

$$\begin{aligned} x - 1 &< \lfloor x \rfloor \leq x < \lfloor x \rfloor + 1 \\ \lfloor x \rfloor + \lfloor y \rfloor &\leq \lfloor x + y \rfloor \leq \lfloor x \rfloor + \lfloor y \rfloor + 1 \\ \lfloor x + y \rfloor &= \begin{cases} \lfloor x \rfloor + \lfloor y \rfloor & \text{si } x + y < \lfloor x \rfloor + \lfloor y \rfloor + 1 \\ \lfloor x \rfloor + \lfloor y \rfloor + 1 & \text{si } x + y \geq \lfloor x \rfloor + \lfloor y \rfloor + 1 \end{cases} \end{aligned}$$

Soit  $0 \leq a < 2^{32}$  et  $0 \leq u < 2^{32}$ . On note  $\alpha = a \lll s$ ,  $b = a + u$ , et  $\beta = (b \bmod 2^{32}) \lll s$ . On veut exprimer  $v = \beta \boxplus \alpha$  en fonction de  $u = b \boxplus a$ .

Si  $b < 2^{32}$ , on a :

$$\begin{aligned} \beta - \alpha &= \left( \left\lfloor \frac{a+u}{2^{32-s}} \right\rfloor + 2^s(a+u) \right) - \left( \left\lfloor \frac{a}{2^{32-s}} \right\rfloor + 2^s a \right) \\ &= \left\lfloor \frac{a+u}{2^{32-s}} \right\rfloor - \left\lfloor \frac{a}{2^{32-s}} \right\rfloor + 2^s u \\ &= \left\lfloor \frac{u}{2^{32-s}} \right\rfloor + 2^s u \quad \text{ou} \quad \left\lfloor \frac{u}{2^{32-s}} \right\rfloor + 2^s u + 1 \\ \beta \boxplus \alpha &= u \lll s \quad \text{ou} \quad (u \lll s) \boxplus 1 \end{aligned}$$

Si non,  $2^{32} \leq b < 2^{33}$ , et dans ce cas :

$$\begin{aligned} \beta - \alpha &= \left( \left\lfloor \frac{a+u-2^{32}}{2^{32-s}} \right\rfloor + 2^s(a+u) \right) - \left( \left\lfloor \frac{a}{2^{32-s}} \right\rfloor + 2^s a \right) \\ &= -2^s + \left( \left\lfloor \frac{a+u}{2^{32-s}} \right\rfloor + 2^s(a+u) \right) - \left( \left\lfloor \frac{a}{2^{32-s}} \right\rfloor + 2^s a \right) \\ \beta \boxplus \alpha &= (u \lll s) \boxplus 2^s \quad \text{ou} \quad (u \lll s) \boxplus 2^s \boxplus 1 \end{aligned}$$

De plus, on peut exprimer précisément  $v = \delta(\alpha, \beta)$  en fonction de  $u = \delta(a, b)$  et  $a$ , ce qui permet, pour un  $u$  donné, de compter le nombre de  $a$  qui donnent chaque  $v$ , et d'exprimer des

conditions suffisantes sur les bits de  $a$  pour être sûr d'obtenir un certain  $v$  :

$$v = \begin{cases} v_1 = (u \lll s) & \text{si } a + u < 2^{32} \text{ et} \\ & (a \bmod 2^{32-s}) + (u \bmod 2^{32-s}) < 2^{32-s} \\ v_2 = (u \lll s) \boxplus 1 & \text{si } a + u < 2^{32} \text{ et} \\ & (a \bmod 2^{32-s}) + (u \bmod 2^{32-s}) \geq 2^{32-s} \\ v_3 = (u \lll s) \boxminus 2^s & \text{si } a + u \geq 2^{32} \text{ et} \\ & (a \bmod 2^{32-s}) + (u \bmod 2^{32-s}) < 2^{32-s} \\ v_4 = (u \lll s) \boxminus 2^s \boxplus 1 & \text{si } a + u \geq 2^{32} \text{ et} \\ & (a \bmod 2^{32-s}) + (u \bmod 2^{32-s}) \geq 2^{32-s} \end{cases}$$

Si  $u$  est fixé, on peut calculer la probabilité d'obtenir chacun des  $v_i$  pour  $a$  choisi aléatoirement (de façon uniforme). En première approximation, on a :

$$\begin{aligned} p_1 &\approx \frac{2^{32} - u}{2^{32}} \frac{2^{32-s} - (u \bmod 2^{32-s})}{2^{32-s}} & p_2 &\approx \frac{2^{32} - u}{2^{32}} \frac{(u \bmod 2^{32-s})}{2^{32-s}} \\ p_3 &\approx \frac{u}{2^{32}} \frac{2^{32-s} - (u \bmod 2^{32-s})}{2^{32-s}} & p_4 &\approx \frac{u}{2^{32}} \frac{(u \bmod 2^{32-s})}{2^{32-s}} \end{aligned}$$

Ainsi, si on moyenne sur tous les  $u \in \mathbb{Z}_{2^{32}}$ , on a  $p_1 = p_2 = p_3 = p_4$ . Mais si on ne considère que les  $u$  de poids de Hamming faible, qui sont ceux qui interviennent majoritairement dans la suite, alors le premier cas est largement majoritaire. Plus précisément, si  $u = 2^j$  on a (les  $p$  correspondent au cas où  $j < 32 - s$ , et les  $p'$  au cas contraire)

$$\begin{array}{lll} v_1 = 2^{j \boxplus s} & p_1 \approx (1 - 2^{j-32})(1 - 2^{j+s-32}) & p'_1 \approx (1 - 2^{j-32}) \\ v_2 = 2^{j \boxplus s} \boxplus 1 & p_2 \approx (1 - 2^{j-32})(2^{j+s-32}) & p'_2 = 0 \\ v_3 = 2^{j \boxplus s} \boxminus 2^s & p_3 \approx (2^{j-32})(1 - 2^{j+s-32}) & p'_3 \approx (2^{j-32}) \\ v_4 = 2^{j \boxplus s} \boxminus 2^s \boxplus 1 & p_4 \approx (2^{j-32})(2^{j+s-32}) & p'_4 = 0 \end{array}$$

### 3.1.2 Interactions entre $\boxplus$ et $\lll$

Pour la recherche de chemin, on va avoir besoin d'exprimer une valeur possible de  $\partial(Q'_i \lll s, Q_i \lll s)$  en fonction de  $\partial(Q'_i, Q_i)$ . On peut faire ce calcul en utilisant le résultat précédent : il suffit de calculer  $u = \delta(Q_i, Q'_i)$  à partir de  $\partial(Q'_i, Q_i)$ , puis de choisir une représentation en  $\partial$  de  $v = \delta(Q_i \lll s, Q'_i \lll s)$ . Cependant, on obtient des meilleurs résultats en essayant de conserver l'information supplémentaire contenue dans  $\partial(Q'_i, Q_i)$ . On va donc essayer de construire une représentation de  $u \lll s$  à partir d'une représentation de  $u$ , puis utiliser cette représentation pour construire  $v$ .

Soit  $u = \sum_{j=0}^{31} \varepsilon_j 2^j \bmod 2^{32}$ , où  $\varepsilon_j \in \{-1, 0, +1\}$ . On note avec  $\boxplus$  l'addition modulo 32 sur les indices.

Si  $\sum_{j=0}^{31} \varepsilon_j 2^j \geq 0$ , on a :

$$\begin{aligned}
u \lll s &= \left\lfloor \frac{\sum_{j=0}^{31} \varepsilon_j 2^j}{2^{32-s}} \right\rfloor + \left( 2^s \sum_{j=0}^{31} \varepsilon_j 2^j \bmod 2^{32} \right) \\
&= \sum_{j=32-s}^{31} \varepsilon_j 2^{j-(32-s)} + \left\lfloor \frac{\sum_{j=0}^{31-s} \varepsilon_j 2^j}{2^{32-s}} \right\rfloor + \sum_{j=0}^{31-s} \varepsilon_j 2^{j+s} \\
&= \sum_{j=0}^{31} \varepsilon_j 2^{j \boxplus s} + \underbrace{\left\lfloor \frac{\sum_{j=0}^{31-s} \varepsilon_j 2^j}{2^{32-s}} \right\rfloor}_{0 \text{ ou } -1} \\
&= \sum_{j=0}^{31} \varepsilon_j 2^{j \boxplus s} \quad \text{ou} \quad \sum_{j=0}^{31} \varepsilon_j 2^{j \boxplus s} \boxplus 1
\end{aligned}$$

Si  $\sum_{j=0}^{31} \varepsilon_j 2^j < 0$ , on a :

$$\begin{aligned}
u \lll s &= \left\lfloor \frac{2^{32} + \sum_{j=0}^{31} \varepsilon_j 2^j}{2^{32-s}} \right\rfloor + \left( 2^s \sum_{j=0}^{31} \varepsilon_j 2^j \bmod 2^{32} \right) \\
&= 2^s + \left\lfloor \frac{\sum_{j=0}^{31} \varepsilon_j 2^j}{2^{32-s}} \right\rfloor + \left( 2^s \sum_{j=0}^{31} \varepsilon_j 2^j \bmod 2^{32} \right) \\
&= 2^s + \sum_{j=0}^{31} \varepsilon_j 2^{j \boxplus s} + \underbrace{\left\lfloor \frac{\sum_{j=0}^{31-s} \varepsilon_j 2^j}{2^{32-s}} \right\rfloor}_{0 \text{ ou } -1} \\
&= \sum_{j=0}^{31} \varepsilon_j 2^{j \boxplus s} \boxplus 2^s \quad \text{ou} \quad \sum_{j=0}^{31} \varepsilon_j 2^{j \boxplus s} \boxplus 2^s \boxplus 1
\end{aligned}$$

Ceci nous permet d'avoir une expression simple de  $(-2^j) \lll s$ , et donc de calculer  $v = \delta(a \lll s, b \lll s)$  en fonction de  $u = \delta(a, b)$  dans le cas où  $u = -2^j$  (les  $v$  et  $p$  correspondent au cas où  $j < 32 - s$ , et les  $v'$  et  $p'$  au cas contraire) :

$$\begin{array}{llll}
v_1 = -2^{j \boxplus s} \boxplus 2^s \boxplus 1 & p_1 \approx (2^{j-32})(2^{j+s-32}) & v'_1 = -2^{j \boxplus s} \boxplus 2^s & p'_1 \approx (2^{j-32}) \\
v_2 = -2^{j \boxplus s} \boxplus 2^s & p_2 \approx (2^{j-32})(1 - 2^{j+s-32}) & v'_2 = -2^{j \boxplus s} \boxplus 2^s \boxplus 1 & p'_2 = 0 \\
v_3 = -2^{j \boxplus s} \boxplus 1 & p_3 \approx (1 - 2^{j-32})(2^{j+s-32}) & v'_3 = -2^{j \boxplus s} & p'_3 \approx (1 - 2^{j-32}) \\
v_4 = -2^{j \boxplus s} & p_4 \approx (1 - 2^{j-32})(1 - 2^{j+s-32}) & v'_4 = -2^{j \boxplus s} \boxplus 1 & p'_4 = 0
\end{array}$$

Ici aussi, le cas majoritaire est celui où la différence semble commuter avec la rotation. Comme les différences qu'on manipulera dans la suite sont creuses, on négligera parfois les interactions entre ces deux opérations pour simplifier les explications, mais l'algorithme devra toujours tenir compte des différents cas possibles.

### 3.2 Choix de la différentielle $\Delta$ sur les messages

La première partie de l'attaque consiste à choisir une différentielle sur les messages. Ce choix se fait essentiellement à la main, en examinant les propriétés des fonctions  $\Phi_i$  utilisées, résumées dans le tableau 4 page 19.

On voit que les fonctions MUX et MAJ permettent d'absorber une différence sur leurs entrées : si une seule entrée varie, on peut toujours ajouter une condition sur les autres qui assure que la sortie ne variera pas. Ainsi, si on a  $\partial Q_{i-4} \neq 0$ , on peut garder  $\partial Q_{i-3} = \partial Q_{i-2} = \partial Q_{i-1} = 0$ , et la différence réapparaîtra juste dans  $\partial Q_i$  par  $Q_i = (Q_{i-4} \boxplus \Phi_i \boxplus m_i \boxplus k_i) \lll s_i$ . On peut donc limiter fortement la propagation des erreurs dans les tours qui utilisent ces fonctions : en ajoutant quelques conditions sur les  $Q_i$ , on assure qu'une erreur n'en génère pas d'autres.

Au contraire, avec la fonction  $\oplus$ , si une seule entrée varie, alors la sortie varie forcément, et cette nouvelle différence va se propager en plus de l'ancienne. De plus, cette fonction est utilisée dans le dernier tour, et il est très difficile de trouver des modifications de messages pour remplir les conditions qu'on a sur ce tour. On va donc choisir des différentielles qui donnent des chemins qui se comportent bien pour le troisième tour, et ensuite essayer de compléter ce chemin pour les premiers tours.

*Remarque 1.* Une étape de la fonction de compression avec un message fixé est inversible (c'est une étape d'une fonction de chiffrement).

*Corollaire 1.* Ainsi, les fonctions MUX et MAJ permettent une diffusion minimale de l'erreur : si on introduit un bit de différence dans l'état interne à travers  $m_i$ , il ne peut pas disparaître sans introduire de nouvelle différence dans  $m_i$  mais il reste un seul bit de différence après plusieurs étapes.

*Corollaire 2.* Soit  $i_0 = \min\{i : \delta m_i \neq 0\}$  et  $i_1 = \max\{i : \delta m_i \neq 0\}$ . Si on a un chemin différentiel qui donne des collisions dans la fonction de compression ( $\partial_{-4} = \partial_{-3} = \partial_{-2} = \partial_{-1} = 0$  et en sortie  $\partial_{44} = \partial_{45} = \partial_{46} = \partial_{47} = 0$ ), alors  $\partial_i = 0$  pour  $i \notin \llbracket i_0, i_1 \rrbracket$ .

#### 3.2.1 La différentielle de [YWZW05]

Une première façon de faire est de choisir une différentielle sur le message qui n'introduise des erreurs qu'au tout début du dernier tour, pour qu'elle soit corrigée dans les étapes des deux premiers tours qui utilisent des fonctions  $\Phi_i$  plus faciles à contrôler.

L'idée la plus naturelle est d'avoir un seul bit de différence entre  $M$  et  $M'$  : c'est ce qui est fait dans [YWZW05]. On trouve des chemins qui marchent en introduisant la différence dans  $M_4$ , et les  $\partial_i$  du dernier tour sont tous nuls. Le tableau 1 détaille le dernier tour : si on introduit une différence sur le bit  $k$ , on a  $\partial Q_{29} = \langle \blacktriangle^{[k]} \rangle$  et  $\partial Q_{30} = \langle \blacktriangledown^{[k]} \rangle$  :

- pour l'étape 32, ces deux différences s'annulent dans le XOR, d'où  $\partial Q_{32} = 0$
- pour l'étape 33, on a  $\Phi_{33}^{[k]} \neq \Phi_{33}^{[k]}$ , mais cela se compense avec la différence sur  $Q_{29}$ , à condition que le signe de  $\partial \Phi^{[k]}$  soit correct :  $Q_{33} = (Q_{29} \boxplus (Q_{30} \oplus Q_{31} \oplus Q_{32})) \boxplus m_{33} \boxplus k_{33} \lll s_{33}$ .
- pour l'étape 34, on a plus de différence dans  $\Phi_i$ , et la différence sur  $m_{34}$  annule la différence sur  $Q_{30}$  :  $Q_{30} \boxplus m_{34} = Q'_{30} \boxplus m'_{34}$

Il ne reste plus qu'à trouver un chemin pour les deux premiers tours qui utilise cette différentielle sur les messages. Le chemin donné dans [YWZW05] utilise  $k = 22$ , et il est reproduit page 38. Ce chemin reste valide pour de nombreuses valeurs de  $k$ , et en prenant  $k = 25$  on économise même 4 conditions.

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
29	5			$\langle \blacktriangle^{[k]} \rangle$	
30	9			$\langle \blacktriangledown^{[k]} \rangle$	
31	13				
32	3		0		
33	9		$\langle \blacktriangledown^{[k]} \rangle$		$Q_{32}^{[k]} = Q_{31}^{[k]}$
34	11	$\langle \blacktriangle^{[k]} \rangle$	0		

TAB. 1: Le dernier tour du chemin de [YWZW05].

### 3.2.2 La différentielle de [WLF<sup>+</sup>05]

Une autre façon de choisir une différentielle sur les messages est d'assurer qu'on a une collision locale à l'intérieur du dernier tour avec une bonne probabilité. Comme on se concentre sur un seul tour, on a une liberté totale sur la différentielle, et on peut donc facilement construire une telle collision locale, comme détaillé dans le tableau 2, où les  $\epsilon_i$  valent au choix  $+1$  ou  $-1$ .

step	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
$i$	$\langle \epsilon_1^{[k \boxminus s_i]} \rangle$		$\langle \epsilon_1^{[k]} \rangle$	
$i+1$	$\langle \epsilon_2^{[k \boxminus s_{i+1}]}, \epsilon_3^{[k]} \rangle$	$\langle -\epsilon_3^{[k]} \rangle$	$\langle \epsilon_2^{[k]} \rangle$	$\begin{cases} Q_{i-2}^{[k]} = Q_{i-1}^{[k]} & \text{si } \epsilon_1 \neq \epsilon_3 \\ Q_{i-2}^{[k]} \neq Q_{i-1}^{[k]} & \text{si } \epsilon_1 = \epsilon_3 \end{cases}$
$i+2$				
$i+3$				
$i+4$		$\langle -\epsilon_1^{[k]} \rangle$		$\begin{cases} Q_{i+2}^{[k]} = Q_{i+3}^{[k]} & \text{si } \epsilon_1 \neq \epsilon_2 \\ Q_{i+2}^{[k]} \neq Q_{i+3}^{[k]} & \text{si } \epsilon_1 = \epsilon_2 \end{cases}$
$i+5$	$\langle -\epsilon_2^{[k]} \rangle$			

TAB. 2: Une collision locale dans le dernier tour

Si on introduit une différence avec  $\delta_i = \epsilon_1 2^{k \boxminus s_i}$  ( $32 \leq i \leq 48$ ), on aura  $\partial_i = \langle \epsilon_1^{[k]} \rangle$ . Si on prend  $\delta_{i+1} = \epsilon_1 2^{k \boxminus s_{i+1}} + \epsilon_2 2^k$ , on a  $\partial\Phi_{i+1} = \pm 2^k$  dont on peut choisir le signe en mettant la bonne condition sur  $Q_{i-2}^{[k]}$  et  $Q_{i-1}^{[k]}$ , et qui va s'annuler avec le  $\epsilon_2 2^k$ ; on obtient  $\partial_{i+1} = \langle \epsilon_1^{[k]} \rangle$ . Pour les étapes  $i+2$  et  $i+3$ , on a alors  $\partial\Phi = 0$ . À l'étape  $i+4$ , on peut aussi choisir le signe de  $\partial\Phi_{i+4} = \pm 2^k$  pour annuler  $\epsilon_1 2^k$ . Enfin, à l'étape  $i+5$ , en prenant  $\delta m = -\epsilon_2 2^k$ , on a  $Q_{i+1} \boxplus m_{i+5} = Q'_{i+1} \boxplus m'_{i+5}$ , d'où la collision.

Ici aussi, il ne reste plus qu'à compléter ce chemin dans les deux premiers tours.

Pour affiner le choix, on peut prendre  $k = 31$ , ce qui évite les deux conditions pour choisir le signe de  $\partial\Phi$ . On a aussi intérêt à choisir  $i$  de sorte que  $m_i$ ,  $m_{i+1}$  et  $m_{i+5}$  soient utilisés le plus tôt possible dans le deuxième tour : cela a des chances de minimiser le nombre de conditions sur le second tour, et même si le nombre de conditions dans le premier tour augmente, ce n'est pas très grave puisqu'on sait très facilement les satisfaire. Le choix optimal pour ce critère est  $i = 35$ . C'est ainsi qu'est construit le chemin de [WLF<sup>+</sup>05]; la collision dans le dernier tour est détaillée dans le tableau 3 et le chemin complet est page 30. Cependant, ce n'est pas le seul choix possible, le chemin 4 page 36 montre un exemple avec  $i = 38$ .

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
35	15	$\langle \blacktriangledown^{[16]} \rangle$		$\langle \blacktriangledown^{[31]} \rangle$	
36	3	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$	$\langle \blacktriangledown^{[31]} \rangle$	$\langle \blacktriangledown^{[31]} \rangle$	
37	9				
38	11				
39	15		$\langle \blacktriangle^{[31]} \rangle$		
40	3	$\langle \blacktriangle^{[31]} \rangle$			

TAB. 3: Le dernier tour du chemin de [WLF<sup>+</sup>05]

### 3.3 Calcul des conditions suffisantes

Pour tenter d'automatiser l'attaque, on va commencer par calculer les conditions suffisantes à partir du chemin différentiel. On suppose donc connu la différentielle sur le message  $\Delta$  et le chemin différentiel  $\partial$ , et on cherche des conditions sur les  $Q_i$  qui, si elles sont remplies pour un message  $M$ , vont assurer que le chemin différentiel  $\partial$  sera suivi par les messages  $M$  et  $M' = M \boxplus \Delta$ , c'est à dire qu'on aura effectivement  $\partial(Q_i, Q'_i) = \partial_i$ . Ainsi, si le chemin est bien choisi (i.e. en entrée  $\partial_{-4} = \partial_{-3} = \partial_{-2} = \partial_{-1} = 0$  et en sortie  $\partial_{44} = \partial_{45} = \partial_{46} = \partial_{47} = 0$ ), un message qui remplit les conditions fournira une collision, et au contraire, si le chemin est impossible, on ne trouvera pas de jeu de conditions.

Les conditions seront calculées en étudiant chaque étape du calcul de MD4, en commençant par la dernière étape et en remontant vers le début : si on suppose que les conditions déjà calculées impliquent  $\partial(Q_k, Q'_k) = \partial_k$  pour  $k > i$ , l'étape  $i + 4$  s'écrit :

$$\begin{aligned} Q_{i+4} &= (Q_i \boxplus \Phi_{i+4}(Q_{i+1}, Q_{i+2}, Q_{i+3}) \boxplus m_{i+4} \boxplus k_{i+4}) \lll s_{i+4} \\ Q'_{i+4} &= (Q'_i \boxplus \Phi_{i+4}(Q'_{i+1}, Q'_{i+2}, Q'_{i+3}) \boxplus m'_{i+4} \boxplus k_{i+4}) \lll s_{i+4} \end{aligned}$$

#### 3.3.1 Fixer le bon $\partial_i$

Pour assurer que  $\partial(Q_i, Q'_i) = \partial_i$ , on va commencer par garantir  $\delta(Q_i, Q'_i) = \delta_i$ . On a vu en section 3.1.1 qu'on sait calculer à partir de  $\delta_{i+4}$  une valeur qu'on notera  $\delta_{i+4}^{\ggg}$  et des conditions suffisantes sur  $Q_{i+4}$  pour que, si ces conditions sont remplies et  $\delta(Q_{i+4}, Q'_{i+4}) = \delta_{i+4}$ , alors  $\delta(Q_{i+4} \ggg s_{i+4}, Q'_{i+4} \ggg s_{i+4}) = \delta_{i+4}^{\ggg}$ .

Or,

$$\begin{aligned} \delta_{i+4}^{\ggg} &= Q'_{i+4} \ggg s_{i+4} \boxplus Q_{i+4} \ggg s_{i+4} \\ &= (Q'_i \boxplus Q_i) \boxplus (\Phi'_{i+4} \boxplus \Phi_{i+4}) \boxplus (m_{i+4} \boxplus m'_{i+4}) \boxplus (k_{i+4} \boxplus k_{i+4}) \\ &= (Q'_i \boxplus Q_i) \boxplus (\Phi'_{i+4} \boxplus \Phi_{i+4}) \boxplus \Delta_{i+4} \\ (Q'_i \boxplus Q_i) &= \delta_{i+4}^{\ggg} \boxplus \Delta_{i+4} \boxplus \Phi_{i+4} \boxplus \Phi'_{i+4} \end{aligned}$$

Il suffit donc d'assurer que  $\Phi'_{i+4} \boxplus \Phi_{i+4} = \delta_i \boxplus \delta_{i+4}^{\ggg} \boxplus \Delta_{i+4}$  (la valeur du membre droit est connue à cette étape de l'algorithme). Pour cela, on choisit une différentielle sur les bits  $\partial(\Phi_{i+4}, \Phi'_{i+4})$  qui correspond au  $\delta(\Phi_{i+4}, \Phi'_{i+4})$  souhaité, puis pour chaque bit on connaît les variations des entrées et la variation souhaitée des sorties, ce qui permet de repérer les cas impossibles ou d'imposer des conditions suffisantes sur  $Q_{i+1}, Q_{i+2}, Q_{i+3}$  (voir table 4 page ci-contre). Le jeu de conditions ainsi construit assure que  $\delta(Q_i, Q'_i) = \delta_i$ .

			$F(x, y, z) = \text{MUX}(x, y, z)$			$G(x, y, z) = \text{MAJ}(x, y, z)$			$H(x, y, z) = x \oplus y \oplus z$			$I(x, y, z) = y \oplus (x \vee \neg z)$		
$\partial x$	$\partial y$	$\partial z$	$\partial F = 0$	$\partial F = 1$	$\partial F = -1$	$\partial G = 0$	$\partial G = 1$	$\partial G = -1$	$\partial H = 0$	$\partial H = 1$	$\partial H = -1$	$\partial I = 0$	$\partial I = 1$	$\partial I = -1$
0	0	0	✓	✗	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗
0	0	+1	$x = 1$	$x = 0$	✗	$x = y$	$x \neq y$	✗	✗	$x = y$	$x \neq y$	$x = 1$	$x, y = 0, 1$	$x, y = 0, 0$
0	0	-1	$x = 1$	✗	$x = 0$	$x = y$	✗	$x \neq y$	✗	$x \neq y$	$x = y$	$x = 1$	$x, y = 0, 0$	$x, y = 0, 1$
0	+1	0	$x = 0$	$x = 1$	✗	$x = z$	$x \neq z$	✗	✗	$x = z$	$x \neq z$	✗	$x, z = 0, 1$	$x, y \neq 1, 0$
0	-1	0	$x = 0$	✗	$x = 1$	$x = z$	✗	$x \neq z$	✗	$x \neq z$	$x = z$	✗	$x, y \neq 1, 0$	$x, z = 0, 1$
+1	0	0	$y = z$	$y, z = 1, 0$	$y, z = 0, 1$	$y = z$	$y \neq z$	✗	✗	$y = z$	$y \neq z$	$z = 0$	$y, z = 0, 1$	$y, z = 1, 1$
-1	0	0	$y = z$	$y, z = 0, 1$	$y, z = 1, 0$	$y = z$	✗	$y \neq z$	✗	$y \neq z$	$y = z$	$z = 0$	$y, z = 1, 1$	$y, z = 0, 1$
0	+1	+1	✗	✓	✗	✗	✓	✗	✓	✗	✗	$x = 0$	✗	$x = 1$
0	-1	+1	✗	$x = 0$	$x = 1$	✓	✗	✗	✓	✗	✗	$x = 0$	$x = 1$	✗
0	+1	-1	✗	$x = 1$	$x = 0$	✓	✗	✗	✓	✗	✗	$x = 0$	✗	$x = 1$
0	-1	-1	✗	✗	✓	✗	✗	✓	✓	✗	✗	$x = 0$	$x = 1$	✗
+1	0	+1	$y = 0$	$y = 1$	✗	✗	✓	✗	✓	✗	✗	✓	✗	✗
-1	0	+1	$y = 1$	$y = 0$	✗	✓	✗	✗	✓	✗	✗	✗	$y = 1$	$y = 0$
+1	0	-1	$y = 1$	✗	$y = 0$	✓	✗	✗	✓	✗	✗	✗	$y = 0$	$y = 1$
-1	0	-1	$y = 0$	✗	$y = 1$	✗	✗	✓	✓	✗	✗	✓	✗	✗
+1	+1	0	$z = 1$	$z = 0$	✗	✗	✓	✗	✓	✗	✗	$z = 1$	✗	$z = 0$
-1	+1	0	$z = 0$	$z = 1$	✗	✓	✗	✗	✓	✗	✗	$z = 1$	✗	$z = 0$
+1	-1	0	$z = 0$	✗	$z = 1$	✓	✗	✗	✓	✗	✗	$z = 1$	$z = 0$	✗
-1	-1	0	$z = 1$	✗	$z = 0$	✗	✗	✓	✓	✗	✗	$z = 1$	$z = 0$	✗
+1	+1	+1	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗	✗	✓
-1	+1	+1	✗	✓	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗
+1	-1	+1	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✓	✗
-1	-1	+1	✓	✗	✗	✗	✗	✓	✗	✓	✗	✓	✗	✗
+1	+1	-1	✓	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗
-1	+1	-1	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✓
+1	-1	-1	✗	✗	✓	✗	✗	✓	✗	✓	✗	✓	✗	✗
-1	-1	-1	✗	✗	✓	✗	✗	✓	✗	✗	✓	✗	✓	✗

TAB. 4: Conditions sur les fonctions  $\Phi_i$ .

Les chemins donnés en appendice donnent des exemples de jeu de conditions suffisantes, et permettent de bien comprendre comment on les obtient. Par exemple, le début du chemin 5 est reproduit dans le tableau 5. On voit qu'aux étapes 5 à 7, 9, et 11 à 14, on a  $\delta\Phi_i = 0$ , et on a des conditions qui le garantissent. À l'étape 8, les entrées de  $\Phi$  ne varient pas, il n'y a donc pas besoin de conditions. À l'étape 10, on a besoin de  $\delta\Phi_{10} = -2^{28}$ . On choisit de le représenter avec  $\partial\Phi_{10} = \langle \blacktriangledown^{[28]} \rangle$ , et on a des conditions suffisantes pour le garantir. On peut remarquer qu'on a ici besoin du signe de  $\partial_8^{[28]}$ , et le chemin utilise une retenue pour avoir  $\delta_8 = 2^{28}$  et  $\partial_8^{[28]} = -1$ .

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
4	3	$\langle \blacktriangle^{[22]} \rangle$		$\langle \blacktriangle^{[25]} \rangle$	
5	7				$Q_3^{[25]} = Q_2^{[25]}$
6	11				$Q_5^{[25]} = 0$
7	19				$Q_6^{[25]} = 1$
8	3			$\langle \blacktriangledown^{[28,29]} \rangle$	
9	7				$Q_7^{[28]} = Q_6^{[28]}, Q_7^{[29]} = Q_6^{[29]}$
10	11		$\langle \blacktriangledown^{[28]} \rangle$	$\langle \blacktriangledown^{[7]} \rangle$	$Q_9^{[28]} = 1, Q_9^{[29]} = 0$
11	19				$Q_9^{[7]} = Q_8^{[7]}, Q_{10}^{[28]} = 1, Q_{10}^{[29]} = 1$
12	3			$\langle \blacktriangle^{[31]} \rangle$	$Q_{11}^{[7]} = 0$
13	7				$Q_{12}^{[7]} = 1, Q_{11}^{[31]} = Q_{10}^{[31]}$
14	11			$\langle \blacktriangledown^{[18]} \rangle$	$Q_{13}^{[31]} = 0$

TAB. 5: Un exemple de morceau de chemin, avec ses conditions suffisantes

Une fois qu'on a  $\delta(Q_i, Q'_i) = \delta_i$ , il suffit d'ajouter quelques conditions sur  $Q_i$  en suivant la proposition 3 pour avoir  $\partial(Q_i, Q'_i) = \partial_i$ .

### 3.3.2 Choix à effectuer

Au final, pour obtenir notre jeux de conditions suffisantes on a deux choix à effectuer :

**On doit choisir un  $\delta_{i+4}^{\ggg}$**  qui correspond à  $\delta_{i+4}$ . Il y a ici au maximum quatre choix possibles, mais la forme creuse de la différentielle fait qu'un de ces choix sera beaucoup plus probable que les autres (voir section 3.1.1), et les mauvais choix donnent souvent des incompatibilités au bout de quelques étapes.

De plus, au moment où on fait ce choix, on a déjà des conditions sur  $Q_{i+4}$  qui limitent encore plus le nombre de choix. Par exemple, si  $\partial_{i+4} = \langle \blacktriangle^{[k]} \rangle$ , alors  $\delta_{i+4} = 2^k$ , et on a la condition  $Q_{i+4}^{[k]} = 0$ , ce qui assure que  $Q_{i+4} + \delta_{i+4} < 2^{32}$  et  $(Q_{i+4} \bmod 2^{32-s_{i+4}}) + (\delta_{i+4} \bmod 2^{32-s_{i+4}}) < 2^{32-s_{i+4}}$  (dans  $\mathbb{Z}$ ); ainsi on a nécessairement  $\delta_{i+4}^{\ggg} = \delta_{i+4} \lll s_{i+4} = 2^{k \boxplus s_{i+4}}$ . De même, avec  $\partial_{i+4} = \langle \blacktriangledown^{[k]} \rangle$ , on a  $\delta_{i+4} = 2^{32} - 2^k$ , et la condition  $Q_{i+4}^{[k]} = 1$ . Alors  $Q_{i+4} + \delta_{i+4} \geq 2^{32}$ , et  $(Q_{i+4} \bmod 2^{32-s_{i+4}}) + (\delta_{i+4} \bmod 2^{32-s_{i+4}}) \geq 2^{32-s_{i+4}}$ , ce qui ne laisse qu'une seule possibilité :  $\delta_{i+4}^{\ggg} = -2^{k \boxplus s_{i+4}}$ . Ceci n'est possible que si on calcule les conditions de la fin vers le début, et justifie ce choix. Les problèmes vont se poser quand on a des retenues.

**On choisit aussi un  $\partial(\Phi_{i+4}, \Phi'_{i+4})$**  qui correspond à  $\delta(\Phi_{i+4}, \Phi'_{i+4})$ . Il peut y avoir un grand nombre de choix, mais seul un petit nombre sera compatible avec les entrées de  $\Phi_{i+4}$  (notamment, pour avoir  $\partial(\Phi_{i+4}, \Phi_{i+4})^{[k]} \neq 0$ , il faut avoir  $\partial_{i+1}^{[k]} \neq 0$ ,  $\partial_{i+2}^{[k]} \neq 0$ , ou  $\partial_{i+3}^{[k]} \neq 0$ ). En pratique,

on commence par choisir une représentation à partir de la représentation signée de  $\delta_{i+4}^{\ggg}$ ,  $\delta_i$ , et  $\Delta_{i+4}$ , puis on cherche les conditions correspondantes, en commençant par le bit de poids faible, et quand il est impossible de satisfaire un bit, on modifie  $\partial\Phi_{i+4}$  sur ce bit. En cas d'erreur, on fait du backtracking pour explorer les autres possibilités pour les choix précédents.

### 3.4 Recherche du chemin

Comme on dispose d'un algorithme de calcul des conditions suffisantes, on peut estimer la probabilité qu'un message suive un chemin différentiel donné, et éliminer les chemins impossibles. Cependant le nombre de chemins potentiels est beaucoup trop grand pour pouvoir tous les tester ; on va devoir trouver un algorithme qui explore les chemins sans trop perdre de temps sur les mauvais chemins.

#### 3.4.1 Idée générale

L'algorithme de recherche de chemin effectue les mêmes étapes que l'algorithme de calcul des conditions suffisantes, mais cette fois on ne connaît pas  $\partial_i$ . On va donc supposer que  $\Phi_{i+4} = \Phi'_{i+4}$ , ce qui donne  $\delta_i = \delta_{i+4}^{\ggg} \boxplus \Delta_{i+4}$ . On a vu que les fonctions  $\Phi_{i+4}$  du premier et du deuxième tour ont des propriétés d'absorptions qui font qu'un chemin avec  $\Phi_{i+4} = \Phi'_{i+4}$  sera souvent réalisable. Ainsi, les  $\delta_i$  seront nuls jusqu'à ce qu'une différence soit introduite par  $\Delta$ , et cette différence se transmettra ensuite toutes les quatre étapes en subissant une rotation.

On prend ensuite pour  $\partial_i$  une représentation simple de  $\delta_i$  (remarquons qu'on a seulement besoin de  $\partial_i$  quand on est remonté à l'étape  $i - 1$ ). Si tout se passe relativement bien, on va obtenir un chemin différentiel, mais on n'aura pas  $\partial_{-1} = \partial_{-2} = \partial_{-3} = \partial_{-4} = 0$  (on retrouvera en fait en sortie les bits de différence introduits par  $\Delta$ , dans l'étape qui correspond modulo 4, et avec une rotation).

L'idée est ensuite de noter les bits sur lesquels le chemin n'a pas la bonne valeur, et de relancer la recherche en utilisant  $\Phi_i$  pour améliorer le chemin en modifiant ces bits. Ceci s'applique aussi si la recherche de conditions suffisantes a décelé un problème dans le chemin avant de remonter jusqu'au bout. On va donc gérer un ensemble  $\mathcal{P}$  de chemins, et chaque passe de l'algorithme sélectionnera un chemin  $P \in \mathcal{P}$  pour essayer de l'améliorer, et insérera les nouveaux chemins trouvés dans cet ensemble.

Pour pouvoir utiliser un  $\Phi_i \neq \Phi'_i$ , on a besoin d'avoir des entrées différentes pour les deux messages. Comme on a rarement des différences sur les bits précis qui nous intéressent, on pourra aussi jouer sur le choix du  $\partial_j$  qui correspond à  $\delta_j$ , pour  $j \in \{i + 1, i + 2, i + 3\}$ . En effet étendre une retenue n'a que peu d'effet sur la suite du chemin<sup>2</sup> mais permet parfois d'avoir une différence sur le bit qui nous intéresse dans  $\Phi_i$ .

Le programme sera écrit avec une fonction qui s'occupe de l'étape  $i$ , qui s'appelle récursivement sur des  $i$  plus petits, et quand on arrive à la première étape, on ajoute le chemin dans  $\mathcal{P}$ . Cela permet de tester plusieurs choix quand le cas se présente, et les différents chemins éventuellement obtenus seront tous considérés. Cette structure générale est décrite par l'algorithme 2.

#### 3.4.2 Structures de données

Pendant le calcul des conditions suffisantes, un chemin sera représenté par la valeur des  $\partial_i$  : pour chaque étape on a besoin de deux mots de 32 bits, qui vont représenter 32 valeurs dans  $\{-1, 0, 1\}$ . Mais quand on veut modifier un chemin existant, cette information est à peu près

<sup>2</sup>les différences sur les entrées de  $\Phi$  peuvent être absorbées, et c'est seulement  $\delta_{i+4}$  qui intervient ensuite à l'étape  $i$ , pas  $\partial_{i+4}$

**Algorithme 2** Recherche de chemin différentiel

---

```

PATHFIND()
1:  $\mathcal{P} \leftarrow \{\epsilon\}$ 
2: loop
3:   extraire  $P$  de  $\mathcal{P}$ 
4:   PATHSTEP( $P, \epsilon, 48$ ) // On lance la recherche à partir de la dernière étape
5: end loop

PATHSTEP( $P_0, P, i$ ) // On étend le chemin  $P$  à l'étape  $i$ , en suivant  $P_0$ 
6: if  $i < 0$  then
7:   ajouter  $P$  à  $\mathcal{P}$ 
8: else
9:   for all  $P'$  choix possible pour l'étape courante do
10:    PATCHTARGET( $P_0, P', i$ )
11:   end for
12: end if

PATCHTARGET( $P_0, P, i$ ) // On modifie l'étape  $i$  de  $P$  pour corriger les erreurs de  $P_0$ 
13: for all  $P'$  choix possible pour l'étape courante do
14:   PATCHCARRIES( $P_0, P', i$ )
15: end for

PATCHCARRIES( $P_0, P, i$ ) // On modifie l'étape  $i$  de  $P$  pour permettre les retenues de  $P$ 
16: for all  $P'$  choix possible pour l'étape courante do
17:   PATHSTEP( $P_0, P', i - 1$ )
18: end for

```

---

inexploitable puisqu'une différence dans le chemin va changer tous les  $\partial_i$  calculés ensuite. Entre deux passes de l'algorithme, on va donc plutôt conserver la valeur des  $\partial\Phi_i$ , qui suffit à reconstruire un chemin, et qui reste valable après une modification locale. De plus, les propagations de retenues utilisées pour les entrées de  $\Phi_i$  pourront se faire différemment en utilisant les nouvelles valeurs de  $\partial_i$  : au moment de choisir un  $\partial_j$ , on fera plusieurs essais avec différentes propagations de retenues en observant les  $\partial\Phi_i$  sur lesquels il peut influencer.

### 3.4.3 Corriger les erreurs

La partie centrale de l'algorithme est donc de trouver sur quels bits de  $Q_i$  il faut agir pour modifier les bits d'erreurs. On effectuera ces tentatives de modifications à chaque étape pour les bits problématiques  $Q_{i_0}^{[k]}$ . L'idée de base est très simple : on a vu que les bits de différence se propagent toutes les 4 étapes en subissant la rotation de l'étape, on va donc corriger  $Q_{i_0}$  en agissant sur  $Q_{i_0+4}$ . On obtient ainsi l'algorithme 3.

Quand on arrive à produire une différence dans  $\Phi_i$  sur un de ces bits, on s'attend à trouver un chemin avec un bit d'erreur en moins, puisque cette modification devrait peu affecter le reste du chemin (on aura néanmoins des problèmes si le bit d'erreur qu'on fait disparaître était déjà utilisé pour en corriger d'autres).

Le tableau 6 montre une branche de l'exécution sur le chemin de [YWZW05]. On voit que les différentes passes permettent de corriger les bits d'erreur (entourés dans le tableau). On voit aussi comment les retenues utilisées pour permettre un certain  $\delta\Phi_i$  peuvent s'adapter aux

step	$s_i$	$\delta m_i$	Chemin initial		Chemin 1		Chemin 2		Chemin 3		Chemin final	
			$\partial\Phi_i$	$\partial Q_i$	$\partial\Phi_i$	$\partial Q_i$	$\partial\Phi_i$	$\partial Q_i$	$\partial\Phi_i$	$\partial Q_i$	$\partial\Phi_i$	$\partial Q_i$
4	3	$\langle \blacktriangle^{[22]} \rangle$	$\langle \blacktriangledown^{[22]} \rangle$		$\langle \blacktriangledown^{[22]} \rangle$		$\langle \blacktriangledown^{[22]} \rangle$			$\langle \blacktriangle^{[25]} \rangle$		$\langle \blacktriangle^{[25]} \rangle$
5	7		$\langle \blacktriangledown^{[1]}, \blacktriangle^{[13]} \rangle$	$\langle \blacktriangledown^{[8]}, \blacktriangle^{[20]} \rangle$	$\langle \blacktriangle^{[13]} \rangle$	$\langle \blacktriangle^{[20]} \rangle$	$\langle \blacktriangle^{[13]} \rangle$	$\langle \blacktriangle^{[20]} \rangle$	$\langle \blacktriangle^{[13]} \rangle$	$\langle \blacktriangle^{[20]} \rangle$		
6	11		$\langle \blacktriangledown^{[17]} \rangle$	$\langle \blacktriangledown^{[28]} \rangle$	$\langle \blacktriangledown^{[17]} \rangle$	$\langle \blacktriangledown^{[28]} \rangle$						
7	19											
8	3									$\langle \blacktriangledown^{[28,29]} \rangle$		$\langle \blacktriangledown^{[28,29]} \rangle$
9	7			$\langle \blacktriangledown^{[15]}, \blacktriangle^{[27]} \rangle$		$\langle \blacktriangle^{[27]} \rangle$		$\langle \blacktriangledown^{[27,28]} \rangle$		$\langle \blacktriangle^{[27]} \rangle$		
10	11			$\langle \blacktriangledown^{[7]} \rangle$		$\langle \blacktriangledown^{[7]} \rangle$	$\langle \blacktriangledown^{[28]} \rangle$	$\langle \blacktriangledown^{[7]} \rangle$	$\langle \blacktriangledown^{[28]} \rangle$	$\langle \blacktriangledown^{[7]} \rangle$	$\langle \blacktriangledown^{[28]} \rangle$	$\langle \blacktriangledown^{[7]} \rangle$
11	19											
12	3									$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangle^{[31]} \rangle$
13	7			$\langle \blacktriangle^{[2]}, \blacktriangledown^{[22]} \rangle$		$\langle \blacktriangle^{[2]} \rangle$		$\langle \blacktriangle^{[2]} \rangle$		$\langle \blacktriangle^{[2]} \rangle$		
14	11			$\langle \blacktriangledown^{[18]} \rangle$		$\langle \blacktriangledown^{[18]} \rangle$		$\langle \blacktriangledown^{[18]} \rangle$		$\langle \blacktriangledown^{[18]} \rangle$		$\langle \blacktriangledown^{[18]} \rangle$
15	19											
16	3									$\langle \blacktriangle^{[2]} \rangle$		$\langle \blacktriangle^{[2]} \rangle$
17	5	$\langle \blacktriangle^{[22]} \rangle$		$\langle \blacktriangle^{[7]} \rangle$		$\langle \blacktriangle^{[7]}, \blacktriangle^{[27]} \rangle$		$\langle \blacktriangle^{[7]}, \blacktriangle^{[27]} \rangle$		$\langle \blacktriangle^{[7]}, \blacktriangle^{[27]} \rangle$	$\langle \blacktriangle^{[2]} \rangle$	$\langle \blacktriangle^{[7]}, \blacktriangle^{[27]} \rangle$
18	9			$\langle \blacktriangledown^{[27]} \rangle$		$\langle \blacktriangledown^{[27]} \rangle$		$\langle \blacktriangledown^{[27]} \rangle$		$\langle \blacktriangledown^{[27]} \rangle$		$\langle \blacktriangledown^{[27]} \rangle$
19	13											
20	3									$\langle \blacktriangle^{[5]} \rangle$		$\langle \blacktriangle^{[5]} \rangle$
21	5			$\langle \blacktriangle^{[12]} \rangle$	$\langle \blacktriangledown^{[27]} \rangle$	$\langle \blacktriangle^{[12]} \rangle$	$\langle \blacktriangledown^{[27]} \rangle$	$\langle \blacktriangle^{[12]} \rangle$	$\langle \blacktriangledown^{[27]} \rangle$	$\langle \blacktriangle^{[12]} \rangle$	$\langle \blacktriangledown^{[27]} \rangle$	$\langle \blacktriangle^{[12]} \rangle$
22	9			$\langle \blacktriangledown^{[4]} \rangle$		$\langle \blacktriangledown^{[4]} \rangle$		$\langle \blacktriangledown^{[4]} \rangle$		$\langle \blacktriangle^{[4,5]} \rangle$		$\langle \blacktriangle^{[4,5]} \rangle$
23	13											
24	3								$\langle \blacktriangledown^{[5]} \rangle$		$\langle \blacktriangledown^{[5]} \rangle$	
25	5			$\langle \blacktriangle^{[17]} \rangle$		$\langle \blacktriangle^{[17]} \rangle$		$\langle \blacktriangle^{[17]} \rangle$		$\langle \blacktriangle^{[17]} \rangle$		$\langle \blacktriangle^{[17]} \rangle$
26	9			$\langle \blacktriangledown^{[13]} \rangle$		$\langle \blacktriangledown^{[13]} \rangle$		$\langle \blacktriangledown^{[13]} \rangle$		$\langle \blacktriangledown^{[13]} \rangle$		$\langle \blacktriangledown^{[13]} \rangle$
27	13											
28	3											
29	5			$\langle \blacktriangle^{[22]} \rangle$		$\langle \blacktriangle^{[22]} \rangle$		$\langle \blacktriangle^{[22]} \rangle$		$\langle \blacktriangle^{[22]} \rangle$		$\langle \blacktriangle^{[22]} \rangle$
30	9			$\langle \blacktriangledown^{[22]} \rangle$		$\langle \blacktriangledown^{[22]} \rangle$		$\langle \blacktriangledown^{[22]} \rangle$		$\langle \blacktriangledown^{[22]} \rangle$		$\langle \blacktriangledown^{[22]} \rangle$
31	13											
32	3											
33	9			$\langle \blacktriangledown^{[22]} \rangle$		$\langle \blacktriangledown^{[22]} \rangle$		$\langle \blacktriangledown^{[22]} \rangle$		$\langle \blacktriangledown^{[22]} \rangle$		$\langle \blacktriangledown^{[22]} \rangle$
34	11	$\langle \blacktriangle^{[22]} \rangle$										

TAB. 6: Exécution de l'algorithme sur le chemin de [WLF+ 05]

**Algorithme 3** Calcul du bit à modifier

---

PATCHTARGET( $P_0, P, i$ )

- 1: **for all**  $Q_{i_0}^{[k]}$  bit d'erreur dans  $P_0$  **do**
- 2:   PATCHTARGETBIT( $P_0, P, i, i_0, k$ )
- 3: **end for**

---

PATCHTARGETBIT( $P_0, P, i, i_0, k$ ) //  $k$  est le bit de  $Q_{i_0}$  que l'on veut modifier à l'étape  $i$ 

- 4: **if**  $i < i_0$  **then**
  - 5:   **return**
  - 6: **else if**  $i = i_0$  **then**
  - 7:   modifier  $P$  sur le bit  $k$  de l'étape  $i$
  - 8:   PATCHCARRIES( $P_0, P, i$ )
  - 9: **else**
  - 10:   PATCHTARGETBIT( $P_0, P, i, i_0 + 4, k + s_{i_0} \bmod 32$ )
  - 11: **end if**
- 

modifications du reste du chemin.

En utilisant cette fonction très simple, on arrive ainsi à retrouver les chemins de [YWZW05] en une fraction de seconde. Cependant cela ne suffit pas pour retrouver des chemins utilisant la différentielle de [WLF<sup>+</sup>05], on va donc aussi utiliser des modifications plus complexes.

#### 3.4.4 Corrections indirectes

Pour trouver des chemins plus complexes, on va utiliser des corrections indirectes : au lieu d'agir sur un bit qui va directement modifier le bit erroné, on va agir sur  $Q_i$  de façon à introduire une différence dans  $Q_{i_1}$  qui elle-même ira corriger le bit qui nous intéresse dans  $Q_{i_0}$ . Ceci aura pour effet de remplacer un bit erroné par un autre, mais si un bit ne peut pas être corrigé de façon directe, le nouveau bit introduit sera peut-être plus facile à corriger.

La structure récursive de l'algorithme permet de gérer facilement ces corrections en ajoutant un cas où on agit sur  $Q_{i_0}^{[k]}$  en utilisant  $\Phi_{i_0+4}$  et  $Q_{i_0+a}^{[k]}$  ; on obtient ainsi l'algorithme 4.

Comme cette recherche est beaucoup plus coûteuse que la recherche en utilisant que des corrections directes, on ne lancera une recherche indirecte au niveau  $\eta$  sur un chemin qu'après l'avoir fait au niveau  $\eta - 1$ , et avoir traité les éventuels chemins obtenus.

Le tableau 7 propose un exemple de modification indirecte. On voit que le chemin de départ comporte trois erreurs, ainsi que le chemin modifié, mais ces erreurs ne sont pas les mêmes. Ce chemin modifié permet ensuite d'obtenir le chemin 3 avec des modifications directes, alors que le chemin de départ ne le permet pas.

Enfin, dans certains cas, on obtient de bons résultats avec une correction un peu plus indirecte : au lieu de faire en sorte de modifier un bit de  $Q_{i_1}$  qui permet une modification directe sur  $Q_{i_0}$ , on va viser un bit qui permet, *après une extension de retenue* de corriger l'erreur sur  $Q_{i_0}$ .

step	$s_i$	$\delta m_i$	Chemin initial		Chemin modifié	
			$\partial\Phi_i$	$\partial Q_i$	$\partial\Phi_i$	$\partial Q_i$
0	3					
1	7	$\langle \blacktriangle[31] \rangle$	$\langle \blacktriangledown[30] \rangle$	$\langle \blacktriangle[5,6] \rangle$	$\langle \blacktriangledown[30] \rangle$	$\langle \blacktriangledown\blacktriangle[5,6] \rangle$
2	11	$\langle \blacktriangledown[28], \blacktriangle[31] \rangle$	$\langle \blacktriangle[28], \blacktriangledown[31] \rangle$		$\langle \blacktriangle[28] \rangle$	$\langle \blacktriangle[10] \rangle$
3	19					
4	3				$\langle \blacktriangledown[7] \rangle$	$\langle \blacktriangle[10,11] \rangle$
5	7			$\langle \blacktriangledown\blacktriangle[12,13] \rangle$		$\langle \blacktriangledown\blacktriangle[12,13] \rangle$
6	11				$\langle \blacktriangledown\blacktriangle[10,11] \rangle$	
7	19					
8	3		$\langle \blacktriangle[13] \rangle$	$\langle \blacktriangle[16] \rangle$	$\langle \blacktriangle[13] \rangle$	$\langle \blacktriangledown[13], \blacktriangle[16] \rangle$
9	7			$\langle \blacktriangle[19] \rangle$		$\langle \blacktriangle[19] \rangle$
10	11					
11	19					
12	3	$\langle \blacktriangledown[16] \rangle$	$\langle \blacktriangle[19] \rangle$	$\langle \blacktriangle[22] \rangle$	$\langle \blacktriangle[19] \rangle$	$\langle \blacktriangledown[16], \blacktriangle[22] \rangle$
13	7			$\langle \blacktriangledown\blacktriangledown\blacktriangle[26...28] \rangle$		$\langle \blacktriangledown\blacktriangledown\blacktriangle[26...28] \rangle$
14	11					
15	19		$\langle \blacktriangle[28] \rangle$	$\langle \blacktriangledown\blacktriangle[15,16] \rangle$	$\langle \blacktriangle[28] \rangle$	$\langle \blacktriangledown\blacktriangle[15,16] \rangle$
16	3			$\langle \blacktriangle[25] \rangle$	$\langle \blacktriangle[16] \rangle$	$\langle \blacktriangle[25] \rangle$
17	5			$\langle \blacktriangle[31] \rangle$		$\langle \blacktriangle[31] \rangle$
18	9					
19	13	$\langle \blacktriangledown[16] \rangle$		$\langle \blacktriangledown[28] \rangle$		$\langle \blacktriangledown[28] \rangle$
20	3	$\langle \blacktriangle[31] \rangle$	$\langle \blacktriangledown[28], \blacktriangle[31] \rangle$	$\langle \blacktriangle[28], \blacktriangledown[31] \rangle$	$\langle \blacktriangledown[28], \blacktriangle[31] \rangle$	$\langle \blacktriangle[28], \blacktriangledown[31] \rangle$
21	5		$\langle \blacktriangledown[31] \rangle$		$\langle \blacktriangledown[31] \rangle$	
22	9					
23	13		$\langle \blacktriangle[28] \rangle$		$\langle \blacktriangle[28] \rangle$	
24	3	$\langle \blacktriangledown[28], \blacktriangle[31] \rangle$				
25	5					
26	9					
27	13					
28	3					
29	5					
30	9					
31	13					

TAB. 7: Une modification indirecte

---

**Algorithme 4** Calcul du bit à modifier avec modifications indirectes

---

PATCHTARGET( $P_0, P, i$ )

- 1: **for all**  $Q_{i_0}^{[k]}$  bit d'erreur dans  $P_0$  **do**
- 2:   PATCHTARGETBIT( $P_0, P, i, i_0, k, \eta$ )
- 3: **end for**

PATCHTARGETBIT( $P_0, P, i, i_0, k, \eta$ ) // On autorise  $\eta$  indirections

- 4: **if**  $i < i_0$  **then**
  - 5:   **return**
  - 6: **else if**  $i = i_0$  **then**
  - 7:   modifier  $P$  sur le bit  $k$  de l'étape  $i$
  - 8:   PATCHCARRIES( $P_0, P, i$ )
  - 9: **else**
  - 10:   PATCHTARGETBIT( $P_0, P, i, i_0 + 4, k + s_{i_0} \bmod 32, \eta$ )
  - 11: **end if**
  - 12: **if**  $\eta > 0$  **then**
  - 13:   modifier  $P_0$  sur le bit  $k$  de l'étape  $i_0$
  - 14:   **for**  $a \in \{1, 2, 3\}$  **do**
  - 15:     PATCHTARGETBIT( $P_0, P, i, i_0 + a, k, \eta - 1$ )
  - 16:   **end for**
  - 17: **end if**
- 

### 3.4.5 Diriger la recherche

Il est donc important de savoir comparer deux chemins pour trouver le plus prometteur, et ne pas lancer de longues passes de recherche pour rien. Parmi les critères que l'on peut utiliser pour comparer des chemins, on peut penser à :

- l'étape jusqu'à laquelle le calcul est remonté
- le nombre de bits d'erreurs
- le nombre de conditions
- le niveau d'indirection déjà tenté
- la profondeur dans l'arbre de recherche des chemins

## 4 Conclusion

Ainsi, les objectifs du stage ont été atteints, puisque j'ai acquis une bonne compréhension de l'attaque sur MD4, et que j'ai réussi à automatiser une des parties les plus obscures. L'algorithme est efficace et trouve de meilleurs chemins que ceux qui sont déjà connus.

### 4.1 Résultats

Avec cette méthode complètement automatisée, on obtient de meilleurs résultats que Schl  ffer et Oswald [SO06] :   partir de la diff erentielle de Wang, on trouve en un quelques heures un chemin avec 94 conditions, dont seulement 2 dans le troisi eme tour, et 16 dans le deuxi eme. Le tableau 8 compare ces diff erents chemins.

Nombre de conditions	1 <sup>er</sup> tour	2 <sup>e</sup> tour	3 <sup>e</sup> tour	total
chemin 1 page 30 : Wang [WLF <sup>+</sup> 05]	96	25	2	123
chemin 2 page 32 : Schl��ffer et Oswald [SO06]	122	22	2	146
chemin 3 page 34 : d�ecrit dans ce rapport	76	16	2	94

TAB. 8: Comparaison des diff erents chemins avec la diff erentielle de Wang

De plus, cet algorithme peut  tre utilis  pour trouver des chemins qui permettent de construire d'autres attaques que de simples collisions : il a permis de trouver un chemin qui ne marche que si une certaine condition sur l'IV est vraie, ce qui permet des attaques contre certains protocoles utilisant une fonction de hachage comme brique de base, par exemple HMAC.

Ce travail sera l'objet d'une soumission   FSE (*Fast Software Encryption*).

Enfin, la suite logique est d' tudier l'attaque sur MD5, et de voir si l'algorithme peut  tre adapt .

## Références

- [Cra05] Ronald Cramer, editor. *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*. Springer, 2005.
- [Jou04] Antoine Joux. Multicollisions in iterated hash functions. application to cascaded constructions. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 306–316. Springer, 2004.
- [Riv90] Ronald L. Rivest. The MD4 message digest algorithm. In Alfred Menezes and Scott A. Vanstone, editors, *CRYPTO*, volume 537 of *Lecture Notes in Computer Science*, pages 303–311. Springer, 1990.
- [Sho05] Victor Shoup, editor. *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*. Springer, 2005.
- [SO06] Martin Schl affer and Elisabeth Oswald. Searching for differential paths in MD4. In *FSE*, 2006.
- [WLF<sup>+</sup>05] Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, and Xiuyuan Yu. Cryptanalysis of the hash functions MD4 and RIPEMD. In Cramer [Cra05], pages 1–18.
- [WY05] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Cramer [Cra05], pages 19–35.
- [WYY05a] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In Shoup [Sho05], pages 17–36.
- [WYY05b] Xiaoyun Wang, Hongbo Yu, and Yiqun Lisa Yin. Efficient collision search attacks on SHA-0. In Shoup [Sho05], pages 1–16.
- [YWZW05] Hongbo Yu, Gaoli Wang, Guoyan Zhang, and Xiaoyun Wang. The second-preimage attack on MD4. In Yvo Desmedt, Huaxiong Wang, Yi Mu, and Yongqing Li, editors, *CANS*, volume 3810 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2005.

## A Chemins différentiels

Dans cette partie, nous donnons le détail de quelques chemins différentiels, en utilisant les notations de ce rapport. Les conditions nécessaires sont données à l'étape à laquelle elles sont utiles, et pour gagner en lisibilité, les conditions pour assurer que  $\delta Q_i$  donne bien  $\partial Q_i$  ne sont pas indiquées, car elles peuvent se déduire facilement de  $\partial Q_i$  (cf. proposition 3). Les  $\partial \Phi_i$  sont aussi donnés dans les tableau, pour une lecture plus facile.

Les chemins 1 à 3 pages 30–34 sont obtenus avec la même différentielle sur les messages :

- le chemin 1 page suivante est celui donné par Wang dans [WLF<sup>+</sup>05]
- le chemin 2 page 32 est obtenu dans [SO06], de façon presque automatique. Il a moins de conditions sur le deuxième tour que le chemin de Wang (22 au lieu de 25).
- le chemin 3 page 34 a été trouvé par mon algorithme, de façon complètement automatisée. Il a seulement 16 conditions sur le deuxième tour, et possède aussi moins de conditions sur le premier tour que les deux autres chemins.

Le chemin 4 page 36 utilise une autre différentielle avec une collision locale sur le dernier tour. Il a plus de conditions sur le deuxième tour que les chemins précédents, mais moins de conditions au total.

Le chemin 5 page 38 est celui de [YWZW05]; son intérêt est d'avoir un nombre total de conditions très faible : 62. En fait on peut faire un petit peu mieux en choisissant bien le bit sur lequel on introduit une différence : le chemin 6 page 39 a seulement 58 conditions.

Enfin, en page 40, on trouve une paire de message qui suivent le chemin différentiel 3 trouvé grâce à mon algorithme. Le tableau 13 montre les états internes pendant le calcul de MD4, et met en évidence les différences. Les deux message utilisés sont :

Message $M$															
4a	b9	00	39	dc	1d	20	7e	37	77	70	4a	77	98	ec	13
a4	3d	b1	48	ae	55	3c	76	5b	3e	9d	0d	b4	62	d9	5c
6d	15	3d	69	38	ab	14	11	aa	8e	1d	3e	07	df	69	48
af	3f	ba	09	c6	12	bf	38	ab	34	31	1c	13	e9	82	74
Message $M'$															
42	c9	00	29	db	1d	23	fe	37	f7	70	ba	77	98	7a	12
64	be	b0	c8	76	55	3c	76	d6	3e	9d	8d	b4	62	d9	58
6d	77	3e	69	18	a9	14	11	ca	8e	21	3e	0e	db	69	58
af	5f	b1	0d	c6	12	a7	38	db	34	31	1c	d3	ec	82	64

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
0	3				
1	7	$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangle^{[6]} \rangle$	
2	11	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$		$\langle \blacktriangledown^{[7]}, \blacktriangle^{[10]} \rangle$	$Q_0^{[6]} = Q_{-1}^{[6]}$
3	19		$\langle \blacktriangle^{[6]} \rangle$	$\langle \blacktriangle^{[25]} \rangle$	$Q_2^{[6]} = 1, Q_1^{[7]} = Q_0^{[7]}, Q_1^{[10]} = Q_0^{[10]}$
4	3				$Q_3^{[6]} = 1, Q_3^{[7]} = 0, Q_3^{[10]} = 0, Q_2^{[25]} = Q_1^{[25]}$
5	7			$\langle \blacktriangle^{[13]} \rangle$	$Q_4^{[7]} = 1, Q_4^{[10]} = 1, Q_4^{[25]} = 0$
6	11			$\langle \blacktriangle\blacktriangledown^{[18\dots 20]}, \blacktriangle^{[21]} \rangle$	$Q_4^{[13]} = Q_3^{[13]}, Q_5^{[25]} = 1$
7	19			$\langle \blacktriangledown\blacktriangledown^{[12\dots 14]} \rangle$	$Q_6^{[13]} = 0, Q_5^{[18]} = Q_4^{[18]}, Q_5^{[19]} = Q_4^{[19]}, Q_5^{[20]} = Q_4^{[20]}, Q_5^{[21]} = Q_4^{[21]}$
8	3		$\langle \blacktriangle^{[13]} \rangle$	$\langle \blacktriangle^{[16]} \rangle$	$Q_6^{[12]} = Q_5^{[12]}, Q_6^{[13]} = 0, Q_6^{[14]} = Q_5^{[14]}, Q_7^{[18]} = 0, Q_7^{[19]} = 0, Q_7^{[20]} = 0, Q_7^{[21]} = 0$
9	7	$\langle \blacktriangledown\blacktriangledown\blacktriangle^{[12\dots 14]}, \blacktriangle\blacktriangledown^{[18\dots 20]} \rangle$		$\langle \blacktriangle^{[19]}, \blacktriangledown\blacktriangle^{[20\dots 22]}, \blacktriangledown^{[25]} \rangle$	$Q_8^{[12]} = 1, Q_8^{[13]} = 1, Q_8^{[14]} = 1, Q_7^{[16]} = Q_6^{[16]}, Q_8^{[18]} = 0, Q_8^{[19]} = 0, Q_8^{[20]} = 0, Q_8^{[21]} = 1$
10	11		$\langle \blacktriangledown^{[21]} \rangle$	$\langle \blacktriangledown^{[29]} \rangle$	$Q_9^{[12]} = 1, Q_9^{[13]} = 1, Q_9^{[14]} = 1, Q_9^{[16]} = 0, Q_8^{[19]} = Q_7^{[19]}, Q_8^{[20]} = Q_7^{[20]}, Q_8^{[21]} = 1, Q_7^{[21]} = 0, Q_8^{[22]} = Q_7^{[22]}, Q_8^{[25]} = Q_7^{[25]}$
11	19			$\langle \blacktriangle^{[31]} \rangle$	$Q_{10}^{[16]} = 1, Q_{10}^{[19]} = 0, Q_{10}^{[20]} = 0, Q_{10}^{[21]} = 0, Q_{10}^{[22]} = 0, Q_{10}^{[25]} = 0, Q_9^{[29]} = Q_8^{[29]}$
12	3	$\langle \blacktriangledown^{[16]} \rangle$	$\langle \blacktriangle^{[19]}, \blacktriangle^{[22]} \rangle$	$\langle \blacktriangle^{[22]}, \blacktriangle^{[25]} \rangle$	$Q_{11}^{[19]} = 0, Q_{11}^{[20]} = 1, Q_{11}^{[21]} = 1, Q_{11}^{[22]} = 0, Q_{11}^{[25]} = 1, Q_{11}^{[29]} = 0, Q_{10}^{[31]} = Q_9^{[31]}$
13	7		$\langle \blacktriangle^{[25]} \rangle$	$\langle \blacktriangledown^{[26]}, \blacktriangledown\blacktriangle^{[28, 29]} \rangle$	$Q_{11}^{[22]} = Q_{10}^{[22]}, Q_{10}^{[25]} = 0, Q_{11}^{[25]} = 1, Q_{12}^{[29]} = 1, Q_{12}^{[31]} = 0$
14	11		$\langle \blacktriangle^{[29]} \rangle$		$Q_{13}^{[22]} = 0, Q_{13}^{[25]} = 0, Q_{12}^{[26]} = Q_{11}^{[26]}, Q_{12}^{[28]} = Q_{11}^{[28]}, Q_{11}^{[29]} = 0, Q_{12}^{[29]} = 1, Q_{13}^{[31]} = 1$
15	19			$\langle \blacktriangle^{[18]} \rangle$	$Q_{14}^{[22]} = 1, Q_{14}^{[25]} = 1, Q_{14}^{[26]} = 0, Q_{14}^{[28]} = 0, Q_{14}^{[29]} = 0$
16	3	$\langle \blacktriangledown^{[26]}, \blacktriangledown^{[28]} \rangle$		$\langle \blacktriangledown\blacktriangle^{[25, 26]}, \blacktriangledown^{[28]}, \blacktriangledown^{[31]} \rangle$	$Q_{14}^{[18]} = Q_{13}^{[18]}, Q_{15}^{[26]} \neq Q_{14}^{[26]}, Q_{15}^{[28]} \neq Q_{14}^{[28]}, Q_{15}^{[29]} = Q_{14}^{[29]}$
17	5	$\langle \blacktriangle^{[26]}, \blacktriangledown^{[28]} \rangle$			$Q_{16}^{[18]} = Q_{14}^{[18]}, Q_{15}^{[25]} = Q_{14}^{[25]}, Q_{15}^{[26]} \neq Q_{14}^{[26]}, Q_{15}^{[28]} \neq Q_{14}^{[28]}, Q_{15}^{[31]} = Q_{14}^{[31]}$
18	9				$Q_{17}^{[18]} = Q_{16}^{[18]}, Q_{17}^{[25]} = Q_{15}^{[25]}, Q_{17}^{[26]} = Q_{15}^{[26]}, Q_{17}^{[28]} = Q_{15}^{[28]}, Q_{17}^{[31]} = Q_{15}^{[31]}$
19	13	$\langle \blacktriangledown^{[16]} \rangle$		$\langle \blacktriangledown^{[29]}, \blacktriangle^{[31]} \rangle$	$Q_{18}^{[25]} = Q_{17}^{[25]}, Q_{18}^{[26]} = Q_{17}^{[26]}, Q_{18}^{[28]} = Q_{17}^{[28]}, Q_{18}^{[31]} = Q_{17}^{[31]}$

suite page de droite...

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
16	3		$\langle \blacktriangledown^{[26]}, \blacktriangledown^{[28]} \rangle$	$\langle \blacktriangledown \blacktriangle^{[25,26]}, \blacktriangledown^{[28]}, \blacktriangledown^{[31]} \rangle$	$Q_{14}^{[18]} = Q_{13}^{[18]}, Q_{15}^{[26]} \neq Q_{14}^{[26]}, Q_{15}^{[28]} \neq Q_{14}^{[28]}, Q_{15}^{[29]} = Q_{14}^{[29]}$
17	5		$\langle \blacktriangle^{[26]}, \blacktriangledown^{[28]} \rangle$		$Q_{16}^{[18]} = Q_{14}^{[18]}, Q_{15}^{[25]} = Q_{14}^{[25]}, Q_{15}^{[26]} \neq Q_{14}^{[26]}, Q_{15}^{[28]} \neq Q_{14}^{[28]},$ $Q_{15}^{[31]} = Q_{14}^{[31]}$
18	9				$Q_{17}^{[18]} = Q_{16}^{[18]}, Q_{17}^{[25]} = Q_{15}^{[25]}, Q_{17}^{[26]} = Q_{15}^{[26]}, Q_{17}^{[28]} = Q_{15}^{[28]},$ $Q_{17}^{[31]} = Q_{15}^{[31]}$
19	13	$\langle \blacktriangledown^{[16]} \rangle$		$\langle \blacktriangledown^{[29]}, \blacktriangle^{[31]} \rangle$	$Q_{18}^{[25]} = Q_{17}^{[25]}, Q_{18}^{[26]} = Q_{17}^{[26]}, Q_{18}^{[28]} = Q_{17}^{[28]}, Q_{18}^{[31]} = Q_{17}^{[31]}$
20	3	$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangledown \blacktriangle^{[28,29]}, \blacktriangledown^{[31]} \rangle$	$Q_{18}^{[29]} = Q_{17}^{[29]}, Q_{18}^{[31]} = Q_{17}^{[31]}$
21	5				$Q_{19}^{[28]} = Q_{18}^{[28]}$
22	9				$Q_{21}^{[28]} = Q_{19}^{[28]}$
23	13		$\langle \blacktriangle^{[29]}, \blacktriangledown^{[31]} \rangle$		$Q_{22}^{[28]} = Q_{21}^{[28]}, Q_{22}^{[29]} \neq Q_{21}^{[29]}, Q_{22}^{[31]} \neq Q_{21}^{[31]}$
24	3	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$			
25	5				
26	9				
27	13				
28	3				
29	5				
30	9				
31	13				
32	3				
33	9				
34	11				
35	15	$\langle \blacktriangledown^{[16]} \rangle$		$\langle \blacktriangledown^{[31]} \rangle$	
36	3	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$	$\langle \blacktriangledown^{[31]} \rangle$	$\langle \blacktriangledown^{[31]} \rangle$	
37	9				
38	11				
39	15		$\langle \blacktriangle^{[31]} \rangle$		
40	3	$\langle \blacktriangle^{[31]} \rangle$			
41	9				
42	11				
43	15				
44	3				
45	9				
46	11				
47	15				

123 conditions : 96 + 25 + 2

**Chemin 2** Le chemin de [SO06]

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
0	3				
1	7	$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangle^{[6]} \rangle$	
2	11	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$		$\langle \blacktriangledown^{[7]}, \blacktriangle^{[10]} \rangle$	$Q_0^{[6]} = Q_{-1}^{[6]}$
3	19		$\langle \blacktriangle^{[6]} \rangle$	$\langle \blacktriangle^{[25]} \rangle$	$Q_2^{[6]} = 1, Q_1^{[7]} = Q_0^{[7]}, Q_1^{[10]} = Q_0^{[10]}$
4	3				$Q_3^{[6]} = 1, Q_3^{[7]} = 0, Q_3^{[10]} = 0, Q_2^{[25]} = Q_1^{[25]}$
5	7			$\langle \blacktriangledown\blacktriangledown\blacktriangledown\blacktriangle^{[13\dots16]} \rangle$	$Q_4^{[7]} = 1, Q_4^{[10]} = 1, Q_4^{[25]} = 0$
6	11			$\langle \blacktriangledown^{[18]}, \blacktriangledown\blacktriangledown\blacktriangle^{[21\dots23]} \rangle$	$Q_4^{[13]} = Q_3^{[13]}, Q_4^{[14]} = Q_3^{[14]}, Q_4^{[15]} = Q_3^{[15]}, Q_4^{[16]} = Q_3^{[16]}, Q_5^{[25]} = 1$
7	19		$\langle \blacktriangle^{[23]} \rangle$	$\langle \blacktriangle^{[10]}, \blacktriangle^{[12]} \rangle$	$Q_6^{[13]} = 0, Q_6^{[14]} = 0, Q_6^{[15]} = 0, Q_6^{[16]} = 0, Q_5^{[18]} = Q_4^{[18]}, Q_5^{[21]} = Q_4^{[21]}, Q_5^{[22]} = Q_4^{[22]}, Q_4^{[23]} = 0, Q_5^{[23]} = 1$
8	3		$\langle \blacktriangle^{[16]}, \blacktriangledown\blacktriangle^{[22,23]} \rangle$	$\langle \blacktriangle^{[19]}, \blacktriangledown\blacktriangle^{[25,26]} \rangle$	$Q_6^{[10]} = Q_5^{[10]}, Q_6^{[12]} = Q_5^{[12]}, Q_7^{[13]} = 1, Q_7^{[14]} = 1, Q_7^{[15]} = 1, Q_7^{[16]} = 0, Q_7^{[18]} = 0, Q_7^{[21]} = 0, Q_7^{[22]} = 1, Q_7^{[23]} = 1$
9	7			$\langle \blacktriangledown\blacktriangledown\blacktriangle^{[20\dots23]} \rangle$	$Q_8^{[10]} = 0, Q_8^{[12]} = 0, Q_8^{[18]} = 1, Q_7^{[19]} = Q_6^{[19]}, Q_8^{[21]} = 1, Q_8^{[22]} = 1, Q_8^{[23]} = 1, Q_7^{[25]} = Q_6^{[25]}, Q_7^{[26]} = Q_6^{[26]}$
10	11		$\langle \blacktriangledown^{[21]} \rangle$	$\langle \blacktriangledown^{[29]} \rangle$	$Q_9^{[10]} = 1, Q_9^{[12]} = 1, Q_9^{[19]} = 0, Q_8^{[20]} = Q_7^{[20]}, Q_8^{[21]} = 1, Q_7^{[21]} = 0, Q_8^{[22]} = Q_7^{[22]}, Q_8^{[23]} = Q_7^{[23]}, Q_9^{[25]} = 0, Q_9^{[26]} = 0$
11	19			$\langle \blacktriangle^{[0]}, \blacktriangle^{[29]}, \blacktriangledown^{[31]} \rangle$	$Q_{10}^{[19]} = 1, Q_{10}^{[20]} = 0, Q_{10}^{[21]} = 0, Q_{10}^{[22]} = 0, Q_{10}^{[23]} = 0, Q_{10}^{[25]} = 1, Q_{10}^{[26]} = 1, Q_9^{[29]} = Q_8^{[29]}$
12	3	$\langle \blacktriangledown^{[16]} \rangle$	$\langle \blacktriangledown^{[22]} \rangle$	$\langle \blacktriangledown\blacktriangle^{[19,20]}, \blacktriangle^{[22]}, \blacktriangledown^{[25]}, \blacktriangledown\blacktriangle^{[28,29]} \rangle$	$Q_{10}^{[0]} = Q_9^{[0]}, Q_{11}^{[20]} = 1, Q_{11}^{[21]} = 1, Q_{11}^{[22]} = 0, Q_{11}^{[23]} = 1, Q_9^{[29]} = 0, Q_{10}^{[31]} = Q_9^{[31]}$
13	7		$\langle \blacktriangledown^{[20]} \rangle$		$Q_{12}^{[0]} = 0, Q_{11}^{[19]} = Q_{10}^{[19]}, Q_{11}^{[20]} = 1, Q_{10}^{[20]} = 0, Q_{11}^{[22]} = Q_{10}^{[22]}, Q_{11}^{[25]} = Q_{10}^{[25]}, Q_{11}^{[28]} = Q_{10}^{[28]}, Q_{10}^{[29]} = 1, Q_{12}^{[31]} = 0$
14	11		$\langle \blacktriangle^{[29]} \rangle$		$Q_{13}^{[0]} = 1, Q_{13}^{[19]} = 0, Q_{13}^{[20]} = 0, Q_{13}^{[22]} = 0, Q_{13}^{[25]} = 0, Q_{13}^{[28]} = 0, Q_{13}^{[31]} = 1$
15	19			$\langle \blacktriangle^{[16]}, \blacktriangledown\blacktriangle^{[18,19]} \rangle$	$Q_{14}^{[19]} = 1, Q_{14}^{[20]} = 1, Q_{14}^{[22]} = 1, Q_{14}^{[25]} = 1, Q_{14}^{[28]} = 1, Q_{14}^{[29]} = 1$
16	3		$\langle \blacktriangle^{[19]} \rangle$	$\langle \blacktriangle^{[25]}, \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$	$Q_{14}^{[16]} = Q_{13}^{[16]}, Q_{14}^{[18]} = Q_{13}^{[18]}, Q_{14}^{[19]} \neq Q_{13}^{[19]}$
17	5				$Q_{16}^{[16]} = Q_{14}^{[16]}, Q_{16}^{[18]} = Q_{14}^{[18]}, Q_{16}^{[19]} = Q_{14}^{[19]}, Q_{15}^{[25]} = Q_{14}^{[25]}, Q_{15}^{[28]} = Q_{14}^{[28]}, Q_{15}^{[31]} = Q_{14}^{[31]}$
18	9				$Q_{17}^{[16]} = Q_{16}^{[16]}, Q_{17}^{[18]} = Q_{16}^{[18]}, Q_{17}^{[19]} = Q_{16}^{[19]}, Q_{17}^{[25]} = Q_{15}^{[25]}, Q_{17}^{[28]} = Q_{15}^{[28]}, Q_{17}^{[31]} = Q_{15}^{[31]}$
19	13	$\langle \blacktriangledown^{[16]} \rangle$		$\langle \blacktriangle^{[31]} \rangle$	$Q_{18}^{[25]} = Q_{17}^{[25]}, Q_{18}^{[28]} = Q_{17}^{[28]}, Q_{18}^{[31]} = Q_{17}^{[31]}$

suite page de droite...

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
16	3		$\langle \blacktriangle^{[19]} \rangle$	$\langle \blacktriangle^{[25]}, \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$	$Q_{14}^{[16]} = Q_{13}^{[16]}, Q_{14}^{[18]} = Q_{13}^{[18]}, Q_{14}^{[19]} \neq Q_{13}^{[19]}$
17	5				$Q_{16}^{[16]} = Q_{14}^{[16]}, Q_{16}^{[18]} = Q_{14}^{[18]}, Q_{16}^{[19]} = Q_{14}^{[19]}, Q_{15}^{[25]} = Q_{14}^{[25]}, Q_{15}^{[28]} = Q_{14}^{[28]},$ $Q_{15}^{[31]} = Q_{14}^{[31]}$
18	9				$Q_{17}^{[16]} = Q_{16}^{[16]}, Q_{17}^{[18]} = Q_{16}^{[18]}, Q_{17}^{[19]} = Q_{16}^{[19]}, Q_{17}^{[25]} = Q_{15}^{[25]}, Q_{17}^{[28]} = Q_{15}^{[28]},$ $Q_{17}^{[31]} = Q_{15}^{[31]}$
19	13	$\langle \blacktriangledown^{[16]} \rangle$		$\langle \blacktriangle^{[31]} \rangle$	$Q_{18}^{[25]} = Q_{17}^{[25]}, Q_{18}^{[28]} = Q_{17}^{[28]}, Q_{18}^{[31]} = Q_{17}^{[31]}$
20	3	$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangle^{[28]}, \blacktriangledown^{[31]} \rangle$	$Q_{18}^{[31]} = Q_{17}^{[31]}$
21	5				$Q_{19}^{[28]} = Q_{18}^{[28]}$
22	9				$Q_{21}^{[28]} = Q_{19}^{[28]}$
23	13		$\langle \blacktriangledown^{[31]} \rangle$		$Q_{22}^{[28]} = Q_{21}^{[28]}, Q_{22}^{[31]} \neq Q_{21}^{[31]}$
24	3	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$			
25	5				
26	9				
27	13				
28	3				
29	5				
30	9				
31	13				
32	3				
33	9				
34	11				
35	15	$\langle \blacktriangledown^{[16]} \rangle$		$\langle \blacktriangledown^{[31]} \rangle$	
36	3	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$	$\langle \blacktriangledown^{[31]} \rangle$	$\langle \blacktriangledown^{[31]} \rangle$	
37	9				
38	11				
39	15		$\langle \blacktriangle^{[31]} \rangle$		
40	3	$\langle \blacktriangle^{[31]} \rangle$			
41	9				
42	11				
43	15				
44	3				
45	9				
46	11				
47	15				

146 conditions : 122 + 22 + 2

---

**Chemin 3** Un meilleur chemin avec la même différentielle
 

---

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
0	3				
1	7	$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangle^{[6]} \rangle$	
2	11	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$		$\langle \blacktriangledown^{[7]}, \blacktriangle^{[10]} \rangle$	$Q_0^{[6]} = Q_{-1}^{[6]}$
3	19				$Q_2^{[6]} = 0, Q_1^{[7]} = Q_0^{[7]}, Q_1^{[10]} = Q_0^{[10]}$
4	3		$\langle \blacktriangle\blacktriangledown^{[6,7]} \rangle$	$\langle \blacktriangle\blacktriangledown^{[9\dots11]} \rangle$	$Q_3^{[6]} = 0, Q_3^{[7]} = 1, Q_3^{[10]} = 0$
5	7			$\langle \blacktriangle^{[13]} \rangle$	$Q_4^{[7]} = 1, Q_3^{[9]} = Q_2^{[9]}, Q_3^{[10]} = 0, Q_3^{[11]} = Q_2^{[11]}$
6	11		$\langle \blacktriangle\blacktriangledown^{[10,11]} \rangle$	$\langle \blacktriangledown^{[18]} \rangle$	$Q_5^{[9]} = 0, Q_5^{[10]} = 1, Q_5^{[11]} = 1, Q_4^{[13]} = Q_3^{[13]}$
7	19				$Q_6^{[9]} = 1, Q_6^{[10]} = 1, Q_6^{[11]} = 1, Q_6^{[13]} = 0, Q_5^{[18]} = Q_4^{[18]}$
8	3		$\langle \blacktriangle^{[13]} \rangle$	$\langle \blacktriangledown^{[12]}, \blacktriangle^{[16]} \rangle$	$Q_7^{[13]} = 0, Q_7^{[18]} = 0$
9	7		$\langle \blacktriangledown^{[12]} \rangle$	$\langle \blacktriangle^{[19]} \rangle$	$Q_7^{[12]} = 1, Q_6^{[12]} = 0, Q_7^{[16]} = Q_6^{[16]}, Q_8^{[18]} = 1$
10	11			$\langle \blacktriangledown^{[29]} \rangle$	$Q_9^{[12]} = 0, Q_9^{[16]} = 0, Q_8^{[19]} = Q_7^{[19]}$
11	19				$Q_{10}^{[12]} = 1, Q_{10}^{[16]} = 1, Q_{10}^{[19]} = 0, Q_9^{[29]} = Q_8^{[29]}$
12	3	$\langle \blacktriangledown^{[16]} \rangle$	$\langle \blacktriangle^{[19]} \rangle$	$\langle \blacktriangle\blacktriangledown^{[15,16]}, \blacktriangle^{[22]} \rangle$	$Q_{11}^{[19]} = 0, Q_{11}^{[29]} = 0$
13	7			$\langle \blacktriangledown\blacktriangledown\blacktriangle^{[26\dots29]} \rangle$	$Q_{11}^{[15]} = Q_{10}^{[15]}, Q_{11}^{[16]} = Q_{10}^{[16]}, Q_{11}^{[22]} = Q_{10}^{[22]}, Q_{12}^{[29]} = 1$
14	11		$\langle \blacktriangle^{[29]} \rangle$		$Q_{13}^{[15]} = 0, Q_{13}^{[16]} = 0, Q_{13}^{[22]} = 0, Q_{12}^{[26]} = Q_{11}^{[26]}, Q_{12}^{[27]} = Q_{11}^{[27]},$ $Q_{12}^{[28]} = Q_{11}^{[28]}, Q_{11}^{[29]} = 0, Q_{12}^{[29]} = 1$
15	19		$\langle \blacktriangledown\blacktriangle^{[28,29]} \rangle$	$\langle \blacktriangle^{[15]} \rangle$	$Q_{14}^{[15]} = 1, Q_{14}^{[16]} = 1, Q_{14}^{[22]} = 1, Q_{14}^{[26]} = 0, Q_{14}^{[27]} = 0, Q_{14}^{[28]} = 1,$ $Q_{14}^{[29]} = 1$
16	3		$\langle \blacktriangle^{[15]} \rangle$	$\langle \blacktriangle^{[25]} \rangle$	$Q_{14}^{[15]} \neq Q_{13}^{[15]}, Q_{15}^{[26]} = Q_{14}^{[26]}, Q_{15}^{[27]} = Q_{14}^{[27]}, Q_{15}^{[28]} = Q_{14}^{[28]},$ $Q_{15}^{[29]} = Q_{14}^{[29]}$
17	5			$\langle \blacktriangle^{[31]} \rangle$	$Q_{16}^{[15]} = Q_{14}^{[15]}, Q_{15}^{[25]} = Q_{14}^{[25]}$
18	9				$Q_{17}^{[15]} = Q_{16}^{[15]}, Q_{17}^{[25]} = Q_{15}^{[25]}, Q_{16}^{[31]} = Q_{15}^{[31]}$
19	13	$\langle \blacktriangledown^{[16]} \rangle$		$\langle \blacktriangledown^{[28]} \rangle$	$Q_{18}^{[25]} = Q_{17}^{[25]}, Q_{18}^{[31]} = Q_{16}^{[31]}$
20	3	$\langle \blacktriangle^{[31]} \rangle$	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$	$\langle \blacktriangle^{[28]}, \blacktriangledown^{[31]} \rangle$	$Q_{18}^{[28]} \neq Q_{17}^{[28]}, Q_{19}^{[31]} \neq Q_{18}^{[31]}$
21	5		$\langle \blacktriangledown^{[31]} \rangle$		$Q_{19}^{[31]} \neq Q_{18}^{[31]}$

suite page de droite...

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
18	9				$Q_{17}^{[15]} = Q_{16}^{[15]}, Q_{17}^{[25]} = Q_{15}^{[25]}, Q_{16}^{[31]} = Q_{15}^{[31]}$
19	13	$\langle \blacktriangledown^{[16]} \rangle$		$\langle \blacktriangledown^{[28]} \rangle$	$Q_{18}^{[25]} = Q_{17}^{[25]}, Q_{18}^{[31]} = Q_{16}^{[31]}$
20	3	$\langle \blacktriangle^{[31]} \rangle$	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$	$\langle \blacktriangle^{[28]}, \blacktriangledown^{[31]} \rangle$	$Q_{18}^{[28]} \neq Q_{17}^{[28]}, Q_{19}^{[31]} \neq Q_{18}^{[31]}$
21	5		$\langle \blacktriangledown^{[31]} \rangle$		$Q_{19}^{[31]} \neq Q_{18}^{[31]}$
22	9				$Q_{21}^{[31]} = Q_{19}^{[31]}$
23	13		$\langle \blacktriangle^{[28]} \rangle$		$Q_{22}^{[28]} \neq Q_{21}^{[28]}, Q_{22}^{[31]} = Q_{21}^{[31]}$
24	3	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$			
25	5				
26	9				
27	13				
28	3				
29	5				
30	9				
31	13				
32	3				
33	9				
34	11				
35	15	$\langle \blacktriangledown^{[16]} \rangle$		$\langle \blacktriangledown^{[31]} \rangle$	
36	3	$\langle \blacktriangledown^{[28]}, \blacktriangle^{[31]} \rangle$	$\langle \blacktriangledown^{[31]} \rangle$	$\langle \blacktriangledown^{[31]} \rangle$	
37	9				
38	11				
39	15		$\langle \blacktriangle^{[31]} \rangle$		
40	3	$\langle \blacktriangle^{[31]} \rangle$			
41	9				
42	11				
43	15				
44	3				
45	9				
46	11				
47	15				

94 conditions : 76 + 16 + 2

**Chemin 4** Un chemin avec une autre collision locale dans le dernier tour

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
0	3				
1	7				
2	11				
3	19				
4	3				
5	7				
6	11	$\langle \blacktriangledown^{[20]} \rangle$		$\langle \blacktriangledown^{[0]}, \blacktriangle^{[31]} \rangle$	
7	19				$Q_5^{[0]} = Q_4^{[0]}, Q_5^{[31]} = Q_4^{[31]}$
8	3				$Q_7^{[0]} = 0, Q_7^{[31]} = 0$
9	7				$Q_8^{[0]} = 1, Q_8^{[31]} = 1$
10	11			$\langle \blacktriangle\blacktriangle\blacktriangle\blacktriangle\blacktriangledown^{[10\dots 16]} \rangle$	
11	19	$\langle \blacktriangledown^{[12]}, \blacktriangle^{[16]} \rangle$	$\langle \blacktriangledown^{[3]}, \blacktriangle^{[31]} \rangle$		$Q_9^{[10]} = Q_8^{[10]}, Q_9^{[11]} = Q_8^{[11]}, Q_8^{[12]} = 1, Q_9^{[12]} = 0, Q_9^{[13]} = Q_8^{[13]},$ $Q_9^{[14]} = Q_8^{[14]}, Q_9^{[15]} = Q_8^{[15]}, Q_9^{[16]} = 0, Q_8^{[16]} = 1$
12	3				$Q_{10}^{[3]} = Q_9^{[3]}, Q_{11}^{[10]} = 0, Q_{11}^{[11]} = 0, Q_{11}^{[12]} = 0, Q_{11}^{[13]} = 0, Q_{11}^{[14]} = 0,$ $Q_{11}^{[15]} = 0, Q_{11}^{[16]} = 0, Q_{10}^{[31]} = Q_9^{[31]}$
13	7	$\langle \blacktriangle^{[31]} \rangle$	$\langle \blacktriangle^{[3]}, \blacktriangledown^{[31]} \rangle$	$\langle \blacktriangle^{[10]} \rangle$	$Q_{12}^{[3]} = 1, Q_{12}^{[10]} = 1, Q_{12}^{[11]} = 1, Q_{12}^{[12]} = 1, Q_{12}^{[13]} = 1, Q_{12}^{[14]} = 1,$ $Q_{12}^{[15]} = 1, Q_{12}^{[16]} = 1, Q_{12}^{[31]} = 1$
14	11	$\langle \blacktriangledown^{[16]}, \blacktriangledown^{[31]} \rangle$	$\langle \blacktriangle^{[10]}, \blacktriangledown^{[31]} \rangle$	$\langle \blacktriangledown^{[27]} \rangle$	$Q_{13}^{[3]} = 1, Q_{11}^{[10]} = 0, Q_{12}^{[10]} = 1, Q_{13}^{[31]} = 0$
15	19			$\langle \blacktriangledown^{[18]}, \blacktriangle^{[22]} \rangle$	$Q_{14}^{[10]} = 0, Q_{13}^{[27]} = Q_{12}^{[27]}$
16	3				$Q_{15}^{[10]} = Q_{14}^{[10]}, Q_{14}^{[18]} = Q_{13}^{[18]}, Q_{14}^{[22]} = Q_{13}^{[22]}, Q_{15}^{[27]} = Q_{13}^{[27]}$
17	5			$\langle \blacktriangle^{[15]} \rangle$	$Q_{16}^{[18]} = Q_{14}^{[18]}, Q_{16}^{[22]} = Q_{14}^{[22]}, Q_{16}^{[27]} = Q_{15}^{[27]}$
18	9			$\langle \blacktriangledown\blacktriangledown^{[4,5]} \rangle$	$Q_{16}^{[15]} = Q_{15}^{[15]}, Q_{17}^{[18]} = Q_{16}^{[18]}, Q_{17}^{[22]} = Q_{16}^{[22]}$
19	13	$\langle \blacktriangledown^{[5]} \rangle$	$\langle \blacktriangledown\blacktriangledown\blacktriangle^{[3\dots 5]}, \blacktriangledown^{[18]}, \blacktriangledown^{[31]} \rangle$		$Q_{17}^{[4]} = Q_{16}^{[4]}, Q_{17}^{[5]} \neq Q_{16}^{[5]}, Q_{18}^{[15]} = Q_{16}^{[15]}$
20	3				$Q_{18}^{[3]} = Q_{17}^{[3]}, Q_{19}^{[15]} = Q_{18}^{[15]}, Q_{18}^{[18]} = Q_{17}^{[18]}, Q_{18}^{[31]} = Q_{17}^{[31]}$
21	5			$\langle \blacktriangle^{[20]} \rangle$	$Q_{20}^{[3]} = Q_{18}^{[3]}, Q_{20}^{[18]} = Q_{18}^{[18]}, Q_{20}^{[31]} = Q_{18}^{[31]}$

suite page de droite...

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
18	9			$\langle \blacktriangle^{[4,5]} \rangle$	$Q_{16}^{[15]} = Q_{15}^{[15]}, Q_{17}^{[18]} = Q_{16}^{[18]}, Q_{17}^{[22]} = Q_{16}^{[22]}$
19	13		$\langle \blacktriangledown^{[5]} \rangle$	$\langle \blacktriangledown\blacktriangledown^{[3\dots 5]}, \blacktriangledown^{[18]}, \blacktriangledown^{[31]} \rangle$	$Q_{17}^{[4]} = Q_{16}^{[4]}, Q_{17}^{[5]} \neq Q_{16}^{[5]}, Q_{18}^{[15]} = Q_{16}^{[15]}$
20	3				$Q_{18}^{[3]} = Q_{17}^{[3]}, Q_{19}^{[15]} = Q_{18}^{[15]}, Q_{18}^{[18]} = Q_{17}^{[18]}, Q_{18}^{[31]} = Q_{17}^{[31]}$
21	5			$\langle \blacktriangle^{[20]} \rangle$	$Q_{20}^{[3]} = Q_{18}^{[3]}, Q_{20}^{[18]} = Q_{18}^{[18]}, Q_{20}^{[31]} = Q_{18}^{[31]}$
22	9		$\langle \blacktriangledown\blacktriangle^{[4,5]} \rangle$		$Q_{21}^{[3]} = Q_{20}^{[3]}, Q_{21}^{[4]} \neq Q_{20}^{[4]}, Q_{21}^{[5]} \neq Q_{20}^{[5]}, Q_{21}^{[18]} = Q_{20}^{[18]}, Q_{20}^{[20]} = Q_{19}^{[20]}, Q_{21}^{[31]} = Q_{20}^{[31]}$
23	13	$\langle \blacktriangle^{[31]} \rangle$		$\langle \blacktriangle^{[16]}, \blacktriangledown^{[31]} \rangle$	$Q_{22}^{[20]} = Q_{20}^{[20]}$
24	3				$Q_{22}^{[16]} = Q_{21}^{[16]}, Q_{23}^{[20]} = Q_{22}^{[20]}, Q_{22}^{[31]} = Q_{21}^{[31]}$
25	5	$\langle \blacktriangledown^{[20]} \rangle$			$Q_{24}^{[16]} = Q_{22}^{[16]}, Q_{24}^{[31]} = Q_{22}^{[31]}$
26	9				$Q_{25}^{[16]} = Q_{24}^{[16]}, Q_{25}^{[31]} = Q_{24}^{[31]}$
27	13	$\langle \blacktriangledown^{[16]}, \blacktriangledown^{[31]} \rangle$			
28	3				
29	5				
30	9				
31	13				
32	3				
33	9				
34	11				
35	15				
36	3				
37	9				
38	11	$\langle \blacktriangledown^{[20]} \rangle$		$\langle \blacktriangledown^{[31]} \rangle$	
39	15	$\langle \blacktriangledown^{[16]}, \blacktriangledown^{[31]} \rangle$	$\langle \blacktriangle^{[31]} \rangle$	$\langle \blacktriangledown^{[31]} \rangle$	
40	3				
41	9				
42	11		$\langle \blacktriangle^{[31]} \rangle$		
43	15	$\langle \blacktriangle^{[31]} \rangle$			
44	3				
45	9				
46	11				
47	15				

99 conditions : 56 + 41 + 2

**Chemin 5** Le chemin de [YWZW05]

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
0	3				
1	7				
2	11				
3	19				
4	3	$\langle \blacktriangle^{[22]} \rangle$		$\langle \blacktriangle^{[25]} \rangle$	
5	7				$Q_3^{[25]} = Q_2^{[25]}$
6	11				$Q_5^{[25]} = 0$
7	19				$Q_6^{[25]} = 1$
8	3			$\langle \blacktriangledown^{[28,29]} \rangle$	
9	7				$Q_7^{[28]} = Q_6^{[28]}, Q_7^{[29]} = Q_6^{[29]}$
10	11		$\langle \blacktriangledown^{[28]} \rangle$	$\langle \blacktriangledown^{[7]} \rangle$	$Q_9^{[28]} = 1, Q_9^{[29]} = 0$
11	19				$Q_9^{[7]} = Q_8^{[7]}, Q_{10}^{[28]} = 1, Q_{10}^{[29]} = 1$
12	3			$\langle \blacktriangle^{[31]} \rangle$	$Q_{11}^{[7]} = 0$
13	7				$Q_{12}^{[7]} = 1, Q_{11}^{[31]} = Q_{10}^{[31]}$
14	11			$\langle \blacktriangledown^{[18]} \rangle$	$Q_{13}^{[31]} = 0$
15	19				$Q_{13}^{[18]} = Q_{12}^{[18]}, Q_{14}^{[31]} = 1$
16	3			$\langle \blacktriangle^{[2]} \rangle$	$Q_{15}^{[18]} = Q_{13}^{[18]}$
17	5	$\langle \blacktriangle^{[22]} \rangle$	$\langle \blacktriangle^{[2]} \rangle$	$\langle \blacktriangle^{[7]}, \blacktriangle^{[27]} \rangle$	$Q_{15}^{[2]} \neq Q_{14}^{[2]}, Q_{16}^{[18]} = Q_{15}^{[18]}$
18	9			$\langle \blacktriangledown^{[27]} \rangle$	$Q_{17}^{[2]} = Q_{15}^{[2]}, Q_{16}^{[7]} = Q_{15}^{[7]}, Q_{16}^{[27]} = Q_{15}^{[27]}$
19	13				$Q_{18}^{[2]} = Q_{17}^{[2]}, Q_{18}^{[7]} = Q_{16}^{[7]}$
20	3			$\langle \blacktriangle^{[5]} \rangle$	$Q_{19}^{[7]} = Q_{18}^{[7]}$
21	5		$\langle \blacktriangledown^{[27]} \rangle$	$\langle \blacktriangle^{[12]} \rangle$	$Q_{19}^{[5]} = Q_{18}^{[5]}, Q_{20}^{[27]} \neq Q_{19}^{[27]}$
22	9			$\langle \blacktriangle^{[4,5]} \rangle$	$Q_{21}^{[5]} = Q_{19}^{[5]}, Q_{20}^{[12]} = Q_{19}^{[12]}$
23	13				$Q_{21}^{[4]} = Q_{20}^{[4]}, Q_{22}^{[12]} = Q_{20}^{[12]}$
24	3		$\langle \blacktriangledown^{[5]} \rangle$		$Q_{23}^{[4]} = Q_{21}^{[4]}, Q_{23}^{[5]} \neq Q_{21}^{[5]}, Q_{23}^{[12]} = Q_{22}^{[12]}$
25	5			$\langle \blacktriangle^{[17]} \rangle$	$Q_{24}^{[4]} = Q_{23}^{[4]}, Q_{24}^{[5]} = Q_{23}^{[5]}$
26	9			$\langle \blacktriangledown^{[13]} \rangle$	$Q_{24}^{[17]} = Q_{23}^{[17]}$
27	13				$Q_{25}^{[13]} = Q_{24}^{[13]}, Q_{26}^{[17]} = Q_{24}^{[17]}$
28	3				$Q_{27}^{[13]} = Q_{25}^{[13]}, Q_{27}^{[17]} = Q_{26}^{[17]}$
29	5			$\langle \blacktriangle^{[22]} \rangle$	$Q_{28}^{[13]} = Q_{27}^{[13]}$
30	9			$\langle \blacktriangledown^{[22]} \rangle$	$Q_{28}^{[22]} = Q_{27}^{[22]}$
31	13				
32	3				
33	9		$\langle \blacktriangledown^{[22]} \rangle$		$Q_{32}^{[22]} = Q_{31}^{[22]}$
34	11	$\langle \blacktriangle^{[22]} \rangle$			
35	15				
36	3				
37	9				
38	11				

62 conditions : 24 + 37 + 1

**Chemin 6** Un meilleur chemin en introduisant la différence sur le bit 25

step	$s_i$	$\delta m_i$	$\partial \Phi_i$	$\partial Q_i$	conditions
0	3				
1	7				
2	11				
3	19				
4	3	$\langle \blacktriangle^{[25]} \rangle$		$\langle \blacktriangle^{[28]} \rangle$	
5	7				$Q_3^{[28]} = Q_2^{[28]}$
6	11				$Q_5^{[28]} = 0$
7	19				$Q_6^{[28]} = 1$
8	3			$\langle \blacktriangle^{[31]} \rangle$	
9	7				$Q_7^{[31]} = Q_6^{[31]}$
10	11		$\langle \blacktriangle^{[31]} \rangle$	$\langle \blacktriangledown^{[10]} \rangle$	$Q_9^{[31]} = 1$
11	19				$Q_9^{[10]} = Q_8^{[10]}, Q_{10}^{[31]} = 1$
12	3			$\langle \blacktriangle^{[2]} \rangle$	$Q_{11}^{[10]} = 0$
13	7				$Q_{11}^{[2]} = Q_{10}^{[2]}, Q_{12}^{[10]} = 1$
14	11			$\langle \blacktriangledown^{[21]} \rangle$	$Q_{13}^{[2]} = 0$
15	19				$Q_{14}^{[2]} = 1, Q_{13}^{[21]} = Q_{12}^{[21]}$
16	3			$\langle \blacktriangle^{[5]} \rangle$	$Q_{15}^{[21]} = Q_{13}^{[21]}$
17	5	$\langle \blacktriangle^{[25]} \rangle$	$\langle \blacktriangle^{[5]} \rangle$	$\langle \blacktriangle^{[10]}, \blacktriangle^{[30]} \rangle$	$Q_{15}^{[5]} \neq Q_{14}^{[5]}, Q_{16}^{[21]} = Q_{15}^{[21]}$
18	9			$\langle \blacktriangledown^{[30]} \rangle$	$Q_{17}^{[5]} = Q_{15}^{[5]}, Q_{16}^{[10]} = Q_{15}^{[10]}, Q_{16}^{[30]} = Q_{15}^{[30]}$
19	13				$Q_{18}^{[5]} = Q_{17}^{[5]}, Q_{18}^{[10]} = Q_{16}^{[10]}$
20	3			$\langle \blacktriangle^{[8]} \rangle$	$Q_{19}^{[10]} = Q_{18}^{[10]}$
21	5		$\langle \blacktriangledown^{[30]} \rangle$	$\langle \blacktriangle^{[15]} \rangle$	$Q_{19}^{[8]} = Q_{18}^{[8]}, Q_{20}^{[30]} \neq Q_{19}^{[30]}$
22	9			$\langle \blacktriangledown^{[7,8]} \rangle$	$Q_{21}^{[8]} = Q_{19}^{[8]}, Q_{20}^{[15]} = Q_{19}^{[15]}$
23	13				$Q_{21}^{[7]} = Q_{20}^{[7]}, Q_{22}^{[15]} = Q_{20}^{[15]}$
24	3		$\langle \blacktriangledown^{[8]} \rangle$		$Q_{23}^{[7]} = Q_{21}^{[7]}, Q_{23}^{[8]} \neq Q_{21}^{[8]}, Q_{23}^{[15]} = Q_{22}^{[15]}$
25	5			$\langle \blacktriangle^{[20]} \rangle$	$Q_{24}^{[7]} = Q_{23}^{[7]}, Q_{24}^{[8]} = Q_{23}^{[8]}$
26	9			$\langle \blacktriangledown^{[16]} \rangle$	$Q_{24}^{[20]} = Q_{23}^{[20]}$
27	13				$Q_{25}^{[16]} = Q_{24}^{[16]}, Q_{26}^{[20]} = Q_{24}^{[20]}$
28	3				$Q_{27}^{[16]} = Q_{25}^{[16]}, Q_{27}^{[20]} = Q_{26}^{[20]}$
29	5			$\langle \blacktriangle^{[25]} \rangle$	$Q_{28}^{[16]} = Q_{27}^{[16]}$
30	9			$\langle \blacktriangledown^{[25]} \rangle$	$Q_{28}^{[25]} = Q_{27}^{[25]}$
31	13				
32	3				
33	9		$\langle \blacktriangledown^{[25]} \rangle$		$Q_{32}^{[25]} = Q_{31}^{[25]}$
34	11	$\langle \blacktriangle^{[25]} \rangle$			
35	15				
36	3				
37	9				
38	11				

58 conditions : 20 + 37 + 1

step	message	$s_i$	$Q_i$
0	$m_0=00101001000000001100100101000010$	3	01001000000001100100101000001001
1	$m_1=\blacktriangle 1111110001000110001110111011011$	7	1000100100001000101010000 $\blacktriangle$ 110011
2	$m_2=\blacktriangledown\blacktriangledown\blacktriangle 1010011100001111011100110111$	11	100001101111110111110 $\blacktriangle$ 10 $\blacktriangledown$ 0001111
3	$m_3=00010010011110101001100001110111$	19	01100000000111100101001010010111
4	$m_4=11001000101100001011111001100100$	3	1100111010100000001 $\blacktriangledown\blacktriangle\blacktriangle$ 010100100
5	$m_5=01110110001111000101010101110110$	7	110100010111100000 $\blacktriangle$ 1110000011111
6	$m_6=10001101100111010011111011010110$	11	0000110001011 $\blacktriangledown$ 110100111110100110
7	$m_7=01011000110110010110001010110100$	19	10001010100010111101111101111110
8	$m_8=01101001001111100111011101101101$	3	100010101100111 $\blacktriangle$ 111 $\blacktriangledown$ 100111000100
9	$m_9=00010001000101001010100100011000$	7	100101000101 $\blacktriangle$ 0100100111010111000
10	$m_{10}=00111110001000011000111011001010$	11	01 $\blacktriangledown$ 00101110000011011011010101010
11	$m_{11}=01011000011010011101101100001110$	19	01010011110000111000101000100000
12	$m_{12}=00001101101100\blacktriangledown0101111110101111$	3	111100101 $\blacktriangle$ 01100 $\blacktriangledown$ 000000101011010
13	$m_{13}=00111000101001110001001011000110$	7	01 $\blacktriangle\blacktriangledown\blacktriangledown$ 01100011000000111100010010
14	$m_{14}=00011100001100010011010011011011$	11	11110011011001011011111010100101
15	$m_{15}=01100100100000101110110011010011$	19	0011001001101000 $\blacktriangle$ 100111100010100
16	$m_0=00101001000000001100100101000010$	3	010011 $\blacktriangle$ 0010000101001101001001111
17	$m_4=11001000101100001011111001100100$	5	$\blacktriangle$ 1100011111111001010001010011110
18	$m_8=01101001001111100111011101101101$	9	00011110011100111001001000110011
19	$m_{12}=00001101101100\blacktriangledown0101111110101111$	13	111 $\blacktriangledown$ 0111010011110111110100100001
20	$m_1=\blacktriangle 1111110001000110001110111011011$	3	$\blacktriangledown$ 11 $\blacktriangle$ 0011001111110001111110110100
21	$m_5=01110110001111000101010101110110$	5	10000111010100100001101111000101
22	$m_9=00010001000101001010100100011000$	9	11010011101010010001001011100010
23	$m_{13}=00111000101001110001001011000110$	13	10000100101011001000100110110110
24	$m_2=\blacktriangledown\blacktriangledown\blacktriangle 1010011100001111011100110111$	3	01111110110101010110001101010000
25	$m_6=10001101100111010011111011010110$	5	11000011110110110000010011001000
26	$m_{10}=00111110001000011000111011001010$	9	01010100001110100010101001100110
27	$m_{14}=00011100001100010011010011011011$	13	01101011010011010100101001000111
28	$m_3=00010010011110101001100001110111$	3	0111100101101011111110100110001
29	$m_7=01011000110110010110001010110100$	5	00010100010010010110111110011110
30	$m_{11}=01011000011010011101101100001110$	9	11011111110111000100100100000000
31	$m_{15}=01100100100000101110110011010011$	13	10000011101110000111000011110011
32	$m_0=00101001000000001100100101000010$	3	11001011101000000100010000001010
33	$m_8=01101001001111100111011101101101$	9	01001100101000010100101100001000
34	$m_4=11001000101100001011111001100100$	11	00000011100101111011000011100001
35	$m_{12}=00001101101100\blacktriangledown0101111110101111$	15	$\blacktriangledown$ 0111110000100110100001001101101
36	$m_2=\blacktriangledown\blacktriangledown\blacktriangle 1010011100001111011100110111$	3	$\blacktriangledown$ 0110000100001110000001100110011
37	$m_{10}=00111110001000011000111011001010$	9	01000001011011100110010000001101
38	$m_6=10001101100111010011111011010110$	11	01001000000001010101101010000000
39	$m_{14}=00011100001100010011010011011011$	15	01010000010100111000000110000101
40	$m_1=\blacktriangle 1111110001000110001110111011011$	3	10110101111001100101110110111111
41	$m_9=00010001000101001010100100011000$	9	00011010111111110000000011011110
42	$m_5=01110110001111000101010101110110$	11	00110011110000111101100101100011
43	$m_{13}=00111000101001110001001011000110$	15	10000001111101110100101001010111
44	$m_3=00010010011110101001100001110111$	3	00000000001100111010111000001111
45	$m_{11}=01011000011010011101101100001110$	9	10010100000010011001000100101000
46	$m_7=01011000110110010110001010110100$	11	00100100111010010100000010001010
47	$m_{15}=01100100100000101110110011010011$	15	11010001001111000000001100010011

TAB. 13: Détail d'une collision

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
0	3	$\langle \blacktriangle^{[0]} \rangle$		$\langle \blacktriangle^{[3]} \rangle$	
1	7				$Q_{-1}^{[3]} = Q_{-2}^{[3]}$
2	11				$Q_1^{[3]} = 0$
3	19				$Q_2^{[3]} = 1$
4	3			$\langle \blacktriangledown^{[6,7]} \rangle$	
5	7				$Q_3^{[6]} = Q_2^{[6]}, Q_3^{[7]} = Q_2^{[7]}$
6	11				$Q_5^{[6]} = 0, Q_5^{[7]} = 0$
7	19		$\langle \blacktriangle^{[7]} \rangle$	$\langle \blacktriangle^{[26]} \rangle$	$Q_6^{[6]} = 1, Q_6^{[7]} = 0$
8	3		$\langle \blacktriangledown^{[26]} \rangle$	$\langle \blacktriangle^{[9]}, \blacktriangledown^{[29]} \rangle$	$Q_5^{[26]} = 1, Q_6^{[26]} = 0$
9	7				$Q_7^{[9]} = Q_6^{[9]}, Q_8^{[26]} = 0, Q_7^{[29]} = Q_6^{[29]}$
10	11				$Q_9^{[9]} = 0, Q_9^{[26]} = 1, Q_9^{[29]} = 0$
11	19			$\langle \blacktriangle^{[13]} \rangle$	$Q_{10}^{[9]} = 1, Q_{10}^{[29]} = 1$
12	3			$\langle \blacktriangledown^{[0]}, \blacktriangle^{[12]} \rangle$	$Q_{10}^{[13]} = Q_9^{[13]}$
13	7				$Q_{11}^{[0]} = Q_{10}^{[0]}, Q_{11}^{[12]} = Q_{10}^{[12]}, Q_{11}^{[13]} = 0$
14	11		$\langle \blacktriangledown^{[0]} \rangle$	$\langle \blacktriangle\blacktriangle\blacktriangledown^{[11\dots13]} \rangle$	$Q_{13}^{[0]} = 1, Q_{13}^{[12]} = 0, Q_{13}^{[13]} = 1$
15	19		$\langle \blacktriangledown^{[13]} \rangle$		$Q_{14}^{[0]} = 1, Q_{13}^{[11]} = Q_{12}^{[11]}, Q_{13}^{[12]} = 0, Q_{13}^{[13]} = 1, Q_{12}^{[13]} = 0$
16	3	$\langle \blacktriangle^{[0]} \rangle$	$\langle \blacktriangle\blacktriangledown^{[12,13]} \rangle$		$Q_{15}^{[11]} = Q_{13}^{[11]}, Q_{15}^{[12]} \neq Q_{13}^{[12]}, Q_{15}^{[13]} \neq Q_{13}^{[13]}$
17	5				$Q_{16}^{[11]} = Q_{15}^{[11]}, Q_{16}^{[12]} = Q_{15}^{[12]}, Q_{16}^{[13]} = Q_{15}^{[13]}$
18	9			$\langle \blacktriangle\blacktriangle\blacktriangledown^{[20\dots23]} \rangle$	
19	13				$Q_{17}^{[20]} = Q_{16}^{[20]}, Q_{17}^{[21]} = Q_{16}^{[21]}, Q_{17}^{[22]} = Q_{16}^{[22]}, Q_{17}^{[23]} = Q_{16}^{[23]}$
20	3		$\langle \blacktriangledown^{[23]} \rangle$	$\langle \blacktriangledown^{[26]} \rangle$	$Q_{19}^{[20]} = Q_{17}^{[20]}, Q_{19}^{[21]} = Q_{17}^{[21]}, Q_{19}^{[22]} = Q_{17}^{[22]}, Q_{19}^{[23]} \neq Q_{17}^{[23]}$
21	5				$Q_{20}^{[20]} = Q_{19}^{[20]}, Q_{20}^{[21]} = Q_{19}^{[21]}, Q_{20}^{[22]} = Q_{19}^{[22]}, Q_{20}^{[23]} = Q_{19}^{[23]}, Q_{19}^{[26]} = Q_{19}^{[26]}$
22	9			$\langle \blacktriangledown^{[29]} \rangle$	$Q_{21}^{[26]} = Q_{19}^{[26]}$
23	13				$Q_{22}^{[26]} = Q_{21}^{[26]}, Q_{21}^{[29]} = Q_{20}^{[29]}$
24	3			$\langle \blacktriangle\blacktriangledown^{[29,30]} \rangle$	$Q_{23}^{[29]} = Q_{21}^{[29]}$
25	5				$Q_{23}^{[30]} = Q_{22}^{[30]}$
26	9		$\langle \blacktriangle^{[29]} \rangle$		$Q_{25}^{[29]} \neq Q_{23}^{[29]}, Q_{25}^{[30]} = Q_{23}^{[30]}$
27	13				$Q_{26}^{[29]} = Q_{25}^{[29]}, Q_{26}^{[30]} = Q_{25}^{[30]}$
28	3			$\langle \blacktriangledown^{[0]} \rangle$	
29	5				$Q_{27}^{[0]} = Q_{26}^{[0]}$
30	9				$Q_{29}^{[0]} = Q_{27}^{[0]}$
31	13				$Q_{30}^{[0]} = Q_{29}^{[0]}$
32	3	$\langle \blacktriangle^{[0]} \rangle$			

TAB. 14: Un chemin pour la première partie de l'attaque sur HMAC-MD4

step	$s_i$	$\delta m_i$	$\partial\Phi_i$	$\partial Q_i$	conditions
-4	0			$\langle \blacktriangle^{[30]} \rangle$	
-3	0				
-2	0				
-1	0				
0	3			$\langle \blacktriangle^{[1]} \rangle$	
1	7				$Q_{-1}^{[1]} = Q_{-2}^{[1]}$
2	11				$Q_1^{[1]} = 0$
3	19				$Q_2^{[1]} = 1$
4	3	$\langle \blacktriangle^{[22]} \rangle$		$\langle \blacktriangle^{[4]}, \blacktriangle^{[25]} \rangle$	
5	7				$Q_3^{[4]} = Q_2^{[4]}, Q_3^{[25]} = Q_2^{[25]}$
6	11				$Q_5^{[4]} = 0, Q_5^{[25]} = 0$
7	19				$Q_6^{[4]} = 1, Q_6^{[25]} = 1$
8	3			$\langle \blacktriangle^{[7]}, \blacktriangledown^{[28,29]} \rangle$	
9	7				$Q_7^{[7]} = Q_6^{[7]}, Q_7^{[28]} = Q_6^{[28]}, Q_7^{[29]} = Q_6^{[29]}$
10	11		$\langle \blacktriangledown^{[28]} \rangle$	$\langle \blacktriangle\blacktriangledown^{[7,8]} \rangle$	$Q_9^{[7]} = 0, Q_9^{[28]} = 1, Q_9^{[29]} = 0$
11	19				$Q_9^{[7]} = 0, Q_9^{[8]} = Q_8^{[8]}, Q_{10}^{[28]} = 1, Q_{10}^{[29]} = 1$
12	3		$\langle \blacktriangle\blacktriangledown^{[7,8]} \rangle$	$\langle \blacktriangle^{[31]} \rangle$	$Q_{11}^{[7]} = 1, Q_{11}^{[8]} = 1$
13	7				$Q_{12}^{[7]} = 1, Q_{12}^{[8]} = 1, Q_{11}^{[31]} = Q_{10}^{[31]}$
14	11			$\langle \blacktriangledown^{[18]} \rangle$	$Q_{13}^{[31]} = 0$
15	19				$Q_{13}^{[18]} = Q_{12}^{[18]}, Q_{14}^{[31]} = 1$
16	3			$\langle \blacktriangle^{[2]} \rangle$	$Q_{15}^{[18]} = Q_{13}^{[18]}$
17	5	$\langle \blacktriangle^{[22]} \rangle$	$\langle \blacktriangle^{[2]} \rangle$	$\langle \blacktriangle^{[7]}, \blacktriangle^{[27]} \rangle$	$Q_{15}^{[2]} \neq Q_{14}^{[2]}, Q_{16}^{[18]} = Q_{15}^{[18]}$
18	9			$\langle \blacktriangledown^{[27]} \rangle$	$Q_{17}^{[2]} = Q_{15}^{[2]}, Q_{16}^{[7]} = Q_{15}^{[7]}, Q_{16}^{[27]} = Q_{15}^{[27]}$
19	13				$Q_{18}^{[2]} = Q_{17}^{[2]}, Q_{18}^{[7]} = Q_{16}^{[7]}$
20	3			$\langle \blacktriangle^{[5]} \rangle$	$Q_{19}^{[7]} = Q_{18}^{[7]}$
21	5		$\langle \blacktriangledown^{[27]} \rangle$	$\langle \blacktriangle^{[12]} \rangle$	$Q_{19}^{[5]} = Q_{18}^{[5]}, Q_{20}^{[27]} \neq Q_{19}^{[27]}$
22	9			$\langle \blacktriangle\blacktriangledown^{[4,5]} \rangle$	$Q_{21}^{[5]} = Q_{19}^{[5]}, Q_{20}^{[12]} = Q_{19}^{[12]}$
23	13				$Q_{21}^{[4]} = Q_{20}^{[4]}, Q_{22}^{[12]} = Q_{20}^{[12]}$
24	3		$\langle \blacktriangledown^{[5]} \rangle$		$Q_{23}^{[4]} = Q_{21}^{[4]}, Q_{23}^{[5]} \neq Q_{21}^{[5]}, Q_{23}^{[12]} = Q_{22}^{[12]}$
25	5			$\langle \blacktriangle^{[17]} \rangle$	$Q_{24}^{[4]} = Q_{23}^{[4]}, Q_{24}^{[5]} = Q_{23}^{[5]}$
26	9			$\langle \blacktriangledown^{[13]} \rangle$	$Q_{24}^{[17]} = Q_{23}^{[17]}$
27	13				$Q_{25}^{[13]} = Q_{24}^{[13]}, Q_{26}^{[17]} = Q_{24}^{[17]}$
28	3				$Q_{27}^{[13]} = Q_{25}^{[13]}, Q_{27}^{[17]} = Q_{26}^{[17]}$
29	5			$\langle \blacktriangle^{[22]} \rangle$	$Q_{28}^{[13]} = Q_{27}^{[13]}$
30	9			$\langle \blacktriangledown^{[22]} \rangle$	$Q_{28}^{[22]} = Q_{27}^{[22]}$
31	13				
32	3				
33	9		$\langle \blacktriangledown^{[22]} \rangle$		$Q_{32}^{[22]} = Q_{31}^{[22]}$
34	11	$\langle \blacktriangle^{[22]} \rangle$			
35	15				
36	3				

TAB. 15: Un chemin pour la deuxième partie de l'attaque sur HMAC-MD4