

# Automatic Search of Differential Path in MD4

Pierre-Alain Fouque, Gaëtan Leurent, Phong Nguyen

Laboratoire d'Informatique de l'École Normale Supérieure,  
Département d'Informatique,  
45 rue d'Ulm, 75230 Paris Cedex 05, France

Ecrypt Hash Workshop, May 2007

# Motivation

## Why do we need an algorithm?

- Understanding
- Improving
- New attacks

## Results

- Some improvement of known attacks
- New attack against NMAC-MD4

# Outline

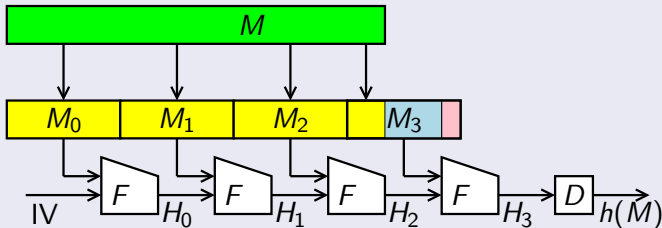
- 1 Introduction
  - The MD4 hash function
  - Wang's attack
- 2 Understand and automate
  - Sufficient conditions
    - Step operation
    - SC Algorithm
  - Differential Path
  - Message difference
- 3 Results
  - Collisions
  - Second preimage
  - NMAC Attack
- 4 Conclusion

# The MD4 hash function

## General design

### MD4 Design

- Merkle-Damgård
- Block size: 512 bits
- Internal state: 128 bits
- MD Strengthening



# The MD4 hash function

## Compression function

G. Leurent

Introduction

MD4

Wang's attack

Understand and automate

Sufficient conditions

Step operation

SC Algorithm

Differential Path

Message difference

Results

Collisions

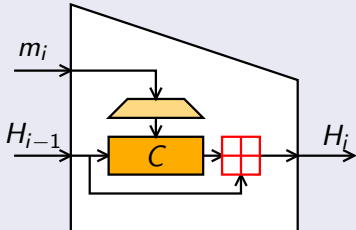
2<sup>nd</sup> preimage

NMAC Attack

Conclusion

## Compression Function Design

- Davies-Meyer with a Feistel-like cipher.

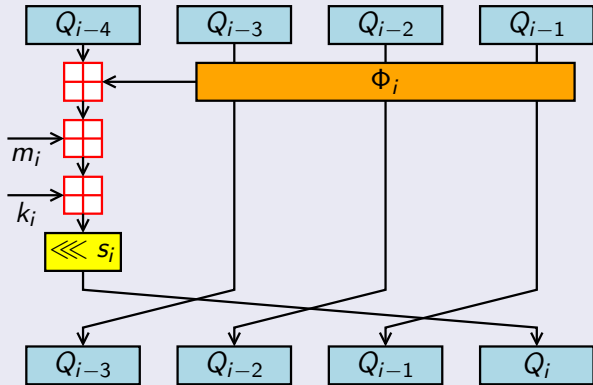


- Designed to be fast: 32 bit words, and operations available in hardware:
  - additions mod  $2^{32}$ :  $\boxplus$
  - boolean functions:  $\Phi_i$
  - rotations  $\lll s_i$
- Message expansion  $M = \langle M_0, \dots, M_{15} \rangle \mapsto \langle m_0, \dots, m_{47} \rangle$
- 4 words of internal state  $Q_i$  updated in rounds of 16 steps

# The MD4 hash function

## Compression function

### MD4 Step Update



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

# MD4 Collisions

Automatic  
Search of  
Differential  
Path in MD4

G. Leurent

Introduction

MD4

Wang's attack

Understand  
and automate

Sufficient  
conditions

Step operation

SC Algorithm

Differential Path

Message  
difference

Results

Collisions

2<sup>nd</sup> preimage

NMAC Attack

Conclusion

## Wang in a nutshell

- 1 Precomputation:
  - Choose a message difference.
  - Compute a differential path.
  - Derive a set of sufficient conditions.
- 2 Collision search:
  - Find a message that satisfies the set of conditions.

## Main result

We know a difference  $\Delta$  and a set of conditions on the internal state variables  $Q_i$ 's, such that:

*If all the conditions are satisfied by the internal state variable in the computation of  $H(M)$ ,  
then  $H(M) = H(M + \Delta)$ .*

# What is a differential path?

## Description

- Specifies how the computations of  $H(M)$  and  $H(M + \Delta)$  are related.
- The differences introduced in the message evolve in the internal state.
- Differential attack with the modular difference.
- Most of the work is modulo  $2^{32}$ , but we also need to control bit differences.

# What is a differential path?

## Notations

G. Leurent

### Notations

- Modular difference:  $\delta(x, y) = y \boxplus x$
- Wang's difference:  $\partial(x, y) = \langle y^{[31]} - x^{[31]}, \dots, y^{[0]} - x^{[0]} \rangle$
- $\blacktriangle$  and  $\blacktriangledown$  for  $+1$  and  $-1$ .
- $x^{[k]}$  for the  $k + 1$ -st bit of  $x$ .
- Compact notation:  $\langle \blacktriangle^{[0]}, \blacktriangledown^{[3,4]}, \blacktriangle\blacktriangle^{[30,31]} \rangle$

### Differential path notations

- We consider a message  $M$ .  $M' = M \boxplus \Delta$ .
- The differential path specifies  $\partial Q_i = \partial(Q_i, Q'_i)$ .
- The desired values are  $\partial_i$ .

Introduction

MD4

Wang's attack

Understand  
and automate

Sufficient  
conditions

Step operation

SC Algorithm

Differential Path

Message  
difference

Results

Collisions

2<sup>nd</sup> preimage

NMAC Attack

Conclusion

# Understanding Wang

Automatic  
Search of  
Differential  
Path in MD4

G. Leurent

Introduction

MD4

Wang's attack

Understand  
and automate

Sufficient  
conditions

Step operation

SC Algorithm

Differential Path

Message  
difference

Results

Collisions

2<sup>nd</sup> preimage

NMAC Attack

Conclusion

## Question

How to compute the set of conditions?

- 1 Derive a set of sufficient conditions from a differential path.
- 2 Compute a differential path from a message difference.
- 3 Choose a message difference.

- 1 Introduction
  - The MD4 hash function
  - Wang's attack
- 2 Understand and automate
  - Sufficient conditions
    - Step operation
    - SC Algorithm
  - Differential Path
  - Message difference
- 3 Results
  - Collisions
  - Second preimage
  - NMAC Attack
- 4 Conclusion

# Sufficient conditions computations

## Goal

- We are given a differential path  $\langle \partial_i \rangle$ .
- We want to compute a set of conditions so that:

*If  $Q(M)$  satisfies the conditions,  
then  $Q(M)$  and  $Q(M')$  follows the path.*

## Strategy

- We will iteratively add conditions for the current state, assuming the previous ones are satisfied.
- First, study the step operation and the  $\partial$ -difference. (Differential attack)

# Remarks about the $\partial$ -difference

## The $\delta$ -difference and the $\partial$ -difference

- If we know  $\partial(x, y)$ , we can compute  $\delta(x, y)$ .
- If we know  $\delta(x, y)$ , many  $\partial(x, y)$  are possible.

For instance, if  $\delta(x, y) = 2^k$ ,  $33 - k$  possibilities:

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

# Remarks about the $\partial$ -difference

## Theorem

$$\partial(x, y) = \langle \varepsilon_{31}, \varepsilon_{30}, \dots, \varepsilon_0 \rangle \iff \begin{cases} \sum_{j=0}^{31} \varepsilon_j 2^j = \delta(x, y) \\ \forall j, \varepsilon_j \in \{-1, 0, +1\} \\ \forall j : \varepsilon_j = +1 \implies x^{[j]} = 0 \\ \forall j : \varepsilon_j = -1 \implies x^{[j]} = 1 \end{cases}$$

- If we know  $\delta(x, y)$ , we can fix one  $\partial(x, y)$  by adding some conditions on  $x$ .
- We can switch between  $\delta$ -difference and  $\partial$ -difference.

# Rotation and modular difference

## Four cases

- We have an algebraic expression of the rotation:

$$u \lll s = \lfloor \frac{u}{2^{32-s}} \rfloor + (2^s u \bmod 2^{32})$$

- We can express  $v = \delta(a \lll s, b \lll s)$  from  $u = \delta(a, b)$

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

→ bit conditions, probabilities

# Rotation and modular difference

## Four cases

- We have an algebraic expression of the rotation:

$$u \lll s = \lfloor \frac{u}{2^{32-s}} \rfloor + (2^s u \bmod 2^{32})$$

- We can express  $v = \delta(a \lll s, b \lll s)$  from  $u = \delta(a, b)$

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

→ bit conditions, probabilities

G. Leurent

Introduction

MD4

Wang's attack

Understand  
and automate

Sufficient  
conditions

**Step operation**

SC Algorithm

Differential Path

Message  
difference

Results

Collisions

2<sup>nd</sup> preimage

NMAC Attack

Conclusion

## Important remark

- The conditions are on the input (or output) of the rotation.
- In MD4, we will use this backwards:

$$Q_{i+4} = (Q_i \boxplus \Phi_{i+4} \boxplus m_{i+4} \boxplus k_{i+4}) \lll S_{i+4}$$

# Wang difference and Boolean functions

Automatic  
Search of  
Differential  
Path in MD4

G. Leurent

Introduction  
MD4  
Wang's attack

Understand  
and automate

Sufficient  
conditions

Step operation  
SC Algorithm  
Differential Path

Message  
difference

Results

Collisions  
2<sup>nd</sup> preimage  
NMAC Attack

Conclusion

## The Boolean function

- Bitwise Boolean functions:

- First round:

$$F(x, y, z) = (x \wedge y) \vee (\neg x \wedge z)$$

- Second round:

$$G(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$$

- Third round:

$$H(x, y, z) = x \oplus y \oplus z$$

- For each bit, if we know the input differences we can add conditions to select one output difference.
- Motivation for  $\partial$ -difference.

# $\Phi_i$ conditions

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

# Step operations summary

G. Leurent

Introduction

MD4

Wang's attack

Understand  
and automate

Sufficient  
conditions

**Step operation**

SC Algorithm

Differential Path

Message  
difference

Results

Collisions

2<sup>nd</sup> preimage

NMAC Attack

Conclusion

For each operation, we can add conditions on  $Q_i$  to make it behave nicely.  
→ Sufficient conditions algorithm.

# Computing sufficient conditions

## Goal

At step  $i + 4$ , we have:

$$Q_{i+4} = (Q_i \boxplus \Phi_{i+4}(Q_{i+1}, Q_{i+2}, Q_{i+3}) \boxplus m_{i+4} \boxplus k_{i+4}) \lll s_{i+4}$$

$$Q'_{i+4} = (Q'_i \boxplus \Phi_{i+4}(Q'_{i+1}, Q'_{i+2}, Q'_{i+3}) \boxplus m'_{i+4} \boxplus k_{i+4}) \lll s_{i+4}$$

We want  $\partial(Q_i, Q'_i) = \partial_i$ .

## Part one: $\delta(Q_i, Q'_i) = \delta_i$

- Choose  $\delta_{i+4}^{\ggg} = \delta(Q_{i+4} \ggg s_{i+4}, Q'_{i+4} \ggg s_{i+4})$   
that match  $\delta_{i+4} = \delta(Q_{i+4}, Q'_{i+4})$ .  
→  $\lll$ -conditions on  $Q_{i+4}$ .
- We just need  $\Phi'_{i+4} \boxplus \Phi_{i+4} = \delta_i \boxplus \delta_{i+4}^{\ggg} \boxplus \Delta_{i+4}$ .  
Choose  $\partial(\Phi_{i+4}, \Phi'_{i+4})$ .  
→  $\Phi$ -conditions on  $Q_{i+1}, Q_{i+2}, Q_{i+3}$

## Part two: $\partial(Q_i, Q'_i) = \partial_i$

→  $\partial$ -conditions on  $Q_i$

## Computing sufficient conditions

## Goal

At step  $i + 4$ , we have:

$$Q_{i+4} = (Q_i \boxplus \Phi_{i+4}(Q_{i+1}, Q_{i+2}, Q_{i+3}) \boxplus m_{i+4} \boxplus k_{i+4}) \lll s_{i+4}$$

$$Q'_{i+4} = (Q'_i \boxplus \Phi_{i+4}(Q'_{i+1}, Q'_{i+2}, Q'_{i+3}) \boxplus m'_{i+4} \boxplus k_{i+4}) \lll s_{i+4}$$

We want  $\partial(Q_i, Q'_i) = \partial_i$ .

Part one:  $\delta(Q_i, Q'_i) = \delta_i$ 

- **Choose**  $\delta_{i+4}^{\ggg} = \delta(Q_{i+4} \ggg s_{i+4}, Q'_{i+4} \ggg s_{i+4})$   
that match  $\delta_{i+4} = \delta(Q_{i+4}, Q'_{i+4})$ .  
→  $\lll$ -conditions on  $Q_{i+4}$ .
- We just need  $\Phi'_{i+4} \boxplus \Phi_{i+4} = \delta_i \boxplus \delta_{i+4}^{\ggg} \boxplus \Delta_{i+4}$ .  
**Choose**  $\partial(\Phi_{i+4}, \Phi'_{i+4})$ .  
→  $\Phi$ -conditions on  $Q_{i+1}, Q_{i+2}, Q_{i+3}$

Part two:  $\partial(Q_i, Q'_i) = \partial_i$ 

→  $\partial$ -conditions on  $Q_i$

# Computing sufficient conditions

## Goal

At step  $i + 4$ , we have:

$$Q_{i+4} = (Q_i \boxplus \Phi_{i+4}(Q_{i+1}, Q_{i+2}, Q_{i+3}) \boxplus m_{i+4} \boxplus k_{i+4}) \lll s_{i+4}$$

$$Q'_{i+4} = (Q'_i \boxplus \Phi_{i+4}(Q'_{i+1}, Q'_{i+2}, Q'_{i+3}) \boxplus m'_{i+4} \boxplus k_{i+4}) \lll s_{i+4}$$

We want  $\partial(Q_i, Q'_i) = \partial_i$ .

## Part one: $\delta(Q_i, Q'_i) = \delta_i$

- Choose  $\delta_{i+4}^{\ggg} = \delta(Q_{i+4} \ggg s_{i+4}, Q'_{i+4} \ggg s_{i+4})$   
that match  $\delta_{i+4} = \delta(Q_{i+4}, Q'_{i+4})$ .  
→  **$\lll$ -conditions on  $Q_{i+4}$ .**
- We just need  $\Phi'_{i+4} \boxplus \Phi_{i+4} = \delta_i \boxplus \delta_{i+4}^{\ggg} \boxplus \Delta_{i+4}$ .  
Choose  $\partial(\Phi_{i+4}, \Phi'_{i+4})$ .  
→  **$\Phi$ -conditions on  $Q_{i+1}, Q_{i+2}, Q_{i+3}$**

## Part two: $\partial(Q_i, Q'_i) = \partial_i$

→  **$\partial$ -conditions on  $Q_i$**

# SC Algorithm

Automatic  
Search of  
Differential  
Path in MD4

G. Leurent

Introduction

MD4

Wang's attack

Understand  
and automate

Sufficient  
conditions

Step operation

**SC Algorithm**

Differential Path

Message  
difference

Results

Collisions

2<sup>nd</sup> preimage

NMAC Attack

Conclusion

## Result

- SC Algorithm works
- Next step: how to compute the differential path?

# Absorbing the differences

G. Leurent

Introduction

MD4

Wang's attack

Understand and automate

Sufficient conditions

Step operation

SC Algorithm

Differential Path

Message difference

Results

Collisions

2<sup>nd</sup> preimage

NMAC Attack

Conclusion

## Important observation

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

$$Q_{i+1} = (Q_{i-3} \boxplus \Phi_{i+1}(Q_i, Q_{i-1}, Q_{i-2}) \boxplus m_{i+1} \boxplus k_{i+1}) \lll s_{i+1}$$

$$Q_{i+2} = (Q_{i-2} \boxplus \Phi_{i+2}(Q_{i+1}, Q_i, Q_{i-1}) \boxplus m_{i+2} \boxplus k_{i+2}) \lll s_{i+2}$$

$$Q_{i+3} = (Q_{i-1} \boxplus \Phi_{i+3}(Q_{i+2}, Q_{i+1}, Q_i) \boxplus m_{i+3} \boxplus k_{i+3}) \lll s_{i+3}$$

$$Q_{i+4} = (Q_i \boxplus \Phi_{i+4}(Q_{i+3}, Q_{i+2}, Q_{i+1}) \boxplus m_{i+4} \boxplus k_{i+4}) \lll s_{i+4}$$

$$Q_{i+5} = (Q_{i+1} \boxplus \Phi_{i+5}(Q_{i+4}, Q_{i+3}, Q_{i+2}) \boxplus m_{i+5} \boxplus k_{i+5}) \lll s_{i+5}$$

- We introduce a difference in  $Q_i$ .
- If  $\Phi_i$  can absorb the difference, it will not multiply.
- It only appears every 4 round, with a rotation.

## The trivial path

This is the basis for MD4 differential paths:  
absorb the message differences

# Absorbing the differences

## Important observation

$$\begin{aligned} Q_i &= (Q_{i-4} \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}) \boxplus m_i \boxplus k_i) \lll S_i \\ Q_{i+1} &= (Q_{i-3} \boxplus \Phi_{i+1}(Q_i, Q_{i-1}, Q_{i-2}) \boxplus m_{i+1} \boxplus k_{i+1}) \lll S_{i+1} \\ Q_{i+2} &= (Q_{i-2} \boxplus \Phi_{i+2}(Q_{i+1}, Q_i, Q_{i-1}) \boxplus m_{i+2} \boxplus k_{i+2}) \lll S_{i+2} \\ Q_{i+3} &= (Q_{i-1} \boxplus \Phi_{i+3}(Q_{i+2}, Q_{i+1}, Q_i) \boxplus m_{i+3} \boxplus k_{i+3}) \lll S_{i+3} \\ Q_{i+4} &= (Q_i \boxplus \Phi_{i+4}(Q_{i+3}, Q_{i+2}, Q_{i+1}) \boxplus m_{i+4} \boxplus k_{i+4}) \lll S_{i+4} \\ Q_{i+5} &= (Q_{i+1} \boxplus \Phi_{i+5}(Q_{i+4}, Q_{i+3}, Q_{i+2}) \boxplus m_{i+5} \boxplus k_{i+5}) \lll S_{i+5} \end{aligned}$$

- We introduce a difference in  $Q_i$ .
- If  $\Phi_i$  can absorb the difference, it will not multiply.
- It only appears every 4 round, with a rotation.

## The trivial path

This is the basis for MD4 differential paths:  
absorb the message differences

# Absorbing the differences

G. Leurent

Introduction  
MD4  
Wang's attack

Understand and automate

Sufficient conditions  
Step operation  
SC Algorithm  
Differential Path  
Message difference

Results

Collisions  
2<sup>nd</sup> preimage  
NMAC Attack

Conclusion

## Important observation

$$\begin{aligned}
 Q_i &= (Q_{i-4} \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}) \boxplus m_i \boxplus k_i) \lll S_i \\
 Q_{i+1} &= (Q_{i-3} \boxplus \Phi_{i+1}(Q_i, Q_{i-1}, Q_{i-2}) \boxplus m_{i+1} \boxplus k_{i+1}) \lll S_{i+1} \\
 Q_{i+2} &= (Q_{i-2} \boxplus \Phi_{i+2}(Q_{i+1}, Q_i, Q_{i-1}) \boxplus m_{i+2} \boxplus k_{i+2}) \lll S_{i+2} \\
 Q_{i+3} &= (Q_{i-1} \boxplus \Phi_{i+3}(Q_{i+2}, Q_{i+1}, Q_i) \boxplus m_{i+3} \boxplus k_{i+3}) \lll S_{i+3} \\
 Q_{i+4} &= (Q_i \boxplus \Phi_{i+4}(Q_{i+3}, Q_{i+2}, Q_{i+1}) \boxplus m_{i+4} \boxplus k_{i+4}) \lll S_{i+4} \\
 Q_{i+5} &= (Q_{i+1} \boxplus \Phi_{i+5}(Q_{i+4}, Q_{i+3}, Q_{i+2}) \boxplus m_{i+5} \boxplus k_{i+5}) \lll S_{i+5}
 \end{aligned}$$

- We introduce a difference in  $Q_i$ .
- If  $\Phi_i$  can absorb the difference, it will not multiply.
- It only appears every 4 round, with a rotation.

## The trivial path

This is the basis for MD4 differential paths:  
absorb the message differences

# Absorbing the differences

G. Leurent

Introduction

MD4

Wang's attack

Understand and automate

Sufficient conditions

Step operation

SC Algorithm

Differential Path

Message difference

Results

Collisions

2<sup>nd</sup> preimage

NMAC Attack

Conclusion

## Important observation

$$\begin{aligned}
 Q_i &= (Q_{i-4} \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}) \boxplus m_i \boxplus k_i) \lll S_i \\
 Q_{i+1} &= (Q_{i-3} \boxplus \Phi_{i+1}(Q_i, Q_{i-1}, Q_{i-2}) \boxplus m_{i+1} \boxplus k_{i+1}) \lll S_{i+1} \\
 Q_{i+2} &= (Q_{i-2} \boxplus \Phi_{i+2}(Q_{i+1}, Q_i, Q_{i-1}) \boxplus m_{i+2} \boxplus k_{i+2}) \lll S_{i+2} \\
 Q_{i+3} &= (Q_{i-1} \boxplus \Phi_{i+3}(Q_{i+2}, Q_{i+1}, Q_i) \boxplus m_{i+3} \boxplus k_{i+3}) \lll S_{i+3} \\
 Q_{i+4} &= (Q_i \boxplus \Phi_{i+4}(Q_{i+3}, Q_{i+2}, Q_{i+1}) \boxplus m_{i+4} \boxplus k_{i+4}) \lll S_{i+4} \\
 Q_{i+5} &= (Q_{i+1} \boxplus \Phi_{i+5}(Q_{i+4}, Q_{i+3}, Q_{i+2}) \boxplus m_{i+5} \boxplus k_{i+5}) \lll S_{i+5}
 \end{aligned}$$

- We introduce a difference in  $Q_i$ .
- If  $\Phi_i$  can absorb the difference, it will not multiply.
- It only appears every 4 round, with a rotation.

## The trivial path

This is the basis for MD4 differential paths:  
absorb the message differences

# Absorbing the differences

G. Leurent

Introduction

MD4

Wang's attack

Understand and automate

Sufficient conditions

Step operation

SC Algorithm

Differential Path

Message difference

Results

Collisions

2<sup>nd</sup> preimage

NMAC Attack

Conclusion

## Important observation

$$\begin{aligned}
 Q_i &= (Q_{i-4} \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}) \boxplus m_i \boxplus k_i) \lll S_i \\
 Q_{i+1} &= (Q_{i-3} \boxplus \Phi_{i+1}(Q_i, Q_{i-1}, Q_{i-2}) \boxplus m_{i+1} \boxplus k_{i+1}) \lll S_{i+1} \\
 Q_{i+2} &= (Q_{i-2} \boxplus \Phi_{i+2}(Q_{i+1}, Q_i, Q_{i-1}) \boxplus m_{i+2} \boxplus k_{i+2}) \lll S_{i+2} \\
 Q_{i+3} &= (Q_{i-1} \boxplus \Phi_{i+3}(Q_{i+2}, Q_{i+1}, Q_i) \boxplus m_{i+3} \boxplus k_{i+3}) \lll S_{i+3} \\
 Q_{i+4} &= (Q_i \boxplus \Phi_{i+4}(Q_{i+3}, Q_{i+2}, Q_{i+1}) \boxplus m_{i+4} \boxplus k_{i+4}) \lll S_{i+4} \\
 Q_{i+5} &= (Q_{i+1} \boxplus \Phi_{i+5}(Q_{i+4}, Q_{i+3}, Q_{i+2}) \boxplus m_{i+5} \boxplus k_{i+5}) \lll S_{i+5}
 \end{aligned}$$

- We introduce a difference in  $Q_i$ .
- If  $\Phi_i$  can absorb the difference, it will not multiply.
- It only appears every 4 round, with a rotation.

## The trivial path

This is the basis for MD4 differential paths:  
absorb the message differences.

# Absorbing the differences

## MD4 Boolean functions

$$F(x, y, z) = (x \wedge y) \vee (\neg x \wedge z)$$

MD4 Boolean function  $F$  can absorb one input difference:

$$F(x, y, z) = IF(x, y, z)$$

$\partial x$	$\partial y$	$\partial z$	$\partial F = 0$	$\partial F = 1$	$\partial F = -1$
0	0	0	✓	✗	✗
0	0	+1	$x = 1$	$x = 0$	✗
0	0	-1	$x = 1$	✗	$x = 0$
0	+1	0	$x = 0$	$x = 1$	✗
0	-1	0	$x = 0$	✗	$x = 1$
+1	0	0	$y = z$	$y, z = 1, 0$	$y, z = 0, 1$
-1	0	0	$y = z$	$y, z = 0, 1$	$y, z = 1, 0$

# Absorbing the differences

## MD4 Boolean functions

$$G(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$$

MD4 Boolean function  $G$  can absorb one input difference:

$$G(x, y, z) = \text{MAJ}(x, y, z)$$

$\partial x$	$\partial y$	$\partial z$	$\partial G = 0$	$\partial G = 1$	$\partial G = -1$
0	0	0	✓	✗	✗
0	0	+1	$x = y$	$x \neq y$	✗
0	0	-1	$x = y$	✗	$x \neq y$
0	+1	0	$x = z$	$x \neq z$	✗
0	-1	0	$x = z$	✗	$x \neq z$
+1	0	0	$y = z$	$y \neq z$	✗
-1	0	0	$y = z$	✗	$y \neq z$

# Absorbing the differences

## MD4 Boolean functions

$$H(x, y, z) = x \oplus y \oplus z$$

MD4 Boolean function H can **not** absorb one input difference:

$$H(x, y, z) = x \oplus y \oplus z$$

$\partial x$	$\partial y$	$\partial z$	$\partial H = 0$	$\partial H = 1$	$\partial H = -1$
0	0	0	✓	✗	✗
0	0	+1	✗	$x = y$	$x \neq y$
0	0	-1	✗	$x \neq y$	$x = y$
0	+1	0	✗	$x = z$	$x \neq z$
0	-1	0	✗	$x \neq z$	$x = z$
+1	0	0	✗	$y = z$	$y \neq z$
-1	0	0	✗	$y \neq z$	$y = z$

Note: Wang use a local collision in round 3,  
no need to search path.

# Differential Path Search

Automatic  
Search of  
Differential  
Path in MD4

G. Leurent

Introduction

MD4

Wang's attack

Understand  
and automate

Sufficient  
conditions

Step operation

SC Algorithm

Differential Path

Message  
difference

Results

Collisions

2<sup>nd</sup> preimage

NMAC Attack

Conclusion

## Basic Idea

- Follow the sufficient conditions algorithm.
- $Q_{i+4} = (Q_i \boxplus \Phi_{i+4} \boxplus m_{i+4} \boxplus k_{i+4}) \lll S_{i+4}$   
 $Q'_{i+4} = (Q'_i \boxplus \Phi'_{i+4} \boxplus m'_{i+4} \boxplus k_{i+4}) \lll S_{i+4}$
- We do not know  $\partial Q_i$ , so we assume  $\Phi'_i = \Phi_i$ ,  
ie. absorb the difference.  $\rightarrow \delta_{i+4}^{\ggg} = \delta_i$ .
- Goes from the last step to the first.
- When we have a path up to the first round,  
there might be a difference in the IV, we will fix it later.

# Differential Path Search

## Turning pseudo-collision path into collision path

- We run the algorithm again, using the previous path as a hint for the values of  $\delta\Phi_i$ .
- We try to modify the path on the bits that will become the IV differences.

## Path representation

- During the computation, the path is represented by  $\partial_i$ 's.
- To modify the path later, we will rather use the  $\delta\Phi_i$ 's.

## Pseudo-code

```
1: function PATHFIND
2:    $\mathcal{P} \leftarrow \{\epsilon\}$ 
3:   loop
4:     extract  $P$  from  $\mathcal{P}$ 
5:     PATHSTEP( $P, \epsilon, 48$ )
6:   function PATHSTEP( $P_0, P, i$ )
7:     if  $i < 0$  then
8:       add  $P$  in  $\mathcal{P}$ 
9:     else
10:      for all possible choice  $P'$  do
11:        PATCHTARGET( $P_0, P', i$ )
12:      function PATCHTARGET( $P_0, P, i$ )
13:        for all possible choice  $P'$  do
14:          PATCHCARRIES( $P_0, P', i$ )
15:        function PATCHCARRIES( $P_0, P, i$ )
16:          for all possible choice  $P'$  do
17:            PATHSTEP( $P_0, P', i - 1$ )
```

## Pseudo-code

```
1: function PATHFIND
2:    $\mathcal{P} \leftarrow \{\epsilon\}$ 
3:   loop
4:     extract  $P$  from  $\mathcal{P}$ 
5:     PATHSTEP( $P, \epsilon, 48$ )
6:   function PATHSTEP( $P_0, P, i$ )
7:     if  $i < 0$  then
8:       add  $P$  in  $\mathcal{P}$ 
9:     else
10:      for all possible choice  $P'$  do
11:        PATCHTARGET( $P_0, P', i$ )
12:   function PATCHTARGET( $P_0, P, i$ )
13:     for all possible choice  $P'$  do
14:       PATCHCARRIES( $P_0, P', i$ )
15:   function PATCHCARRIES( $P_0, P, i$ )
16:     for all possible choice  $P'$  do
17:       PATHSTEP( $P_0, P', i - 1$ )
```

### PATHFIND

- Starts with the trivial path
- Pick a path and try to improve it

# Pseudo-code

```
1: function PATHFIND
2:    $\mathcal{P} \leftarrow \{\epsilon\}$ 
3:   loop
4:     extract  $P$  from  $\mathcal{P}$ 
5:     PATHSTEP( $P, \epsilon, 48$ )
6:   function PATHSTEP( $P_0, P, i$ )
7:     if  $i < 0$  then
8:       add  $P$  in  $\mathcal{P}$ 
9:     else
10:      for all possible choice  $P'$  do
11:        PATCHTARGET( $P_0, P', i$ )
12:      function PATCHTARGET( $P_0, P', i$ )
13:        for all possible choice  $P''$  do
14:          PATCHCARRIES( $P_0, P', P'', i$ )
15:        function PATCHCARRIES( $P_0, P', P'', i$ )
16:          for all possible choice  $P'''$  do
17:            PATHSTEP( $P_0, P'', i - 1, P'''$ )
```

## PATHSTEP

- Choose  $\delta_{i+4}^{\ggg}$  from  $\delta_{i+4}$  and  $\partial\Phi_{i+4}$  from  $\delta\Phi_{i+4}$
- Compute  $\delta Q_i$  from  $\delta_{i+4}^{\ggg}$  and  $\partial\Phi_{i+4}$

# Pseudo-code

```
1: function PATHFIND
2:    $\mathcal{P} \leftarrow \{\epsilon\}$ 
3:   loop
4:     extract  $P$  from  $\mathcal{P}$ 
5:     PATHSTEP( $P, \epsilon, 48$ )
6:   function PATHSTEP( $P_0, P, i$ )
7:     if  $i < 0$  then
8:       add  $P$  in  $\mathcal{P}$ 
9:     else
10:      for all possible choice  $P'$  do
11:        PATCHTARGET( $P_0, P', i$ )
12:      function PATCHTARGET( $P_0, P, i$ )
13:        for all possible choice  $P'$  do
14:          PATCHCARRIES( $P_0, P', i$ )
15:        function PATCHCARRIES( $P_0, P, i$ )
16:          for all possible choice  $P'$  do
17:            PATHSTEP( $P_0, P', i - 1$ )
```

## PATCHTARGET

- Modify  $\partial\Phi_i$   
from the path  $P$ .

# Pseudo-code

```
1: function PATHFIND
2:    $\mathcal{P} \leftarrow \{\epsilon\}$ 
3:   loop
4:     extract  $P$  from  $\mathcal{P}$ 
5:     PATHSTEP( $P, \epsilon, 48$ )
6: function PATHSTEP( $P_0, P, i$ )
7:   if  $i < 0$  then
8:     add  $P$  in  $\mathcal{P}$ 
9:   else
10:    for all possible choice  $P'$  do
11:      PATCHTARGET( $P_0, P', i$ )
12: function PATCHTARGET( $P_0, P, i$ )
13:   for all possible choice  $P'$  do
14:     PATCHCARRIES( $P_0, P', i$ )
15: function PATCHCARRIES( $P_0, P, i$ )
16:   for all possible choice  $P'$  do
17:     PATHSTEP( $P_0, P', i - 1$ )
```

## PATCHCARRIES

- Choose  $\partial Q_i$   
from  $\delta Q_i$

# Correcting the differences

## Direct correction

- $Q_i = (Q_{i-4} \boxplus \Phi_i \boxplus m_i \boxplus k_i) \lll s_i$
- Differences do not multiply: each difference in the IV has to be fixed in exactly one place.
- Possible places: every 4 rounds.
- We use  $\Phi_i$  to modify the bit.

## Indirect Corrections

- $Q_{i+a} = (Q_{i+a-4} \boxplus \Phi_{i+a}(Q_i) \boxplus m_i \boxplus k_i) \lll s_i$
- $Q_i = (Q_{i-4} \boxplus \Phi_i \boxplus m_i \boxplus k_i) \lll s_i$
- We use  $Q_i$  to modify  $Q_{i+a-4}$ .
- This introduces a new difference in  $Q_{i-4}$ .
- Hopefully, the new difference is easier to remove...

# Message difference

## Message difference

- We can try many message differences and run the algorithm
- Interesting message differences depend on the application...

# Overview of the algorithm

## Advantages of indirect corrections

- No need to manually add some differences.
- Use freedom in  $\Phi$  rather than carry expansions.
- Fewer conditions.

## Adaptation to MD5?

- $Q_i = Q_{i-1} \boxplus (Q_{i-4} \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}) \boxplus m_i \boxplus k_i) \lll s_i$
- No easy way to stop difference multiplications.  
Use den Boer-Bosselaers's path?
- No easy way to express the rotation conditions.

## 1 Introduction

- The MD4 hash function
- Wang's attack

## 2 Understand and automate

- Sufficient conditions
  - Step operation
  - SC Algorithm
- Differential Path
- Message difference

## 3 Results

- Collisions
- Second preimage
- NMAC Attack

## 4 Conclusion

# Collisions

## Collision path

- We want to minimize the search complexity
- Few conditions in 3<sup>rd</sup> (and 2<sup>nd</sup>) round: local collision.
- Our algorithm works with Wang's message difference, not (yet?) with Sasaki *et al.*'s.

## Comparison of collision paths

Number of conditions	round 1	round 2	round 3	total
<i>With Wang's message difference:</i>				
Wang <i>et al.</i>	96	25	2	123
Schläffer and Oswald	122	22	2	146
Our path	72	16	2	90
<i>With Sasaki's message difference:</i>				
Sasaki <i>et al.</i>	167	9	1	177

# Second preimage

## Second preimage paths

- Second preimage for weak message
- If  $c$  conditions, a message is weak with probability  $2^{-c}$
- We want to minimize the number of conditions

## Results on Yu's path

- Yu *et al.* gave a path with one bit difference in  $m_4$
- Authors claim 32 path using rotations of the path.  
**Actually, only 28 paths (fails on bit 17,20,26 and 28).**
- Using bit 25, **only 58 conditions** instead of 62.  
Good if you need only one path with very few conditions  
(eg. Contini Yin HMAC-MD4 attacks).

# A New NMAC Attack

## Main idea

- We search for a differential path with the message difference in  $m_0$ :

step	$s_i$	$\delta m_i$	$\partial \Phi_i$	$\partial Q_i$	conditions
0	3	$\langle \triangle^{[0]} \rangle$		$\langle \triangle^{[3]} \rangle$	
1	7				$Q_{-1}^{[3]} = Q_{-2}^{[3]} \text{ (X)}$
2	11				$Q_1^{[3]} = 0$
3	19				$Q_2^{[3]} = 1$
4	3			$\langle \triangle^{[6]} \rangle$	

- The beginning of the path depends on a condition (X) of the IV.
- $\Pr[H(M) = H(M + \Delta) | X] = p \gg 2^{-128}$ .
- $\Pr[H(M) = H(M + \Delta) | \neg X] \approx 2^{-128}$ .
- We learn one bit of the IV with about  $2/p$  message pairs.

# A New NMAC Attack

How to recover the outer key

## NMAC Description

- $\text{NMAC}_{k_1, k_2}(M) = H_{k_1}(H_{k_2}(M))$
- To recover  $k_1$ , we have to control  $H_{k_2}(M)$ .
- We need about  $2/p$  message pairs such that  $H_{k_2}(M_2) = H_{k_2}(M_1) + \Delta$ .
- $\Delta$  must be only in the first 128 bits.
- We can use the birthday paradox:  
we need to hash about  $2^{\frac{n-\log p}{2}}$  messages.

## Advantage

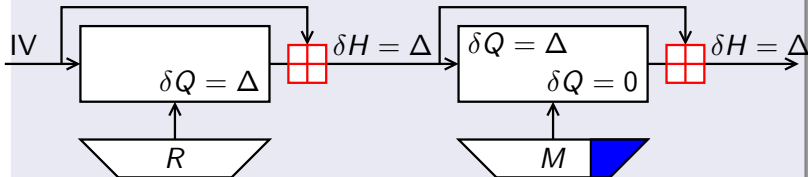
- In Contini-Yin attack, you need to control the **value** of  $H_{k_2}(M)$  (related messages).
- We only need to control the **differences** of  $H_{k_2}(M)$ .

# A New NMAC Attack

How to recover the outer key

## Efficient computation of message pairs

- We start with *one* message pair  $(R_1, R_2)$  such that  $H_{k_2}(R_2) = H_{k_2}(R_1) + \Delta$  (birthday paradox).
- We compute second blocks  $(M_1, M_2)$  such that  $H_{k_2}(R_2||M_2) = H_{k_2}(R_1||M_1) + \Delta$
- This is essentially a collision search with the padding inside the block.



# The Attack against NMAC-MD4

## Differential paths

- We need paths with a difference in  $m_0$  and no difference in  $m_4 \dots m_{15}$ .
- We found 22 paths with one bit difference in  $m_0$  and  $p \approx 2^{-79}$ .
- Unlikely to find such paths in MD5.

## Complexity

- We can recover the full NMAC key  $(k_1, k_2)$
- $2^{88}$  online request to the NMAC oracle.
- $2^{105}$  offline hash computations.  
 $2^{94}$  by using more than one bit of information per path.

G. Leurent

Introduction  
MD4  
Wang's attack

Understand  
and automate

Sufficient  
conditions  
Step operation  
SC Algorithm  
Differential Path  
Message  
difference

Results

Collisions  
2<sup>nd</sup> preimage  
NMAC Attack

Conclusion

## Improving the algorithm

- Using ideas from Stevens *et al.* and Sasaki *et al.*...

## Other uses

- Try to find new kind of attack based on new types of path...