MD4

MD4

- ***Message Digest 4***
- Invented by Rivest, ca 1990
- Weaknesses found by 1992
  - Rivest proposed improved version (MD5), 1992
- Dobbertin found 1st MD4 collision in 1998
  - Clever and efficient attack
  - Nonlinear equation solving and differential

MD4 Algorithm

- Assumes 32-bit **words**
- Little-endian convention
  - Leftmost byte is low-order (relevant when generating "meaningful" collisions)
- Let M be message to hash
- Pad M so length is 448 (mod 512)
  - Single "1" bit followed by "0" bits
  - At least one bit of padding, at most 512
  - Length before padding (64 bits) is appended

MD4 Algorithm

- After padding message is a multiple of the 512-bit **block** size
  - Also a multiple of 32 bit word size
- Let N be number of 32-bit words
  - Then N is a multiple of 16
- Message $M = (Y_0, Y_1, \ldots, Y_{N-1})$
  - Each Y_i is a 32-bit word

MD4 Algorithm

- For 32-bit words A,B,C, define

  $F(A,B,C) = (A \wedge B) \vee (\neg A \wedge C)$

  $G(A,B,C) = (A \wedge B) \vee (A \wedge C) \vee (B \wedge C)$

  $H(A,B,C) = A \oplus B \oplus C$

  where $\wedge$, $\vee$, $\neg$, $\oplus$ are AND, OR, NOT, XOR
- Define constants: K_0 = 0x00000000, K_1 = 0x5a827999, K_2 = 0x6ed9eba1
- Let W_i, i = 0,1,…47 be (permuted) inputs, Y_j

MD4 Algorithm

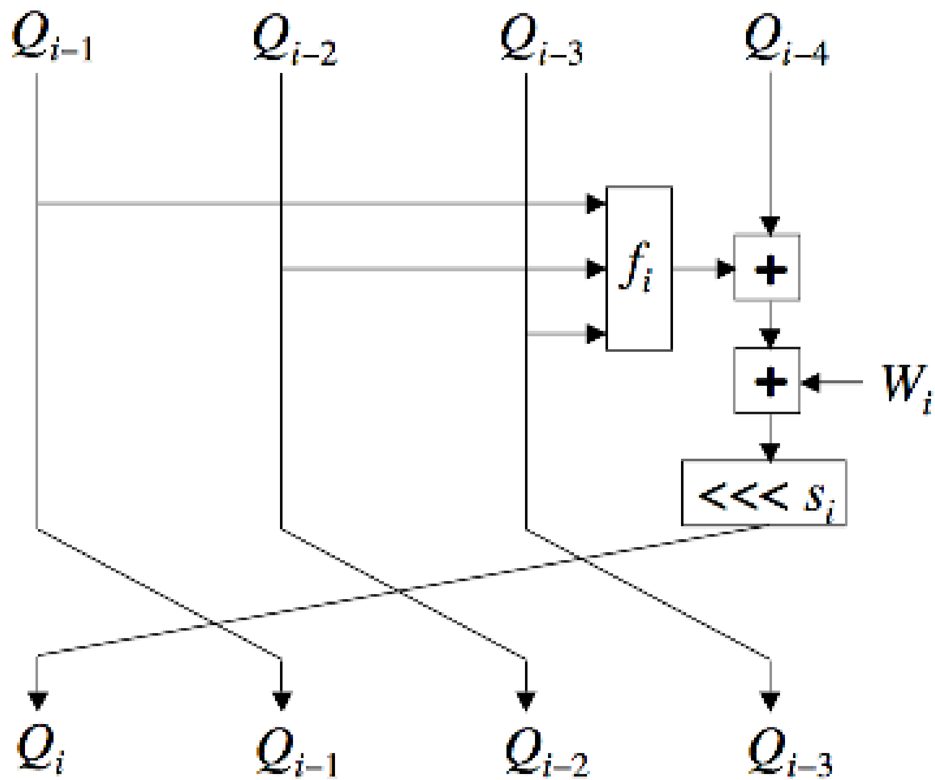
```
// M = (Y_0, Y_1, ..., Y_{N-1}), message to hash, after padding
// Each Y_i is a 32-bit word and N is a multiple of 16
MD4(M)
    // initialize (A, B, C, D) = IV
    (A, B, C, D) = (0x67452301, 0xefcdab89, 0x98badcfe, 0x10325476)
    for i = 0 to N/16 - 1
        // Copy block i into X
        X_j = Y_{16i+j}, for j = 0 to 15
        // Copy X to W
        W_j = X_{σ(j)}, for j = 0 to 47
        // initialize Q
        (Q_{-4}, Q_{-3}, Q_{-2}, Q_{-1}) = (A, D, C, B)
        // Rounds 0, 1 and 2
        Round0(Q, X)
        Round1(Q, X)
        Round2(Q, X)
        // Each addition is modulo 2^32
        (A, B, C, D) = (Q_44 + Q_{-4}, Q_47 + Q_{-1}, Q_46 + Q_{-2}, Q_45 + Q_{-3})
    next i
    return A, B, C, D
end MD4
```

MD4 Algorithm

$$
\begin{aligned}
&\text{Round0}(Q, W) \\
&\quad // \text{ steps 0 through 15} \\
&\quad \texttt{for } i = 0 \texttt{ to } 15 \\
&\qquad Q_i = (Q_{i-4} + F(Q_{i-1}, Q_{i-2}, Q_{i-3}) + W_i + K_0) \lll s_i \\
&\quad \texttt{next } i \\
&\texttt{end } \text{Round0}
\end{aligned}
$$

- Round 0: Steps 0 thru 15, uses F function
- Round 1: Steps 16 thru 31, uses G function
- Round 2: Steps 32 thru 47, uses H function

MD4: One Step



- Where $f_i(A,B,C) = \begin{cases} F(A,B,C) + K_0 & \text{if } 0 \le i \le 15 \\ G(A,B,C) + K_1 & \text{if } 16 \le i \le 31 \\ H(A,B,C) + K_2 & \text{if } 32 \le i \le 47 \end{cases}$

Notation

- Let $MD4_{i...j}$(A,B,C,D,M) be steps i thru j
  - "Initial value" (A,B,C,D) at step i, message M
- Note that $MD4_{0...47}$(IV,M) $\neq$ h(M)
  - Due to padding and final transformation
- Let f(IV,M) = $(Q_{44},Q_{47},Q_{46},Q_{45})$ + IV
  - Where "+" is addition mod 2^{32}, per 32-bit word
- Then f is the MD4 **compression function**

MD4 Attack: Outline

- Dobbertin's attack strategy
  - Specify a differential condition
  - If condition holds, probability of collision
  - Derive system of nonlinear equations: solution satisfies differential condition
  - Find efficient method to solve equations
  - Find enough solutions to yield a collision

MD4 Attack: Motivation

- Find one-block collision, where $M = (X_0, X_1, \ldots, X_{15})$, $M' = (X'_0, X'_1, \ldots, X'_{15})$
- Difference is subtraction mod 2^{32}
- Blocks differ in only 1 word
  - Difference in that word is exactly 1
- Limits avalanche effect to steps 12 thru 19
  - Only 8 of the 48 steps are critical to attack!
  - System of equations applies to these 8 steps

More Notation

- Spse $(Q_j, Q_{j-1}, Q_{j-2}, Q_{j-3}) = \text{MD4}_{0\ldots j}(\text{IV}, M)$ and $(Q'_j, Q'_{j-1}, Q'_{j-2}, Q'_{j-3}) = \text{MD4}_{0\ldots j}(\text{IV}, M')$
- Define

  $\Delta_j = (Q_j - Q'_j,\ Q_{j-1} - Q'_{j-1},\ Q_{j-2} - Q'_{j-2},\ Q_{j-3} - Q'_{j-3})$

  where subtraction is modulo 2^{32}
- Let $\pm 2^n$ denote $\pm 2^n \bmod 2^{32}$, for example, $2^{25} = \text{0x02000000}$ and $-2^5 = \text{0x}\mathtt{ffffffe0}$

MD4 Attack

- All arithmetic is modulo 2^{32}
- Denote $M = (X_0, X_1, \ldots, X_{15})$
- Define M' by $X'_i = X_i$ for $i \neq 12$ and

  $X'_{12} = X_{12} + 1$
- Word X_{12} last appears in step 35
- So, if $\Delta_{35} = (0,0,0,0)$ we have a collision
- Goal is to find pair M and M' with $\Delta_{35} = 0$

MD4 Attack

- Analyze attack in three phases

1. Show: $\Delta_{19} = (2^{25},-2^{5},0,0)$ implies probability at least $1/2^{30}$ that the Δ_{35} condition holds
   - Uses differential cryptanalysis
2. "Backup" to step 12: We can start at step 12 and have Δ_{19} condition hold
   - By solving system of nonlinear equations
3. "Backup" to step 0: And find collision

MD4 Attack

- In each phase of attack, some words of M are determined
- When completed, have M and M′
  - Where $M \neq M'$ but $h(M) = h(M')$
- Equation solving step is tricky part
  - Nonlinear system of equations
  - Must be able to solve efficiently

Steps 19 to 35

- Differential phase of the attack
- Suppose M and M′ as given above
  - Only differ in word 12
- Assume that $\Delta_{19} = (2^{25}, -2^{5}, 0, 0)$
  - And $G(Q_{19},Q_{18},Q_{17}) = G(Q'_{19},Q'_{18},Q'_{17})$
- Then we compute probabilities of "Δ" conditions at steps 19 thru 35

Steps 19 to 35

j	Δ_j: ΔQ_j	ΔQ_{j-1}	ΔQ_{j-2}	ΔQ_{j-3}	i	s_j	p	Input
19	2^{25}	-2^5	0	0	*	*	*	*
20	0	2^{25}	-2^5	0	1	3	1	X_1
21	0	0	2^{25}	-2^5	1	5	1/9	X_5
22	-2^{14}	0	0	2^{25}	1	9	1/3	X_9
23	2^6	-2^{14}	0	0	1	13	1/3	X_{13}
24	0	2^6	-2^{14}	0	1	3	1/9	X_2
25	0	0	2^6	-2^{14}	1	5	1/9	X_6
26	-2^{23}	0	0	2^6	1	9	1/3	X_{10}
27	2^{19}	-2^{23}	0	0	1	13	1/3	X_{14}
28	0	2^{19}	-2^{23}	0	1	3	1/9	X_3
29	0	0	2^{19}	-2^{23}	1	5	1/9	X_7
30	-1	0	0	2^{19}	1	9	1/3	X_{11}
31	1	-1	0	0	1	13	1/3	X_{15}
32	0	1	-1	0	2	3	1/3	X_0
33	0	0	1	-1	2	9	1/3	X_8
34	0	0	0	1	2	11	1/3	X_4
35	0	0	0	0	2	15	1	$X_{12}, X_{12}+1$

- Differential and probabilities

Steps 19 thru 35

- For example, consider Δ_{35}
- Spse j = 34 holds: Then Δ_{34} = (0,0,0,1) and

$$\begin{aligned} Q_{35} &= (Q_{31} + H(Q_{34}, Q_{33}, Q_{32}) + X_{12} + K_2) \lll 15 \\ &= ((Q'_{31} + 1) + H(Q'_{34}, Q'_{33}, Q'_{32}) + X_{12} + K_2) \lll 15 \\ &= (Q'_{31} + H(Q'_{34}, Q'_{33}, Q'_{32}) + (X_{12} + 1) + K_2) \lll 15 \\ &= Q'_{35} \end{aligned}$$

- Implies Δ_{35} = (0,0,0,0) with probability 1
  - As summarized in j = 35 row of table

Steps 12 to 19

- Analyze steps 12 to 19, find conditions that ensure $\Delta_{19} = (2^{25}, -2^{5}, 0, 0)$
  - And $G(Q_{19}, Q_{18}, Q_{17}) = G(Q'_{19}, Q'_{18}, Q'_{17})$, as required in differential phase
- Step 12 to 19—equation solving phase
- This is most complex part of attack
  - Last phase, steps 0 to 11, is easy

Steps 12 to 19

- Info for steps 12 to 19 given here
- If i = 0, function F, if i = 1, function G

j	i	s_j	M Input	M' Input
12	0	3	X_{12}	$X_{12}+1$
13	0	7	X_{13}	X_{13}
14	0	11	X_{14}	X_{14}
15	0	19	X_{15}	X_{15}
16	1	3	X_0	X_0
17	1	5	X_4	X_4
18	1	9	X_8	X_8
19	1	13	X_{12}	$X_{12}+1$

Steps 12 to 19

- To apply differential phase, must have $\Delta_{19} = (2^{25}, -2^{5}, 0, 0)$ which states that

$$Q_{19} = Q'_{19} + 2^{25}$$
$$Q_{18} + 2^{5} = Q'_{18}$$
$$Q_{17} = Q'_{17}$$
$$Q_{16} = Q'_{16}$$

- Derive equations for steps 12 to 19…

Step 12

- At step 12 we have
  $Q_{12} = (Q_8 + F(Q_{11},Q_{10},Q_9) + X_{12}) \lll 3$
  $Q'_{12} = (Q'_8 + F(Q'_{11},Q'_{10},Q'_9) + X'_{12}) \lll 3$
- Since $X'_{12} = X_{12} + 1$ and
  $(Q_8,Q_9,Q_{10},Q_{11}) = (Q'_8,Q'_9,Q'_{10},Q'_{11})$
  it follows that
  $(Q'_{12} \lll 29) - (Q_{12} \lll 29) = 1$

Steps 12 to 19

- Similar analysis for remaining steps yields system of equations:

$$
\begin{aligned}
1 &= (Q'_{12} \lll 29) - (Q_{12} \lll 29) \\
F(Q'_{12}, Q_{11}, Q_{10}) - F(Q_{12}, Q_{11}, Q_{10}) &= (Q'_{13} \lll 25) - (Q_{13} \lll 25) \\
F(Q'_{13}, Q'_{12}, Q_{11}) - F(Q_{13}, Q_{12}, Q_{11}) &= (Q'_{14} \lll 21) - (Q_{14} \lll 21) \\
F(Q'_{14}, Q'_{13}, Q'_{12}) - F(Q_{14}, Q_{13}, Q_{12}) &= (Q'_{15} \lll 13) - (Q_{15} \lll 13) \\
G(Q'_{15}, Q'_{14}, Q'_{13}) - G(Q_{15}, Q_{14}, Q_{13}) &= Q_{12} - Q'_{12} \\
G(Q_{16}, Q'_{15}, Q'_{14}) - G(Q_{16}, Q_{15}, Q_{14}) &= Q_{13} - Q'_{13} \\
G(Q_{17}, Q_{16}, Q'_{15}) - G(Q_{17}, Q_{16}, Q_{15}) &= Q_{14} - Q'_{14} + (Q'_{18} \lll 23) \\
&\quad - (Q_{18} \lll 23) \\
G(Q'_{18}, Q_{17}, Q_{16}) - G(Q_{18}, Q_{17}, Q_{16}) &= Q_{15} - Q'_{15} + (Q'_{19} \lll 19) \\
&\quad - (Q_{19} \lll 19) - 1
\end{aligned}
$$

Steps 12 to 19

- To solve this system must find

  $(Q_{10}, Q_{11}, Q_{12}, Q_{13}, Q_{14}, Q_{15}, Q_{16}, Q_{17}, Q_{18}, Q_{19}, Q'_{12}, Q'_{13}, Q'_{14}, Q'_{15})$

  so that all equations hold
- Given such a solution, we determine

  X_j for j = 13,14,15,0,4,8,12

  so that we begin at step 12 and arrive at step 19 with Δ_{19} condition satisfied

Steps 12 to 19

- This phase reduces to solving (nonlinear) system of equations
- Can manipulate the equations so that
  - Choose $(Q_{14},Q_{15},Q_{16},Q_{17},Q_{18},Q_{19})$ arbitrary
  - Which determines $(Q_{10},Q_{13},Q'_{13},Q'_{14},Q'_{15})$
  - See textbook for details
- Result is 3 equations must be satisfied (next slide)

Steps 12 to 19

- Three conditions must be satisfied:

$$G(Q_{15}, Q_{14}, Q_{13}) - G(Q'_{15}, Q'_{14}, Q'_{13}) = 1$$
$$F(Q'_{14}, Q'_{13}, 0) - F(Q_{14}, Q_{13}, -1) - (Q'_{15} \lll 13) + (Q_{15} \lll 13) = 0.$$
$$G(Q_{19}, Q_{18}, Q_{17}) = G(Q'_{19}, Q'_{18}, Q_{17})$$

- First 2 are "check" equations
  - Third is "admissible" condition
- Naïve algorithm: choose six Q_j, yields five Q_j,Q'_j until 3 equations satisfied
- How much work is this?

Continuous Approximation

- Each equation holds with prob $1/2^{32}$
- Appears that 2^{96} iterations required
  - Since three 32-bit check equations
  - Birthday attack on MD4 is only 2^{64} work!
- Dobbertin has a clever solution
  - A "continuous approximation"
  - Small changes, converge to a solution

Continuous Approximation

- Generate random Q_i values until first check equation is satisfied
  - Random one-bit modifications to Q_i
  - Save if 1st check equation still holds **and** 2nd check equation is "closer" to holding
  - Else try different random modifications
- Modifications converge to solution
  - Then 2 check equations satisfied
  - Repeat until admissible condition holds

Continuous Approximation

- For complete details, see textbook
- Why does continuous approx work?
  - Small change to arguments of F (or G) yield small change in function value
- What is the work factor?
  - Not easy to determine analytically
  - Easy to determine empirically (homework)
  - Efficient, and only once per collision

Steps 0 to 11

- At this point, we have (Q_8,Q_9,Q_{10},Q_{11}) and $\text{MD4}_{12\ldots47}(Q_8,Q_9,Q_{10},Q_{11},X) = \text{MD4}_{12\ldots47}(Q_8,Q_9,Q_{10},Q_{11},X')$
- To finish, we must have $\text{MD4}_{0\ldots11}(IV,X) = \text{MD4}_{0\ldots11}(IV,X') = (Q_8,Q_9,Q_{10},Q_{11})$
- Recall, X_{12} is only difference between M, M′
- Also, X_{12} first appears in step 12
- Have already found X_j for $j = 0,4,8,12,13,14,15$
- Free to choose X_j for $j = 1,2,3,5,6,7,9,10,11$ so that $\text{MD4}_{0\ldots11}$ equation holds — very easy!

All Together Now

- Attack proceeds as follows…

1. Steps 12 to 19: Find $(Q_8, Q_9, Q_{10}, Q_{11})$ and X_j for $j = 0,4,8,12,13,14,15$
2. Steps 0 to 11: Find X_j for remaining j
3. Steps 19 to 35: Check $\Delta_{35} = (0,0,0,0)$
   - If so, have found a collision!
   - If not, goto 2.

Meaningful Collision

- MD4 collisions exist where M and M′ have meaning
  - Attack is so efficient, possible to find meaningful collisions
- Let "$*$" represent a "random" byte
  - Inserted for "security" purposes
- Can find collisions on next slide…

Meaningful Collision

- Different contracts, same hash value

******************** CONTRACT At the price of $176,495 Alf Blowfish sells his house to Ann Bonidea …	******************** CONTRACT At the price of $276,495 Alf Blowfish sells his house to Ann Bonidea …

MD4 Conclusions

- MD4 weaknesses exposed early
  - Never widely used
- But took long time to find a collision
- Dobbertin's attack
  - Clever equation solving phase
  - Also includes differential phase
- Next, MD5…