維基百科

MD5

維基百科，自由的百科全書

MD5訊息摘要演算法（英語：MD5 Message-Digest Algorithm），一種被廣泛使用的密碼雜湊函式，可以產生出一個128位元（16位元組）的雜湊值（hash value），用於確保資訊傳輸完整一致。MD5由美國密碼學家羅納德·李維斯特（Ronald Linn Rivest）設計，於1992年公開，用以取代MD4演算法。這套演算法的程式在 RFC 1321 中被加以規範。

將資料（如一段文字）運算變為另一固定長度值，是雜湊演算法的基礎原理。

1996年後被證實存在弱點，可以被加以破解，對於需要高度安全性的資料，專家一般建議改用其他演算法，如SHA-2。2004年，證實MD5演算法無法防止碰撞攻擊，因此不適用於安全性認證，如SSL公開金鑰認證或是數位簽章等用途。

MD5

概述	
設計者	羅納德·李維斯特
首次發布	1992年4月
系列	MD2、MD4、MD5、MD6
密碼細節	
摘要長度	128位元
分組長度	512位元
結構	Merkle–Damgård construction
重複回數	4[1]

目錄

歷史與密碼學

1992年8月，羅納德·李維斯特向網際網路工程任務組（IETF）提交了一份重要檔案，描述了這種演算法的原理。由於這種演算法的公開性和安全性，在90年代被廣泛使用在各種程式語言中，用以確保資料遞移無誤等。[2]

MD5由MD4、MD3、MD2改進而來，主要增強演算法複雜度和不可逆性。

應用

MD5曾被用於檔案校驗、SSL/TLS、IPsec、SSH，但MD5早已被發現有明顯的缺陷。

演算法

MD5是輸入不定長度資訊，輸出固定長度128-bits的演算法。經過程式流程，生成四個32位元資料，最後聯合起來成為一個128-bits雜湊。基本方式為，求餘、取餘、調整長度、與連結變數進行迴圈運算。得出結果。

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$
$$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

$\oplus, \wedge, \vee, \neg$ 是 *XOR*, *AND*, *OR* , *NOT* 的符號。

虛擬碼

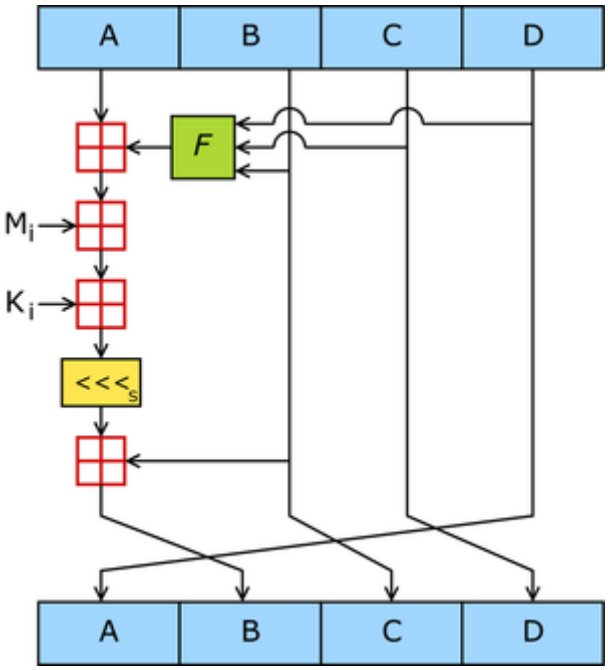

Figure 1. 一個MD5運算— 由類似的64次迴圈構成，分成4組16次。*F* 一個非線性函式；一個函式運算一次。M_i 表示一個 32-bits 的輸入資料，K_i 表示一個 32-bits 常數，用來完成每次不同的計算。

```
//Note: All variables are unsigned 32 bits and wrap modulo 2^32 when calculating
var int[64] r, k

//r specifies the per-round shift amounts
r[ 0..15] := {7, 12, 17, 22,  7, 12, 17, 22,  7, 12, 17, 22,  7, 12, 17, 22}
r[16..31] := {5,  9, 14, 20,  5,  9, 14, 20,  5,  9, 14, 20,  5,  9, 14, 20}
r[32..47] := {4, 11, 16, 23,  4, 11, 16, 23,  4, 11, 16, 23,  4, 11, 16, 23}
r[48..63] := {6, 10, 15, 21,  6, 10, 15, 21,  6, 10, 15, 21,  6, 10, 15, 21}

//Use binary integer part of the sines of integers as constants:
for i from 0 to 63
    k[i] := floor(abs(sin(i + 1)) × 2^32)

//Initialize variables:
var int h0 := 0x67452301
var int h1 := 0xEFCDAB89
var int h2 := 0x98BADCFE
var int h3 := 0x10325476

//Pre-processing:
append "1" bit to message
append "0" bits until message length in bits ≡ 448 (mod 512)
append bit length of message as 64-bit little-endian integer to message

//Process the message in successive 512-bit chunks:
for each 512-bit chunk of message
    break chunk into sixteen 32-bit little-endian words w[i], 0 ≤ i ≤ 15

    //Initialize hash value for this chunk:
    var int a := h0
    var int b := h1
    var int c := h2
    var int d := h3

    //Main loop:
    for i from 0 to 63
        if 0 ≤ i ≤ 15 then
            f := (b and c) or ((not b) and d)
            g := i
        else if 16 ≤ i ≤ 31
            f := (d and b) or ((not d) and c)
            g := (5×i + 1) mod 16
        else if 32 ≤ i ≤ 47
            f := b xor c xor d
            g := (3×i + 5) mod 16
        else if 48 ≤ i ≤ 63
            f := c xor (b or (not d))
            g := (7×i) mod 16

        temp := d
        d := c
        c := b
        b := leftrotate((a + f + k[i] + w[g]),r[i]) + b
        a := temp
    Next i
    //Add this chunk's hash to result so far:
    h0 := h0 + a
    h1 := h1 + b
    h2 := h2 + c
    h3 := h3 + d
End ForEach
var int digest := h0 append h1 append h2 append h3 //(expressed as little-endian)
```

MD5雜湊

一般128位元的MD5雜湊被表示為32位元十六進位數字。以下是一個43位長的僅ASCII字母列的MD5雜湊：

```
MD5("The quick brown fox jumps over the lazy dog")
= 9e107d9d372bb6826bd81d3542a419d6
```

即使在原文中作一個小變化（比如用c取代d）其雜湊也會發生巨大的變化：

```
MD5("The quick brown fox jumps over the lazy cog")
= 1055d3e698d289f2af8663725127bd4b
```

空文的雜湊為：

```
MD5("")
= d41d8cd98f00b204e9800998ecf8427e
```

缺陷

2009年，中國科學院的謝濤和馮登國僅用了$2^{20.96}$的碰撞演算法複雜度，破解了MD5的碰撞抵抗，該攻擊在普通電腦上執行只需要數秒鐘[3]。2011年，RFC 6151 禁止MD5用作金鑰雜湊訊息鑑別碼。

參見

- MD4
- SHA
- AES

參考文獻

1. RFC 1321, section 3.4, "Step 4. Process Message in 16-Word Blocks", page 5.
2. 梁斌. 第3章“搜索引擎的下载系统”第4节“网页抓取原理”. 走进搜索引擎. 孫學瑛 (責任編輯) 第1版. 電子工業出版社. 2007年10月: 51. ISBN 978-7-121-04922-4（中文（中國大陸））.
3. Tao Xie and Dengguo Feng. How To Find Weak Input Differences For MD5 Collision Attacks (PDF). 30 May 2009.

- Berson, Thomas A. Differential Cryptanalysis Mod 2^{32} with Applications to MD5. EUROCRYPT: 71–80. 1992. ISBN 978-3-540-56413-3.
- Bert den Boer; Antoon Bosselaers. Collisions for the Compression Function of MD5. 1993: 293–304. ISBN 978-3-540-57600-6.
- Hans Dobbertin, Cryptanalysis of MD5 compress. Announcement on Internet, May 1996 [1] (http://citeseer.ist.psu.edu/dobbertin96cryptanalysis.html).
- Dobbertin, Hans. The Status of MD5 After a Recent Attack. CryptoBytes. 1996, **2** (2).（原始內容存檔於2006-08-13）.
- Cong, Zijie. 提高哈希安全性的一些错误做法. Clifton Labratory. 2009.

外部連結

- W3C關於MD5的建議 (http://www.w3.org/TR/1998/REC-DSig-label/MD5-1_0)

本頁面最後修訂於2020年7月21日 (星期二) 08:54。