# MD5 加密与密码保护

软件工程系 王嘉威 13331251


2015 年 10 月 27 日

## 目录

# 一、 绪论

## 1.MD5 介绍

**MD5** 即 Message-Digest Algorithm 5（信息-摘要算法 5），用于确保信息传输完整一致。是计算机广泛使用的杂凑算法之一（又译摘要算法、哈希算法），主流编程语言普遍已有 MD5 实现。将数据（如汉字）运算为另一固定长度值，是杂凑算法的基础原理，MD5 的前身有 MD2、MD3 和 MD4

## 2.国内外研究情况

MD5 经受了多年的考验，各国著名的密码学家、数学家和黑客对其进行了多年的分析，提出了“生日攻击”和“袋鼠攻击”等一系列的攻击方法，但仍然没有突破性进展。

然而在 2004 年山东大学的王小云教授在国际密码学术年会作了破译 MD5 算法的报告，MD5 的安全壁垒出现了裂缝（并非真正的破解，只是加速了杂凑冲撞）。

2009 年，冯登国、谢涛二人利用差分攻击，将 MD5 的碰撞算法复杂度从王小云的 2^42 进一步降低到 2^21，极端情况下甚至可以降低至 2^10。仅仅 2^21 的复杂度意味着即便是在 2008 年的计算机上，也只要几秒便可以找到一对碰撞。

# 二、 MD5 算法解析

## 1. 算法原理

对信息的加密分为以下几个步骤：

1） 填充：将二进制的信息后填充 1 个 1 和若干个 0 直到信息位长对 512 求模为 448（注意就是原本位长求模为 448 也要填充 512 位）。

2） 初始化变量：设置 4 个链接变量 A、B、C、D 为 0x01234567、0x89ABCDEF、0xFBDCBA98、0x10325476，注意在程序中以小端规则储存。即 0x01234567 在程序中为 0x67452301。将运算的四个变量 a、b、c、d 分别初始化为 A、B、C、D。

3） 处理分组数据：将文本以 512 位分成若干组，对每一组数据，进行 16 次操作，每次操作对 a,b,c,d 其中三个做一次非线性函数运算，所得结果加上第四个变量，要处理的改组数据和一个常数，所得结果再向左循环移位一个不定的数并加上 a,b,c,d 中的一个，最后将结果赋值给 a,b,c,d 其中之一。

四个非线性函数为：

F( X ,Y ,Z ) = ( X & Y ) | ( (~X) & Z )

G( X ,Y ,Z ) = ( X & Z ) | ( Y & (~Z) )

H( X ,Y ,Z ) =X ^ Y ^ Z

I( X ,Y ,Z ) =Y ^ ( X | (~Z) )

而每轮的操作函数为：

FF(a ,b ,c ,d ,Mj ,s ,ti ) 操作为 a = b + ( (a + F(b,c,d) + Mj + ti) << s)

GG(a ,b ,c ,d ,Mj ,s ,ti ) 操作为 a = b + ( (a + G(b,c,d) + Mj + ti) << s)

HH(a ,b ,c ,d ,Mj ,s ,ti) 操作为 a = b + ( (a + H(b,c,d) + Mj + ti) << s)

II(a ,b ,c ,d ,Mj ,s ,ti) 操作为 a = b + ( (a + I(b,c,d) + Mj + ti) << s

这四轮（共 64 步）是：

第一轮

FF(a ,b ,c ,d ,M0 ,7 ,0xd76aa478 )

FF(d ,a ,b ,c ,M1 ,12 ,0xe8c7b756 )

FF(c ,d ,a ,b ,M2 ,17 ,0x242070db )
FF(b ,c ,d ,a ,M3 ,22 ,0xc1bdceee )
FF(a ,b ,c ,d ,M4 ,7 ,0xf57c0faf )
FF(d ,a ,b ,c ,M5 ,12 ,0x4787c62a )
FF(c ,d ,a ,b ,M6 ,17 ,0xa8304613 )
FF(b ,c ,d ,a ,M7 ,22 ,0xfd469501)
FF(a ,b ,c ,d ,M8 ,7 ,0x698098d8 )
FF(d ,a ,b ,c ,M9 ,12 ,0x8b44f7af )
FF(c ,d ,a ,b ,M10 ,17 ,0xffff5bb1 )
FF(b ,c ,d ,a ,M11 ,22 ,0x895cd7be )
FF(a ,b ,c ,d ,M12 ,7 ,0x6b901122 )
FF(d ,a ,b ,c ,M13 ,12 ,0xfd987193 )
FF(c ,d ,a ,b ,M14 ,17 ,0xa679438e )
FF(b ,c ,d ,a ,M15 ,22 ,0x49b40821 )
第二轮
GG(a ,b ,c ,d ,M1 ,5 ,0xf61e2562 )
GG(d ,a ,b ,c ,M6 ,9 ,0xc040b340 )
GG(c ,d ,a ,b ,M11 ,14 ,0x265e5a51 )
GG(b ,c ,d ,a ,M0 ,20 ,0xe9b6c7aa )
GG(a ,b ,c ,d ,M5 ,5 ,0xd62f105d )
GG(d ,a ,b ,c ,M10 ,9 ,0x02441453 )
GG(c ,d ,a ,b ,M15 ,14 ,0xd8a1e681 )
GG(b ,c ,d ,a ,M4 ,20 ,0xe7d3fbc8 )
GG(a ,b ,c ,d ,M9 ,5 ,0x21e1cde6 )
GG(d ,a ,b ,c ,M14 ,9 ,0xc33707d6 )
GG(c ,d ,a ,b ,M3 ,14 ,0xf4d50d87 )
GG(b ,c ,d ,a ,M8 ,20 ,0x455a14ed )
GG(a ,b ,c ,d ,M13 ,5 ,0xa9e3e905 )
GG(d ,a ,b ,c ,M2 ,9 ,0xfcefa3f8 )
GG(c ,d ,a ,b ,M7 ,14 ,0x676f02d9 )
GG(b ,c ,d ,a ,M12 ,20 ,0x8d2a4c8a )
第三轮
HH(a ,b ,c ,d ,M5 ,4 ,0xfffa3942 )
HH(d ,a ,b ,c ,M8 ,11 ,0x8771f681 )
HH(c ,d ,a ,b ,M11 ,16 ,0x6d9d6122 )
HH(b ,c ,d ,a ,M14 ,23 ,0xfde5380c )
HH(a ,b ,c ,d ,M1 ,4 ,0xa4beea44 )
HH(d ,a ,b ,c ,M4 ,11 ,0x4bdecfa9 )

HH(c ,d ,a ,b ,M7 ,16 ,0xf6bb4b60 )
HH(b ,c ,d ,a ,M10 ,23 ,0xbebfbc70 )
HH(a ,b ,c ,d ,M13 ,4 ,0x289b7ec6 )
HH(d ,a ,b ,c ,M0 ,11 ,0xeaa127fa )
HH(c ,d ,a ,b ,M3 ,16 ,0xd4ef3085 )
HH(b ,c ,d ,a ,M6 ,23 ,0x04881d05 )
HH(a ,b ,c ,d ,M9 ,4 ,0xd9d4d039 )
HH(d ,a ,b ,c ,M12 ,11 ,0xe6db99e5 )
HH(c ,d ,a ,b ,M15 ,16 ,0x1fa27cf8 )
HH(b ,c ,d ,a ,M2 ,23 ,0xc4ac5665 )
第四轮
II(a ,b ,c ,d ,M0 ,6 ,0xf4292244 )
II(d ,a ,b ,c ,M7 ,10 ,0x432aff97 )
II(c ,d ,a ,b ,M14 ,15 ,0xab9423a7 )
II(b ,c ,d ,a ,M5 ,21 ,0xfc93a039 )
II(a ,b ,c ,d ,M12 ,6 ,0x655b59c3 )
II(d ,a ,b ,c ,M3 ,10 ,0x8f0ccc92 )
II(c ,d ,a ,b ,M10 ,15 ,0xffeff47d )
II(b ,c ,d ,a ,M1 ,21 ,0x85845dd1 )
II(a ,b ,c ,d ,M8 ,6 ,0x6fa87e4f )
II(d ,a ,b ,c ,M15 ,10 ,0xfe2ce6e0 )
II(c ,d ,a ,b ,M6 ,15 ,0xa3014314 )
II(b ,c ,d ,a ,M13 ,21 ,0x4e0811a1 )
II(a ,b ,c ,d ,M4 ,6 ,0xf7537e82 )
II(d ,a ,b ,c ,M11 ,10 ,0xbd3af235 )
II(c ,d ,a ,b ,M2 ,15 ,0x2ad7d2bb )
II(b ,c ,d ,a ,M9 ,21 ,0xeb86d391 )
所有这些完成之后，将 a、b、c、d 分别在原来基础上再加上 A、B、C、D。
即 a = a + A , b = b + B , c = c + C , d = d + D
然后用下一分组数据继续运行以上算法。
最后加密的信息为 a,b,c,d 的级联，为 128 位。

## 2. 算法的改进策略

由于 MD5 算法是不可逆的，破解的手段都是通过穷举源码进行 MD5 运算来测试破解。改进的角度从在原来的基础上，附加多次的变换来增加穷举破解的难度。

1） 循环 MD5：对信息进行重复多次的 MD5 加密。
2） 密文分割 MD5：经过一次 MD5 运算后，得到是 32byte 的密文串，将该密文分成若干段，每段都进行一次 MD5 运算，再把得到的若干个密文连接起来最后进行一次 MD5 运算，得到最后的密文。
3） 附加字符串干涉：再加密过程的一个步骤中，附加一个内容确定的字符串，干涉被加密的数据。

# 三、 MD5 加密的应用

## 1. 在数据库中的应用

许多可注册用户的网站往往将账户和密码等信息使用非加密的方式储存到数据库。由于 MD5 算法不可逆，即使数据库信息被他人获取，也难以从中获得用户加密信息的原文。但是一旦用户丢失了密码，数据库也无法找回，只能通过重设密码方式来获得密码。

## 2. 数字签名

MD5 的典型应用是对一段信息产生信息摘要，以防止被篡改。用 MD5 算法将文件信息生成一个 MD5 码作为其数字签名，在该文件传播过程中，若发生任何形式的改变，只要将文件内容用 MD5 算法生成 MD5 码与其数字签名作对比，不相同则可认为文件发生了改变。若有可信的第三方机构，该数字签名还可以用于防止作者的“抵赖”。

## 3. 在电话语音系统软件保护中的应用

电话语音系统实现电话会议的功能, 系统的硬 件由工控机和各种语音卡组成, 软件主要是电话语音处理软件 TSPS(Telephone Speech Process Software)。该系统的软件保护总 体要求是 TSPS 软件、语音卡和工控机配套使用, 即 TSPS 软件 必须在指定的语音卡和工控机上才能正常运行。 综合分析现有的软件保护方法, 本文设计了软件序列号保护软件的方案。设计思想是系统采用序列号的保护方式, 实现 “一机一码”。具体设计是根据机器的硬件特征(也称机器指纹), 通过加密算法对机器指纹进行加密生成软件序列号。用户购买系统时, 获得 TSPS 软件及软件序列号。TSPS 软件每次运行时, 检查序列号的合法性, 决定是否继续运行。 软件保护程序

主要包括序列号生成模块和序列号验证模块。

而序列号的生成则是将机器的硬件信息（含 CPU 序列号和语音卡号）通过某算法加密得到密文 C1，再把 C1 通过 MD5 加密得到最终序列号。主要是通过运行的机器上特征值生成的序列号与文件中存放的序列号进行比较，相等则证明用户合法。

序列号生成程序为独立的程序，由软件开发商使用，不对 外公开。而验证程序存在用户端的机器中，通过跟踪和分析用 户端软件的运行，破解者就有可能发现程序的处理过程，从而 进一步推出序列号。所以对程序进行反汇编处理，以保证序列 号的机密。系统还将语音卡的初始化代码放在保护判断程序段中，这样破解者即使跳过检查代码，系统也无法正常运行。

# 四、 MD5 安全性分析

随着 MD5 弱点的被发现以及计算机能力不断地提升，通过穷举来破解 MD5 的方法已经成为可能。目前主要依靠储存常用密码及其 MD5 值大型字典进行穷举对比的破解手段。但是只要密码原文自身强度足够，还是很难穷举出来的。

并且，由 MD5 基础上改进的一些加入干扰字符串的算法也已经存在，这加大了 MD5 的破解难度。对于一般的安全应用是可以满足其保护需求的。

# 参考文献

[1]MD5 百度百科

[2]段青玲 杨仁刚 李辉 MD5 算法在电话语音系统软件保护中的应用 《微计算机信息》文章编号:1008- 0570(2007)08- 3- 0040-02

[3]基于 MD5 下的密码保护的简单探究 新浪博客