实验原理

一．哈希函数简介

信息安全的核心技术是应用密码技术。密码技术的应用远不止局限于提供机密性服务，密码技术也提供数据完整性服务。密码学上的散列函数(Hash Functions)就是能提供数据完整性保障的一个重要工具。Hash函数常用来构造数据的短“指纹”，消息的发送者使用所有的消息产生一个短“指纹”，并将该短“指纹”与消息一起传输给接收者。即使数据存储在不安全的地方，接收者重新计算数据的指纹，并验证指纹是否改变，就能够检测数据的完整性。这是因为一旦数据在中途被破坏或改变，短指纹就不再正确。

散列函数是一个函数，它以一个变长的报文作为输入，并产生一个定长的散列码，有时也称为报文摘要，作为函数的输出。散列函数最主要的作用是用于鉴别，鉴别在网络安全中起到举足轻重的地位。鉴别的目的有以下两个：第一，验证信息的发送者不是冒充的，同时发信息者也不能抵赖，此为信源识别；第二，验证信息完整性，在传递或存储过程中未被篡改，重放或延迟等。

二．哈希函数特点

密码学哈希函数（cryptography hash function，简称为哈希函数）在现代密码学中起着重要的作用，主要用于数据完整性认证和消息认证。哈希函数的基本思想是对数据进行运算得到一个摘要，运算过程满足：

- 压缩性：任意长度的数据，算出的摘要长度都固定。
- 容易计算：从原数据容易计算出摘要。
- 抗修改性：对原数据进行任何改动，哪怕只修改1个字节，所得到的摘要都有很大区别。
- 弱抗碰撞：已知原数据和其摘要，想找到一个具有相同摘要的数据（即伪造数据），在计算上是困难的。
- 强抗碰撞：想找到两个不同的数据，使它们具有相同的摘要，在计算上是困难的。

三．针对哈希函数的攻击

对散列函数的攻击方法主要有两种：

- 穷举攻击：它可以用于对任何类型的散列函数的攻击，最典型的方式就是所谓的“生日攻击”。采用生日攻击的攻击者将产生许多明文消息，然后计算这些明文消息的摘要，进行比较。
- 利用散列函数的代数结构：攻击其函数的弱性质。通常的有中间相遇攻击、修正分组攻击和差分分析攻击等。

四．MD5哈希函数

1990年R.L.Rivest提出哈希函数MD4。MD4不是建立在其他密码系统和假设之上，而是一种直接构造法。所以计算速度快，特别适合32位计算机软件实现，对于长的信息签名很实用。MD5是MD4的改进版，它比MD4更复杂，但是设计思想相似并且也产生了128位摘要。

五．MD5哈希算法流程 动画演示

对于任意长度的明文，MD5首先对其进行分组，使得每一组的长度为512位，然后对这些明文分组反复重复处理。

对于每个明文分组的摘要生成过程如下：

（1）将512位的明文分组划分为16个子明文分组，每个子明文分组为32位。

（2）申请4个32位的链接变量，记为A、B、C、D。

（3）子明文分组与链接变量进行第1轮运算。

（4）子明文分组与链接变量进行第2轮运算。

（5）子明文分组与链接变量进行第3轮运算。

（6）子明文分组与链接变量进行第4轮运算。

（7）链接变量与初始链接变量进行求和运算。

（8）链接变量作为下一个明文分组的输入重复进行以上操作。

（9）最后，4个链接变量里面的数据就是MD5摘要。

六．MD5分组过程

对于任意长度的明文，MD5可以产生128位的摘要。任意长度的明文首先需要添加位数，使明文总长度为448（mod512）位。在明文后添加位的方法是第一个添加位是1，其余都是0。然后将真正明文的长度（没有添加位以前的明文长度）以64位表示，附加于前面已添加过位的明文后，此时的明文长度正好是512位的倍数。当明文长度大于2的64次方时，仅仅使用低64位比特填充，附加到最后一个分组的末尾。

经过添加处理的明文，其长度正好为512位的整数倍，然后按512位的长度进行分组（block），可以划分成L份明文分组，我们用Y0，Y1，……，YL-1表示这些明文分组。对于每一个明文分组，都要重复反复的处理，如图8-1-1所示。

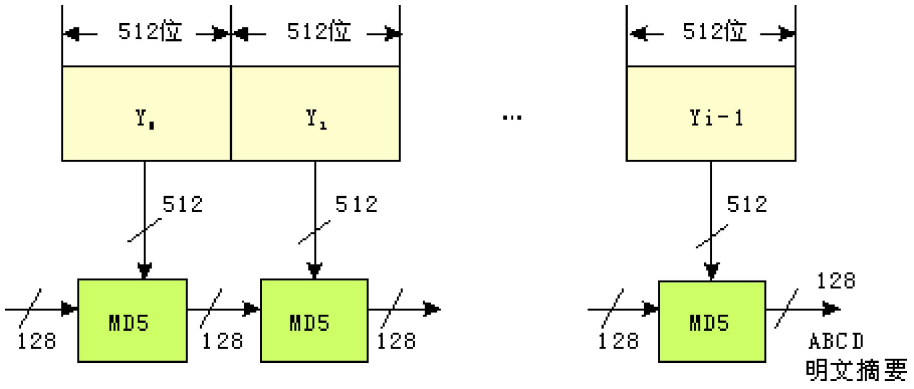

图8-1-1　MD5的分组处理方法

七．MD5子明文分组和链接变量

对于512位的明文分组，MD5将其再分成16份子明文分组（sub-block），每份子明文分组为32位，我们使用M[k]（k= 0, 1,……15）来表示这16份子明文分组。这里的概念要弄清楚，一个添加位后的明文可以划分为L份明文分组，而一个明文分组又可以划分为16份子明文分组。

MD5有4轮非常相似的运算，每一轮包括16个类似的步骤，每一个步骤的数据处理都是针对4个32位记录单元中的数据进行的。这4个链接变量的初始值以16进位制表示如下（低字节优先）A：0x01234567，B：0x89ABCDEF，C：0xFEDCBA98，D：0x76543210，这时A、B、C、D四个链接变量的值为：A=0x67452301，B=0xEFCDAB89，C=0x98BADCFE，D=0x10325476。链接变量用于存放中间散列函数值，经过4轮运算（共64个步骤）之后，链接变量A，B，C，D中的128位即为中间散列函数值。中间散列函数值作为下一个明文分组的输入继续使用，当所有的明文分组都处理完毕后，链接变量A，B，C，D中的128位数据就是摘要。

八．MD5第1轮运算

MD5有4轮非常相似的运算，每一轮包括16个类似的步骤，当第1轮运算中的第1步骤开始处理时，A、B、C、D四个链接变量中的值先赋值到另外4个记录单元A′，B′，C′，D′中。这4个值将保留，用于在第4轮的最后一个步骤完成之后与A，B，C，D进行求和操作。

第1轮的操作程序为FF(a,b,c,d,M[k],S,T[i])

它表示的逻辑为：a←b+((a+F(b,c,d)+M[k]+T[i])<<<S)

其中，a、b、c、d为32位的变量，M[k]表示相应的子明文分组，对于4轮共64步的MD5算法中T[i]是64个不同的固定的数值，S为循环左移的位数，F（x，y，z）是第一轮的逻辑函数，最后将结果存放在链接变量A中，固定值T[i]，循环左移位数和逻辑函数将在以后讨论。

第1轮16步的固定值T[i]的取值如表8-1-1所示。

表8-1-1　MD5第1轮固定数T

T[1]=D76AA478	T[5]=F57C0FAF	T[9]=698098D8	T[13]=6B901122

T[2]=E8C7B756	T[6]=4787C62A	T[10]=8B44F7AF	T[14]=FD987193
T[3]=242070DB	T[7]=A8304613	T[11]=FFFF5BB1	T[15]=A679438E
T[4]=C1BDCEEE	T[8]=FD469501	T[12]=895CD7BE	T[16]=49B40821

MD5规定，第一轮的16步的操作程序如表8-1-2所示。

表8-1-2　MD5第1轮16步运算

步骤数	运算
1	FF(A,B,C,D,M[0],7,0xD76AA478)
2	FF(D,A,B,C,M[1],12,0xE8C7B756)
3	FF(C,D,A,B,M[2],17,0x242070DB)
4	FF(B,C,D,A,M[3],22,0xC1BDCEEE)
5	FF(A,B,C,D,M[4],7,0xF57C0FAF)
6	FF(D,A,B,C,M[5],12,0x4787C62A)
7	FF(C,D,A,B,M[6],17,0xA8304613)
8	FF(B,C,D,A,M[7],22,0xFD469501)
9	FF(A,B,C,D,M[8],7,0x698098D8)
10	FF(D,A,B,C,M[9],12,0x8B44F7AF)
11	FF(C,D,A,B,M[10],17,0xFFFF5BB1)
12	FF(B,C,D,A,M[11],22,0x895CD7BE)
13	FF(A,B,C,D,M[12],7,0x6B901122)
14	FF(D,A,B,C,M[13],12,0xFD987193)
15	FF(C,D,A,B,M[14],17,0xA6794383)
16	FF(B,C,D,A,M[15],22,0x49B40821)

MD5算法中，第一轮的逻辑函数为F（x，y，z）=（x & y）|（~x & z），MD5的算法比较复杂，每一轮包括16步类似的运算，下面我们以第1轮的第1步和第2步为例来展示每一步的运算。

例如，子明文分组M[0] = 0x4368696E，第1轮的操作程序为FF(a,b,c,d,M[k],S,T[i])，它表示的逻辑为：

a←b+((a+F(b,c,d)+M[k]+T[i])<<<S)

第一轮的逻辑函数F（x，y，z）=（x & y）|（~x & z），由表8-1-2知，第1轮第1步的运算为：FF(A,B,C,D,M[0],7,0xD76AA478)，注意到这里的0xD76AA478就是T[1]的值，变量a、b、c、d分别代表链接变量A、B、C、D。首先，b、c、d要经过逻辑函数F，即：

```
(b&c)|(~b&d)=(0xEFCDAB89&0x98BADCFE)|(~0xEFCDAB89&0x10325476)=0x98BADCFE
```

然后得到的值要与A、M[0]和T[1]相加得0x67452301+0x98BADCFE+0x6E696843+0xD76AA478 = 0x45D40CBA，0x45D40CBA要循环左移7位，得到结果：0xEA065D22，0xEA065D22与b相加得：0xEA065D22 + 0xEFCDAB89 = 0xD9D408AB，最后，将这个结果赋值给a，第1步的计算就完成了，只有链接变量A发生了改变，这时链接标量的值为：

A = 0xD9D408AB

B = 0x89ABCDEF

C = 0xFEDCBA98

D = 0x76543210

经过16个步骤之后，MD5的第一轮运算就完成了，链接变量A、B、C、D将携带第1轮运算后的数值进入

第二轮运算。

九．MD5后3轮运算

MD5第2轮、第3轮和第4轮算运与第一轮运算相似，这里给出相应的操作程序、固定数T、每一步运算和逻辑函数。

第2轮的逻辑函数为：G(x, y, z) = (x & z)|(y & ~z)。

第3轮的逻辑函数为：H(x, y, z) = x⊕y⊕z。

第4轮的逻辑函数为：I(x, y, z) = y⊕(x & ~z)。

第2轮的操作程序为：GG(A,B,C,D,M[k],S,T[i])。

它表示的逻辑为：a←b+((a+G(B,C,D)+M[k]+T[i])<<<S)。

第3轮的操作程序为：HH(A,B,C,D,M[k],S,T[i])。

它表示的逻辑为：a←b+((a+H(B,C,D)+M[k]+T[i])<<<S)。

第4轮的操作程序为：II(A,B,C,D,M[k],S,T[i])。

它表示的逻辑为：a←b+((a+I(B,C,D)+M[k]+T[i])<<<S)。

后3轮的每个步骤的运算如表8-1-3所示。

表8-1-3　MD5后3轮16步运算

第二轮	
1	GG(A,B,C,D,M[1],5,0xF61E2562)
2	GG(D,A,B,C,M[6],9,0xC040B340)
3	GG(C,D,A,B,M[11],14,0x275E5A51)
4	GG(B,C,D,A,M[0],20,0xE9B6C7AA)
5	GG(A,B,C,D,M[5],5,0xD62F105D)
6	GG(D,A,B,C,M[10],9,0x02441453)
7	GG(C,D,A,B,M[15],14,0xD8A1E681)
8	GG(B,C,D,A,M[4],20,0xE7D3FBC8)
9	GG(A,B,C,D,M[9],5,0x21E1CDE6)
10	GG(D,A,B,C,M[14],9,0xC33707D6)
11	GG(C,D,A,B,M[3],14,0xF4D50D87)
12	GG(B,C,D,A,M[8],20,0x455A14ED)
13	GG(A,B,C,D,M[13],5,0xA9E3E905)
14	GG(D,A,B,C,M[2],9,0xFCEFA3F8)
15	GG(C,D,A,B,M[7],14,0x676F02D9)
16	GG(B,C,D,A,M[12],20,0x8D2A4C8A)
第三轮	
1	HH(A,B,C,D,M[5],4,0xFFFA3942)
2	HH(D,A,B,C,M[8],11,0x8771F681)
3	HH(C,D,A,B,M[11],16,0x6D9D6122)
4	HH(B,C,D,A,M[14],23,0xFDE5380C)
5	HH(A,B,C,D,M[1],4,0xA4BEEA44)
6	HH(D,A,B,C,M[4],11,0x4BDECFA9)
7	HH(C,D,A,B,M[7],16,0xF6BB4B60)
8	HH(B,C,D,A,M[10],23,0xBEBFBC70)

9	HH(A,B,C,D,M[13],4,0x289B7EC6)
10	HH(D,A,B,C,M[0],11,0xEAA127FA)
11	HH(C,D,A,B,M[3],16,0xD4EF3085)
12	HH(B,C,D,A,M[6],23,0x04881D05)
13	HH(A,B,C,D,M[9],4,0xD9D4D039)
14	HH(D,A,B,C,M[12],11,0xE6DB99E5)
15	HH(C,D,A,B,M[15],16,0x1FA27CF8)
16	HH(B,C,D,A,M[2],23,0xC4AC5665)
第四轮	
1	II(A,B,C,D,M[0],6,0xF4292244)
2	II(D,A,B,C,M[7],10,0x411AFF97)
3	II(C,D,A,B,M[14],15,0xAB9423A7)
4	II(B,C,D,A,M[5],21,0xFC93A039)
5	II(A,B,C,D,M[12],6,0x655B59C3)
6	II(D,A,B,C,M[3],10,0x8F0CCC92)
7	II(C,D,A,B,M[10],15,0xFFEFF47D)
8	II(B,C,D,A,M[1],21,0x85845DD1)
9	II(A,B,C,D,M[8],6,0x6AFA87E4F)
10	II(D,A,B,C,M[15],10,0xFE2CE6E0)
11	II(C,D,A,B,M[6],15,0xA3014314)
12	II(B,C,D,A,M[13],21,0x4E0811A1)
13	II(A,B,C,D,M[4],6,0xF7537E82)
14	II(D,A,B,C,M[11],10,0xBD3AF235)
15	II(C,D,A,B,M[2],15,0x2AD7D2BB)
16	II(B,C,D,A,M[9],21,0xEB86D391

后3轮的固定数T[i]的值如表8-1-4所示。

表8-1-4　后3轮的固定数T[i]

T[17]=F61E2562	T[33]=FFFA3942	T[49]=F4292244
T[18]=C040B340	T[34]=8771F681	T[50]=432AFF97
T[19]=265E5A51	T[35]=699D6122	T[51]=AB9423A7
T[20]=E9B6C7AA	T[36]=FDE5380C	T[52]=FC93A039
T[21]=D62F105D	T[37]=A4BEEA44	T[53]=655B59C3
T[22]=02441453	T[38]=4BDECFA9	T[54]=8F0CCC92
T[23]=D8A1E681	T[39]=F6BB4B60	T[55]=FFEFF47D
T[24]=E7D3FBC8	T[40]=BEBFBC70	T[56]=85845DD1
T[25]=21E1CDE6	T[41]=289B7EC6	T[57]=6FA87E4F
T[26]=C33707D6	T[42]=EAA127FA	T[58]=FE2CE6E0
T[27]=F4D50D87	T[43]=D4EF3085	T[59]=A3014314
T[28]=455A14ED	T[44]=04881D05	T[60]=4E0811A1
T[29]=A9E3E905	T[45]=D9D4D039	T[61]=F7657E82

T[30]=FCEEA3F8	T[46]=E6DB99E5	T[62]=BD3AF235
T[31]=676F02D9	T[47]=1FA27CF8	T[63]=2AD7D2BB
T[32]=8D2A4C8A	T[48]=C4AC5665	T[64]=EB86D391

十．求和运算

第四轮最后一步骤的A，B，C，D输出，将分别与A′，B′，C′，D′记录单元中数值进行求和操作。其结果将成为处理下一个512位明文分组时记录单元A，B，C，D的初始值。当完成了最后一个明文分组运算时，A，B，C，D中的数值就是最后的散列函数值。