实验原理

一．SHA1与MD5差异

SHA1对任意长度明文的预处理和MD5的过程是一样的，即预处理完后的明文长度是512位的整数倍，但是有一点不同，那就是SHA1的原始报文长度不能超过2的64次方，然后SHA1生成160位的报文摘要。SHA1算法简单而且紧凑，容易在计算机上实现。

表8-2-1列出了对MD5及SHA1的比较差异之处。让我们根据各项特性，简要说明其间的不同。

表8-2-1　MD5与SHA1的比较

差异处	MD5	SHA1
摘要长度	128位	160位
运算步骤数	64	80
基本逻辑函数数目	4	4
常数数目	64	4

- 安全性：SHA1所产生的摘要比MD5长32位。若两种散列函数在结构上没有任何问题的话，SHA1比MD5更安全。
- 速度：两种方法都是主要考虑以32位处理器为基础的系统结构。但SHA1的运算步骤比MD5多了16步，而且SHA1记录单元的长度比MD5多了32位。因此若是以硬件来实现SHA1，其速度大约比MD5慢了25%。
- 简易性：两种方法都是相当的简单，在实现上不需要很复杂的程序或是大量存储空间。然而总体上来讲，SHA1对每一步骤的操作描述比MD5简单。

二．SHA1哈希算法流程 动画演示

对于任意长度的明文，SHA1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5个32位的链接变量，记为A、B、C、D、E。

（3）16份子明文分组扩展为80份。

（4）80份子明文分组进行4轮运算。

（5）链接变量与初始链接变量进行求和运算。

（6）链接变量作为下一个明文分组的输入重复进行以上操作。

（7）最后，5个链接变量里面的数据就是SHA1摘要。

三．SHA1的分组过程

对于任意长度的明文，SHA1的明文分组过程与MD5相类似，首先需要对明文添加位数，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。与MD5不同的是SHA1的原始报文长度不能超过2的64次方，另外SHA1的明文长度从低位开始填充。

经过添加位数处理的明文，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Y_0，Y_1，……Y_{L-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，这些与MD5是相同的。

对于512位的明文分组，SHA1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。之后还要将这16份子明文分组扩充到80份子明文分组，我们记为W[k]（k= 0, 1,……79），扩充的方法如下。

$W_t = M_t$，当0≤t≤15

$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <<< 1$，当16≤t≤79

SHA1有4轮运算，每一轮包括20个步骤（一共80步），最后产生160位摘要，这160位摘要存放在5个32位的链接变量中，分别标记为A、B、C、D、E。这5个链接变量的初始值以16进制位表示如下。

A=0x67452301

B=0xEFCDAB89

C=0x98BADCFE

D=0x10325476

E=0xC3D2E1F0

四．SHA1的4轮运算

SHA1有4轮运算，每一轮包括20个步骤，一共80步，当第1轮运算中的第1步骤开始处理时，A、B、C、D、E五个链接变量中的值先赋值到另外5个记录单元A′，B′，C′，D′，E′中。这5个值将保留，用于在第4轮的最后一个步骤完成之后与链接变量A，B，C，D，E进行求和操作。

SHA1的4轮运算，共80个步骤使用同一个操作程序，如下：

$A,B,C,D,E \leftarrow [(A<<<5)+f_t(B,D,C)+E+W_t+K_t],A,(B<<<30),C,D$

其中$f_t(B,D,C)$为逻辑函数，W_t为子明文分组W[t]，K_t为固定常数。这个操作程序的意义为：

- 将$[(A<<<5)+f_t(B,D,C)+E+W_t+K_t]$的结果赋值给链接变量A；
- 将链接变量A初始值赋值给链接变量B；
- 将链接变量B初始值循环左移30位赋值给链接变量C；
- 将链接变量C初始值赋值给链接变量D；
- 将链接变量D初始值赋值给链接变量E。

SHA1规定4轮运算的逻辑函数如表8-2-2所示。

表8-2-2　SHA1的逻辑函数

轮	步骤	函数定义	轮	步骤	函数定义
1	0≤t≤19	$f_t(B,C,D)=(B \cdot C) \vee (\sim B \cdot D)$	3	40≤t≤59	$f_t(B,C,D)=$ $(B \cdot C) \vee (B \cdot D) \vee (C \cdot D)$
2	20≤t≤39	$f_t(B,C,D)=B \oplus C \oplus D$	4	60≤t≤79	$f_t(B,C,D)=B \oplus C \oplus D$

在操作程序中需要使用固定常数Ki（i= 0，1，2，……79），Ki的取值如表8-2-3所示：

表8-2-3　SHA1的常数K取值表

轮	步骤	函数定义	轮	步骤	函数定义
1	0≤t≤19	K_t=5A827999	3	40≤t≤59	K_t=8F188CDC
2	20≤t≤39	K_t=6ED9EBA1	4	60≤t≤79	K_t=CA62C1D6

我们同样举一个例子来说明SHA1哈希算法中的每一步是怎样进行的，比起MD5算法，SHA1相对简单，假设W[1]=0x12345678，此时链接变量的值分别为A=0x67452301、B=0xEFCDAB89、C=0x98BADCFE、D=0x10325476、E=0xC3D2E1F0，那么第1轮第1步的运算过程如下。

（1）将链接变量A循环左移5位，得到的结果为：0xE8A4602C。

（2）将B，C，D经过相应的逻辑函数：

```
(B&C)|(~B&D)=(0xEFCDAB89&0x98BADCFE)|(~0xEFCDAB89&0x10325476)=0x98BADCFE
```

（3）将第（1）步，第（2）步的结果与E，W[1]，和K[1]相加得：

```
0xE8A4602C+0x98BADCFE+0xC3D2E1F0+0x12345678+0x5A827999=0xB1E8EF2B
```

（4）将B循环左移30位得：(B<<<30)=0x7BF36AE2。

（5）将第3步结果赋值给A，A（这里是指A的原始值）赋值给B，步骤4的结果赋值给C，C的原始值赋值给D，D的原始值赋值给E。

（6）最后得到第1轮第1步的结果：

A = 0xB1E8EF2B

B = 0x67452301

C = 0x7BF36AE2

D = 0x98BADCFE

E = 0x10325476

按照这种方法，将80个步骤进行完毕。

第四轮最后一个步骤的A，B，C，D，E输出，将分别与记录单元A′，B′，C′，D′，E′中的数值求和运算。其结果将作为输入成为下一个512位明文分组的链接变量A，B，C，D，E，当最后一个明文分组计算完成以后，A，B，C，D，E中的数据就是最后散列函数值。