MD5实验报告

一、程序流程简介

重要函数

void zip( unsigned char block[64])

我们首先将压缩时的预备函数写出，如四个运算和循环左移等F（）,G（）,H（）,I（）,leftshift（），四轮函数计算等。

然后我们把这些函数整合起来放入。

在能够压缩处理之前，我们还需要对数据进行分组，我们用for循环，将数据分为32位一组，然后再执行一些列计算操作。

encode( unsigned int\* input, unsigned char\* output, int length);

该函数将int数组转换为char，即将十进制转换为十六进制，并且在转换时，我们使用小端序存放结果

init( unsigned char\* input, int len)

初始化函数，我们设定了一个count[2]数组，他负责记录当前字符串[位数]，然后我们获取当前已有的字节数（count[0]>>3再模64），然后我们用count[0]+len<<3，即已有位数加上新增加的位数，然后我们判断是否有溢出，若有，则将高位count[1]+1，接着我们让count[1]获取高位的位数，即先让len右移32得到高位，再左移3位得到位数，即count[1]+=len>>29。然后我们将新加入字符长度与待填充长度进行比较，在第一次调用此函数时，等同于我们将输入的字符串字节数与64字节数即512比特进行比较，若大于等于，则说明我们可以先将数据按64字节分组，先对这些64字节的组执行压缩函数，然后再对最后一组不足64字节的数据执行后续填充操作等。然后函数执行memcpy(&buffer[nowlength], &input[i], len - i);，进行拷贝。

unsigned char\* getDigest()

该函数负责对最后一组进行填充、压缩等操作。首先我们将count中计算的数据长度用encode转换，转换完的数据可以直接填在最后8字节，接着我们获取当前数据已有的字节，然后计算我们的填充字节长度，如果长度小于56字节即448比特，则填充字节=56-数据字节，否则，填充字节=120-数据字。由于我们的填充字节是以1开头，后续全是0，则我们可以构造一个数组tinachong[64]，其开头为0x80即10000000，我们便可以用这个数组进行填充。调用init(tianchong, padLen);将填充数组的内容以一定长度填入我们的最后一个分组，然后我们只需再将最后8字节填入之前算好数据长度，再调用压缩函数进行计算即可。

string getstring()

该函数使用for循环，将数据转化为16进制数。

流程图

输入待加密字符

Getstring（）函数将结果作为16进制输出

若输入数据大于64字节

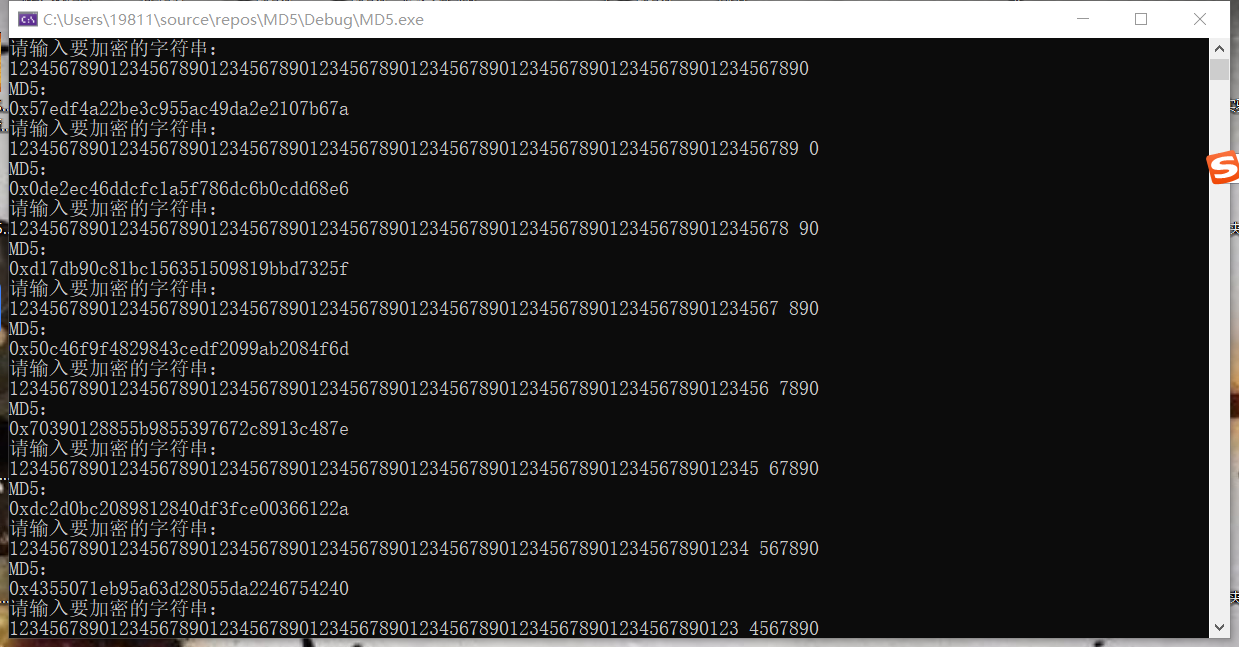
调用zip（）函数对长度满足64字节的分组进行压缩。

getDigest()函数

对不足64字节的一组进行填充

调用init()函数，进行初始化,检查输入数据长度，对数据按64字节进行分组，

雪崩测验



输入：12345678901234567890123456789012345678901234567890123456789012345678901234567890

哈希：0x57edf4a22be3c955ac49da2e2107b67a

雪崩1：1234567890123456789012345678901234567890123456789012345678901234567890123456789 0

哈希0x0de2ec46ddcfc1a5f786dc6b0cdd68e6

变化位数：63

雪崩2：123456789012345678901234567890123456789012345678901234567890123456789012345678 90

哈希0xd17db90c81bc156351509819bbd7325f

变化位数：63

雪崩3：12345678901234567890123456789012345678901234567890123456789012345678901234567 890

哈希0x50c46f9f4829843cedf2099ab2084f6d

变化位数：62

雪崩4：1234567890123456789012345678901234567890123456789012345678901234567890123456 7890

哈希0x70390128855b9855397672c8913c487e

变化位数：63

雪崩5：123456789012345678901234567890123456789012345678901234567890123456789012345 67890

哈希0xdc2d0bc2089812840df3fce00366122a

变化位数：61

雪崩6：12345678901234567890123456789012345678901234567890123456789012345678901234 567890

哈希0x4355071eb95a63d28055da2246754240

变化位数：59

雪崩7：1234567890123456789012345678901234567890123456789012345678901234567890123 4567890

哈希0x960c1fb546d72f07eb6f81936adea0bf

变化位数：67

雪崩8：123456789012345678901234567890123456789012345678901234567890123456789012 34567890

哈希0xe9d58be664aca3a8d4a215837a75d97e

变化位数：76

平均次数：64.25