Summary

Project1

实验原理

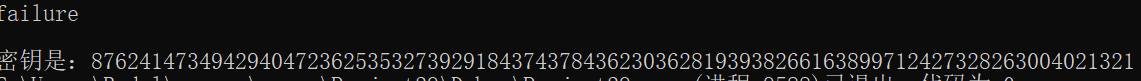
生日攻击原理是根据概率计算算，可以以50%以上的概率找到一个hash碰撞。

这里使用多线程优化了代码

如果寻找更长bit的碰撞，寻找时间也会相应变长。

软件环境：Visual Studio 2019

实验结果



Project2

实验原理

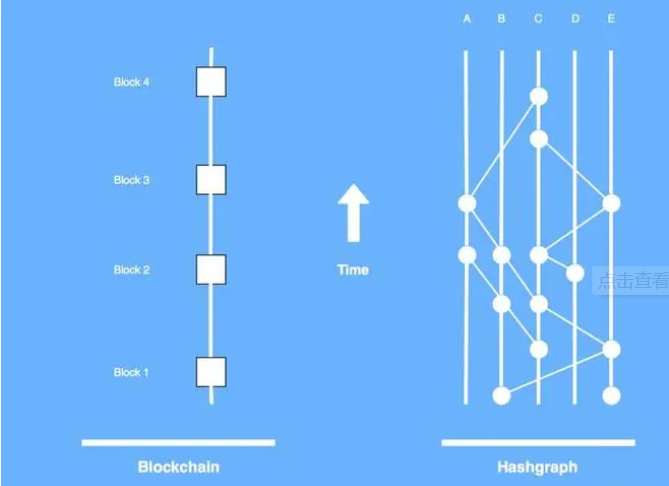
rho方法是SM3算法中用于对消息进行扩展和分块处理的重要步骤，确保输入的消息能够适应SM3算法的运算要求。通过rho方法，可以将任意长度的消息转化为一系列512比特的块，然后对每个块进行进一步的处理，最终生成SM3哈希值。

实验环境

硬件环境： PU:2.0GHz ，5G内存,500G硬盘  
软件环境：开发环境：windows

Project3

实验原理



硬件环境： PU:2.0GHz ，5G内存,500G硬盘  
软件环境：开发环境：windows

Project4

实验原理

利用并行计算和使用表格预计算，以减少运行时间

实验环境

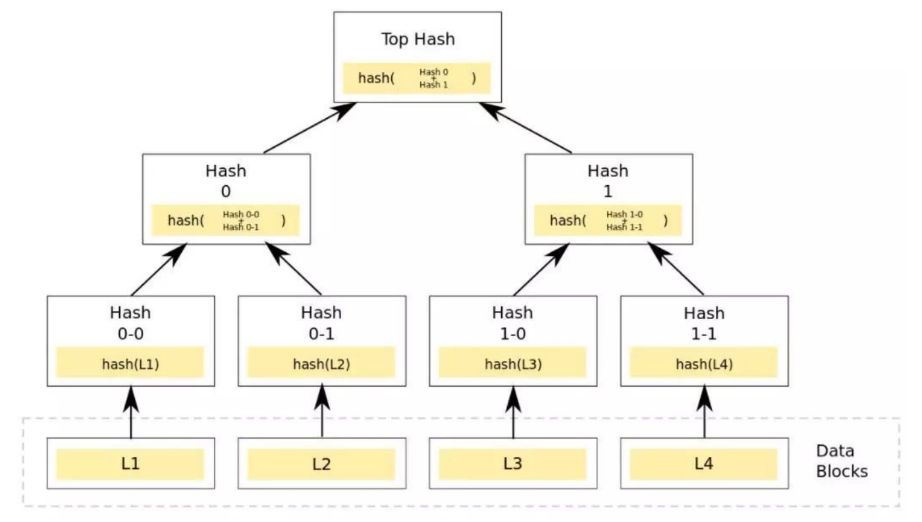
硬件环境： PU:2.0GHz ，5G内存,500G硬盘  
软件环境：开发环境：windows

实验结果



Project5

实验原理



实验环境

硬件环境： PU:2.0GHz ，5G内存,500G硬盘  
软件环境：开发环境：windows

Project9

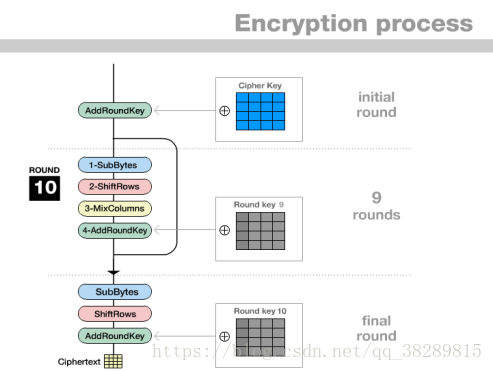
# AES/SM4softwareimplementation

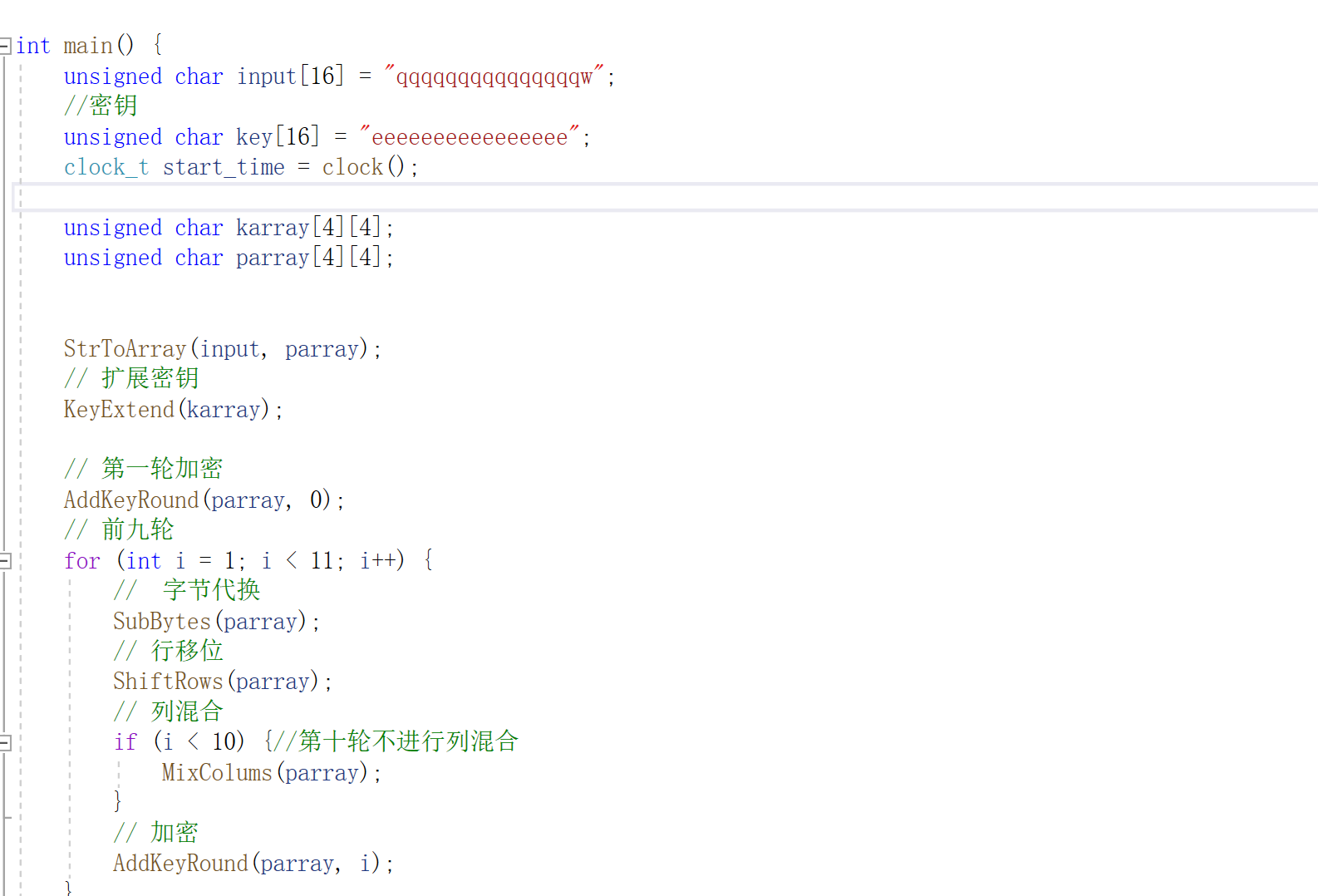
实现方式：

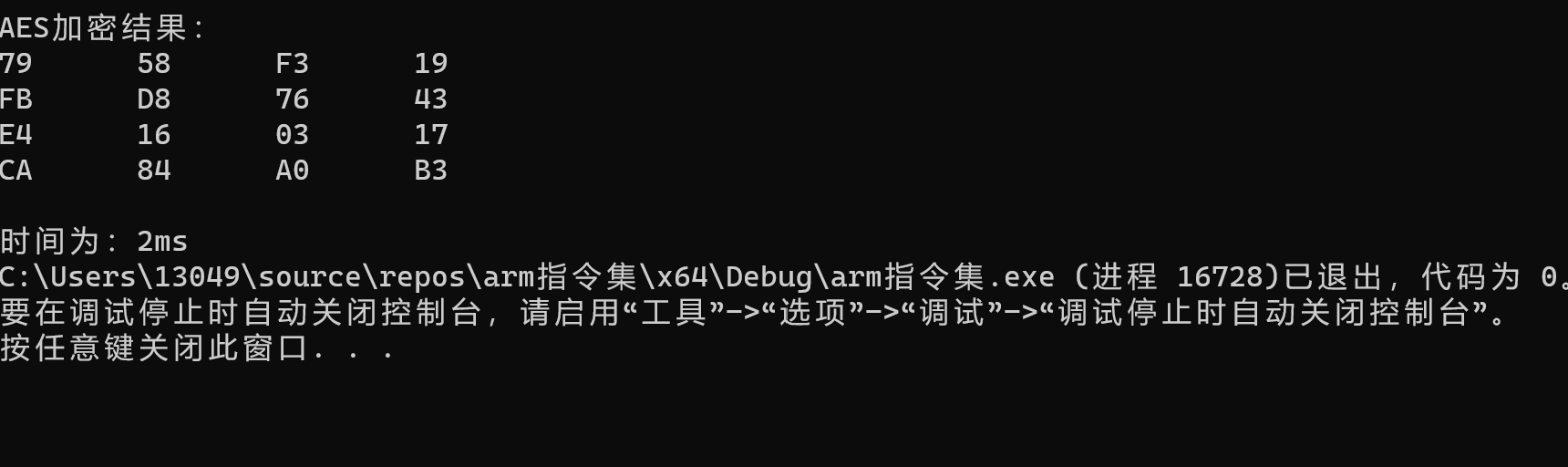
1.字节替换：字节代替的主要功能是通过S盒完成一个字节到另外一个字节的映射。

2.行移位：行移位的功能是实现一个4x4矩阵内部字节之间的置换。

3.列混淆

4.轮密钥加：加密过程中，每轮的输入与轮密钥异或一次（当前分组和扩展密钥的一部分进行按位异或）；因为二进制数连续异或一个数结果是不变的，所以在解密时再异或上该轮的密钥即可恢复输入。首尾使用轮密钥加的理由：若将其他不需要密钥的阶段放在首尾，在不用密钥的情况下就能完成逆过程，这就降低了算法的安全性。





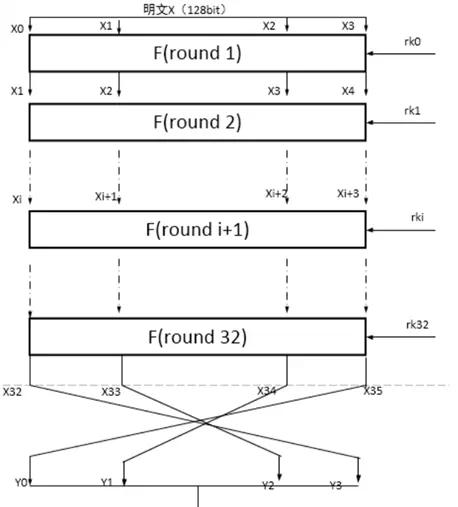
# SM4

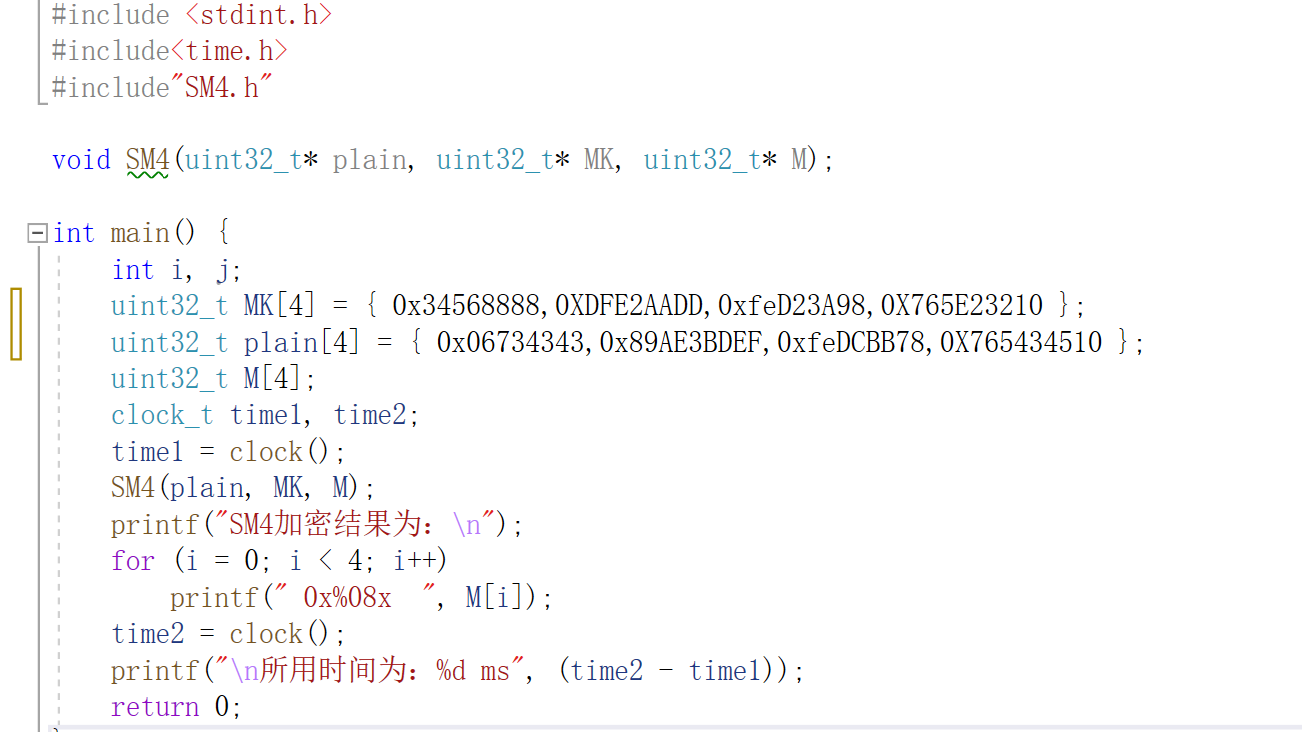
· 与DES和AES算法类似，SM4算法是一种分组密码算法。

· 其分组长度为128bit，密钥长度也为128bit。

· 加密算法与密钥扩展算法均采用32轮非线性迭代结构，以字（32位）为单位进行加密运算，每一次迭代运算均为一轮变换函数F。

· SM4算法加/解密算法的结构相同，只是使用轮密钥相反，其中解密轮密钥是加密轮密钥的逆序。





# 实现结果



Project10

**ECDSA的密钥的计算方式**

使用椭圆曲线E， 其中模数为p，系数为a和b，生成素数阶n的循环群的点G，选择一个随机整数d，且0 < d < q。

计算B=dG，其中d就是ECSDA的私钥，而点B(x，y)就是ECDSA的公钥。



Project11

成功实现sm2的RFC6979方案。运行指导：请直接运行

Project12

成功实现在k值相同情况下还原sm2密钥。运行指导：直接运行，发现还原成功

Project13成功实现sm2的PGP方案。运行指导：直接运行

Project15成功实现在真实网络下，本机网络上实现sm2签名方案。运行指导：先运行B后运行A

Project16

成功实现在真实网络下，本机网络上实现sm2解密方案。运行指导：先运行B后运行A

Project17

比较探究题，见project17

Project18

可以在 <https://live.blockcypher.com/btc-testnet> 中查看详细的交易。

但之后没有使用上面的比特币钱包软件，而是获得了比特币测试地址

https://www.bitaddress.org)

//登录后还需要在网址后加上?testnet=true可获取测试地址

该交易可以在线 <https://live.blockcypher.com/btc-testnet/tx/bdd6a952621cc7653c43ebd6eb8e4d1e6b8b233d820e87aec2db8174675c291f/> 看到。

取得测试地址后可以在水龙头领取测试的比特币

之后便会有一笔交易，打入比特币进这个比特币地址。

在[比特币测试网交易查询]([BlockCypher Testnet Block Explorer |BlockCypher](https://live.blockcypher.com/bcy/)) 网址里面可以查询比特币地址，以及通过txid（交易id，唯一标识某笔交易的交易号）也可以找到某笔交易。

通过查询我的比特币地址：可以看到上面的余额情况：

再通过查询一笔交易的txid可以查到该交易。

交易哈希（Transaction Hash）：每笔比特币交易都有一个唯一的交易哈希，用于标识该交易。通过交易哈希可以在区块链上查找到该交易的详细信息。

输入（Inputs）：交易的输入是之前交易的输出。输入包含了之前某些比特币交易的输出信息，即发送方的地址和发送金额。可以通过解析输入找到发送方的地址和发送的比特币数量。

输出（Outputs）：交易的输出是指交易接收方的地址和接收的金额。可以通过解析输出找到接收方的地址和接收的比特币数量。

手续费（Fee）：交易手续费是发送方支付给矿工的费用，以便将交易添加到区块链中。通过解析交易数据，可以确定交易手续费的大小。

签名（Signatures）：交易数据还包含用于验证交易有效性的数字签名。签名是发送方使用其私钥对交易进行加密的结果，接收方使用发送方的公钥对签名进行解密和验证。

比如我们查询刚才进行的把比特币打入账户的交易，查到这笔交易的

我们的任务是爬取该部分信息并且解析出来，所以可以把网址交给自己写的爬虫程序，它从网址上爬取之后再解析出文本格式。

Project22

探究题，见project22