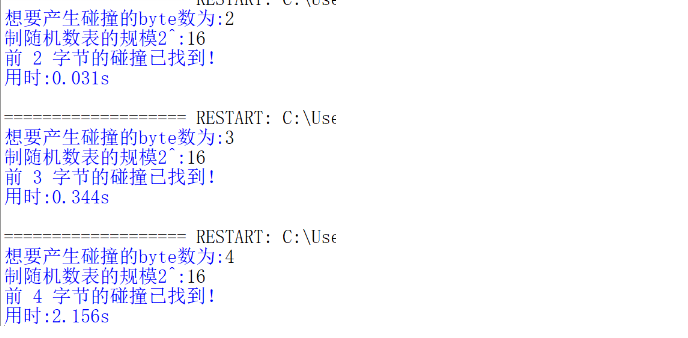
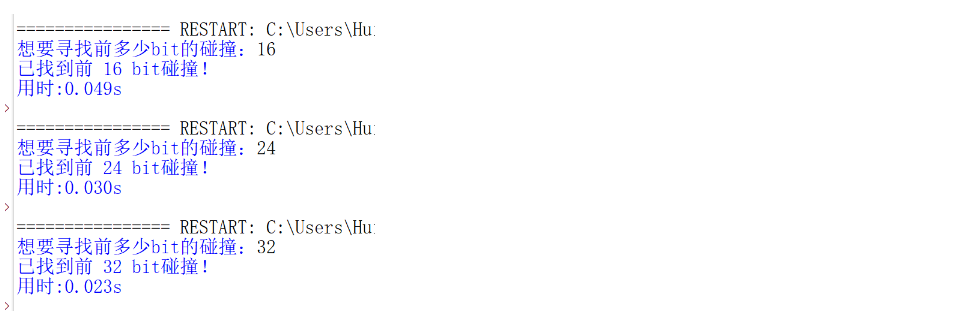
Project1：implement the naïve birthday attack of reduced SM3，借助于制表保存随机生成数据的hash值，通过查找hash值出现的次数来判断是否发生碰撞

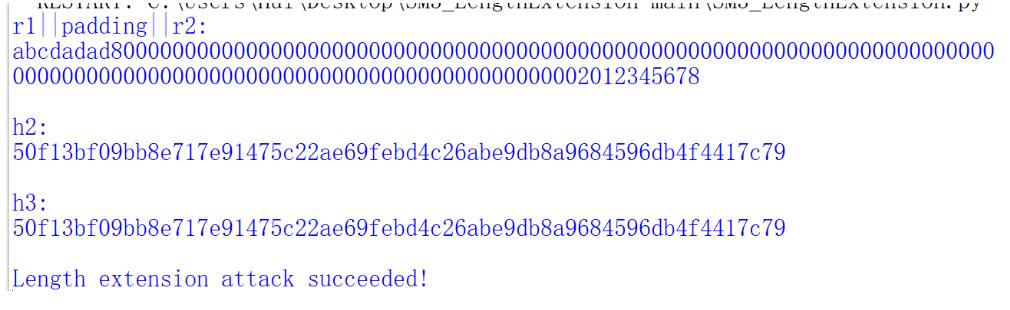
实验截图：



Project2：implement the Rho method of reduced SM3 借助于制表保存随机生成数据的hash值，通过查找hash值出现的次数来判断是否发生碰撞

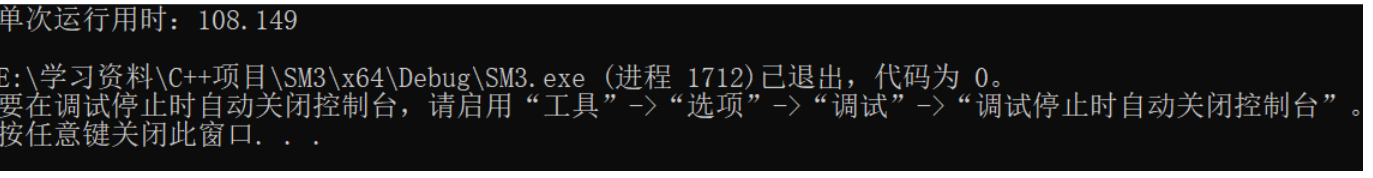


Project3：implement length extension attack for SM3, SHA256, etc. 将初始消息的hash值作为IV输入进下次hash中，即可得到拼接的消息，将此哈希值与待定消息哈希值对比

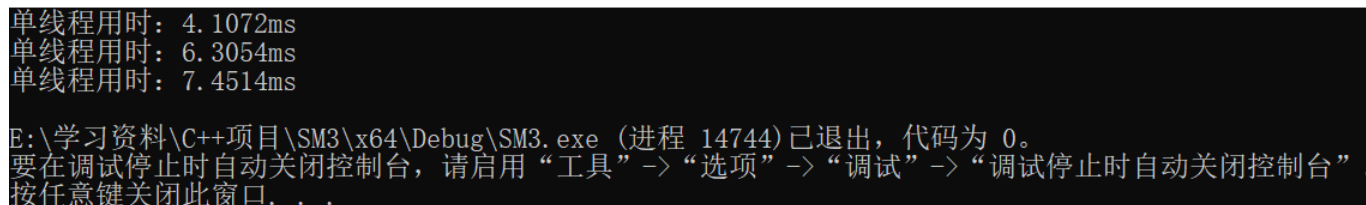


Project4：do your best to optimize SM3 implementation (software) 采用多线程的方法实现SM3加速可以快速提高效率，但是如果设置的进程过多会导致创建进程花费时间过大，由此导致加速效率降低

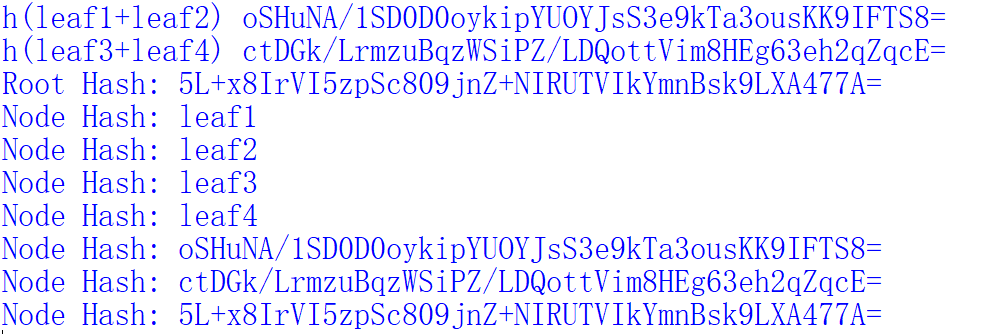
此为SM3原来的用时：



以此为双线程，四线程，八线程用时：



Project5：Impl Merkle Tree following RFC6962 merkle树的实现采用树状结构，消息进行逐层拼接hash

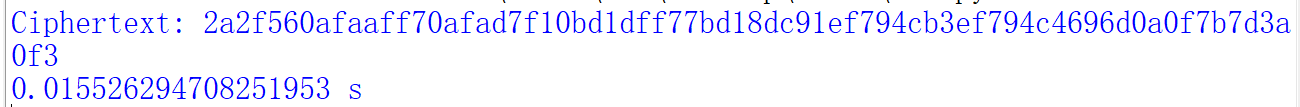




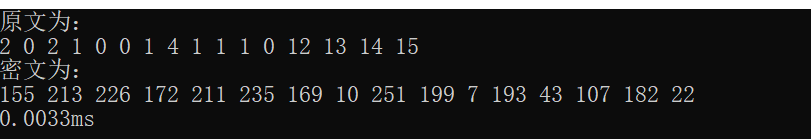
Project8：AES impl with ARM instruction 该实验借用arm编译器实现将写好的C++代码输入arm编译器中即可生成ARM代码

Project9：AES / SM4 software implementation

SM4效率：

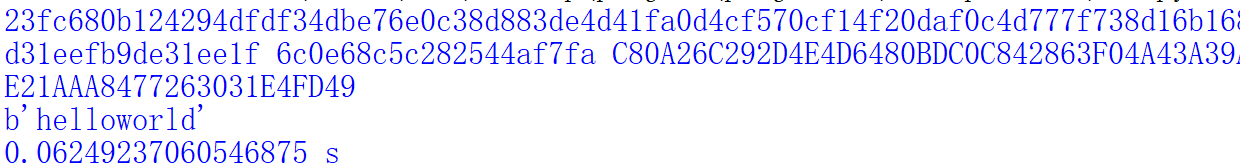


AES效率：

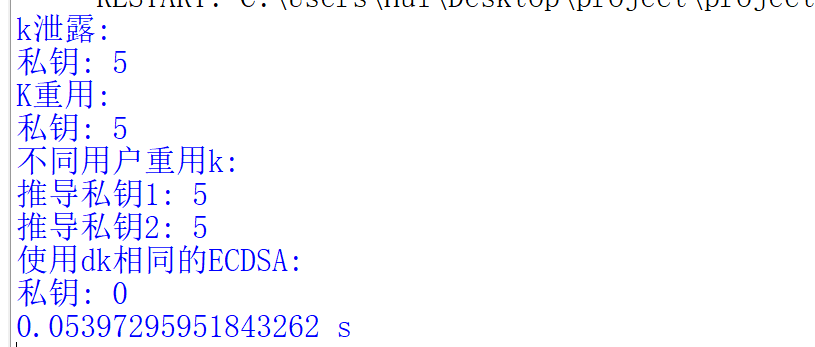


Project10：report on the application of this deduce technique in Ethereum with ECDSA借助于圆锥曲线加密方案实现加密签名策略，应用于区块链技术中

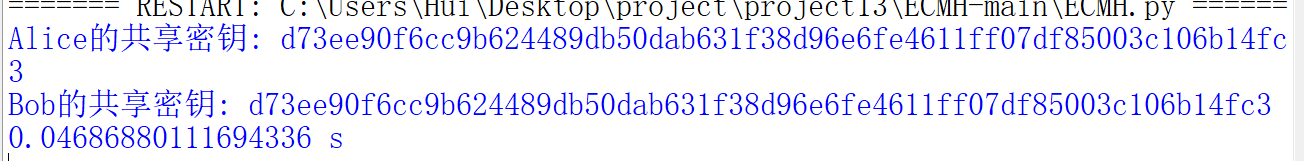
Project11：impl sm2 with RFC6979



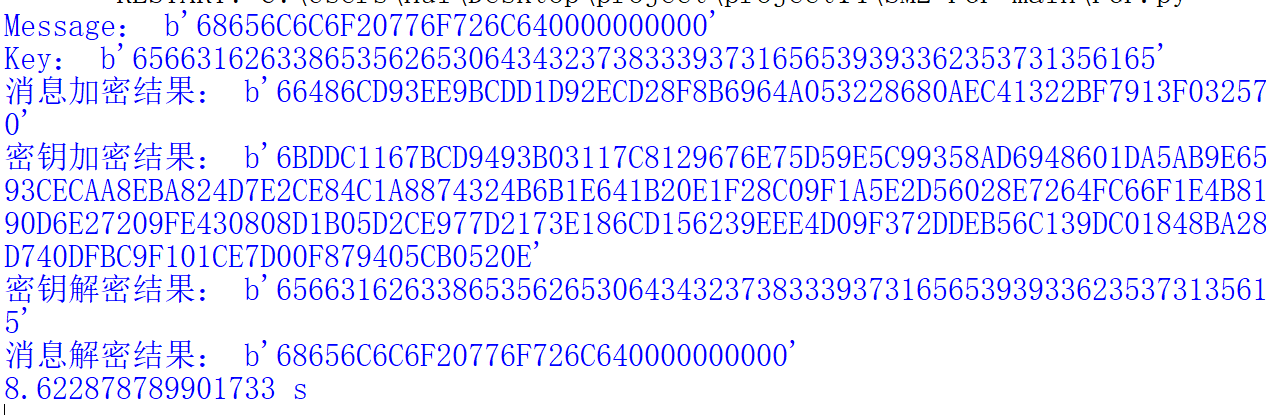
Project12：verify the above pitfalls with proof-of-concept code



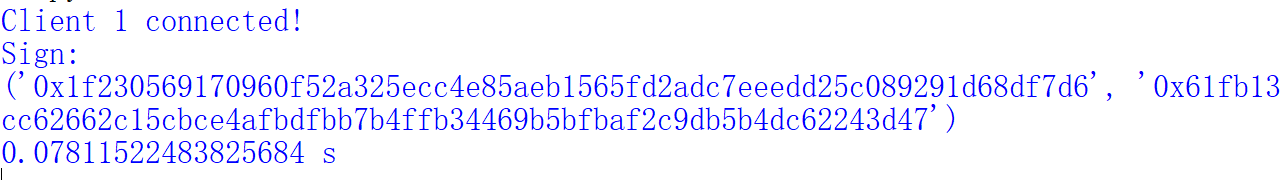
Project13：Implement the above ECMH scheme



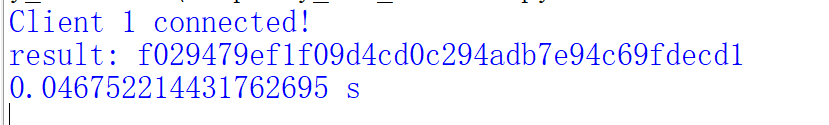
Project14：Implement a PGP scheme with SM2 调用GMSSL库中封装好的SM2/SM4加解密函数。加密时使用SM4加密消息，非SM2加密会话密钥；解密时先使用SM2解密求得会话密钥，再通过SM4和会话密钥求解原消息。



Project15：implement sm2 2P sign with real network communication

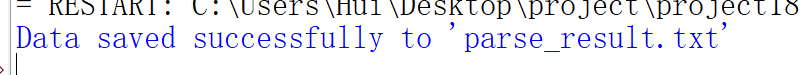


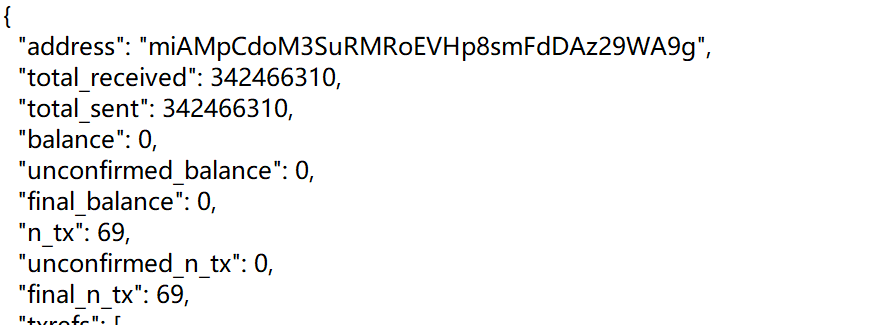
Project16：implement sm2 2P decrypt with real network communication



Project17：比较Firefox和谷歌的记住密码插件的实现区别

Project18：send a tx on Bitcoin testnet, and parse the tx data down to every bit, better write script yourself





Project22：research report on MPT