**密**

**码**

**货**

**币**

**与**

**区**

**块**

**链**

**技**

**术**

**作**

**业**

**报**

**告**

作业名称：《基于盲签名的在线数字货币》

队名：C-moon

队员姓名：钱非凡、贾潇风、陈恺凡

队长：钱非凡

时间：2019/09/21

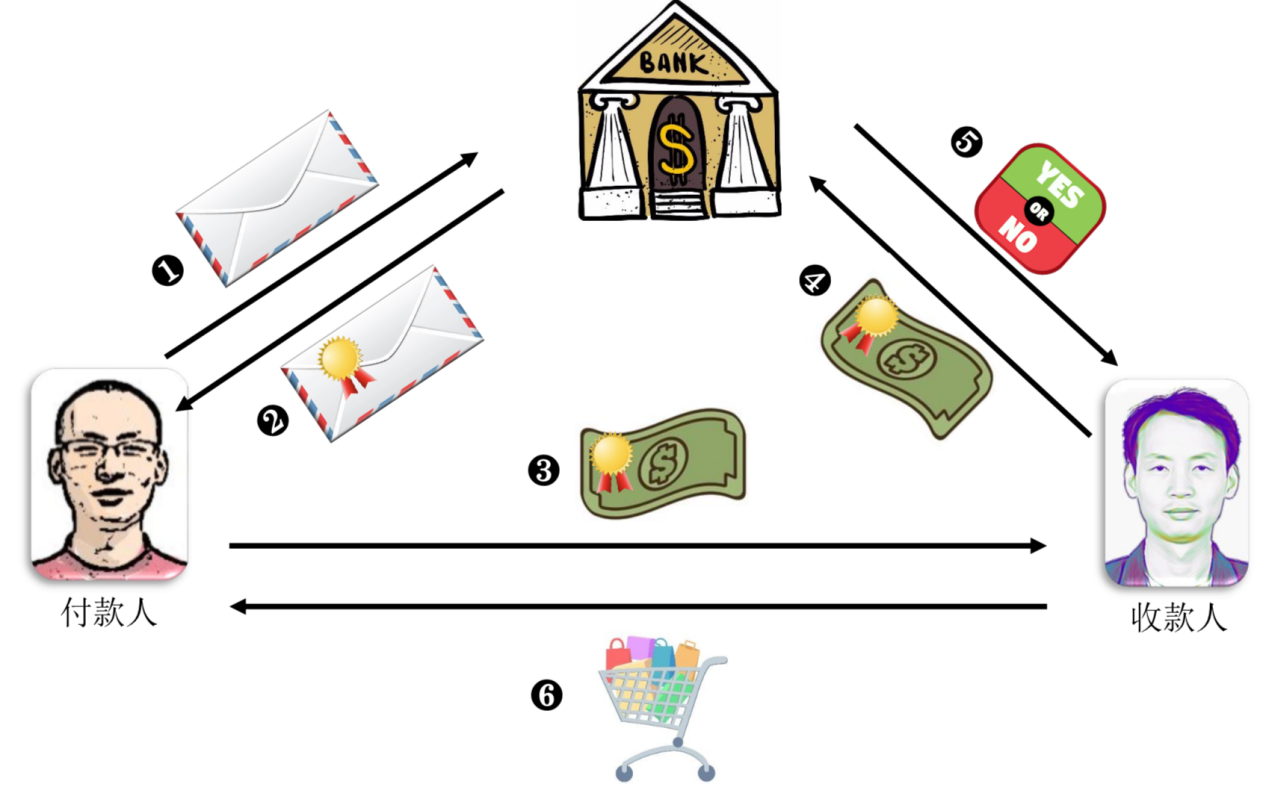
# 一、背景

1.1 《基于盲签名的在线数字货币》背景

盲签名体制是保证参与者匿名性的基本密码协议。自从出现对电子现金技术的研究以来，盲签名已成为其最重要的实现工具之一。

一个盲签名体制是一个协议，包括两个实体:消息发送者和签名者。它允许发送者让签名者对给定的消息签名，并且没有泄露关于消息和消息签名的任何信息。

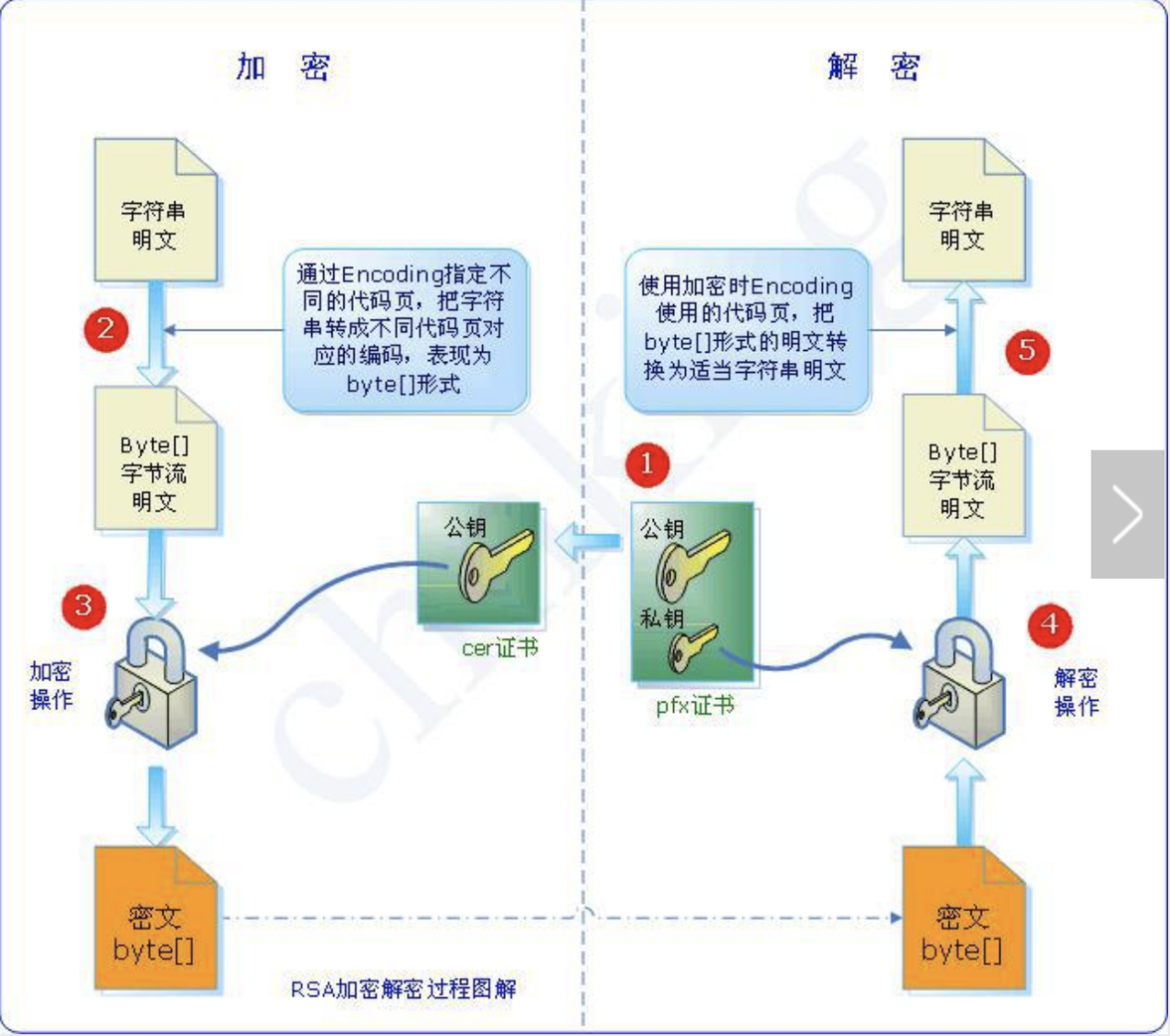
1982年，Chaum首次提出盲签名的概念，并利用盲签名技术提出了第一个电子现金方案。利用盲签名技术可以完全保护用户的隐私权，因此，盲签名技术在诸多电子现金方案中广泛使用。



**图1：盲签名实现的数字货币交易**

1.2 RSA签名算法背景

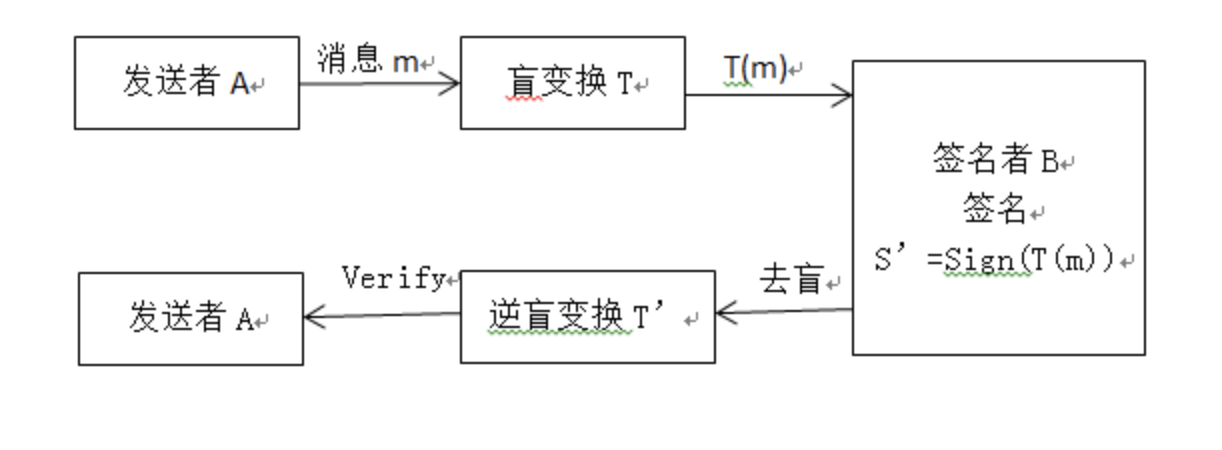
RSA加密算法是一种非对称加密算法。在公开密钥加密和电子商业中RSA被广泛使用。RSA是1977年由罗纳德·李维斯特（Ron Rivest）、阿迪·萨莫尔（Adi Shamir）和伦纳德·阿德曼（Leonard Adleman）一起提出的。对极大整数做因数分解的难度决定了RSA算法的可靠性。



**图2：RSA签名流程**

# 二、基础知识

2.1 RSA盲签名算法介绍



**图3：RSA盲签名流程**

盲数字签名的基本原理是两个可换的加密算法的应用，第一个加密算法是为了隐蔽信息，可称为盲变换，第二个加密算法才是真正的签名算法。此系统用到的是RSA签名算法。

**表1：RSA算法**

|  |  |
| --- | --- |
| 公钥 |  |
| 私钥 |  |
| 密钥对 |  |
| 加密 |  |
| 解密 |  |
| 程序使用（public key） | （为约定的哈希函数类型） |
| 程序所使用（secret key） |  |

1. **系统实现**

3.1 RSA算法实现——、

该部分包含RSA公私钥对生成、签名函数、验证函数三部分。

3.1.1 RSA公私钥对生成

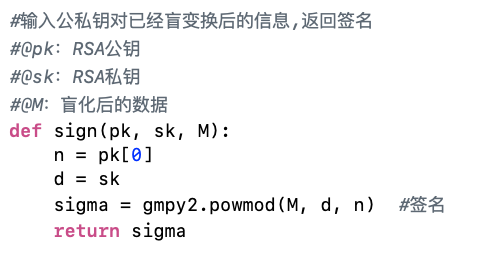
获得随机生成比特长度的素数和，并生成，的值设定为，哈希函数默认为。为模的逆元。



**图4：RSA公私钥对生成**

3.1.1 签名函数

这里使用私钥对数据签名。



**图5：签名函数**

3.1.3 验证函数

使用公钥对获得的签名进行验证。



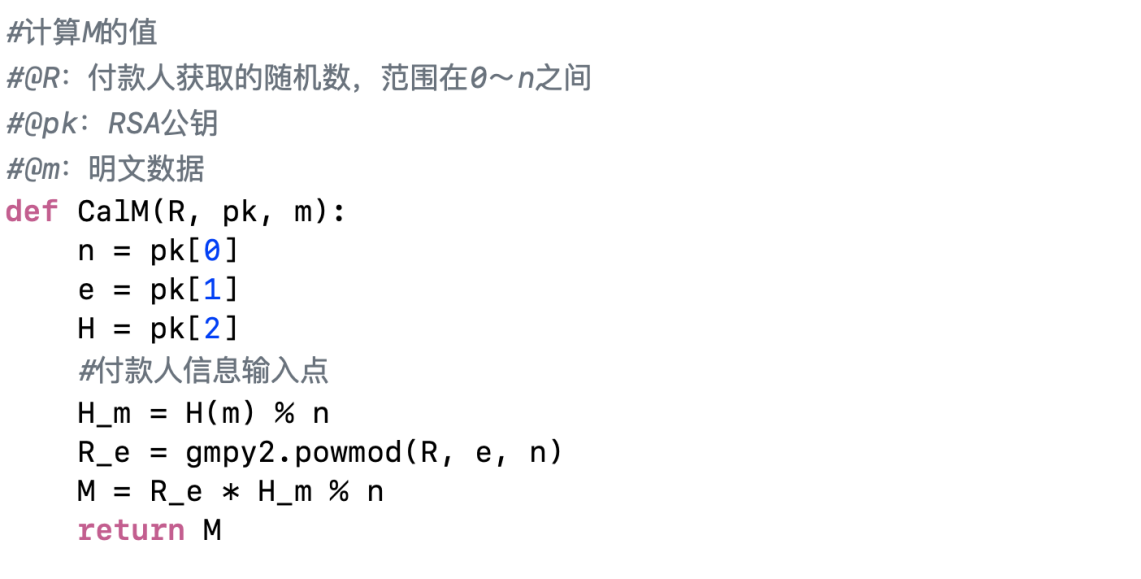
**图6：验证函数**

3.2 盲化与去盲化——client1.py

该部分包含盲化和去盲化两部分。

3.2.1 盲化

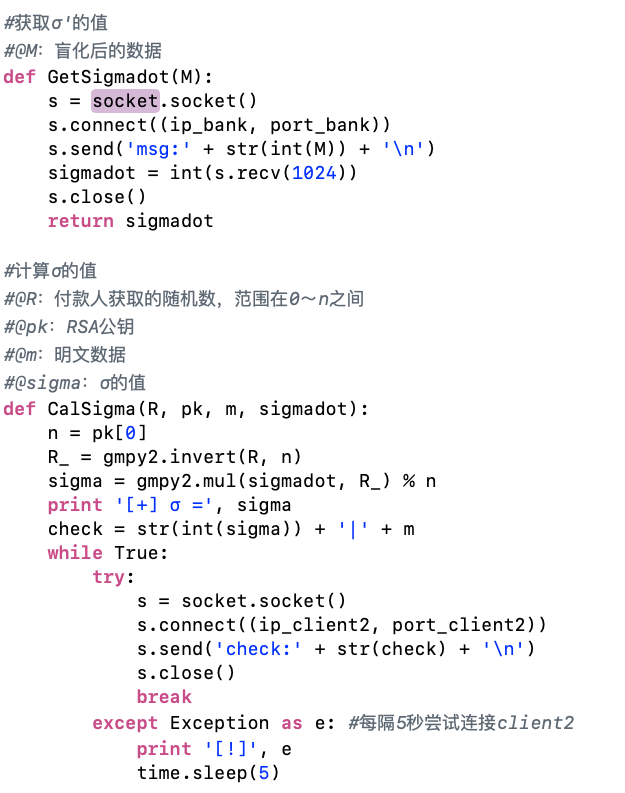
将哈希过后的数据乘上一个随机数的次，得到盲化的结果。



**图7：盲化**

3.2.1 盲化

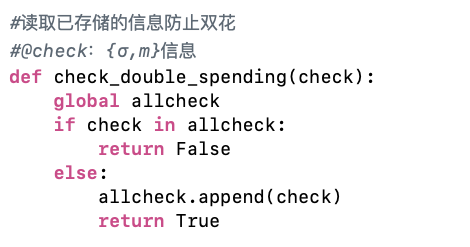
将得到的乘上的逆元去盲，得到签名后的数据。



**图8：去盲化**

3.3 双花检测——bank.py

在allcheck列表中存储了使用过的钱，如果再被使用，返回False；反之，将check加入列表。



**图9：双花检测**

# 四、系统使用

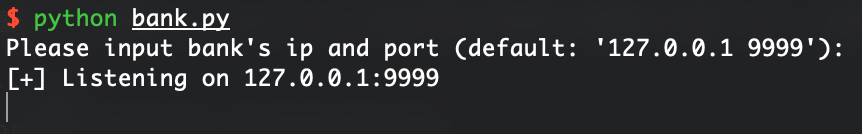
本系统共有三个python程序，分别为用户端（client1.py、client2.py）和银行端（bank.py）。使用套接字socket的形式相互传输所需使用到的数据。其中银行端为服务端，只提供服务（生成公钥、签名、检测双花）。client2.py同时具有用户端和服务端功能（向client1.py提供验证服务）。

**逻辑流程：**

1. bank.py作为服务端，生成公私钥对，并公开公钥对提供用户端使用。
2. client1.py对原消息进行盲变换后得到。同时将发送给服务端bank.py。
3. bank.py获得后进行签名，并将签名后的信息σ’返回给用户端client1.py。
4. client1.py获得签名信息后对其进行去盲变换，得到，将发送给client2.py。
5. client2.py从client1.py处获得，从bank.py获得公钥对。使用公钥对进行签名验证。验证成功后，再将发送给bank.py进行双花验证。
6. 服务端bank.py收到后验证双花，若无双花，则将该数据保存于文件中。

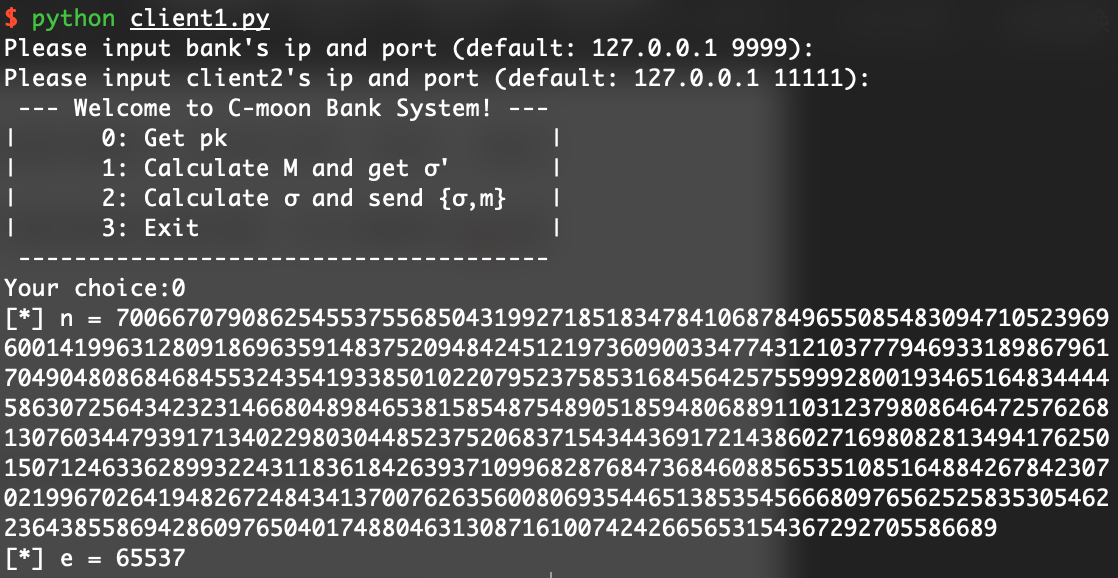
**使用说明：**

1. 三个程序都是使用python2代码编写。用到了gmpy2库，使用前请确认拥有该库或使用pip install 安装该库。
2. 首先运行bank.py，输入ip和端口来开启服务端服务,缺省值已标注。

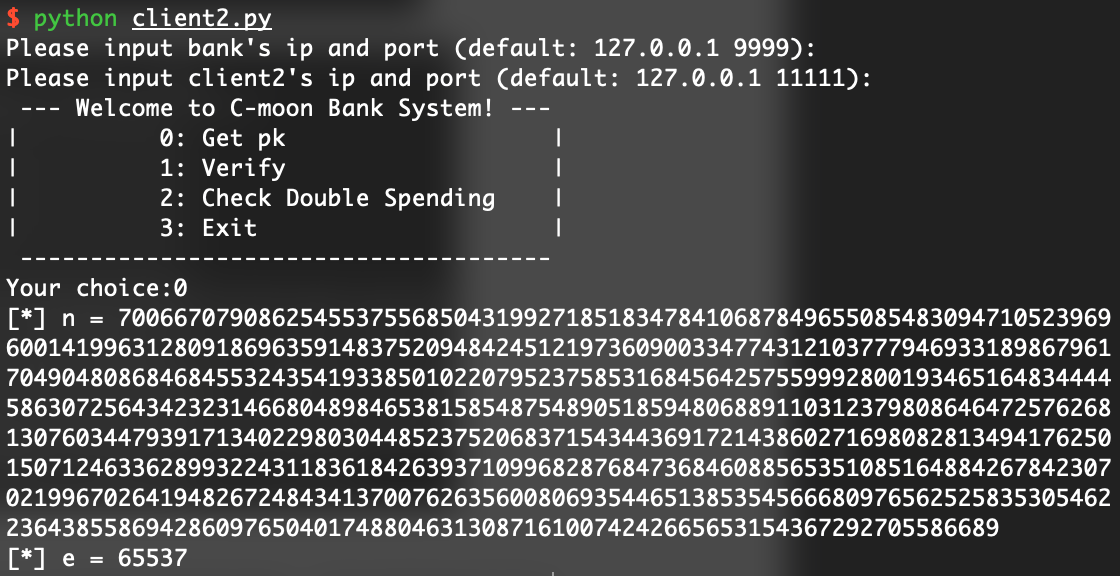


**图10：运行bank.py**

1. 打开client1.py和client2.py来设置对方以及bank.py进程的ip、端口号。缺省值已标注。

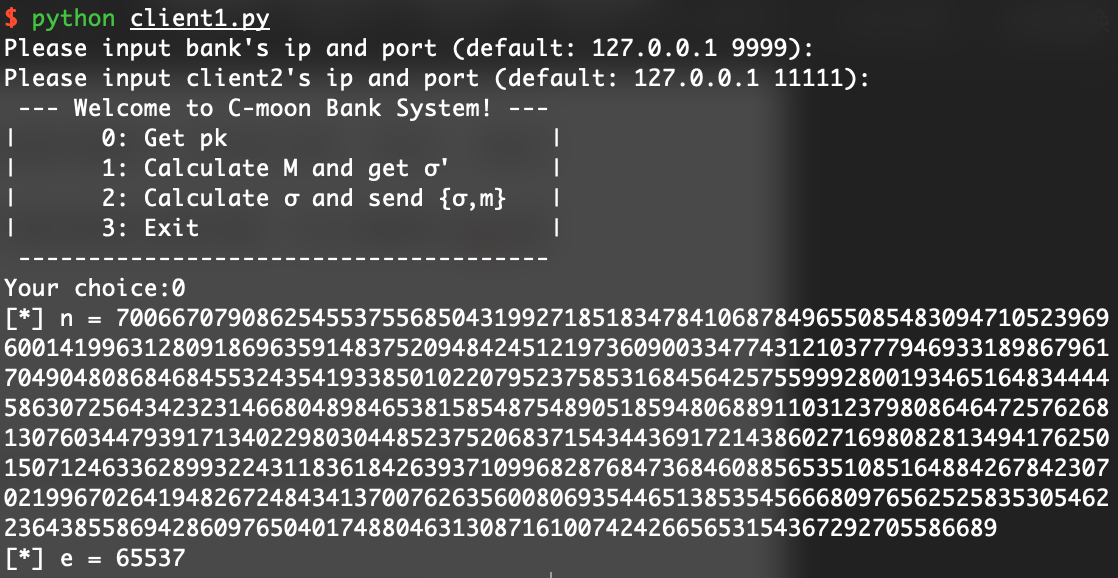


**图11：运行client1.py**

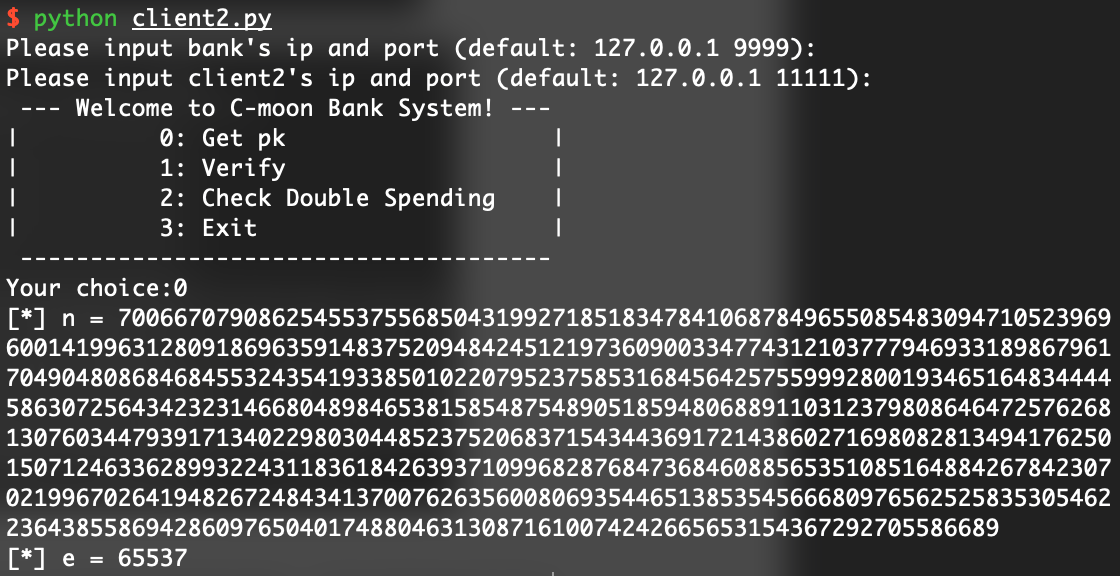


**图12：运行client2.py**

1. 用户端输入0即可获取公钥，在执行任何其他功能以前，都必须先获取公钥。

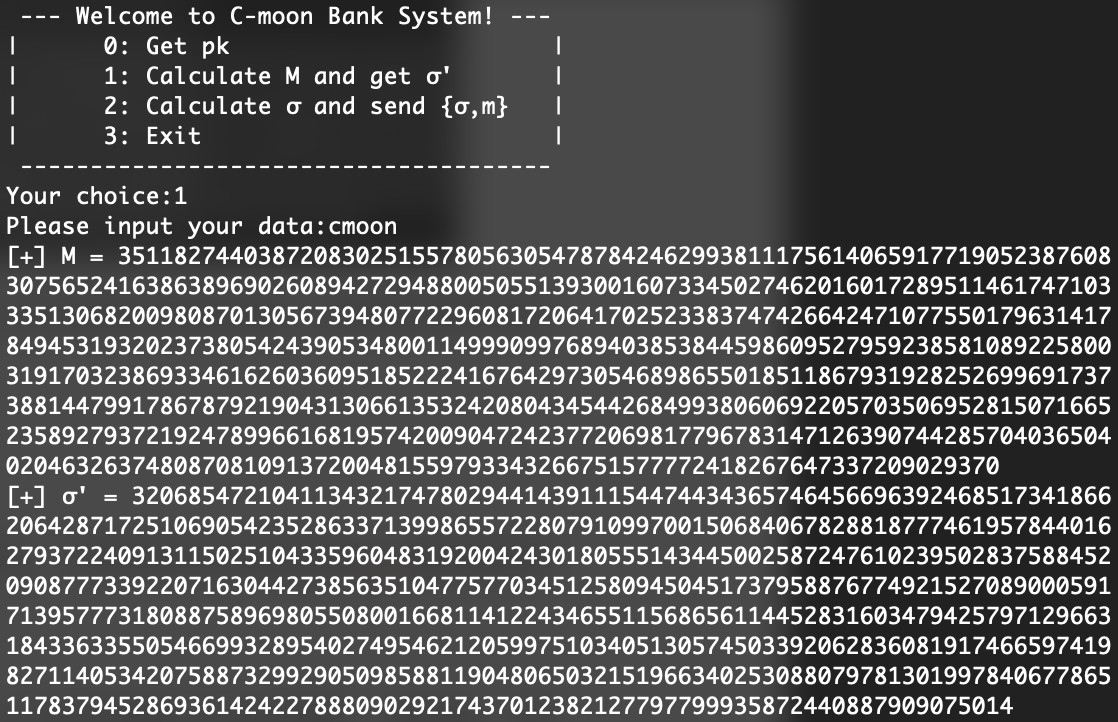


**图13：client1获取公钥**



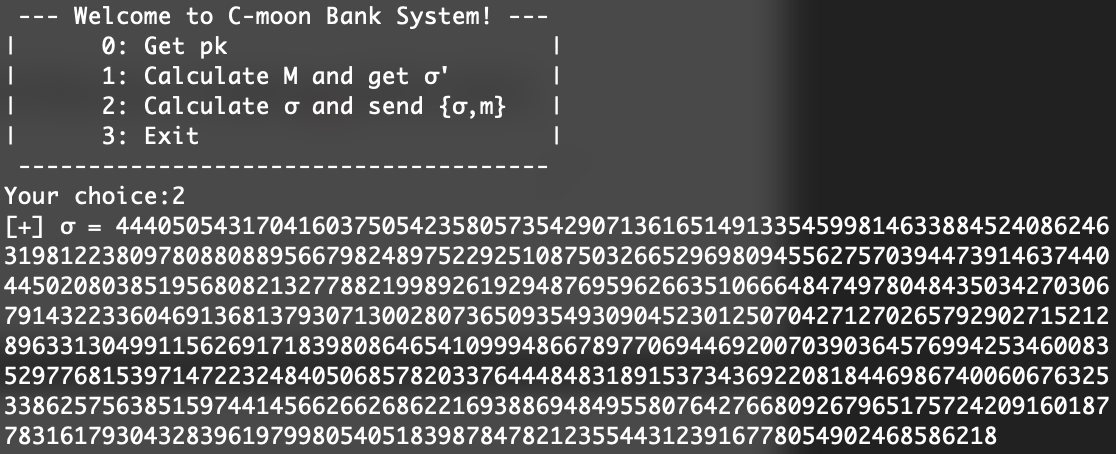
**图14：client2获取公钥**

1. 用户端client1.py（付款人）输入1，输入需要签名的数据，盲化后发送给银行端。银行端盲签名后返回。



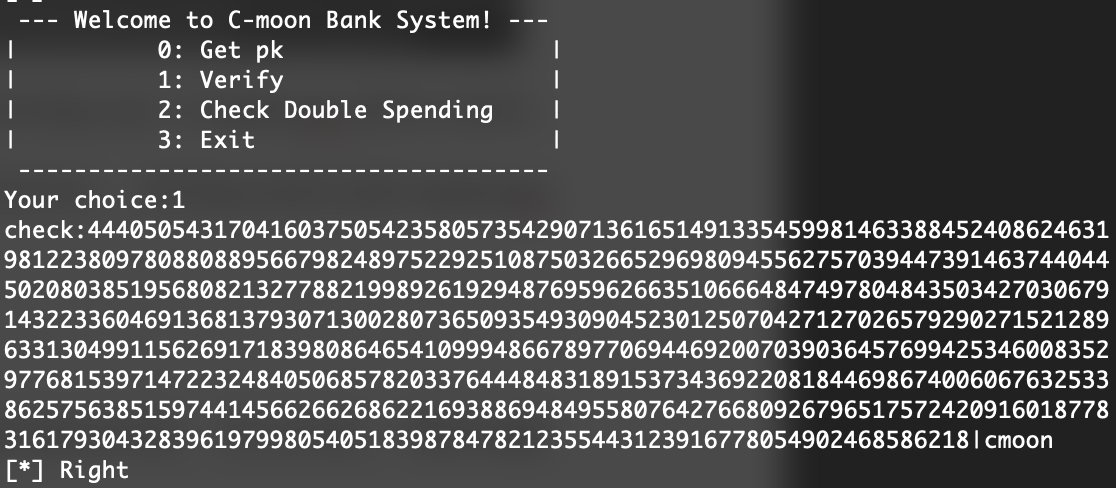
**图15：client1盲化生成并接收银行端返回的**

1. 付款人client1.py输入2，计算得出并发送给收款人client2.py。



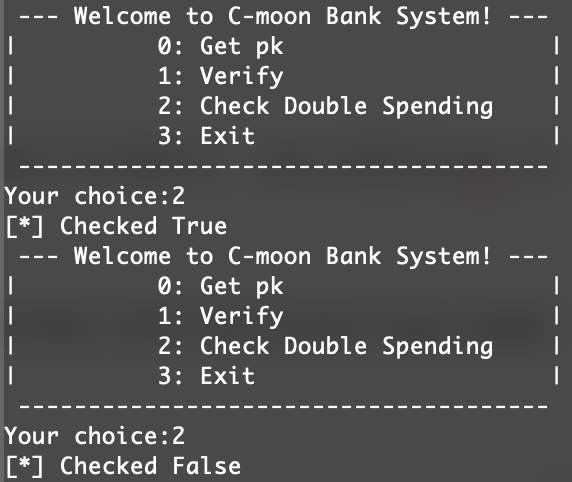
**图16：client1计算出并将发送给client2**

1. 收款人client2.py接收并验证签名是否为真。



**图17：client2接收并对签名进行验证**

1. 收款人client2.py将发送给银行检验是否双花。检验过一次后将的值存入“数据库”，再次检测将认为双花。



**图18：client2将发送给银行检验双花**

# 五、总结

在对签名后的进行去盲变换的时候，除以取余的结果总是不对，后来才发现在取余运算中的除法其实是乘上的逆元，修改后结果正确，这使我们认识到了自己基本功的不足。