《漏洞利用及渗透测试基础》实验报告

姓名：管昀玫 学号：2013750 班级：计科一班

**实验名称：**

基于Paillier算法实现隐私信息获取

**实验要求：**

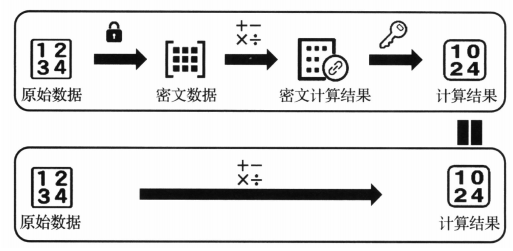
从服务器给定的m个消息中获取其中一个，不得向服务器泄漏获取了哪一个消息，同时客户端能完成获取消息的解密。

扩展实验：

在客户端保存对称密钥k，在服务器端存储m个用对称密钥k加密的密文，通过隐私信息获取方法得到指定密文后解密得到的明文。

**实验过程：**

同态加密(HE，homomorphic encryption)是一种加密算法，它可以通过对密文进行运算得到加密结果，解密后与明文运算的结果一致。



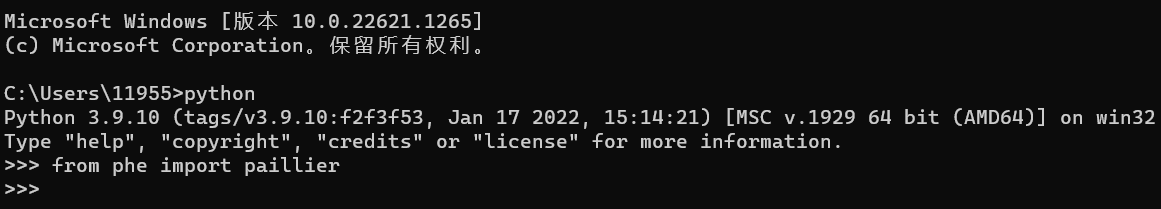
同态加密主要基于公钥密码体制构建，它允许将加密后的密文发给任意的第三方进行

计算，并且在计算前不需要解密，可以在不需要密钥方参与的情况下，在密文上直接进行计

算。

半同态加密(partial homomorphic encryption, 简称 PHE)仅支持单一类型的密文域同态运算(加或乘同态)。注意：加法和乘法同态是相对明文而言所执行的操作，而非密文上执行的运算形式。

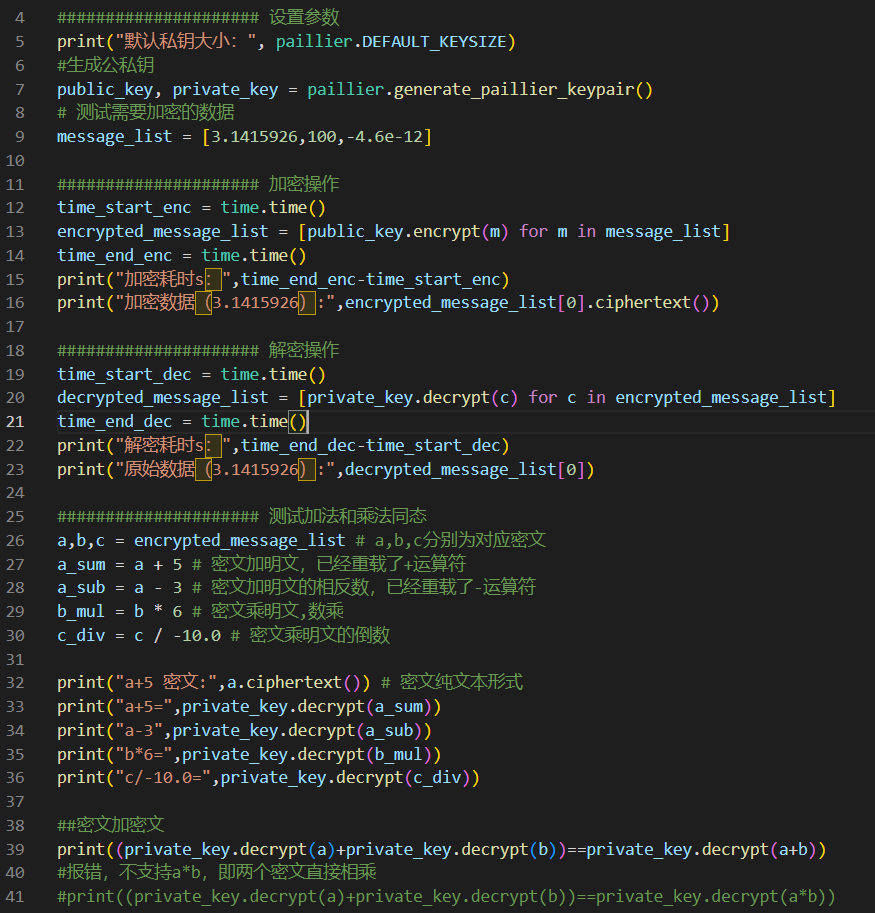
1. 安装pyton和phe库，并验证环境正确性



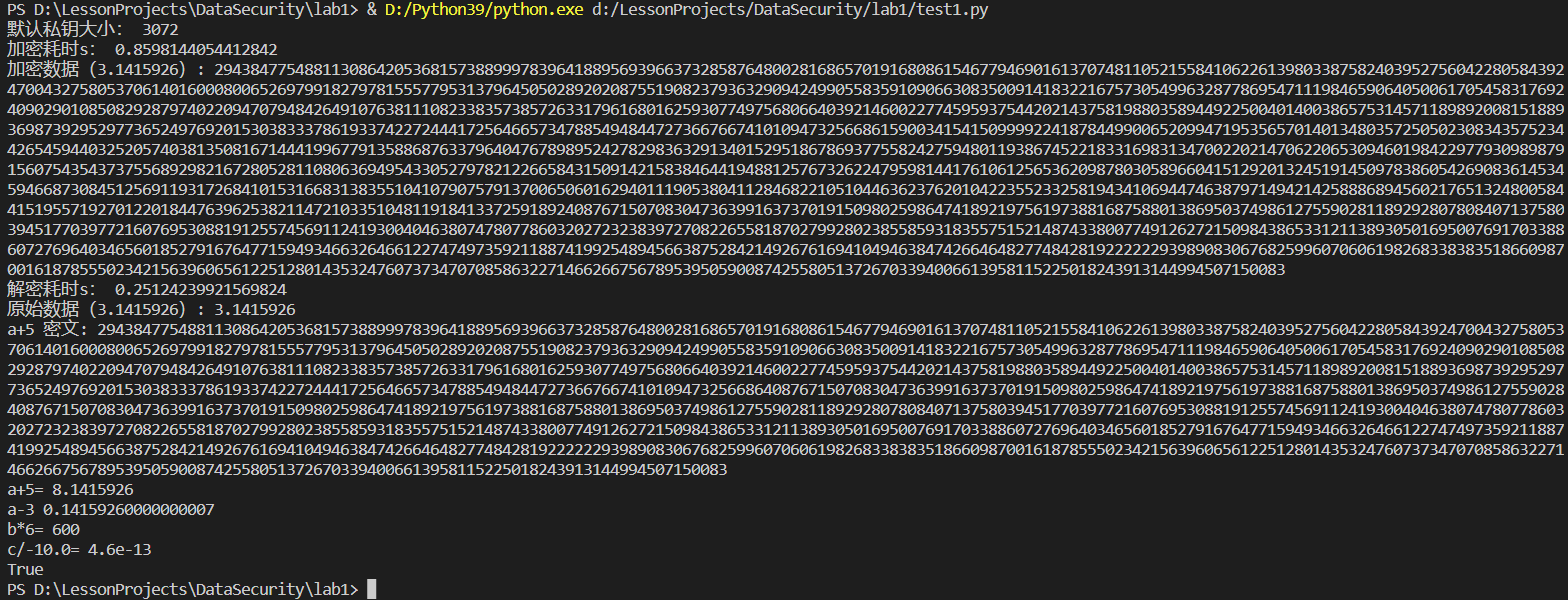
不出现报错信息，则说明安装成功。

1. 运行test1.py，并观察结果

test1.py如下所示：



运行结果如下所示：



程序首先设置了需要加密的数据，生成公钥和密钥，然后分别开始加密和解密。

然后测试加法和乘法同态，分别为密文加明文、密文加明文的相反数、密文乘明文、密文乘明文的倒数。

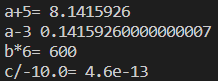
如上述代码所示：第一，python 程序对运算符进行了承载，已经支持直接密文上的运算。第二，只支持明文的加法，不支持明文的乘法，最后一句如果将注释符去掉，将报错。

在此程序中，默认私钥大小为3072，加密耗时为0.8498485088348389 s，解密耗时为 0.26146554946899414 s。可以看到，程序成功解密：



之后程序分别输出了a+5, a-3, b\*6, c/-10.0加法和乘法同态结果：

(a, b, c)分别为[3.1415926,100,-4.6e-12]



在这里可以看出，在密文上相加、相减、相乘、相除与在明文上做同样运算的结果相同，验证了加法与乘法的同态。需要注意的是，a-3小数的尾数有改变，这是由于精度问题。

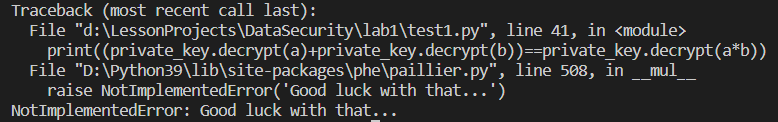
之后测试密文解密加密文解密的结果，测试其是否等于密文直接相加然后解密的结果，实验证明了其相等。

print((private\_key.decrypt(a)+private\_key.decrypt(b))==private\_key.decrypt(a+b))

>>>True

但是需要注意，两个密文直接相乘是不允许的，程序执行后报错：

print((private\_key.decrypt(a)+private\_key.decrypt(b))==private\_key.decrypt(a\*b))



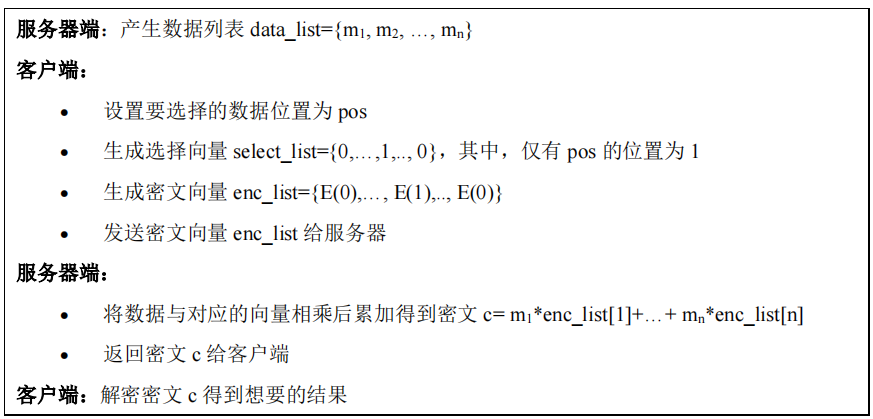
1. 扩展实验

基于 Python 的 phe 库完成隐私信息获取的功能：服务器端拥有多个数值，要

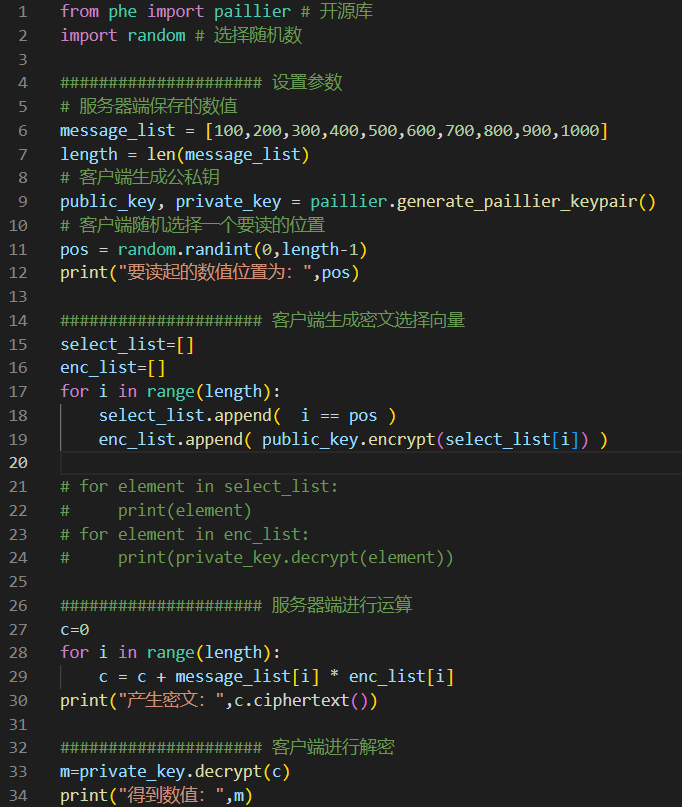
求客户端能基于 Paillier 实现从服务器读取一个指定的数值并正确解密，但服务器不知道所

读取的是哪一个。

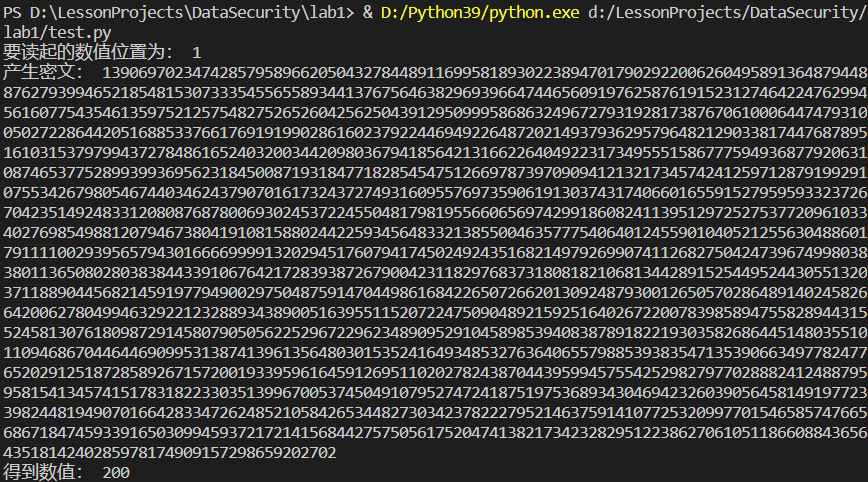
基于 Paillier 协议进行设计：对 Paillier 的标量乘的性质进行扩展，我们知道：数值“0”的密文与任意数值的标量乘也是 0，数值“1”的密文与任意数值的标量乘将是数值本身。



具体程序如下所示：



程序执行的结果如下：



程序首先设置数据列表为[100,200,300,400,500,600,700,800,900,1000]，然后随机选择一个位置（此处为1，即200），并生公钥和密钥。之后，产生一个list，把选中的位置设置为1，其他为0.客户端将整个数据序列都生成密文，服务器进行运算时将密文选择向量与密文原文相乘，得到一个c，这个c便是选中的位置的数据所生成的密文，将其解密就能得到所要的数值。

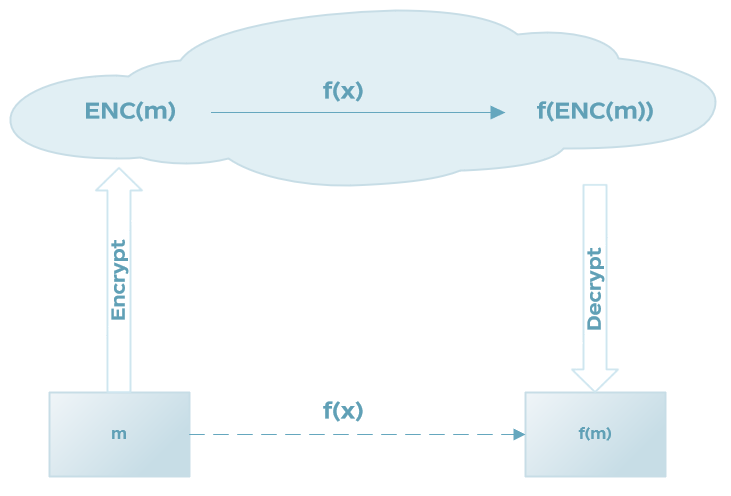




实验得到以下结果，说明实验成功。

**心得体会：**

通过此次实验，我对同态加密有了更深的理解，学会了使用Paillier方案生成明文与密文，并学习到其中的数学原理——卡迈尔克函数与判定复合剩余假设。



半同态加密有一个很典型的应用场景：**连邦学习**。假设Alice和Bob想共同训练一个网络模型，Alice和Bob各自持有一部分训练数据，并且他们不想把自己的数据泄露给对方。那么在训练期间，Alice和Bob需要交互各自训练的梯度数据，并根据双方的梯度数据，共同计算一个对双方都合适的梯度值，用来执行联合梯度下降过程。

2019年，Ligeng Zhu等人发表的“Deep Leakage from Gradients”论文中给出了一种算法，可以从几次迭代的梯度数据中，推断出训练的数据，标签，模型等一系列隐私信息。这使得在分布式机器学习中，通过传输梯度数据来进联合模型训练变得不再安全。那么如果在梯度数据传输的过程中，传输的是加密后的梯度数据，并且这些加密数据可以进行二次计算，那么便可以规避梯度数据传输过程带来的安全风险。

还有一个典型应用场景是**隐私集合求和**。在在线广告投放的场景中，广告主（如商家）在广告平台（如媒体）投放在线广告，并希望计算广告点击的转化收益。然而，广告点击数据集和购买数据集分散在广告主和广告平台两方。使用加法 PHE 结合隐私集合求和（Private Intersection-Sum-with-Cardinality, PIS-C)协议可以在保护双方隐私数据前提下，计算出广告的转化率。协议中的“隐私保护求和”功能依赖于广告主将自己的交易数据用 PHE 加密发送给广告平台，使得广告平台在看不到原始数据的前提下，完成对交集中数据金额的聚合。该方案已被 Google 落地应用。