pailliar.py：生成json格式的用于同态加密的秘钥对

Cloud1.py：发送私钥给Cloud2。

Cloud2\_savekey.py：保存Cloud1发送的私钥到本地的privkey.pub

Cloud2\_dec.py：解密Cloud3传过来的密文。

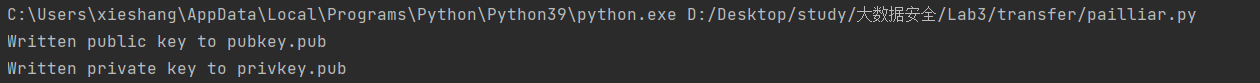
Cloud3.py：接收user.py发送的成绩的密文，并进行计算。

user.py：发送成绩给Cloud3。

本机演示环境：USER、Cloud1为本机(演示用)，Cloud2为Ubuntu 18.04，Cloud3为Kali。

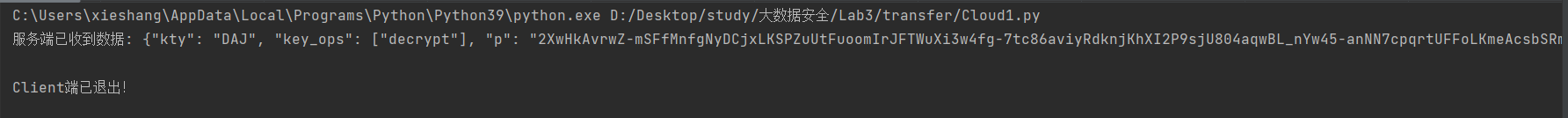
两个Linux端依赖都配好了。

1. 在Cloud1上运行pailliar.py生成key，将publickey复制到机器的代码对应目录下。



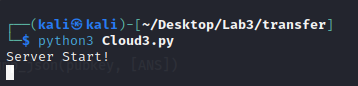
1. 运行Cloud2\_savekey.py和Cloud1.py 将生成的key发送给Cloud2并删除本地的privatekey。

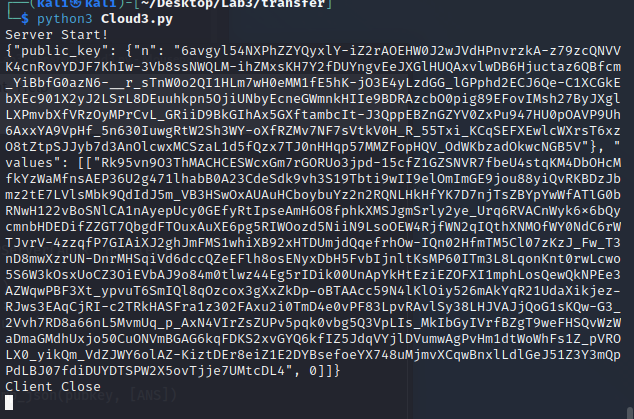
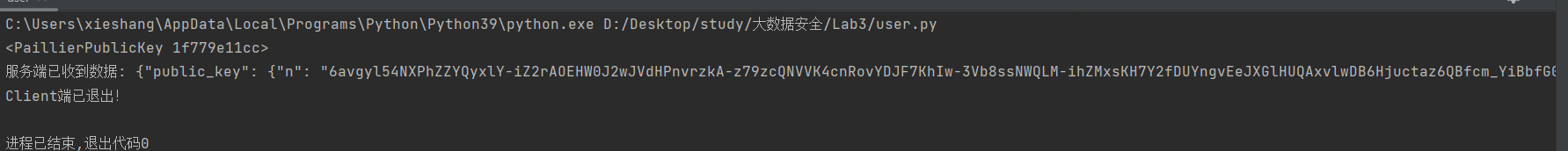


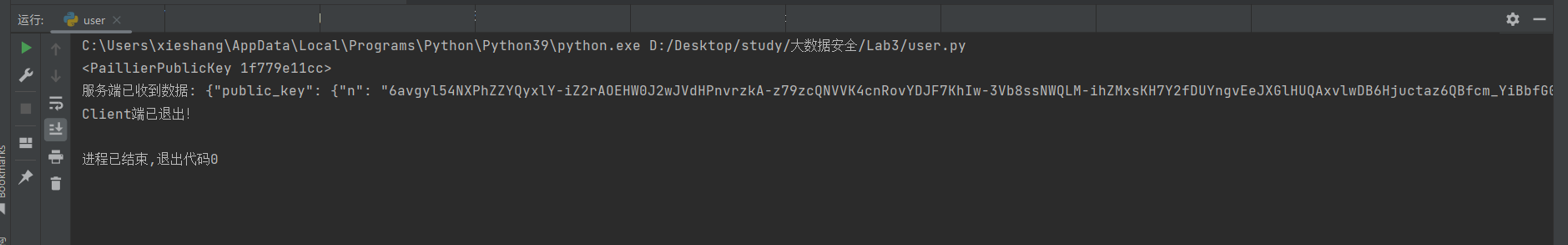




1. 在Cloud2上运行Cloud2\_dec.py。在Cloud3上运行Cloud3.py，用户运行user.py将用公钥加密的数据上传给Cloud3.py，Cloud3进行计算。然后返回加密数据给Cloud2，Cloud2输出结果。









传入了两个31，结果为62，成功。

初始需要的key：

Cloud1需要加密信道的公钥。

Cloud2需要加密信道的公私钥。

Cloud3需要加密信道的公私钥。