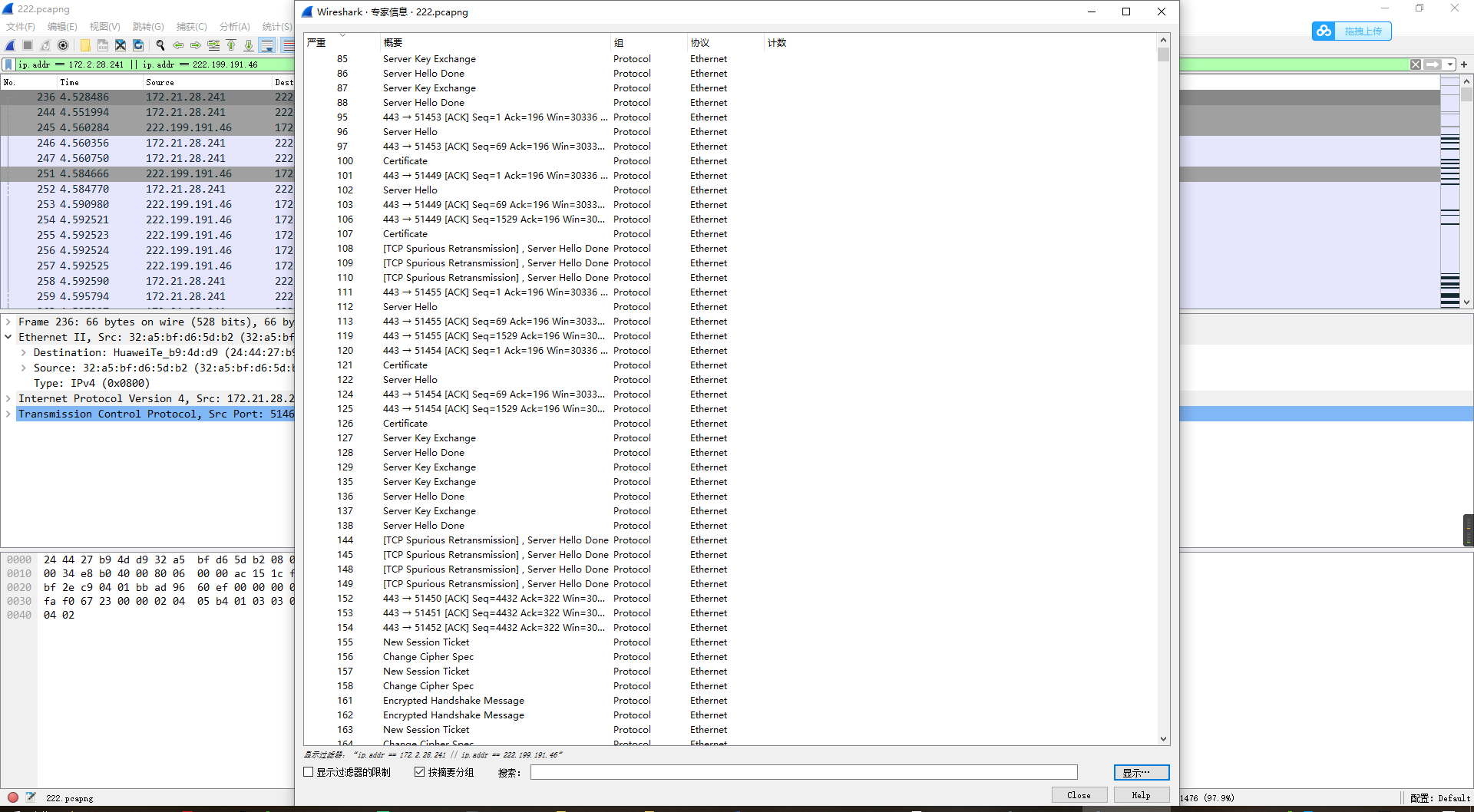
背景：

原本想做个网盘，“污蔑”一下百度云传输不安全，可以让人抓包还原数据。结果



它挺安全的，传输加密了。

那我就不做网盘了，做个第三方存储信息，客户端还原的“秘密线人对话装置”吧。

思路：

做两个电脑应用，一个充当客户端，另一个充当服务器。

服务器在启动时会要求管理员设置一个口令用来验证客户端登录。

客户端在输入只有自己可知的服务器网址，以及约定好的口令后，可以阅读/新建 服务器所存储的信息。

信息只有在用户端解密后才可以浏览，否则会是乱码，甚至读不懂文件名。

具体加密方式：

通过Diffie-Hellman算法，客户端与用户端交互得到公钥K，作为AES的密钥，用来传送文件以及加密解密。每次打开服务端，公钥K就会重新生成，保证了信息的有效信，可靠性。这种加密方式安全性是基于离散对数的难解性。

语言环境：

Python 3

用到的库：

tkinter(python2.5版本后自带，用于创建客户端界面)

re(借用一下正则匹配, 用来过滤网址，端口)

socket(python通讯)

hashlib(实现密码的散列发送)

random(用来在diffie-hellman中产生随机数)

threading(实现服务器端多进程通讯)

getpass(实现服务器端输入预制密码的功能)

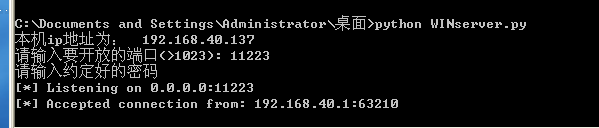
psutil(如果服务器是linux, 需要一个库来获取ip)

pycryptodome(用来实现win下aes加密)

pycrypto(用来实现linux下aes加密)

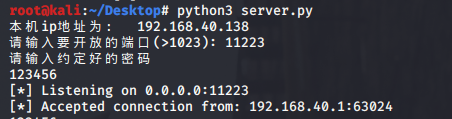
第一步实现:

WINserver 与 client 对话



成功

LINUXserver 与 client 对话



成功

第二步实现:

在客户端实现对服务器端的信息文本进行读取修改操作

第三步：

文本数据加密传输 在本地进行解码

参考网址:

[PythonGUI](https://blog.csdn.net/qq_38251616/article/details/80670714?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-3.compare&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-3.compare)

[正则匹配](https://www.cnblogs.com/chaoqi/p/11215866.html)

[Python获取本机ip](https://jingyan.baidu.com/article/ca00d56c243b54e99eebcfb8.html)

[Python获取多个网卡的主机ip](https://blog.51cto.com/walkerqt/1686735)