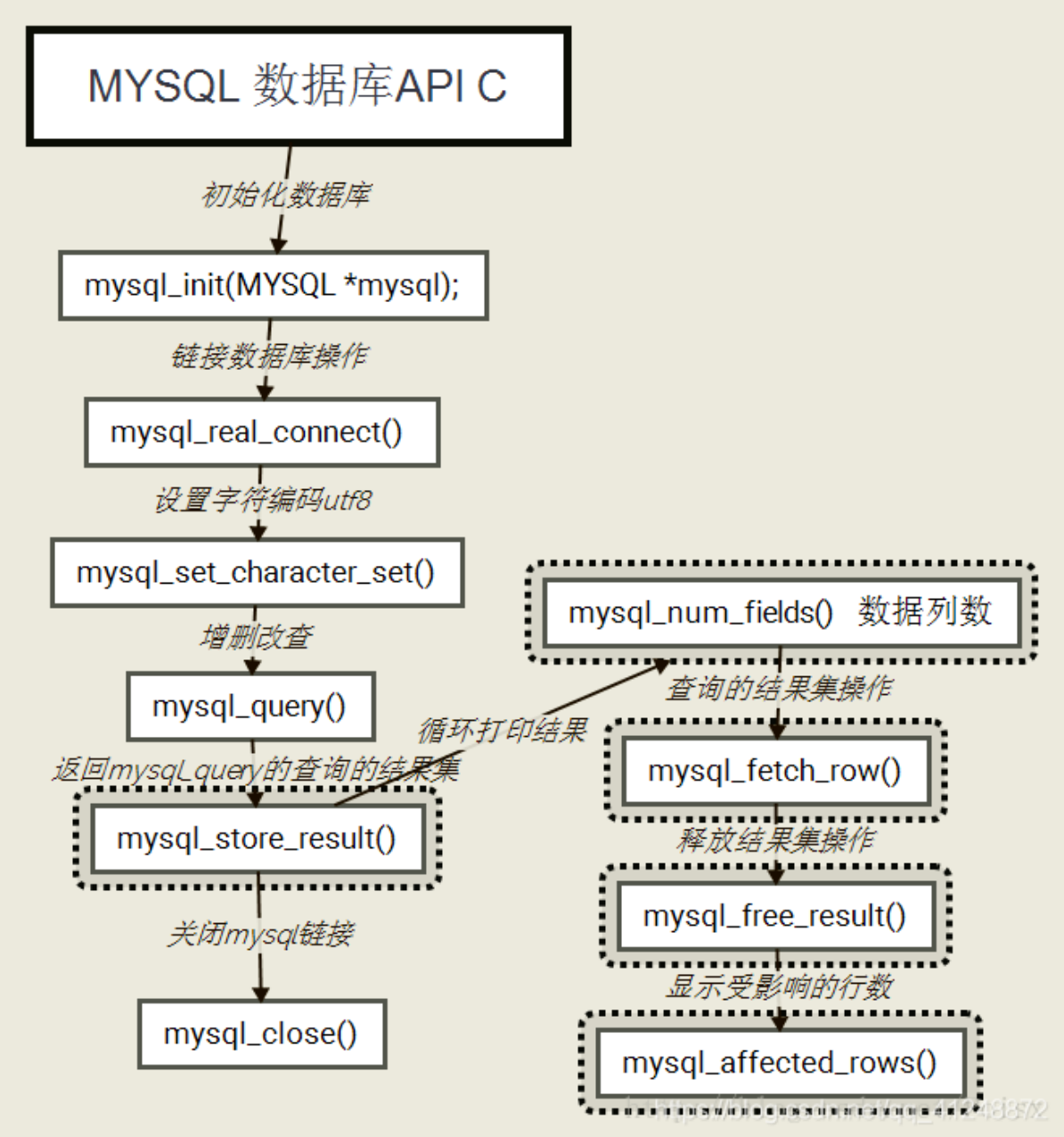
数据加密实训

数据存储加密

安装和使用数据库mysql



这里直接使用的小皮面板自带的mysql管理工具。

建立两个表管理用户和密钥。

```sql
-- 用户表，存储加密的数据
CREATE TABLE users (
    id INT PRIMARY KEY,                 -- 用户 ID
    username VARCHAR(50),               -- 用户名
    password VARBINARY(255)             -- 加密的密码
);

-- 密钥表，存储解密密钥
CREATE TABLE miyao (
    id INT PRIMARY KEY,                 -- 与用户表的 ID 对应
    secret_key VARCHAR(255)             -- 密钥
);
```

加密数据

```sql
SET @secret_key = SUBSTRING(MD5(RAND()) FROM 1 FOR16);
INSERT INTO users (id, username, password)
 VALUES (1, 'xiaowang', AES_ENCRYPT('123456', @secret_key));
```

解密数据

```sql
SELECT secret_key INTO @secret_key FROM miyao WHERE id = 1;
SELECT id, username, AES_DECRYPT(password, @secret_key) AS decrypted_password
 FROM users
 WHERE id =1;
```

磁盘加密（暂时做保留-设备不支持）

数据传输加密实训

P2P通信，即点对点通信，是一种网络通信模式，其中两个或多个计算机直接连接并进行数据交换，而不需要中央服务器的介入。这种通信模式允许网络中的每个节点既是客户端又是服务器，从而实现资源的分布式共享。

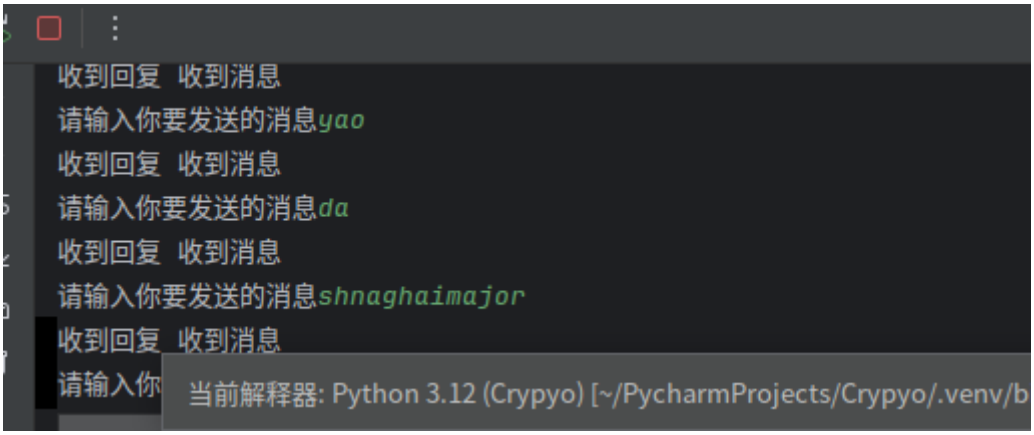
1. 一些Python代码的作用

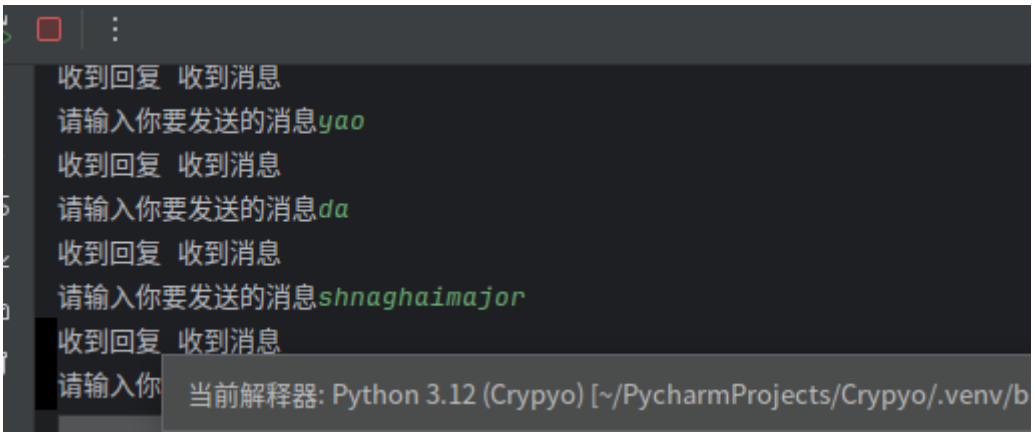
```
import socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)#创建套接字
#socket.AF_INET：表示使用 IPv4 地址。
#socket.SOCK_STREAM：表示使用 TCP 协议（面向连接，可靠传输）。
```

所谓套接字 (Socket)，就是**对网络中不同主机上的应用进程之间进行双向通信的端点的抽象。** 一个套接字就是网络上进程通信的一端，提供了应用层进程利用网络协议交换数据的机制。 从所处的地位来讲，套接字上联应用进程，下联网络协议栈，是应用程序通过网络协议进行通信的接口，是应用程序与网络协议栈进行交互的接口 [1]

```
client_handler = threading.Thread(target=handle_client, args=
(client_socket,))#指定线程运行目标函数，利用args=(client_socket,)将一个元组传递给目标函数
client_socket.sendall(reply.encode('utf-8'))#：Socket 通信只能发送字节数据，而字符串需要先编码为字节。
#.sendall()发送完整的数据，直到所有内容都被传输完成。如果发送失败，会抛出异常。

```



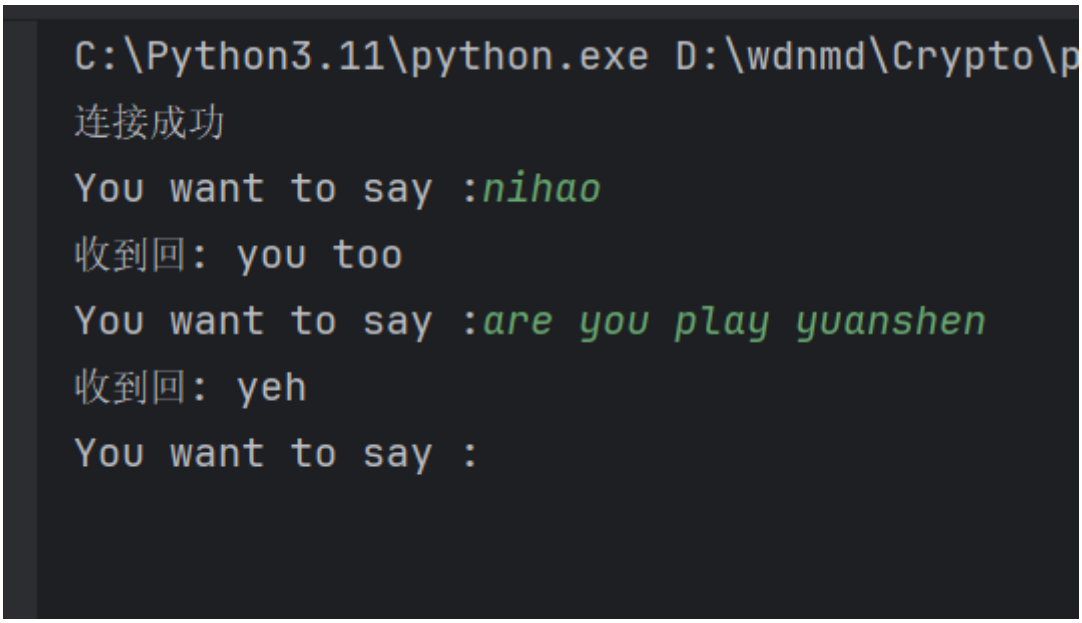


数据rsa加密简单实践，注意多线程运作时，client_socket.close()关闭连接会引发一些错误。所以我舍弃了关闭连接。，当然这只是在pycharm的测试中，接下来尝试使用

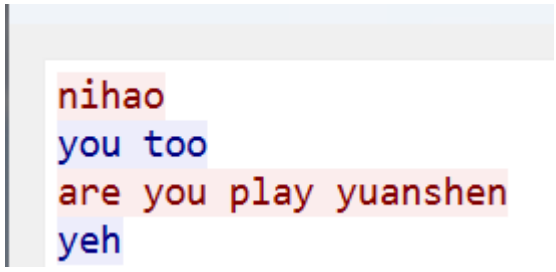
p2p协议

1. 不含加密

连接后进行通信

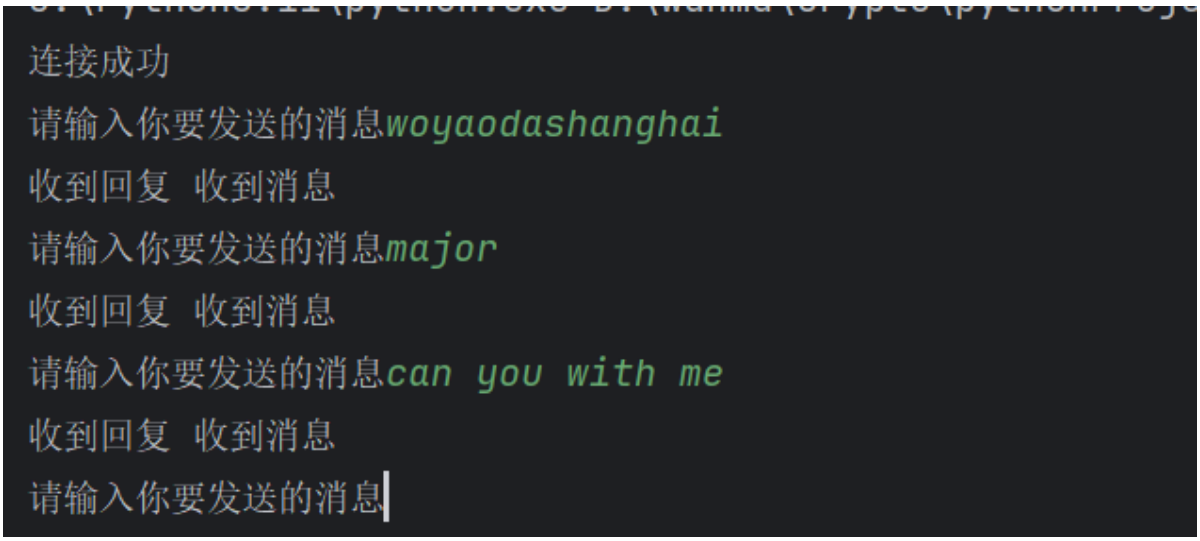


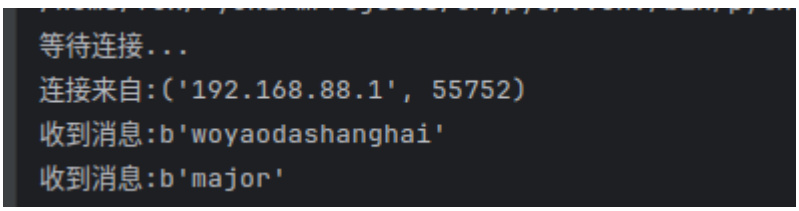
抓包得到明文信息



2. 加密后p2p

连接两端后通信





抓包，得到三条加密后的消息。

```
fONMaFnTxR2dbcW5Q9LHN/VvkMBDGsvGgZ0ta/M2ih0O1f7Zc+VDkR/8s1cHBDlaZKxlSXjPFEmWfwPMIUI5V+BcYjJVEJiLJ+hBIWuHeDdrpP4y5
4ADBq4nrZEafgYeoZWrhqZorb5R/veumYE6ebvoANMnp0Hdj5hmEYOO9G7bqt2lNleg3cKm0U+yH1goWyvVo/2P4Svl53VqQdgixesOHHi5dSWC0D
lKadsehNSwj7NAMjvgnm/1xRdVVaApG7Dz5auh6PBIrdN8deBDCcJD1cUXHQ4BXccp3dGyeZ9BFQiYS+vJbivefJY0jsCI30IEMbrfr+cYjZtgvPa
7Saa97S90xSzAsRb3eTnOB5IeEfFkeaa2aO0Worwn9C+2LKQEZSVrolIgHWWTaJsGUrv+YHtic/K5iLnMKY99fqPyUdxbtn4XHi9tNY/m9qIAjG5K
dU6WaE1eY8PDzTGvBmMAfOSOaPRfvvdMRZgBD03wU9aIAzMZiQc3cbw
...........
P8UDBbtitOGiMuTUMqTmUe1irUz9OLEf4AK6z9TFawz+42NKch0DaAlCjfgUms8qk2VgskMY8wVJ575TJ+cNPgOWUHM9OrYLMN5+cUdooGmVV7SJE
gL/15KqU5UcbgEyxpZwbD5XoyCPpoQDkWGyel+elW8GjK6JOTJDQOmdtbWeJUOmcBPnRZSCK0wKv/R//YLjcJ/wnu0Vmd7qQ8+guoNrmIYnP4HX/J
sRK89GBnko/rmXjp9bKKMMTg0JY/oWvC4jquh5EiBBAxoMtdwmxuYrdreOZr6RWiuwfYfayRlqIryJPXKE5LgRAobizfpe5nYPzv/a02WwXGIdzrE
au8CtArPgUCNiIWRjrK2wXn6u58IRAQMm/H2eQ6b/hLn92//hp/91EReO9jIQDQ2Yfazg9mCs8VqU/jydNO2CyQX6gGk7MGitjiuDNMiMXmQorZ5D
M6AWhN8BaZBEFCG9uSRR2R1oQjcjPnJdqAUEcxtAE4T96oPoTsHneBz
...........
6QIN5giIXlzbpfkg8s6P/l7CxYaZlTGIlSXJhRnjbwdElNrlmFqBnw0nBrxviNB3d11ltNV0a3rauM5qowUxxpTNaD4iXvliFxNXbtXhhBkhT7tx
NnbS3GoO0A8tvVuSkcHy7GJbjYwoN8giNbLaWibubXb7r+PqioO8pmDMdEVgplSRX7DlkMEtXbx26lqzlBg9UdRmESsyXCJAgso27j3v20Sr+l5Sl
eWaQejsHoIvZ0ETTrFIgu3q+23O0PoCADBZt9G5SZLMdmJhTkLt0PMiJbebDXj+Gyw9g8VdYKtLVV5vmFZArAUyxUxGCh0T8oG35pxNWzUIIkvjOg
7pcfRKTTQ3aICL7QMyeiqmcbCuVe2T5Co0mz2ZruV6n0In+p1jxt21wkDMVjcsiHBXI+AVNbsTj/xZiAO4K0h1SQPM29fpT0bbpx4jhfiS5uwrprC
QTbGceyT39AvjBCJ8KnOmt74qbBRSENCqXJU2kql8D9YD5RHS24+XDE
```

实现Telnet和SSH

1、Telnet协议简介

Telnet协议是一种最早的internet应用，telnet协议提供了一种通过终端远程登录到服务器的方式，呈现一个交互式操作界面，用户可以先登录到一台主机，然后再通过telnet的方式远程登录到网络上的其他主机上，而不需要为每一台主机都连接一个硬件终端，然后对设备进行配置和管理。192.168.88.129

连接telnet示例（实验使用的是ubantu2024.2)

利用ifconfig找到虚拟机ip。然后在配置好的telnet上启动服务，在打开windous系统自带的telnet后，连接，并可利用windous运行对虚拟机进行一些命令操作。

```
x@fox-VMware-Virtual-Platform:~$ ls snap
refox  firmware-updater  prompting-client  snapd-desktop-integration  snap-store  thunderbird
x@fox-VMware-Virtual-Platform:~$
x@fox-VMware-Virtual-Platform:~$ touch snap/test.txt
x@fox-VMware-Virtual-Platform:~$
x@fox-VMware-Virtual-Platform:~$ ls snap
refox  firmware-updater  prompting-client  snapd-desktop-integration  snap-store  test.txt  thunderbird
x@fox-VMware-Virtual-Platform:~$
x@fox-VMware-Virtual-Platform:~$
```

在snap中创建test.txt文本文件。

```
[01;34mthunderbird.[0m
.[?2004hfox@fox-VMware-Virtual-Platform:~$
.[?2004l
..[?2004hfox@fox-VMware-Virtual-Platform:~$ touch snap/test.txt
.[?2004l
..[?2004hfox@fox-VMware-Virtual-Platform:~$
.[?2004l
..[?2004hfox@fox-VMware-Virtual-Platform:~$ ls na.[D.[K.[D.[Ksnap
```

抓包得到运行命令过程

```
..%..&..... ..#..'..$
..%..&........ ..#..'..$
.....'.........
....x......'.......ANSI..
........".....!
........".....!
............
...

Linux 6.11.0-9-generic (fox-VMware-Virtual-Platform) (pts/5)

fox-VMware-Virtual-Platform ............
.........f
f
o
o
x
x




.........
[redacted]


Welcome to Ubuntu 24.10 (GNU/Linux 6.11.0-9-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

28 ............................
.........................................apt list --upgradable

.[?2004hfox@fox-VMware-Virtual-Platform:~$
```

得到密码和用户名。

2. ssh实训

这个与上面telnet做出对比，在ssh协商rsa密钥的加持下，wireshark抓包获得的都是加密后的信息。

在ubantu虚拟机上使用

```
$ sudo service ssh start
启动ssh服务，这里默认关闭了防火墙
```

然后再windous端打开ssh服务插件，能使用ssh作为客户端连接。

```
C:\Users\24055>ssh fox@ 192.168.88.129
ignoring bad CNAME "\345\260\217\347\210\254\350\231\253" for host "": domain name "\34
5\260\217\347\210\254\350\231\253" starts with invalid character
ssh: connect to host  port 22: Connection refused

C:\Users\24055>ssh @192.168.88.129
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface] [-b bind_address]
           [-c cipher_spec] [-D [bind_address:]port] [-E log_file]
           [-e escape_char] [-F configfile] [-I pkcs11] [-i identity_file]
           [-J destination] [-L address] [-l login_name] [-m mac_spec]
           [-O ctl_cmd] [-o option] [-P tag] [-p port] [-Q query_option]
           [-R address] [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
           destination [command [argument ...]]

C:\Users\24055>ssh fox@192.168.88.129
The authenticity of host '192.168.88.129 (192.168.88.129)' can't be established.
ED25519 key fingerprint is SHA256:xRMFYuyERmkZqQcIJNtKh548LuA5raZfxIGBVrP39DI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yse
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.88.129' (ED25519) to the list of known hosts.
fox@192.168.88.129's password:
Welcome to Ubuntu 24.10 (GNU/Linux 6.11.0-9-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

0 更新可以立即应用。

Last login: Sun Dec  8 16:21:07 2024 from 192.168.88.129
fox@fox-VMware-Virtual-Platform:~$
```

我们可以在之前船创建服务端rsa协商密钥，可是我后面把密钥删除了再试一次依然能够正常运行[到时候任务结束后问问]。

id_ed25519 × | id_ed25519.pub

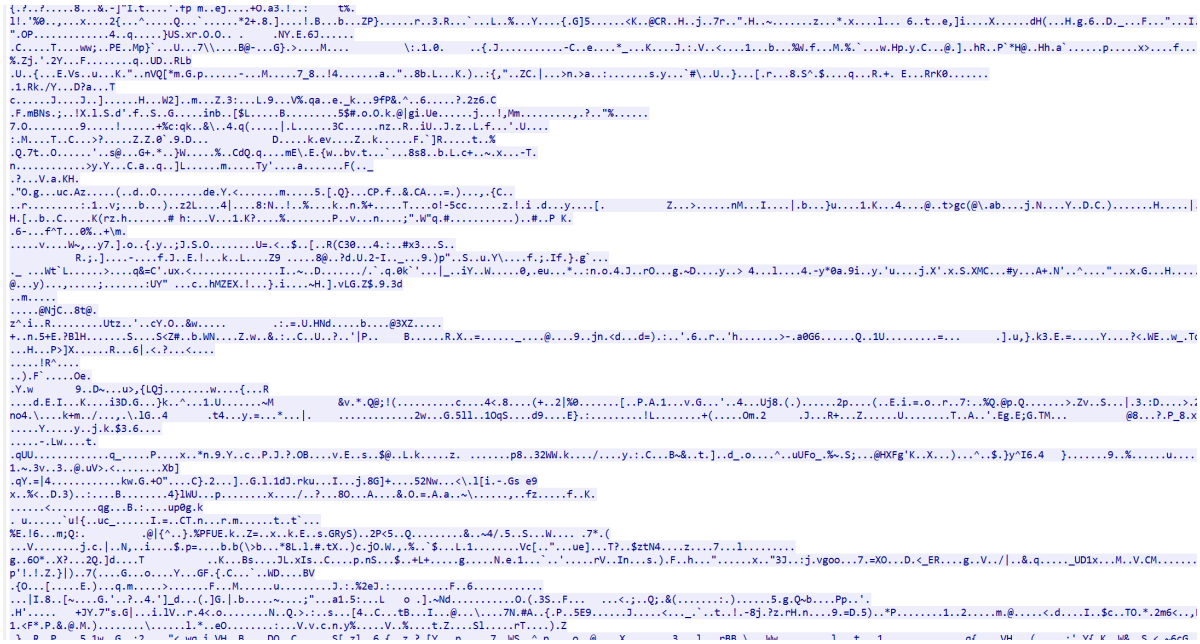
D: > wdnmd > 0101 > .ssh > id_ed25519

```
1  -----BEGIN OPENSSH PRIVATE KEY-----
2  b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAABCAtVRMIN
3  6hTeNpP0PthEynAAAAGAAAAAEAAAAzAAAAC3NzaC1lZDI1NTE5AAAAICaxUEc60RqhQx/A
4  Y/7lii0rzEa+LkHmCuFw8HFCx2wOAAAAkC/XcAa26hQeQh0ePpyhLPJJteCtfyKwPaSZtS
5  1CZTT1CtwNUo0rvPPf4jlH4aj/McNrzwtkIEqIWIdx9uQ8/W77JTguoemF1wGv52rqp49i
6  WzuI14auTUR1ISg5wgBYlixb5beoMqD06LgqT4RLzKl1Cw3Cfpv4Zws+FEN1b66/b8jUst
7  TOiX+L666KOUrBNA==
8  -----END OPENSSH PRIVATE KEY-----
9
```

我们和上述过程一样创建文本文件

```
错误：unknown flag `s'
ox@fox-VMware-Virtual-Platform:~$ sudo -s
sudo] fox 的密码：
oot@fox-VMware-Virtual-Platform:/home/fox# touch test_ssh.txt
oot@fox-VMware-Virtual-Platform:/home/fox#
```

wireshark抓包



得到加密后的结果。