网络与信息安全课内实验三

对称、非对称加解密

### 本次实验目的

〉 1. 加深对非对称，对称加解密算法、散列函数的理解。

〉 2. 了解常用加密工具包以及相关库函数的使用。

〉 3. 了解ssl协议

### 实验平台

ubuntu虚拟机，python3

### 实验步骤

知识回顾

1. 对称加密 采用单钥密码系统的加密方法，同一个密钥可以同时用作信息的加密和解密，这种 加密方法称为对称加密，也称为单密钥加密。在对称加密算法中常用的算法有：DES、 3DES、TDEA、Blowfish、RC2、RC4、RC5、IDEA、SKIPJACK 等。 对称加密算法的缺点是在数据传送前，发送方和接收方必须商定好秘钥，然后使双 方都能保存好秘钥。其次如果一方的秘钥被泄露，那么加密信息也就不安全了。另外， 每对用户每次使用对称加密算法时，都需要使用其他人不知道的唯一秘钥，这会使得收、 发双方所拥有的钥匙数量巨大，密钥管理成为双方的负担。

2. 非对称加密 非对称加密算法需要两个密钥：公开密钥（publickey）和私有密钥（privatekey）。公 开密钥与私有密钥是一对，如果用公开密钥对数据进行加密，只有用对应的私有密钥才 能解密；如果用私有密钥对数据进行加密，那么只有用对应的公开密钥才能解密。因为 加密和解密使用的是两个不同的密钥，所以这种算法叫作非对称加密算法。 非对称加 密算法实现机密信息交换的基本过程是：甲方生成一对密钥并将其中的一把作为公用密 钥向其它方公开；得到该公用密钥的乙方使用该密钥对机密信息进行加密后再发送给甲 方；甲方再用自己保存的另一把专用密钥对加密后的信息进行解密。

3. 散列函数 Hash，一般翻译做"散列"，也有直接音译为"哈希"的，就是把任意长度的输入（又 叫做预映射， pre-image），通过散列算法，变换成固定长度的输出，该输出就是散列值。 这种转换是一种压缩映射，也就是，散列值的空间通常远小于输入的空间，不同的输入 可能会散列成相同的输出，而不可能从散列值来唯一的确定输入值。简单的说就是一种 将任意长度的消息压缩到某一固定长度的消息摘要的函数。

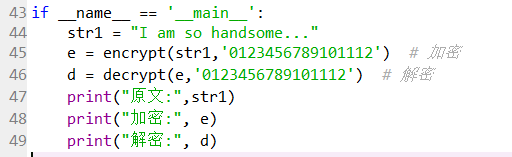
对称加密算法的字符串的加解密

本实验中，需要使用python对字符串进行加解密。

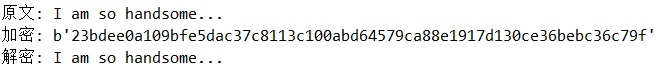
Python中常用来进行加解密相关的库是Crypto，安装：pip3 install pycrypto

本次我们使用AES算法对字符串进行加密。使用的方法很简单，文件在附件中已经给出。



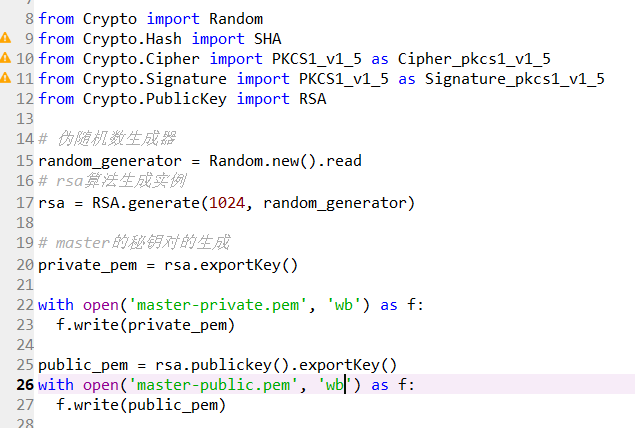


结果如下：

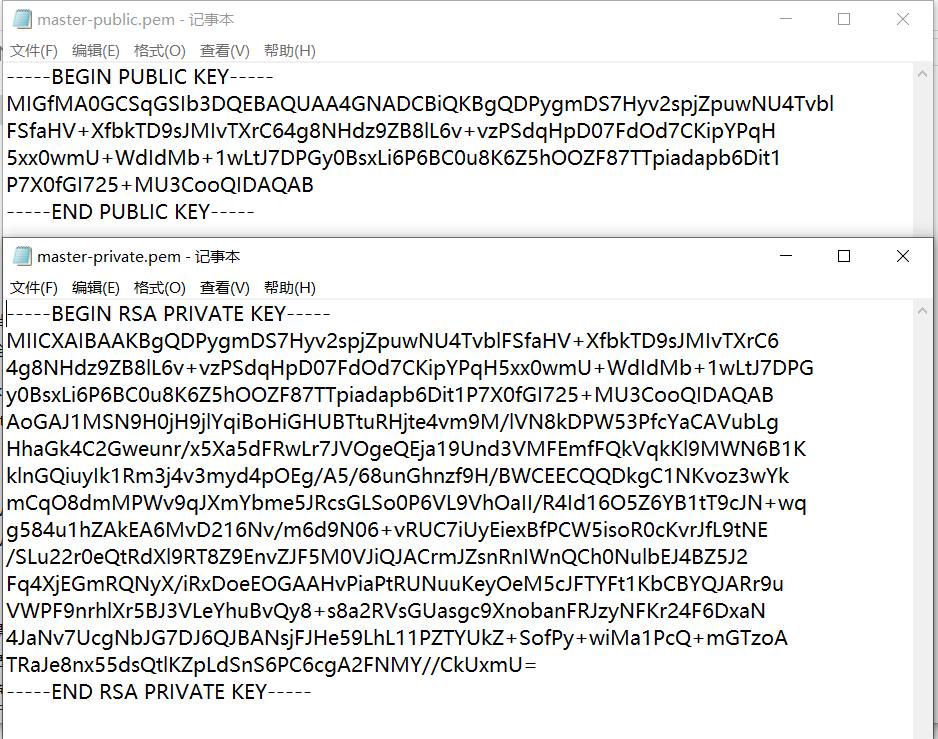


使用RSA的加解密

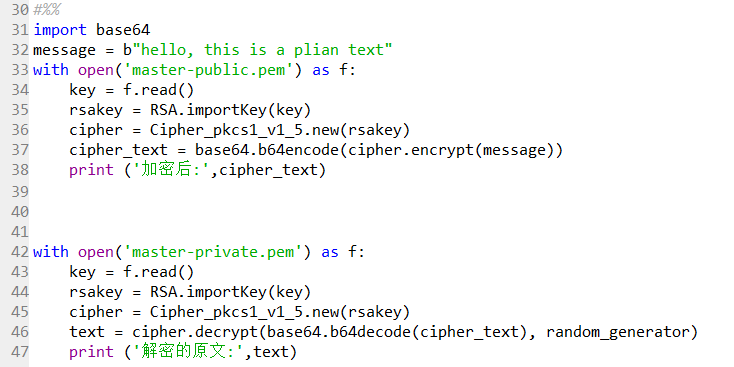
1. 生成公私钥



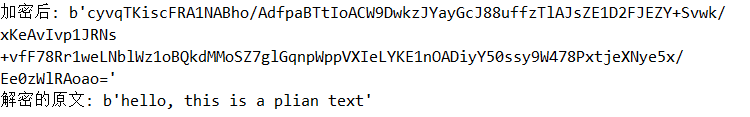
公私钥大概长这样：



1. 使用公钥加密，并使用私钥解密：



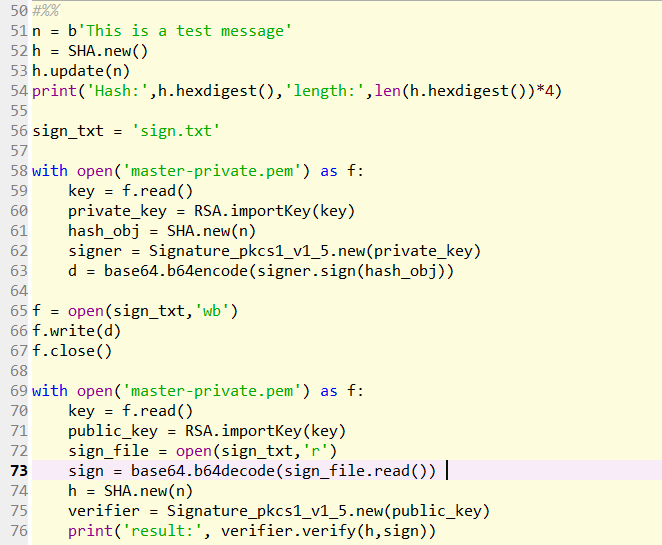
运行结果如下：



数字签名

普通数字签名算法有RSA、ElGamal、Fiat-Shamir、Guillou- Quisquarter、Schnorr、Ong-Schnorr-Shamir数字签名算法、Des/DSA，椭圆曲线数字签名算法和有限自动机数字签名算法等。

此次实验中，我们只尝试使用RSA进行数字签名。原理详见：<https://blog.csdn.net/frank_jb/article/details/79472325>



签名验证结果：

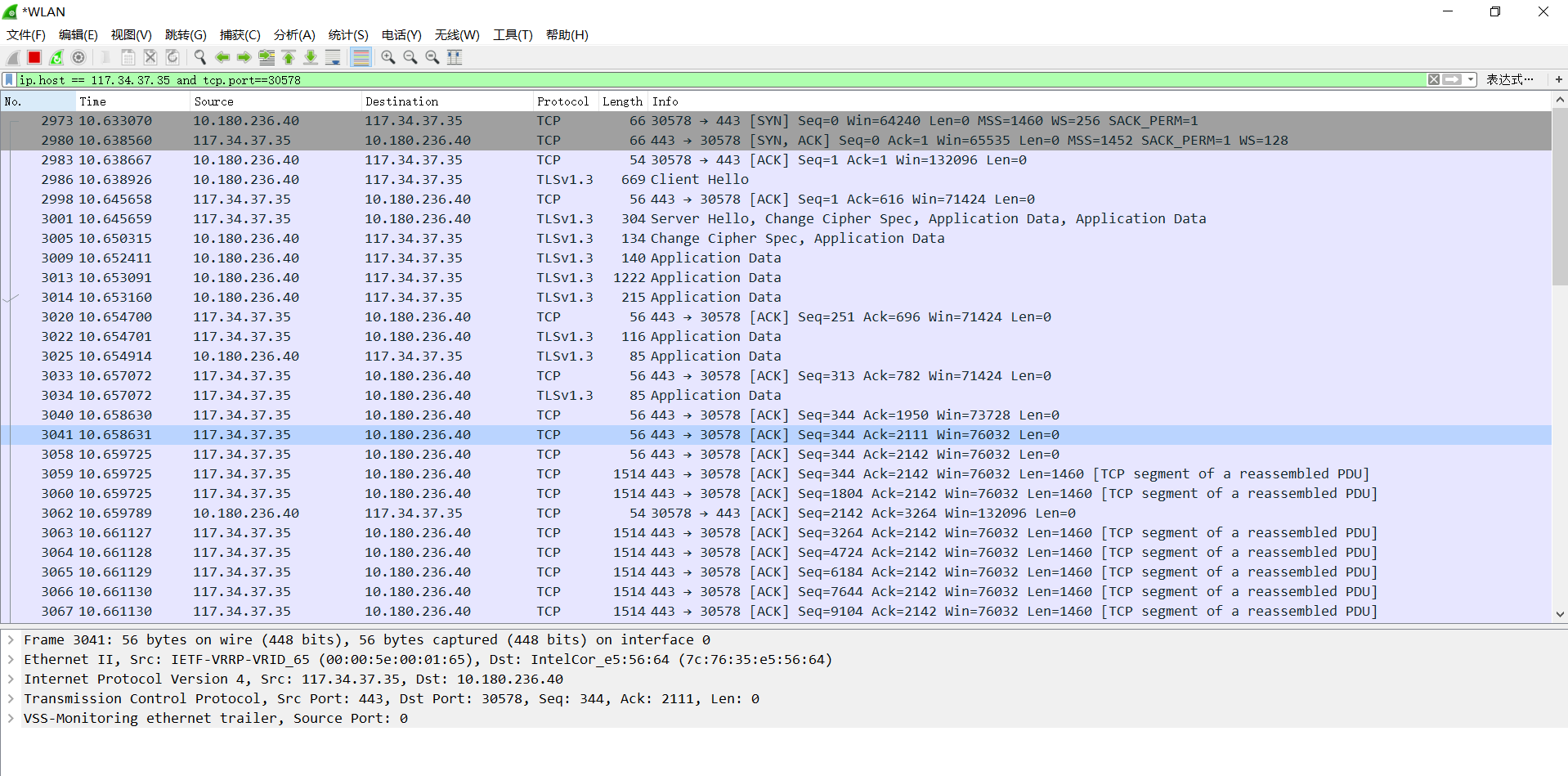
C:\Users\90415\Desktop\捕获.PNG

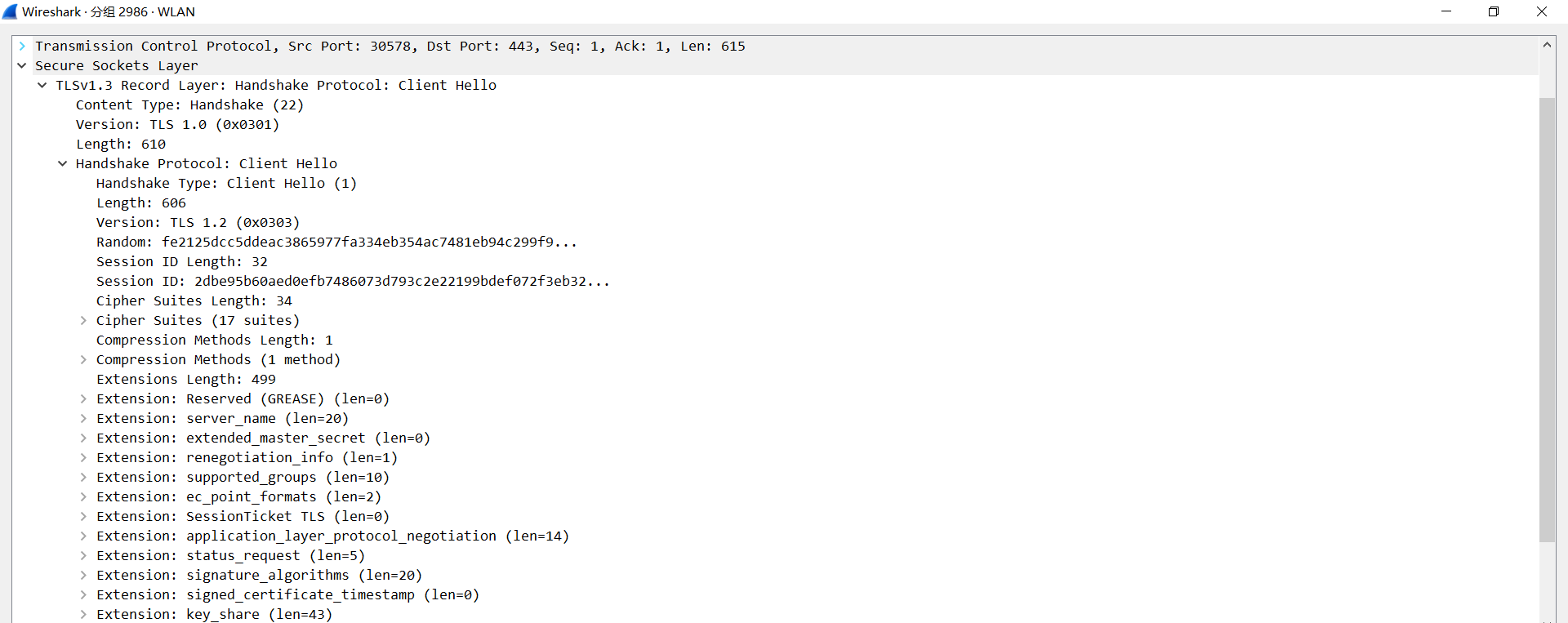
抓包观察 ssl 协议通信握手过程

SSL(Secure Sockets Layer 安全套接层),及其继任者传输层安全（Transport Layer Security，TLS）是为网络通信提供安全及数据完整性的一种安全协议。TLS与SSL在传输层与应用层之间对网络连接进行加密。

为了深入了解ssl协议，我们对其抓包进行分析。深入学习参见<https://www.cnblogs.com/Anker/p/6082966.html>。

最常见的使用方式为https，我们访问一个网站，并使用wireshark进行抓包：





### 实验报告要求+思考题

1. 回答AES加密中iv变量的作用。
2. 思考问题：RSA【公钥加密，私钥解密】和【私钥加密，公钥解密】算法一样吗？为什么？
3. 书写数字签名的注释，每行都干了些什么？并任意举一个例子使得result=False。
4. 详细分析ssl 协议通信握手过程。
5. 写出实验中遇到的问题与心得体会。