

云南大学数学与统计学院

《计算机网络实验》上机实践报告

课程名称：计算机网络实验	年级：2015 级	上机实践成绩：
指导教师：陆正福	姓名：刘鹏	专业：信息与计算科学
上机实践名称：基于 SSL 的安全通信编程实验	学号：20151910042	上机实践日期：2018-11-27
上机实践编号：No.06	组号：	

一、实验目的

1. 熟悉基于 SSL 的通信编程实验；
2. 熟悉教材计算机网络安全与密码学的基本概念

二、实验内容

1. 掌握基于 SSL 的安全通信编程的流程；
2. 查阅 Java 的有关 SSL 的类库文档，使用 Java 实现基于 SSL 的安全通信编程；
3. 使用 Java 和 Android 实现基于 SSL 的安全通信编程（选做）；
4. 使用 Python 实现基于 SSL 的安全通信编程（选做）。

三、实验平台

Windows 10 Pro 1803;
Cygwin GCC 编译器。

四、程序代码

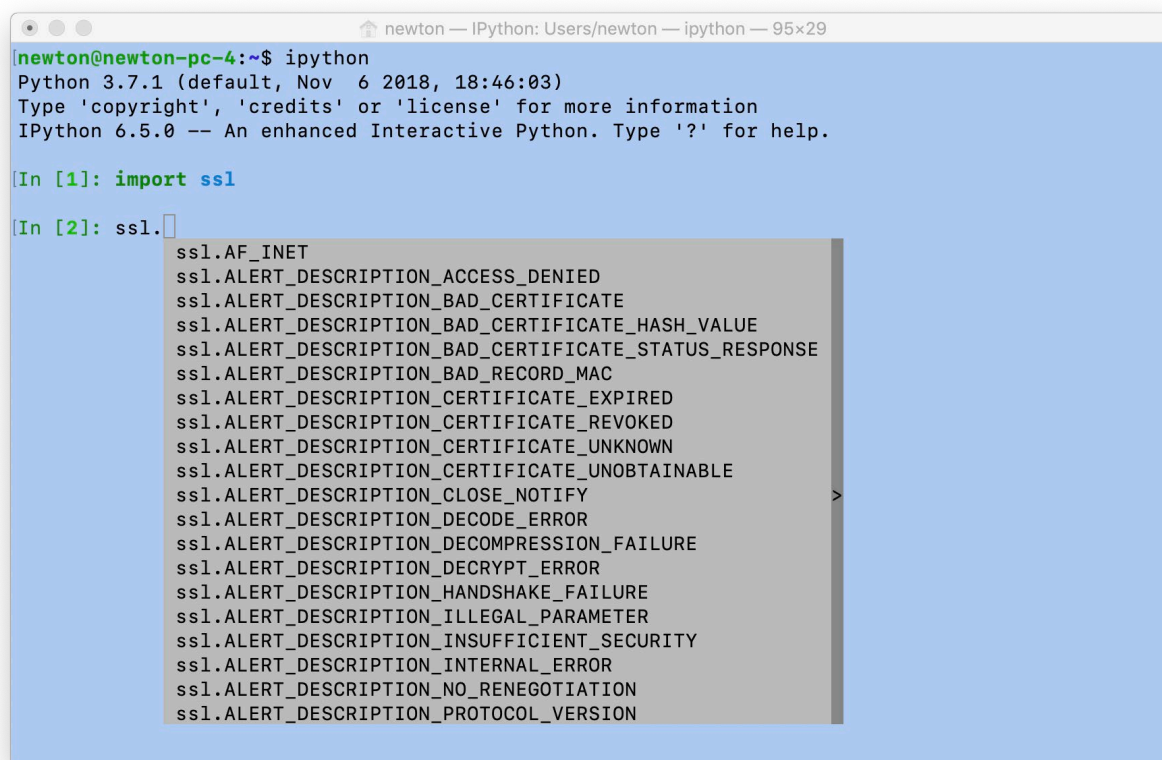
4.1 SSL 环境配置

在计算机网络中，OpenSSL（Open Secure Sockets Layer）是一个开放源代码的软件库包，应用程序可以使用这个包来进行安全通信，避免窃听，同时确认另一端连接者的身份。Netscape 公司在推出第一个 Web 浏览器的同时，提出了 SSL 协议标准。其目标是保证两个应用间通信的保密性和可靠性，可在服务器端和用户端同时实现支持。SSL 已经成为互联网上保密通讯的工业标准。

SSL 能使用户/服务器应用之间的通信不被攻击者窃听，并且始终对服务器进行认证，还可选择对用户进行认证。SSL 协议要求建立在可靠的传输层协议之上，比如 TCP。SSL 协议的优势在于它是与应用层协议独立无关的，高层的应用层协议（例如 HTTP、FTP、TELNET 等）能透明地建立于 SSL 协议之上。SSL 协议在应用层协议通信之前就已经完成加密算法、通信密钥的协商以及服务器认证工作。在此之后应用层协议所传输的数据都会被加密，从而保证通信的私密性。

在编程实验中，对于 SSL 有多种选择，目前比较流行且鲁棒性比较好的是 Python 的 ssl 库（使用 `import ssl` 进行调用）、Java EE 的 net 库的 ssl 类库，值得一提的是，J2EE 使用的 `import` 语句和 J2SE 稍有不同，`java.*` 指的是 `java` J2SDK 里面的类库；`javax` 中的 `x` 值的是 `extension`，即拓展包，它是 J2EE 的 API 集合。openssl 官方给出的是 C/C++ 语言版本的 Library，可以在各种操作系统平台上，针对不同的编译器进行编译、安装。由于之前的所有实验报告均采用了 Java 语言，所以这里选择 Java 版本的类库进行实验。安装了 JDK

之后，不再需要单独配置 Java 环境。



```
newton — IPython: Users/newton — ipython — 95x29
newton@newton-pc-4:~$ ipython
Python 3.7.1 (default, Nov  6 2018, 18:46:03)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.5.0 -- An enhanced Interactive Python. Type '?' for help.

[In [1]: import ssl

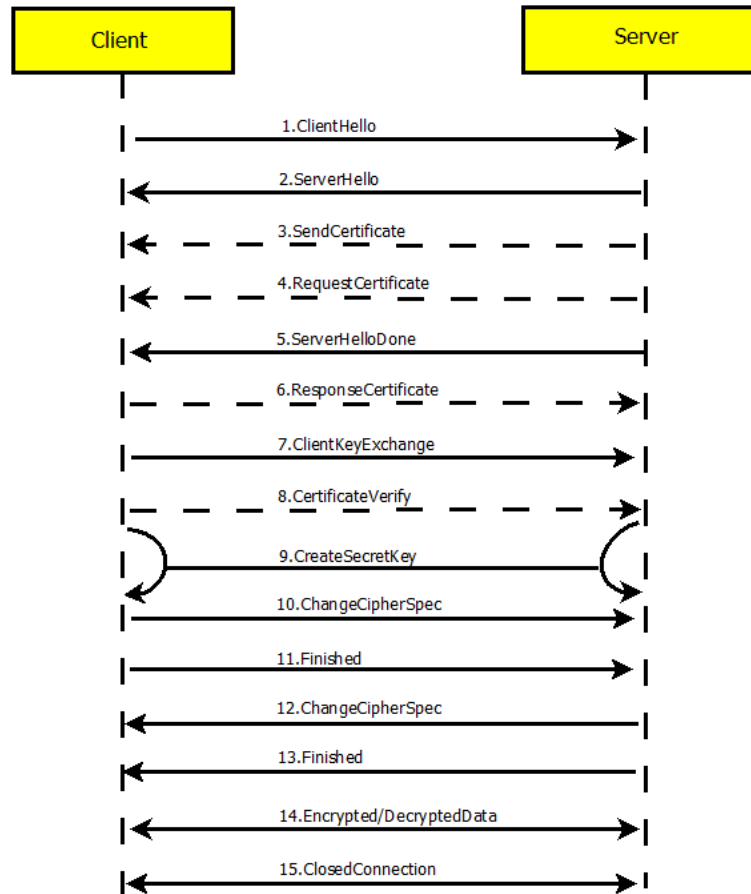
[In [2]: ssl.]
ssl.AF_INET
ssl.ALERT_DESCRIPTION_ACCESS_DENIED
ssl.ALERT_DESCRIPTION_BAD_CERTIFICATE
ssl.ALERT_DESCRIPTION_BAD_CERTIFICATE_HASH_VALUE
ssl.ALERT_DESCRIPTION_BAD_CERTIFICATE_STATUS_RESPONSE
ssl.ALERT_DESCRIPTION_BAD_RECORD_MAC
ssl.ALERT_DESCRIPTION_CERTIFICATE_EXPIRED
ssl.ALERT_DESCRIPTION_CERTIFICATE_REVOKED
ssl.ALERT_DESCRIPTION_CERTIFICATE_UNKNOWN
ssl.ALERT_DESCRIPTION_CERTIFICATE_UNOBTAINABLE
ssl.ALERT_DESCRIPTION_CLOSE_NOTIFY
ssl.ALERT_DESCRIPTION_DECODE_ERROR
ssl.ALERT_DESCRIPTION_DECOMPRESSION_FAILURE
ssl.ALERT_DESCRIPTION_DECRYPT_ERROR
ssl.ALERT_DESCRIPTION_HANDSHAKE_FAILURE
ssl.ALERT_DESCRIPTION_ILLEGAL_PARAMETER
ssl.ALERT_DESCRIPTION_INSUFFICIENT_SECURITY
ssl.ALERT_DESCRIPTION_INTERNAL_ERROR
ssl.ALERT_DESCRIPTION_NO_RENEGOTIATION
ssl.ALERT_DESCRIPTION_PROTOCOL_VERSION
```

除此之外，基于 Wireshark 的抓包实验也需要在本次实验中有所涉及，主要目的是用所捕获的数据包进行信息提取，然后做相关分析。

4.2 基于 SSL 的安全通信编程的流程解析

这里以 Client-Server 模式为基础，介绍基于 SSL 的安全通信的基本流程。

所谓的安全通信，只是在 TCP 协议的基础上，通过客户端与服务器之间的一套标准化动作来交换一些密钥，然后根据密钥和密码算法来进行加密通信。所以在这个层面上，可以把 SSL 狭隘地理解为一种正式通信前的流程，核心仍旧是以数学为基础的密码学原理。从软件工程角度，可以把 SSL 理解为一般明文通信协议与本地数据之间的一个“中间件”，负责在本地数据与可传播的加密信息之间做加密、解密。



五、 实验体会

六、 参考文献

- [1] 林锐. 高质量 C++/C 编程指南 [M]. 1.0 ed., 2001.
- [2] java IO: <https://zhuanlan.zhihu.com/p/21444494>
- [3] java NIO: <https://www.jianshu.com/p/093b7c408dba>
- [4] java NIO: <https://docs.oracle.com/javase/7/docs/api/java/nio/package-summary.html>
- [5] java NET: <https://docs.oracle.com/javase/7/docs/api/java/net/package-summary.html>