基于DTLS的安全服务器设计

# 一、实验目的

通过实验，掌握DTLS的基本原理，掌握python3-dtls库的基本使用。

# 二、实验内容

1. 利用 DTLS 库编写客户端和服务器程序，服务端开启监听，提供数据传输、文件传输功能；

2. 客户端对服务端进行证书认证（单向认证）；

3. 利用相关工具（如 Wireshark）验证 DTLS 通信结果；

# 三、实验原理

DTLS 是 Datagram Transport Level Security 的简称，即数据报安全传输协议。TLS 被广泛应用于保护网页流量和其他应用协议，TLS 提供了一个透明的有向通道，通过在应用层和传输层之间插入 TLS 来保护应用层协议（如 HTTP等）。然而 TLS 必须依赖可靠的传输通道，如 TCP，因此 TLS 无法用于保护 UDP 的数据包流量。

简单来说，DTLS 可以理解为 UDP 上的 TLS 协议，它为 UDP 报文提供了安全传输能力。由于 UDP 协议是不可靠传输协议，其不对丢包、排序等负责，而 TLS 内部也不具备处理这种不可靠性的机制，因此 TLS 无法直接应用于 UDP 上。

**3.1、DTLS握手过程**

对于传输层安全来说，密钥交换机制和数据加密及签名算法决定了整个方案的安全等级。而密钥协商都必须通过握手流程完成，因而这是理解DTLS的关键要点。根据RFC6347定义，一个DTLS握手流程如图1所示：

其流程与TLS概念上是一致的，其中Flight对应一次通过网络发送的数据包；HelloVerifyRequest 用于服务端对客户端实现二次校验；Certificate是交换的证书，由协商后的算法确定是否需要传输；当服务端要求验证客户端身份时，发起CertificateRequest，此时客户端需要发送证书；ChangeCipherSpec是一个简单的标记，标明当前已经完成密钥协商，可以准备传输；Finished消息表示握手结束，通常会携带加密数据由对端进行初次验证。



图1

**3.2、CipherSuite**

由于网络 IO 限制，DTLS 只支持 TLS 的子集：

1. 密钥交换算法 ECDHE\_RSA，这是由 ECC 和 DH 密钥交换算法衍生出来的算法；

2. 动态密钥算法 AES\_128\_GCM，用于实现数据包的加解密；

3. MAC 算法 HMAC\_SHA256，用于创建加密数据块的摘要；

4. 伪随机函数 PRF，TLS1.2 定义其与 MAC 算法一致。

几个常用的 CipherSuite：

* TLS\_PSK\_WITH\_AES\_128\_CBC\_SHA256
* TLS\_PSK\_WITH\_AES\_128\_CCM\_8
* TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256
* TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8

**3.3、防护机制**

**一、握手**

**1. 重传**

DTLS 使用超时重传机制来确保握手消息的到达。

**2. 顺序**

UDP 本身不对消息传输的顺序负责，为了保证握手消息按序传输，每一个 Handshake 报文都分配了一个特殊的序列号，接收方直接处理属于当前步骤的消息，对于提前到达的消息则放到缓存队列中。

**3. 分段**

由于 MTU 限制，DTLS 要求对握手消息实现分段，每一个握手消息都可能包含分段的位移和长度，由接收端组装。

**4. 重复**

DTLS 选择性的支持消息重放检测机制，使用的技术与 IPsec AH/ESP 相同，由接收方维护一个 bitmap 窗口，因老化而不适合窗口的记录和以前接收过的记录都会被悄悄地丢弃。

**二、数据包传输**

DTLS 为每个加密数据包增加了 MAC 鉴权摘要，用于保证数据包的完整性；此外显式附带了一个 SN 号用于排序。

**三、DoS 攻击**

DTLS 定义了基于 cookie 验证的机制来预防攻击，如前面流程中涉及的 HelloVerifyRequest 便是用于进行 cookie 验证。

Cookie 的算法：HMAC(Secret, Client-IP, Client-Parameters)

其中 Secret 由 server 端内置，用于计算 cookie 值，client 端需要在接收到 VerifyRequest 后提供同样的 cookie 值；

server 端根据发送方 IP 计算 cookie 值，一旦发现不一致则判定为非法数据。

**四、会话恢复**

握手流程所占的开销是比较大的，与 TLS 类似，DTLS 也定义了会话恢复机制，如图2所示。

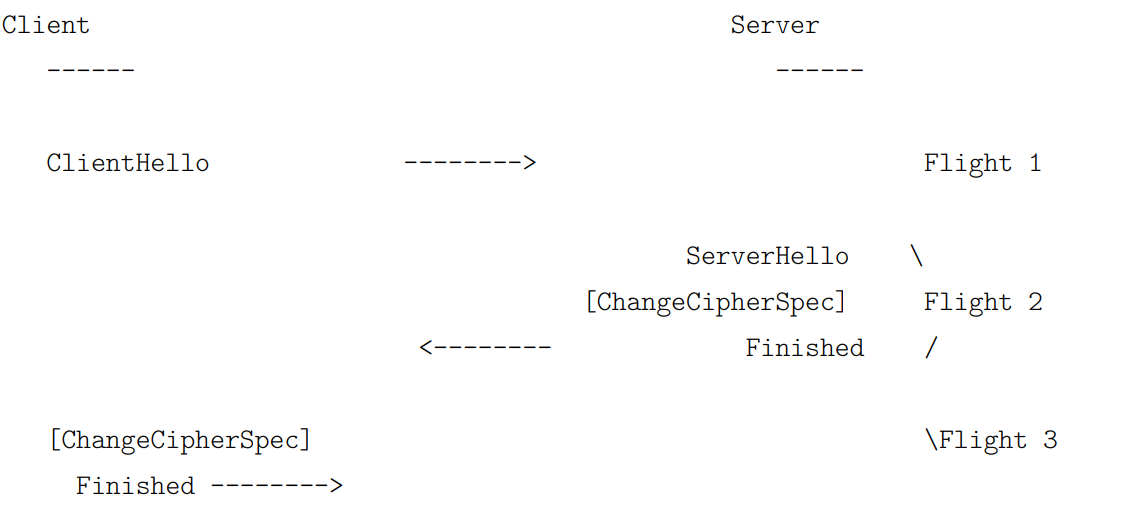


图 2

**简单原理：**

握手成功之后，服务端将生成 SessionID 返回，客户端在下次连接时附带 SessionID；若验证通过，可直接沿用原有的会话数据，包括协商算法和密钥。

# 四、实验条件

1、操作系统：Ubuntu 20.04（其他 Linux 发行版均可，满足内核≥4.8）；

2、工具软件：Python 3，Wireshark；

# 五、实验步骤

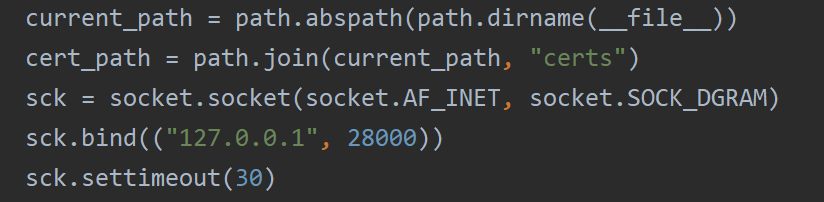
1、安装python-dtls

python3-dtls 是 Python 3版本的 DTLS 实现，其对 Python 原生的 ssl 库进行 patch，使其得以支持 DTLS。执行如下命令以安装：

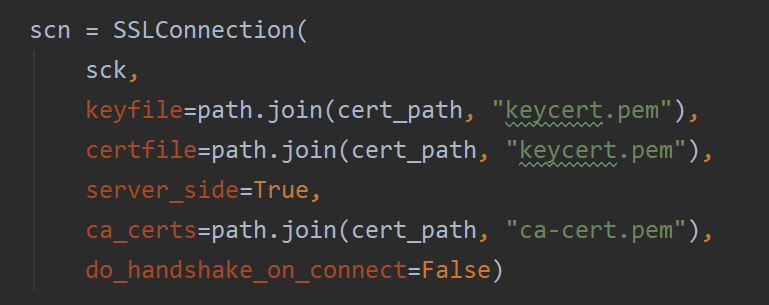
pip install python3-dtls

2、Server模块

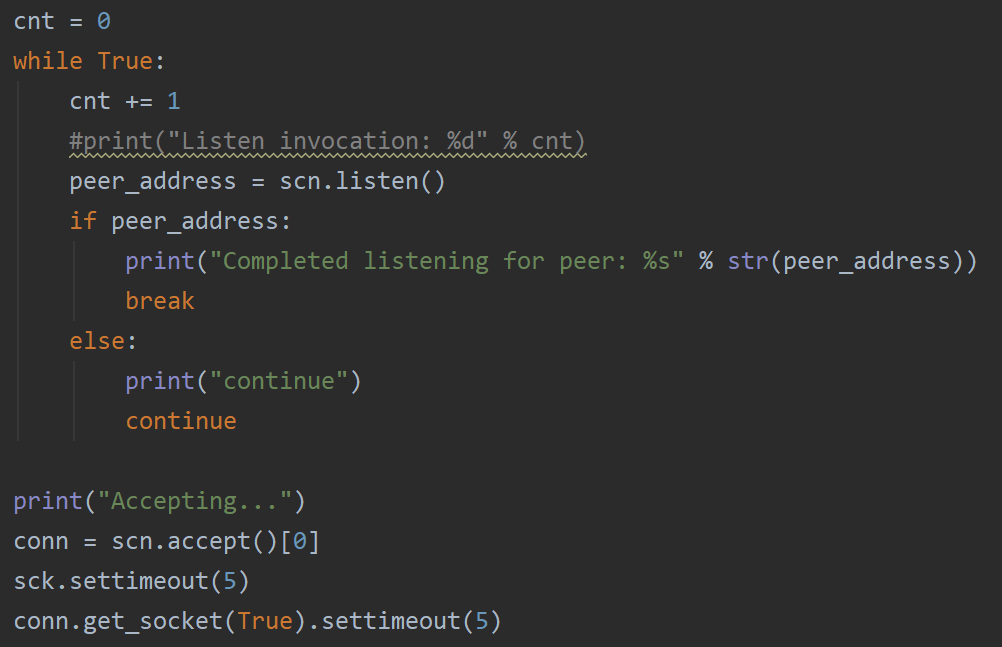
首先是利用socket模块，建立通信。利用socket通信绑定相关IP地址和端口。



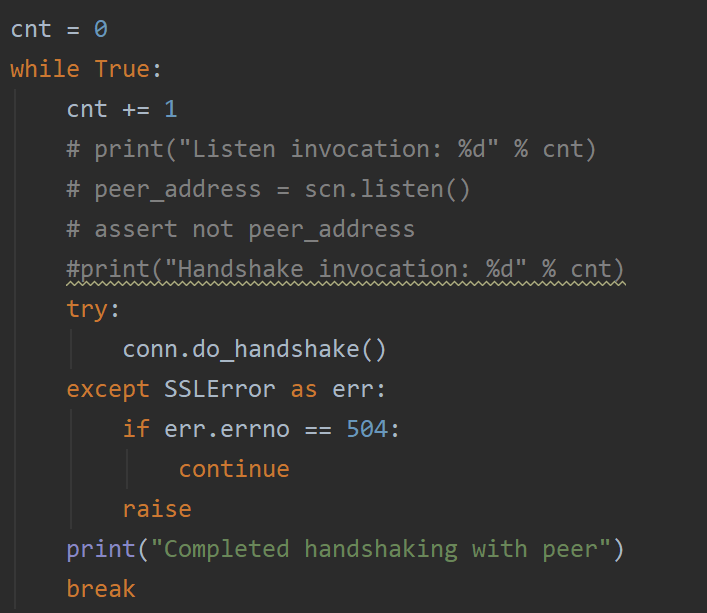
接着是利用其自带的封装函数SSLConnection函数上，这个函数会将SOCKET通信封装到DTLS上去。我们只需要在里面添加证书和密钥参数即可。



接着是等待client响应，首先是利用listen的技术，来找到连接的client端的信息，包括有IP地址以及他开放的端口。这里设计一个while循环进行循环监听。一有client他就会实现连接。

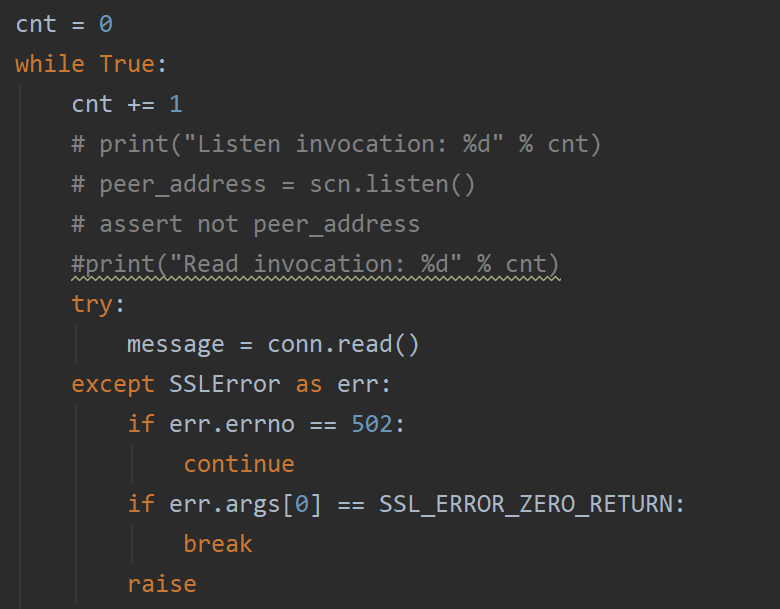


接着是实现握手，这里只需要使用到他封装好的函数do\_handshake()函数即可，具体的我们会在后续抓包的过程中进行分析。握手的过程如3.2的图所示。



接着是传输应用数据阶段：

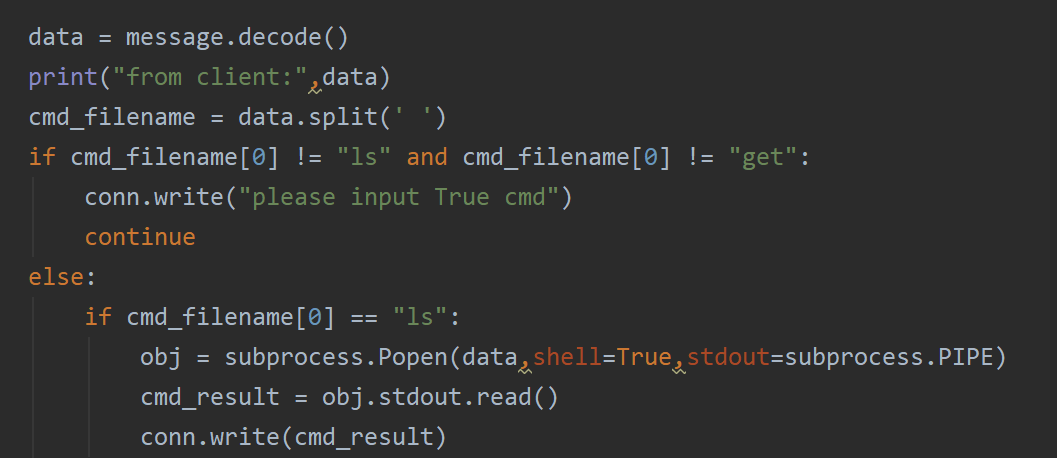
这里读取数据采用的是SOCKET通信中的数据传输，利用封装好的read()函数和write()函数，对数据进行传递。



这其中包括了一些功能，包括有远程命令的执行，远程文件传输。

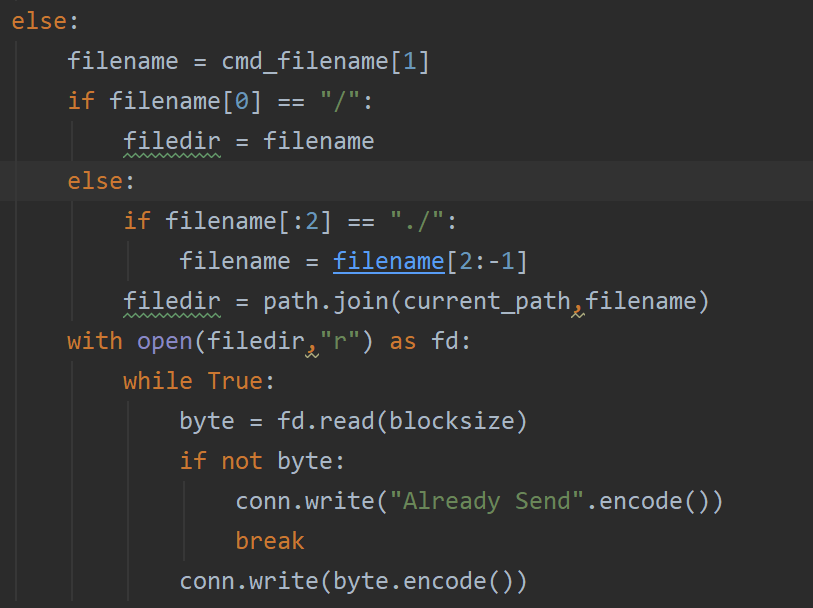
这里主要实现了两个功能，ls功能和get功能，分别是列举出当前目录下的文件以及远程传输文件。具体技术细节如下：

利用传输命令中存在空格，因此利用空格区分cmd和文件，判断cmd为ls还是get功能，如果是需要命令执行的话，就利用subprocess函数封装的命令执行，接着讲生成的结果以write的形式传输给client。如下图所示：

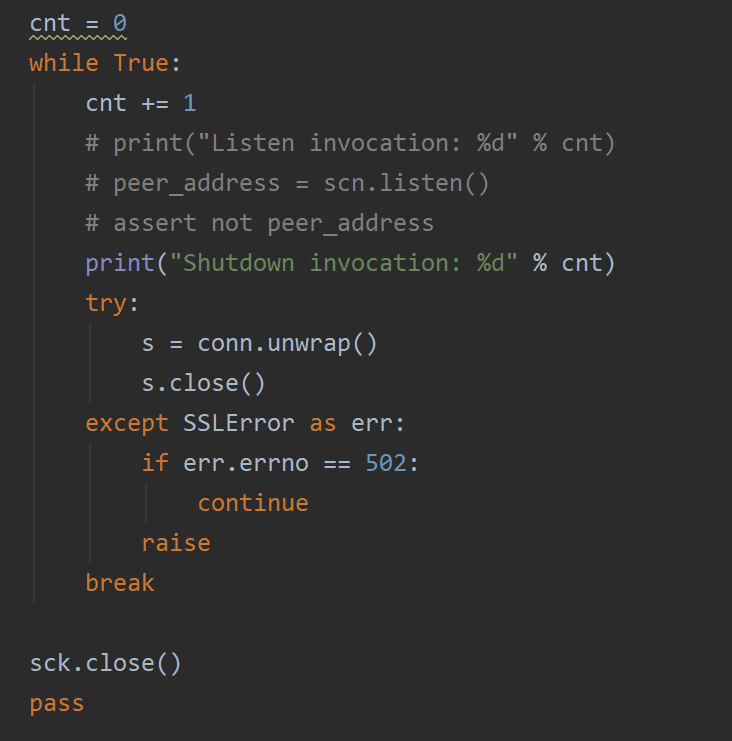


如果是get函数命令的话，那么就获取到他想要读取的文件的路径，这里支持./路径，根目录路径等。通过判断，整合路径，从而获取到文件在哪里，然后利用读取文件的操作。将文件的内容读出来（可以以二进制的形式读出来，这样比较好传输，但是这里我用文本的形式读出来，方便client获取到读完的信号。）

同时设置blocksize为1024B，这样读取数据的时候可以分块进行读取。并使用指针和while循环进行连续读取，直到读完。



数据传输完毕之后就可以结束了，利用unwrap()函数解除连接。



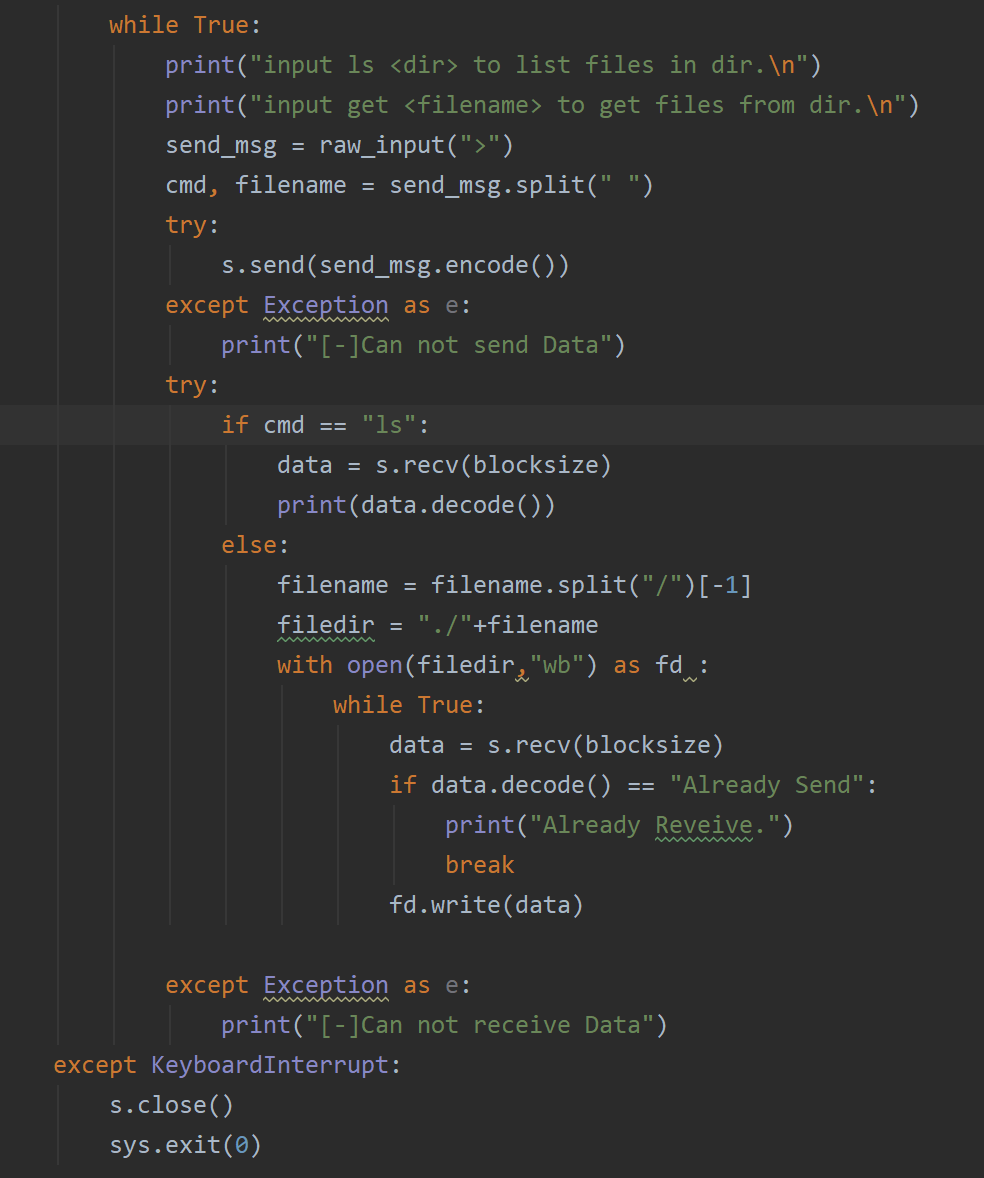
2、Client模块

这里利用到的是ssl中wrap\_socket函数，将dtls函数绑定到socket上，利用do\_patch()函数进行绑定。



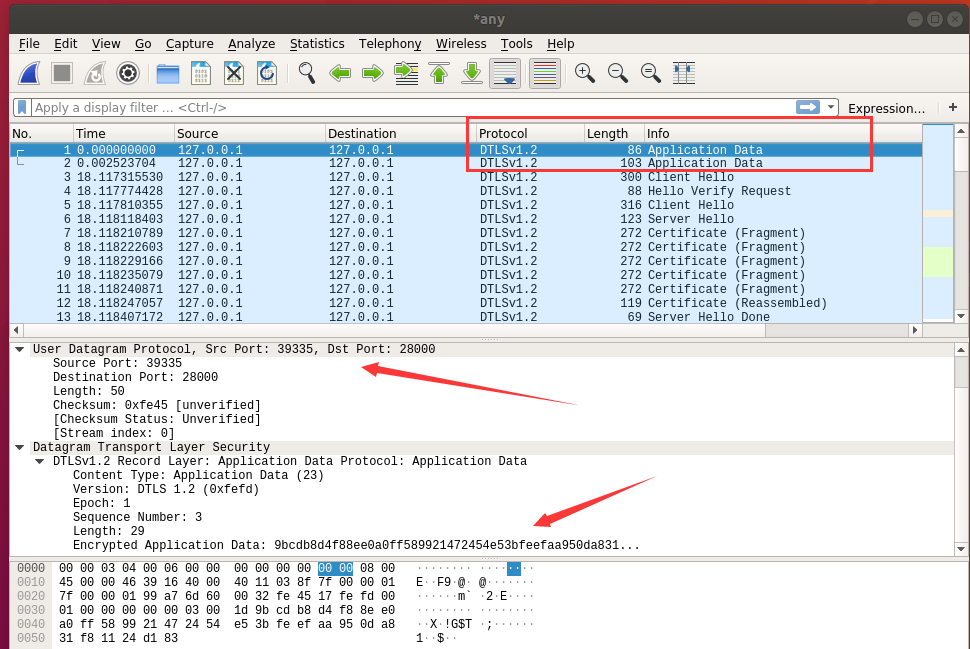
由于是服务器端单向的认证，因此不需要做过多的认证。

接着就是对输入命令的截取，根据不同的情况实现不同的功能，同时在传输文件这个功能上进行区别化对待，写一个while循环将文件内容传输过来。

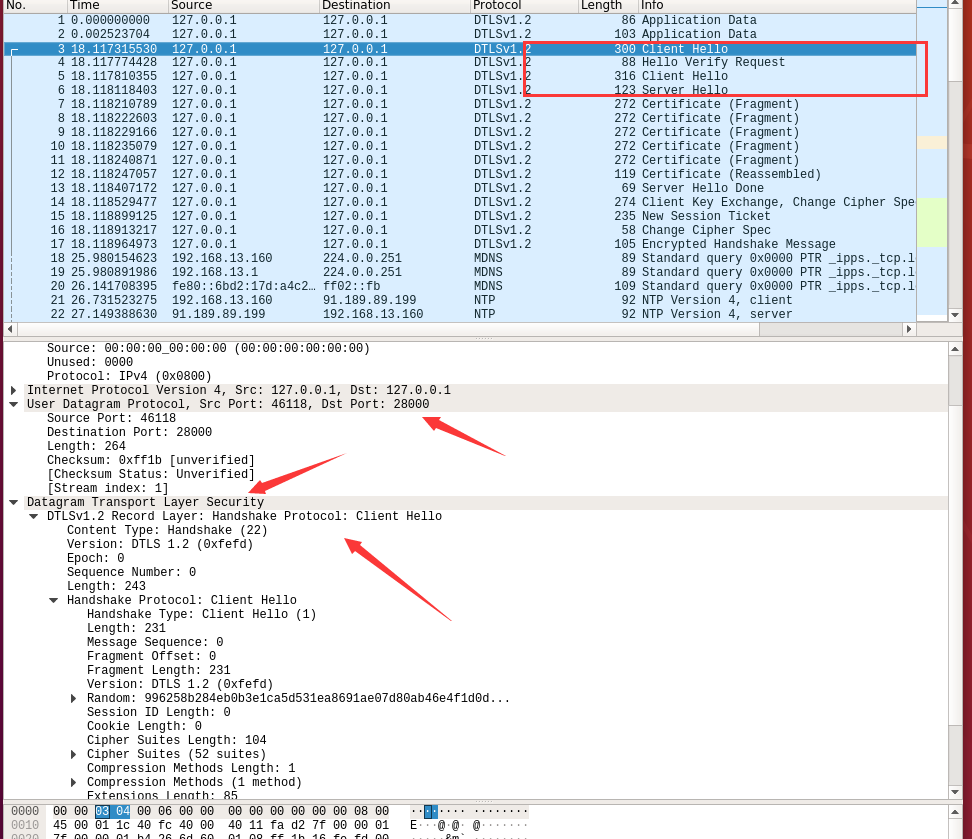


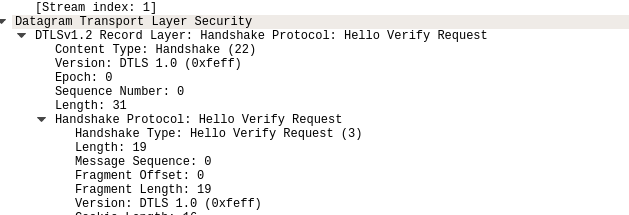
3、Wireshark协议分析

首先是连接，利用UDP协议，告知源端口和目的端口，这里是39335和28000：

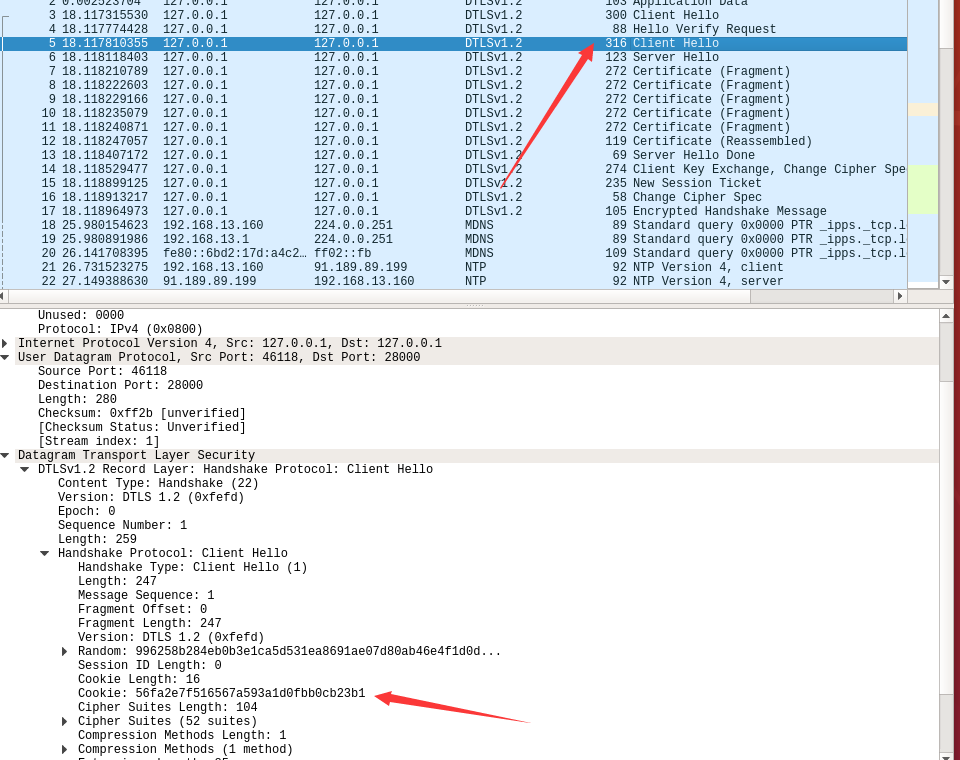


接着是握手协议，根据原理图可知。Client打招呼，然后客户端确认回复，接着两个人相互打招呼。

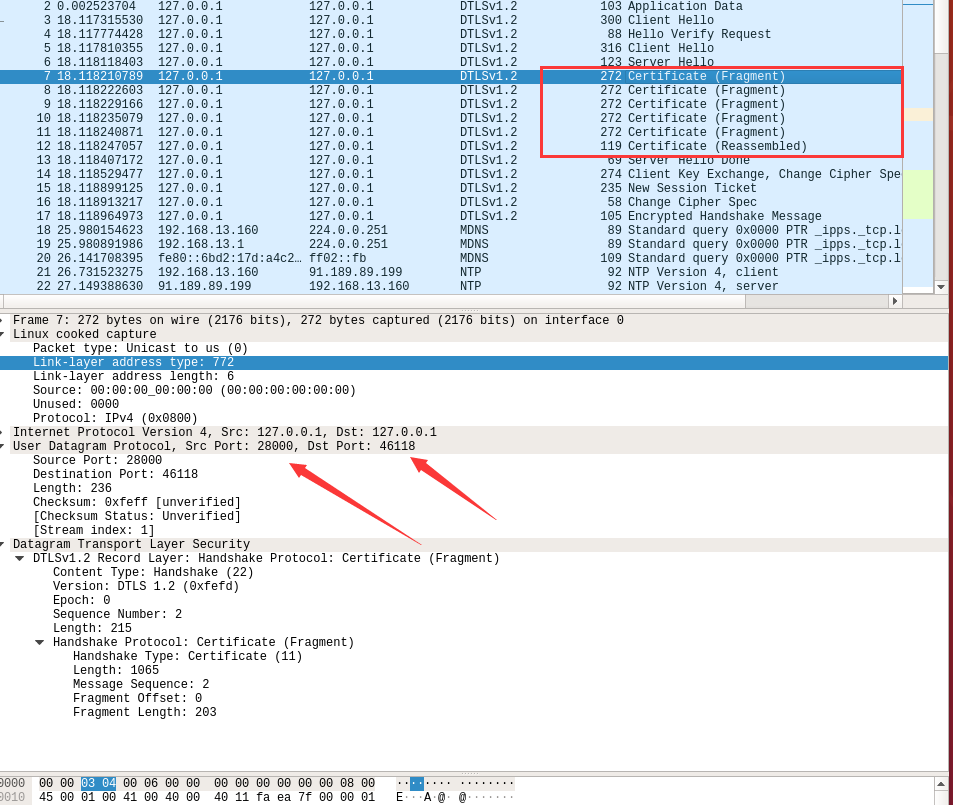




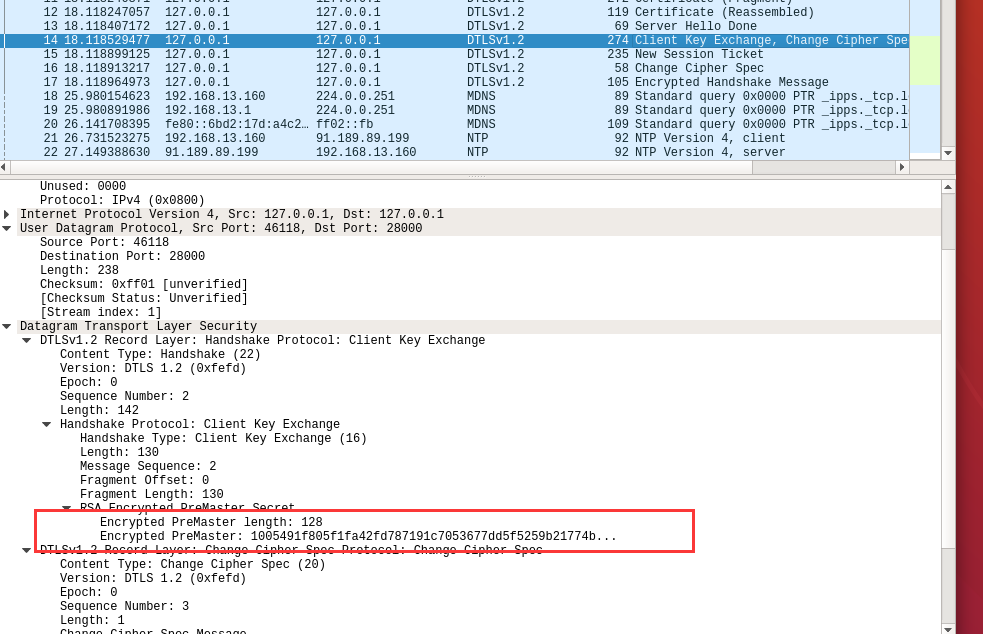
在这过程中传递cookie（为了防止DDOS攻击，client会重复向server发送cookie）：



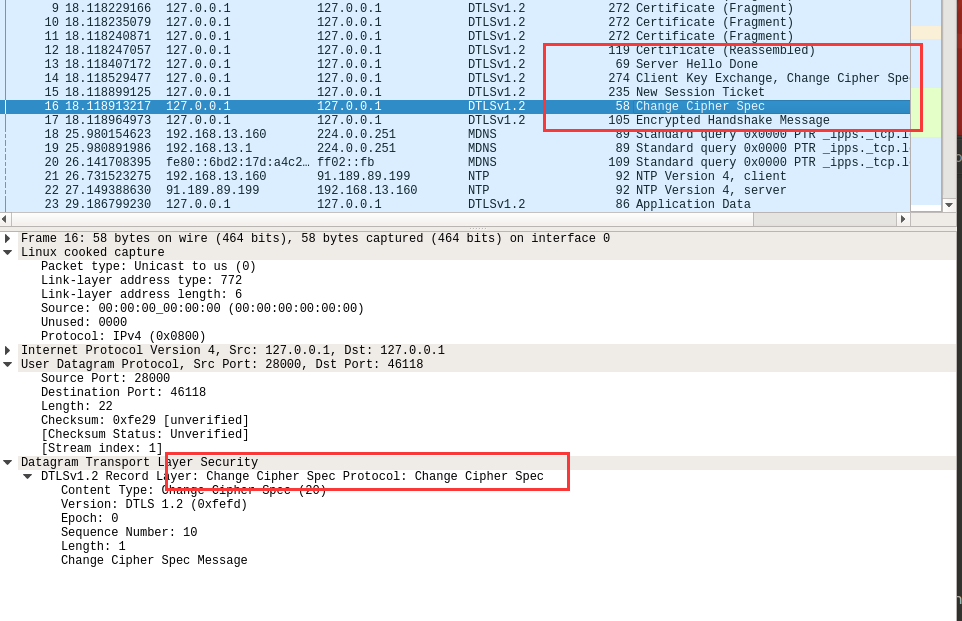
接着发送多个数据包传递证书传递公钥，传递证书都是单向的，从服务器端向客户端进行发送。



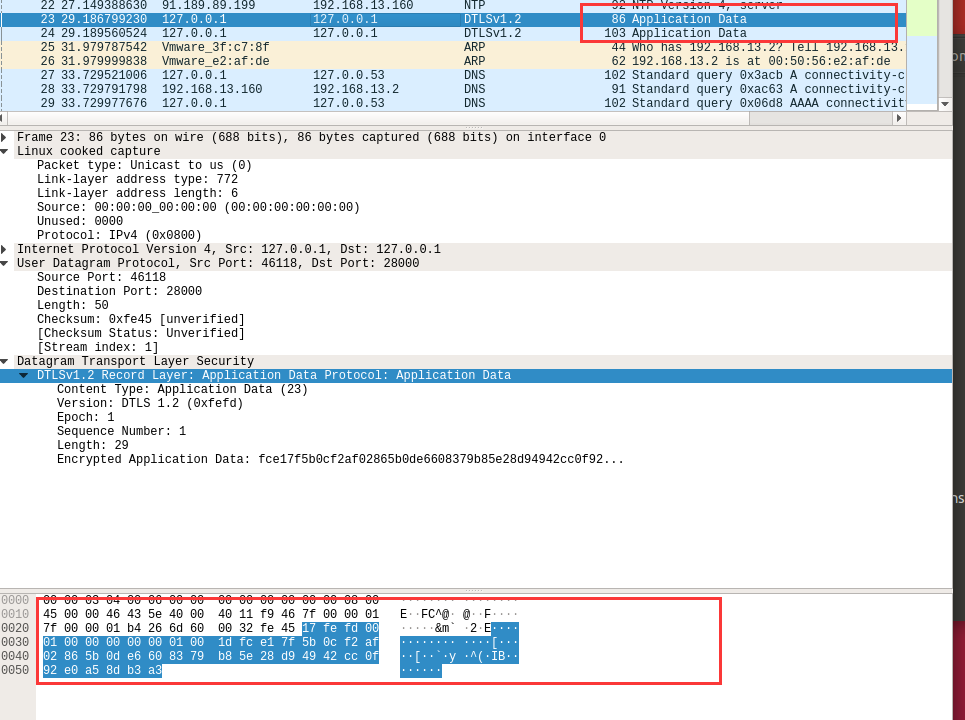
接着传递公钥：128位的RSA公钥



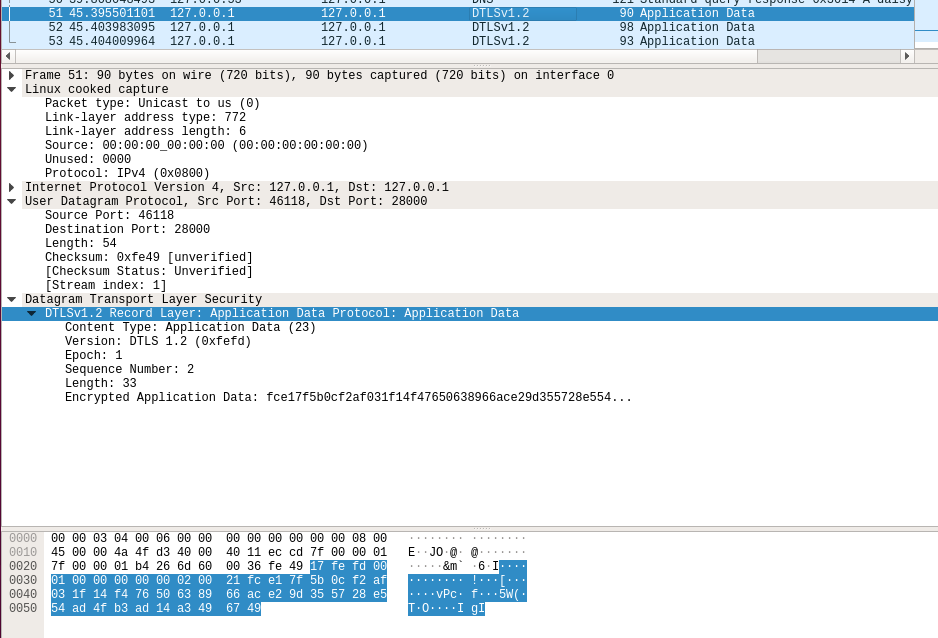
最后实现change cipher spec：



传输数据的过程也是通过加密进行传输的：



最后实现GET传递文件时，由于文件过大分了两次传输，如下图示：



# 六、思考题

1. QUIC 协议（HTTP/3）也是基于 UDP 的，同样提供了安全传输机制，和 DTLS 之间在传输层安全上有何区别？各自优缺点是什么？

2. DTLS 可能存在那些安全缺陷以及如何避免？

**标准：**

本题为加分项。每题答案500字以内，各5分，字数不影响评分。每题答案条理清晰，观点正确得3分，观点正确且新颖得5分，与他人雷同不得分。

# 七、评分细则

1.未完成基本的实验步骤（即客户端、服务端正确通信并握手成功；成功从服务端下载文件到客户端；协议抓包；回答思考题）则不及格；

2．在完成基本实验步骤的基础上，规定课时内未完成，另外验收得60分；最后5人验收得70分；中间8人80分；前5人（视情况调整）验收得90分；

3.根据思考题完成情况进行加分。