Youtube分析报告

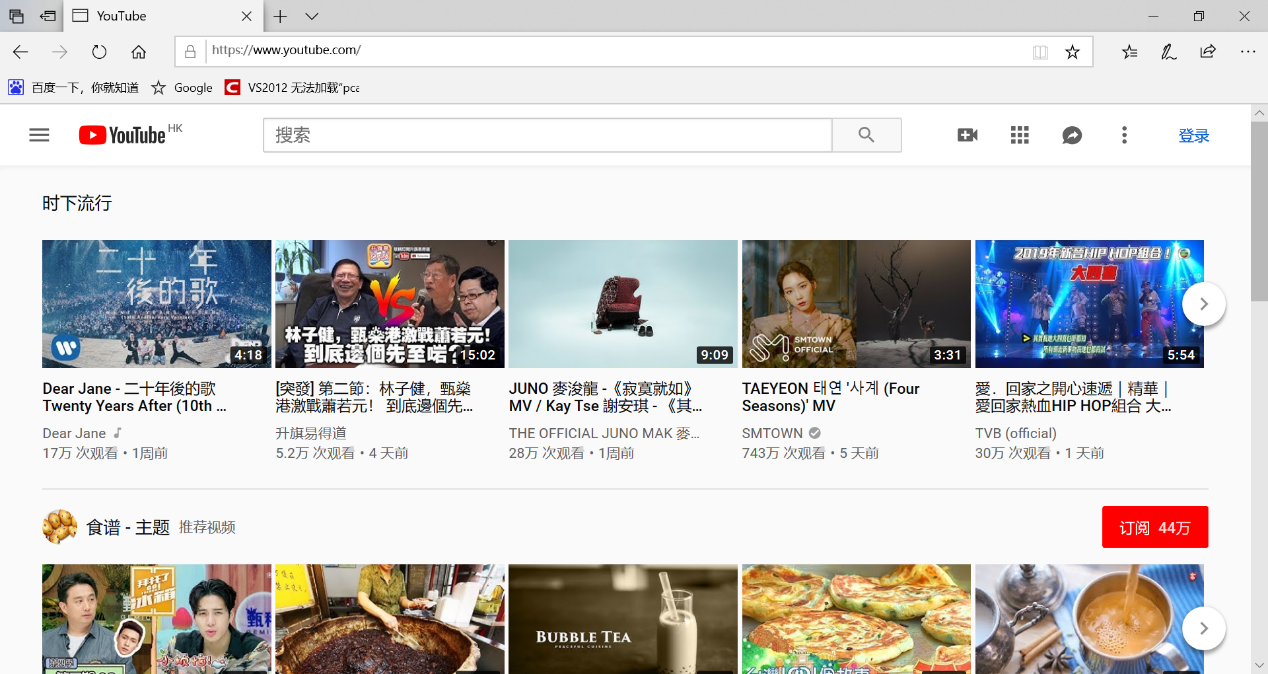
陈力恒 201828015029025 软件所

# 简介

YouTube是一个视频网站，早期公司位于加利福尼亚州的圣布鲁诺。注册于2005年2月15日，由美籍华人陈士骏等人创立，让用户下载、观看及分享影片或短片。

2006年11月，Google公司以16.5亿美元收购了YouTube，并把其当做一家子公司来经营。但是对于如何通过YouTube盈利，Google一直保持非常谨慎的态度。被收购后的YouTube依然风靡全球网民，花旗银行分析师认为，以2012年整年计算，Google可能从YouTube获得24亿美元的净收入。2014年1月3日，YouTube宣布在拉斯维加斯消费电子展(CES)上演示4K高清视频流媒体服务。该服务采用谷歌的视频编解码技术VP9。网站的未注册用户仍可以直接观看视频，而注册用户则可以上传无限制数量的影片。而当影片有可能的冒犯性质的内容时，仅提供给18岁以上的注册用户观看。YouTube作为当前行业内在线视频服务提供商，YouTube的系统每天要处理上千万个视频片段，为全球成千上万的用户提供高水平的视频上传、分发、展示、浏览服务。2015年2月，央视首次把春晚推送到YouTube等境外网站。

Youtube主页如图：



# 软件使用情况

该网址需要科学上网才能访问，本次分析针对PC端的网页版Youtube进行，TLS流量得在本地回传中捕获（浏览器与SS客户端的通信）。

# 测试环境搭建

操作系统：Windows 10

抓包软件：Wireshark Version 3.0.0 (v3.0.0-0-g937e33de)

Fiddler v5.0.20182.28034 for .NET 4

编程软件：Visual Studio 2017

其他：Winpcap库

# 分析工具及要求

使用Wireshark抓包，保存至Pcap文件，用Fiddler对密文解密。利用Visual Studio 开发程序对Pcap文件进行深度分析。对TLS中的可提取的明文信息进行提取。

# 测试条件

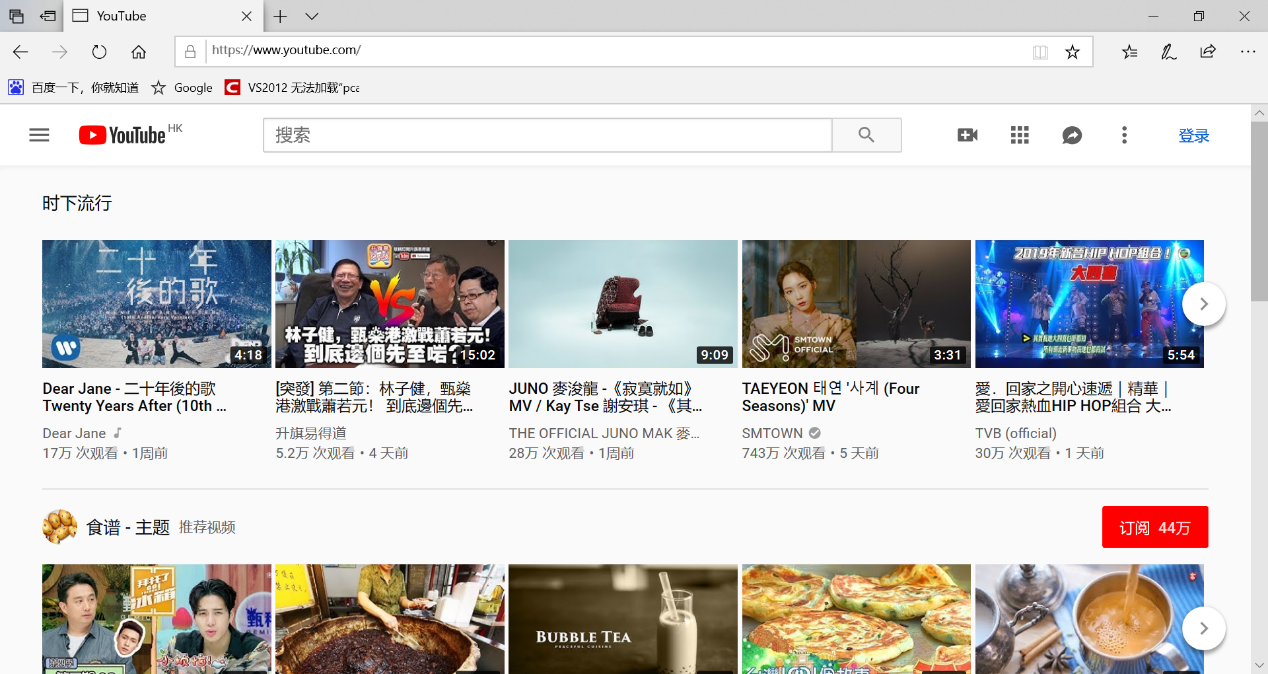
打开SS客户端，开启代理。

# 主要功能及其使用

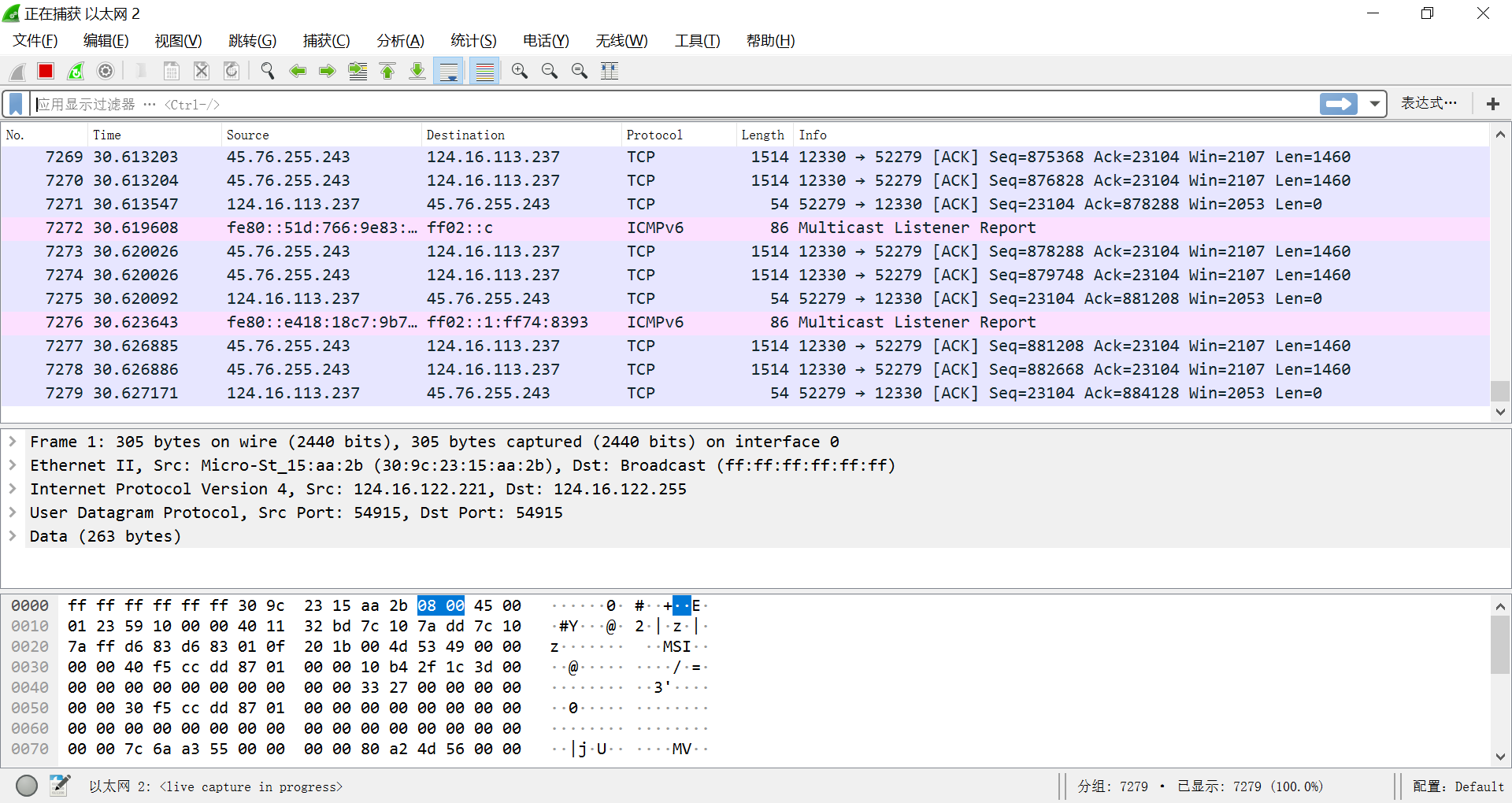
打开Youtube主页，对视频进行播放。

# 抓包过程

* 开启SS客户端，开启代理，开始科学上网。
* 打开Wireshark使用以太网接口抓包。
* 打开Youtube主页。

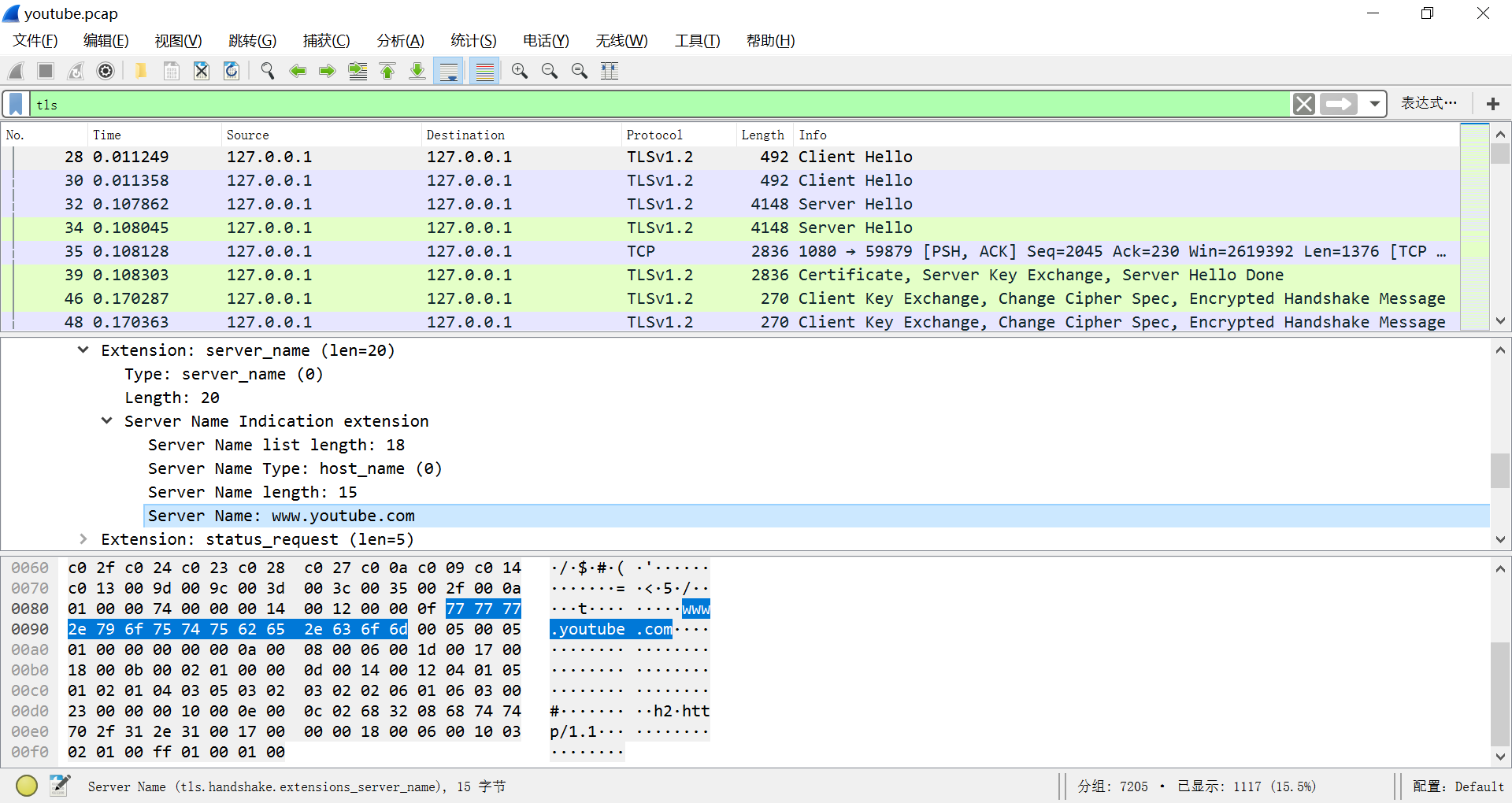


* 发现使用SS之后，所有以太网接口的流量都变成了TCP连接，思考之后便能发现这是因为SS使用了加密传输，才保证我能够科学上网。



* 因此只能够通过本地回传进行具体数据的分析。打开Wireshark Loopback接口，可以发现有我所需要的TLS连接细节。





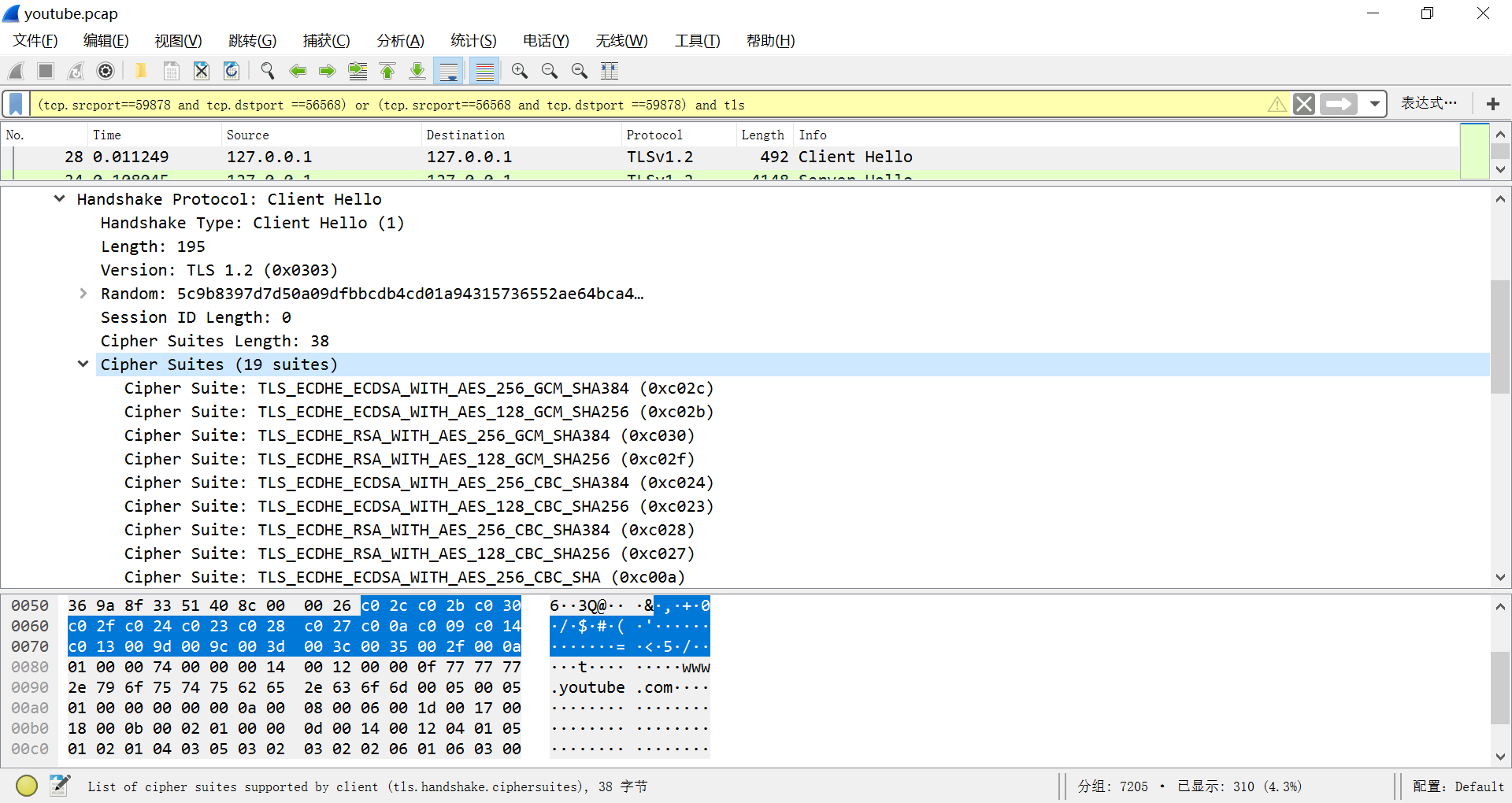
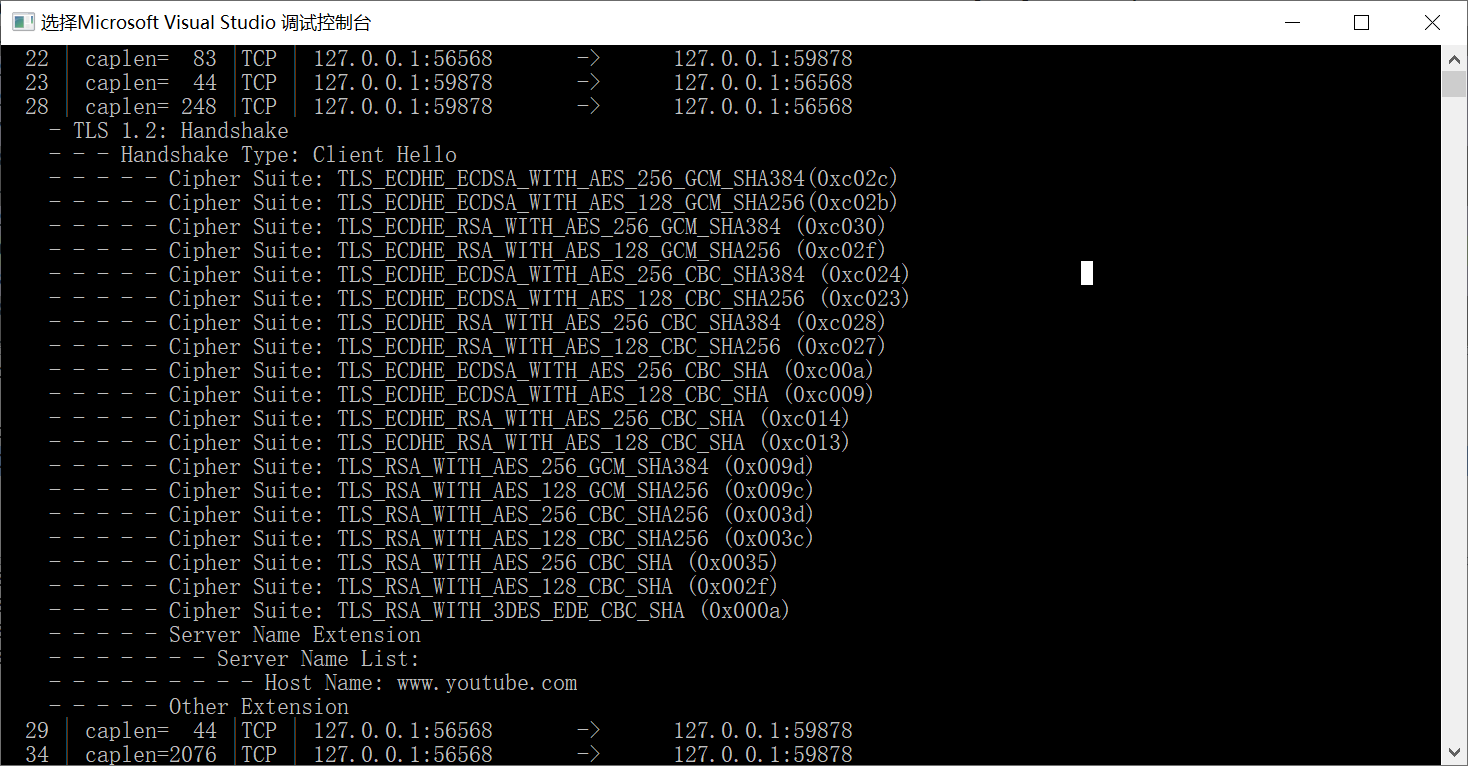
* 保存成youtube.pcap文件。

# 编程分析Pcap

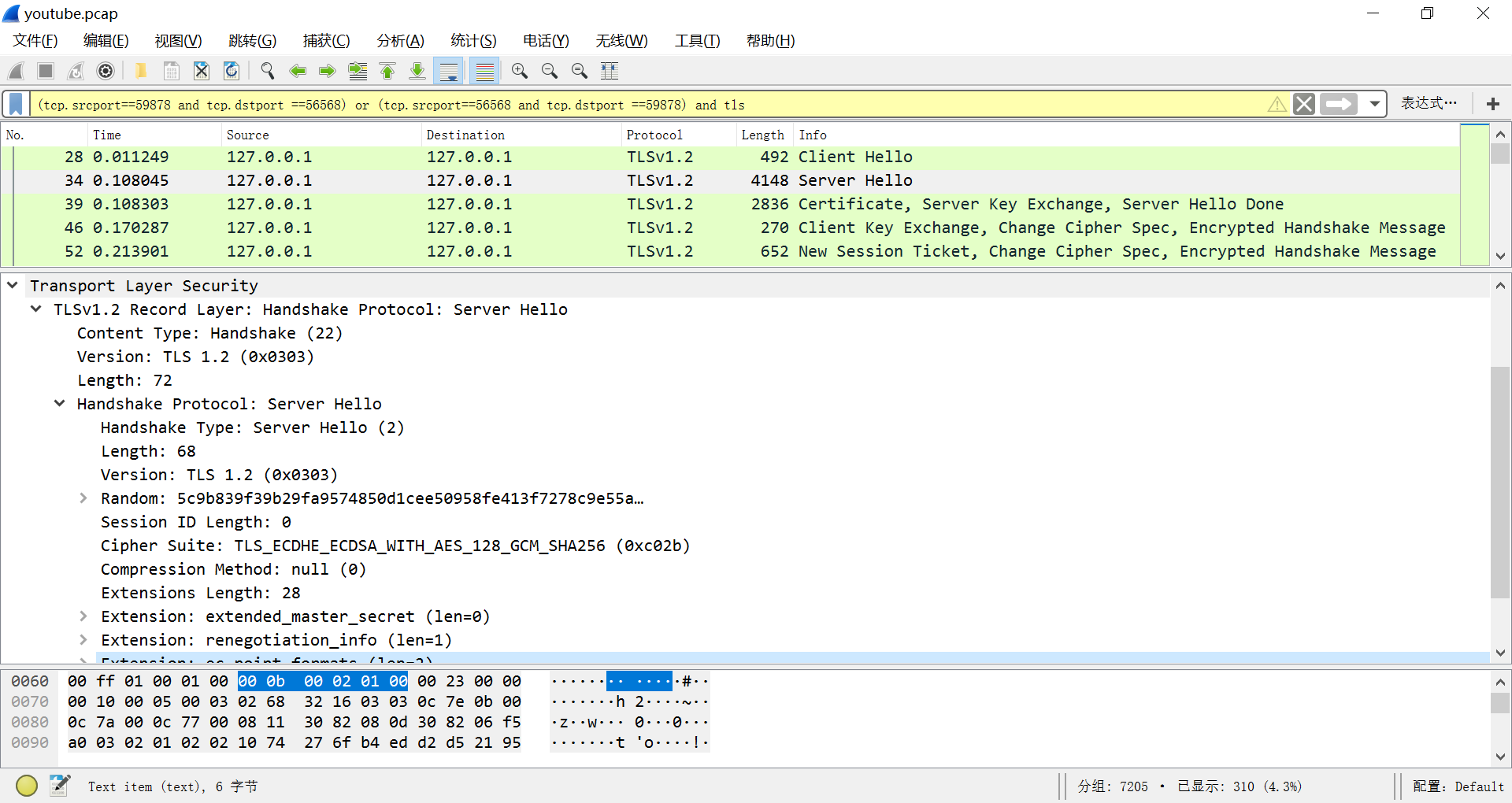
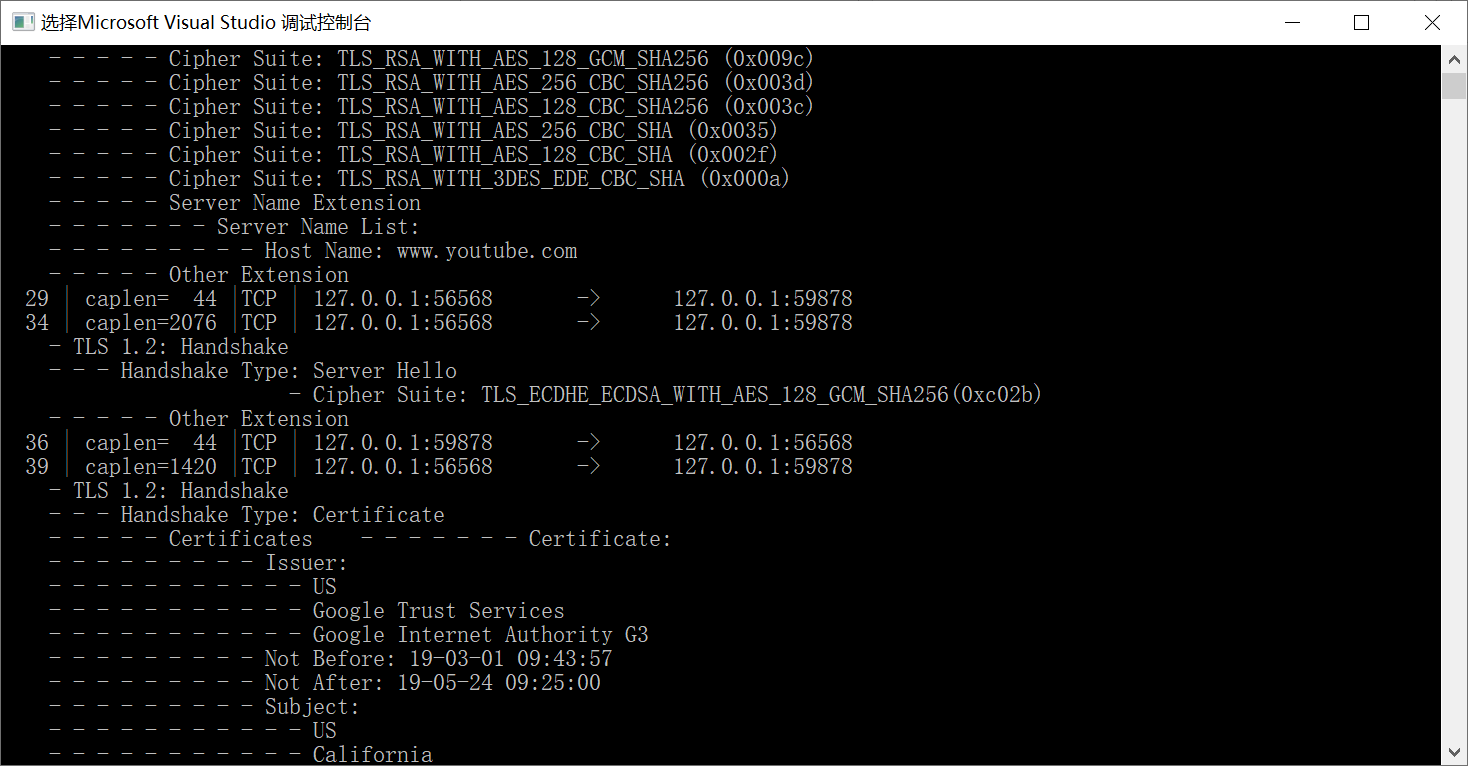
* 通过Winpcap Api，在pcap\_loop和packet\_handler实现对Pcap文件中每一个数据包的捕获。然后利用每一个数据中的数据逐Bit的对协议进行分析，从链接层的Loopback和Ethernet的区分到IP层协议的分析，到TCP层协议的分析，再到Socks、TLS1.2协议的分析。由于TCP层是不标志出应用层的具体协议，因此通过TCP载荷的第一个字节来判断，0x05对应Socks Version5，0x14-0x17对应不同类型的TLS1.2的Record包。
* Socks中关键是对Socks状态的分析，并从中可以提取如请求的域名端口、回复的地址端口。
* 最为困难的还是TLS1.2的分析，因为其过于繁复，需要耐着心一步一步分析。首先可以通过标志0x14-0x17确认TLS包是Change Cipher Spec、Alter、Handshake、Application中的一种。
* 其中Handshake是本次编程中处理细节最多的一个。同样通过标志区分具体的类型0x01对应Client Hello，0x02对应Server Hello等等。接下来依然是对不同的包进行进一层的分析。
* 值得注意的是，在分析Certificate这个Handshake类型的时候，我发现它的证书的数据流是分段的，往往一部分在上一个包Server Hello之后，第二部分在新的一个TLS包中。因此需要设立一个缓冲区将两个段合在一个进行分析。深层的原因应该是：TCP其实并不考虑具体的应用层协议，因此，一个很长的TLS证书，一个TCP包无法装载的下，因此会通过连续若干个TCP包传输。
* 在具体编程过程中，由于Byte众多，一开始经常看错位，因此后来使用Offset来保存并指明当前分析到哪一个字节。这也可以算一个小技巧。
* 另一个值得注意的是，协议类型中的抓包工具显示的很多可见都是从一个字节去对应具体的值（如0x01对应Client Hello），然后显示出来，往往只有域名才在协议中用字符串的ASCII进行传输。

# 编程分析结果的验证

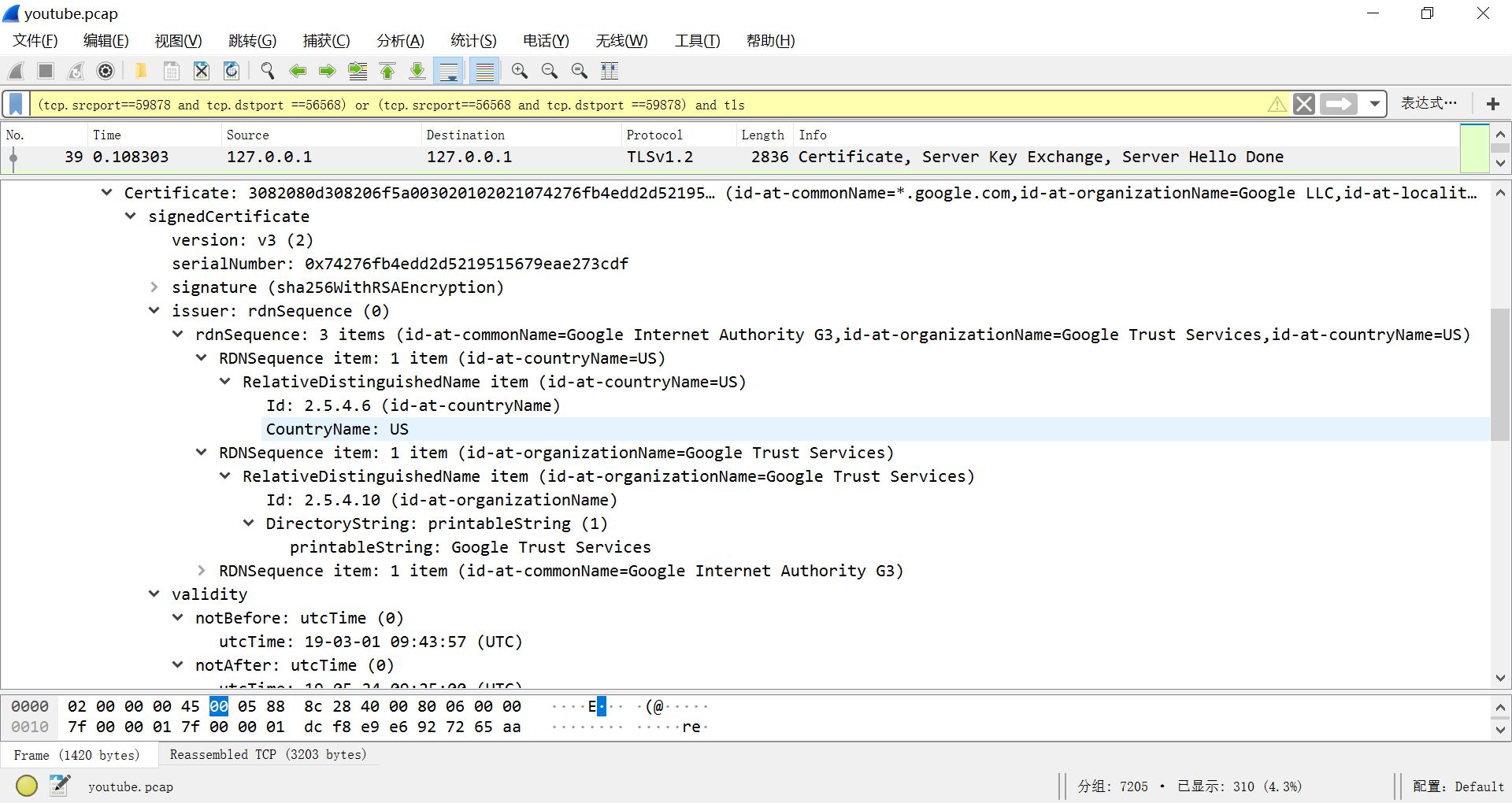
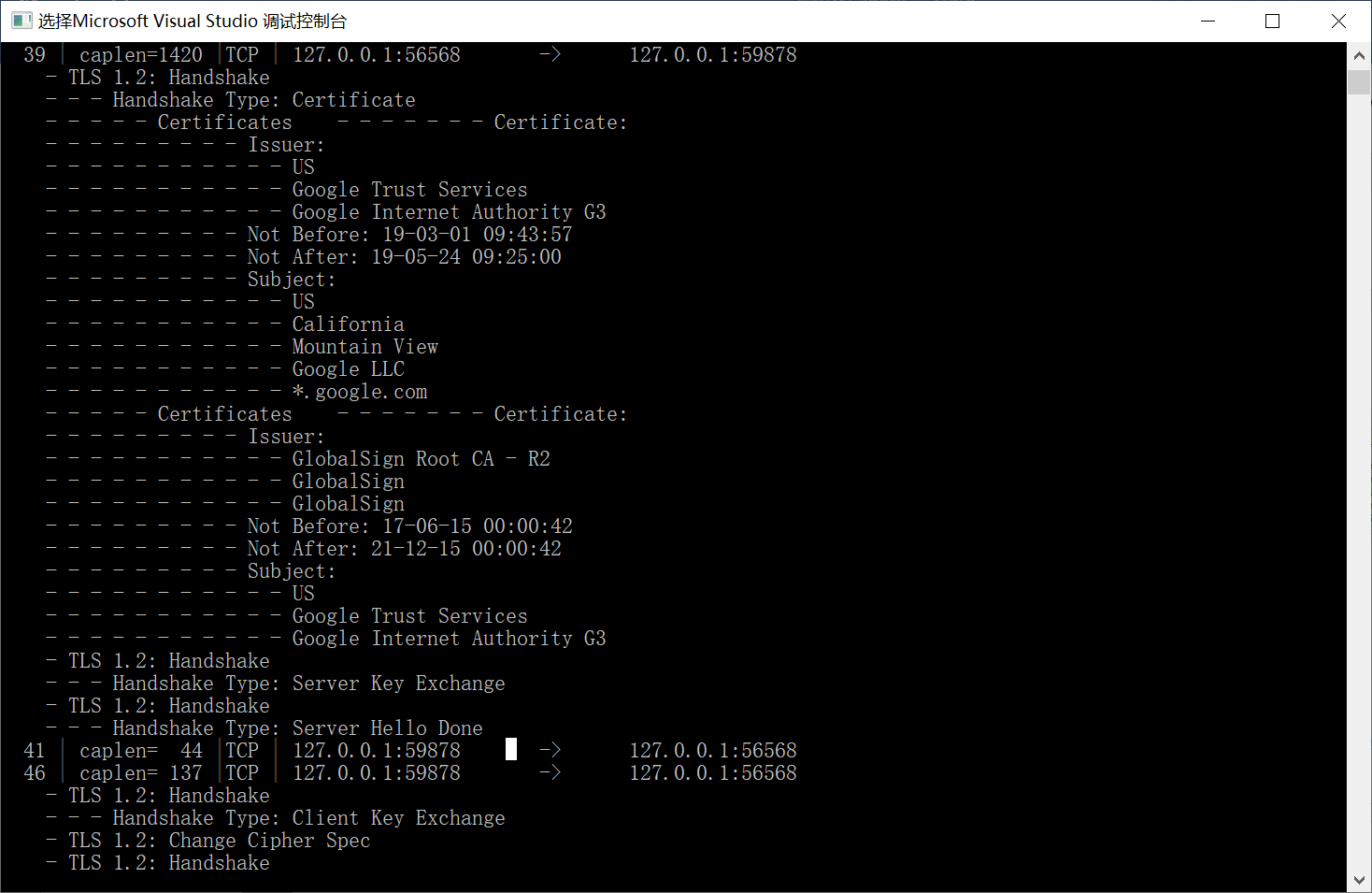
* 其中的Client Hello、Cipher Suite、Server Name都成功识别正确。



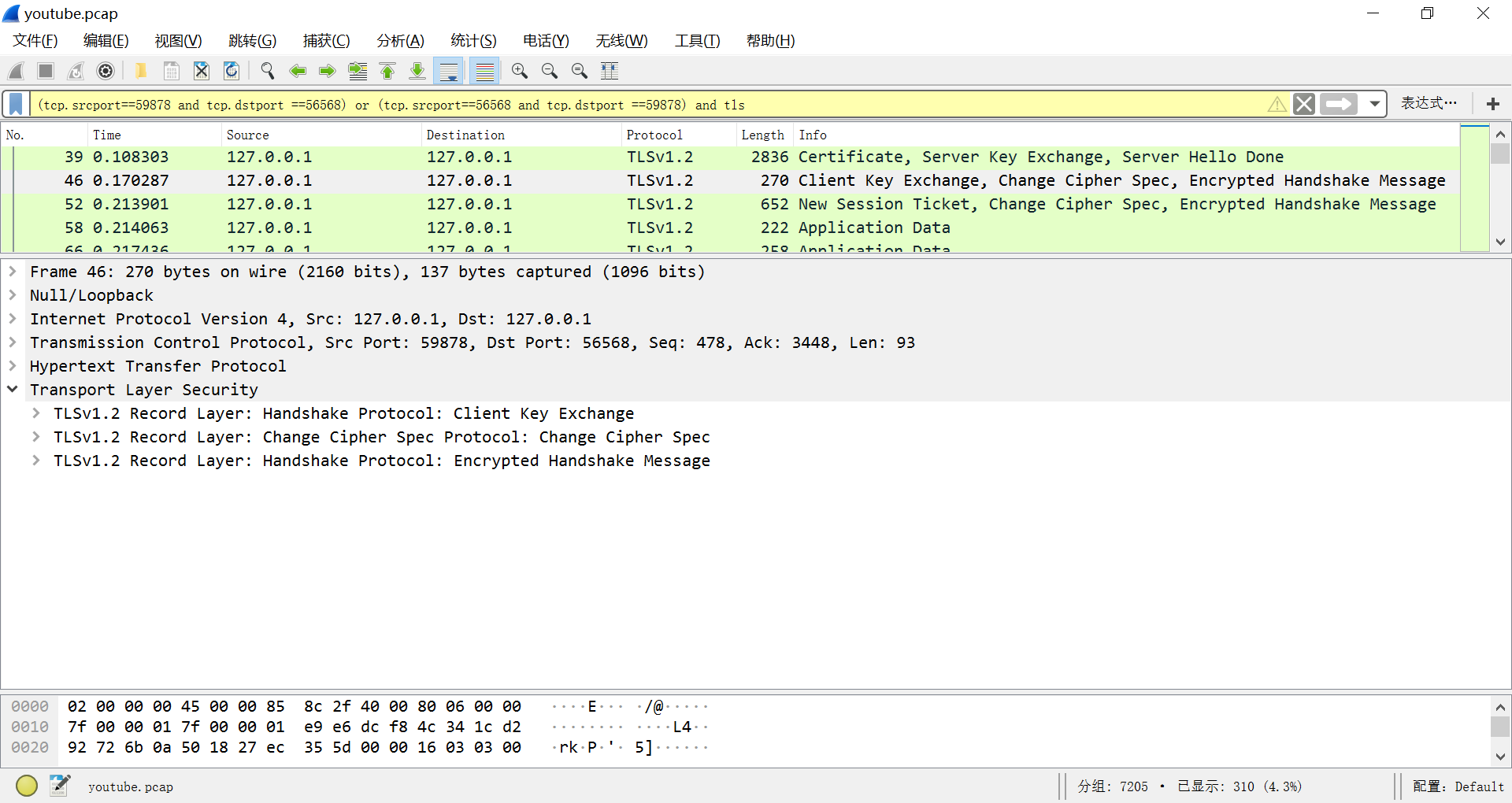
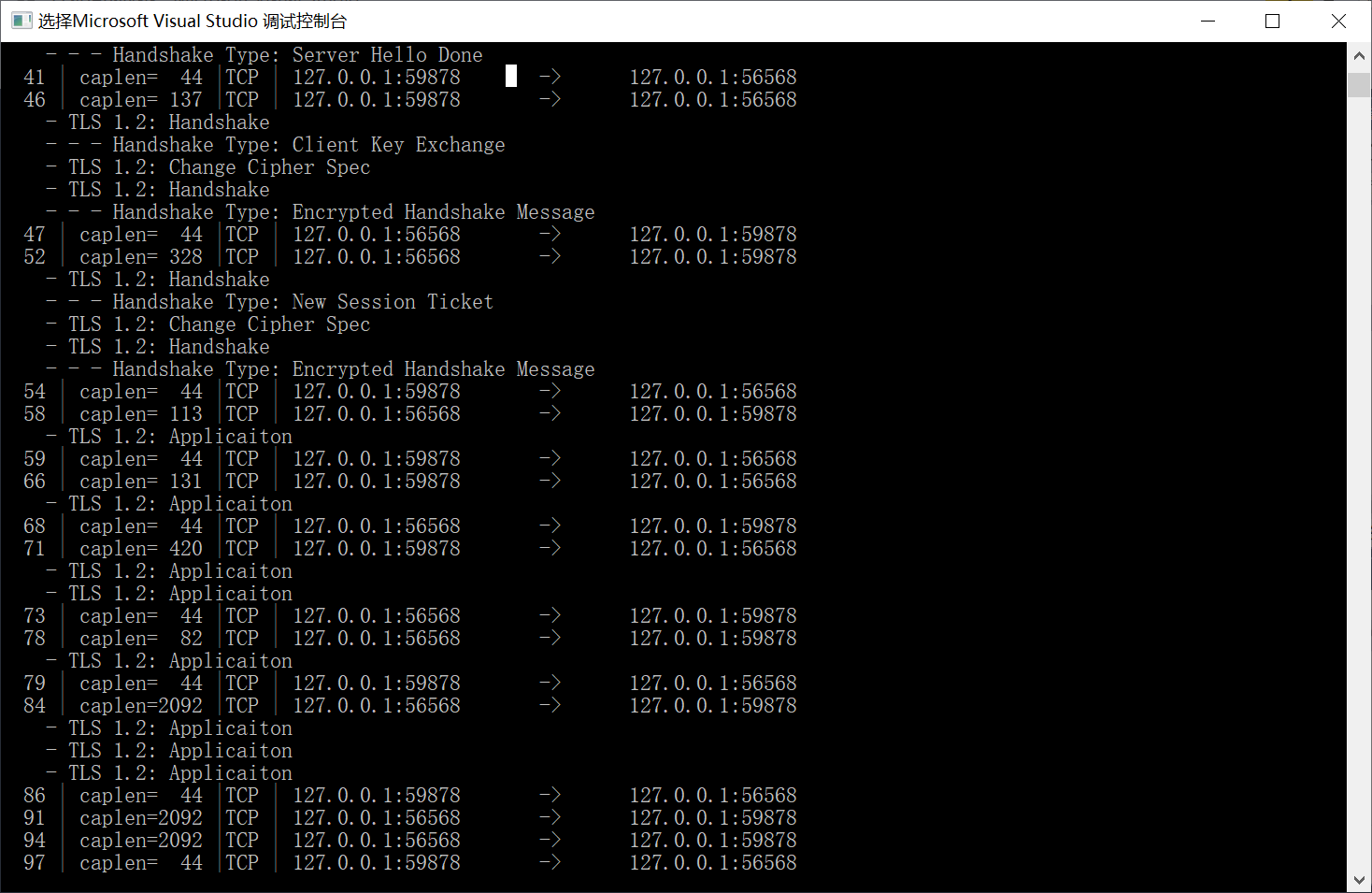
* Server Hello、Cipher Suite均识别正确。



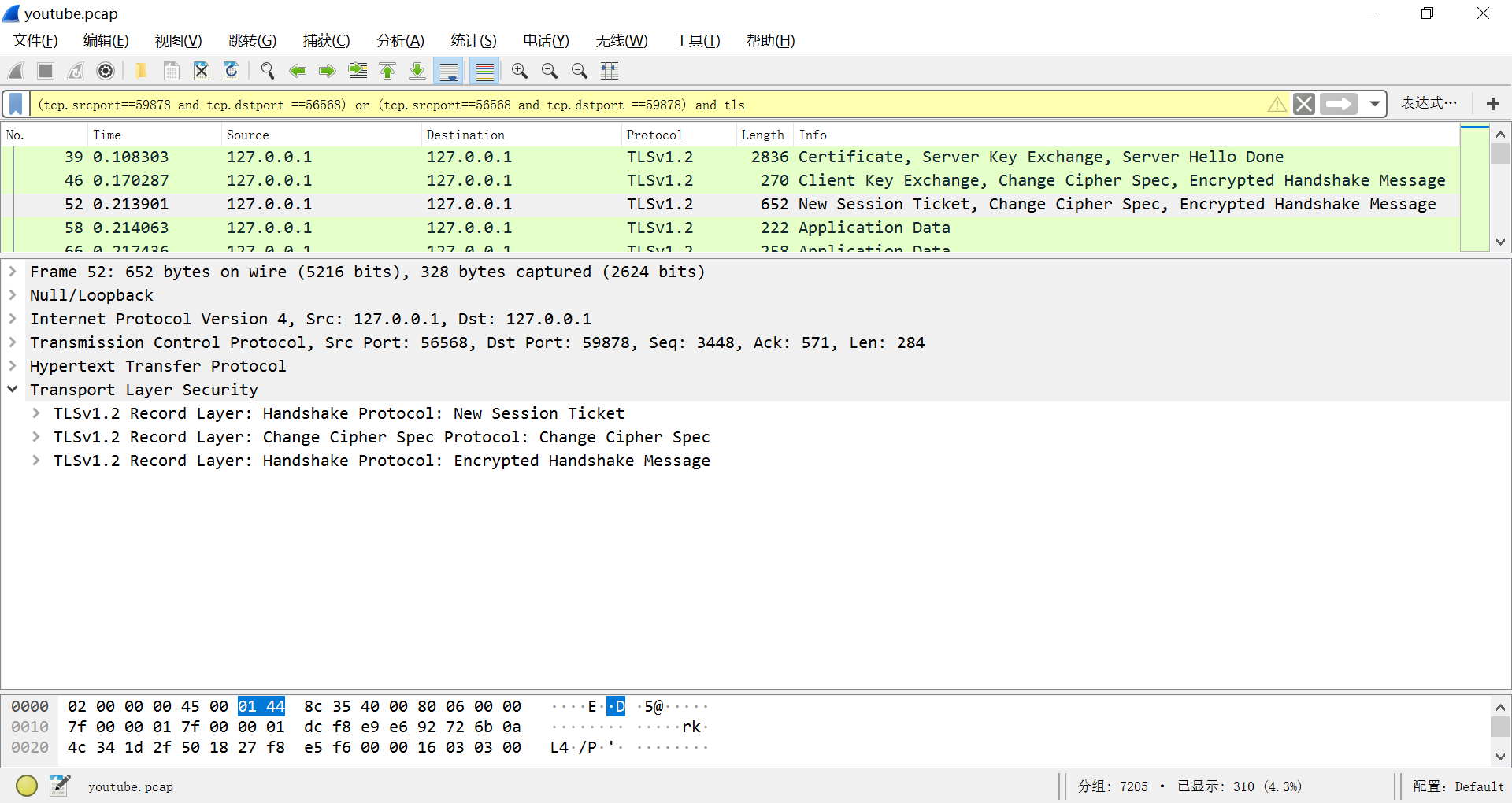
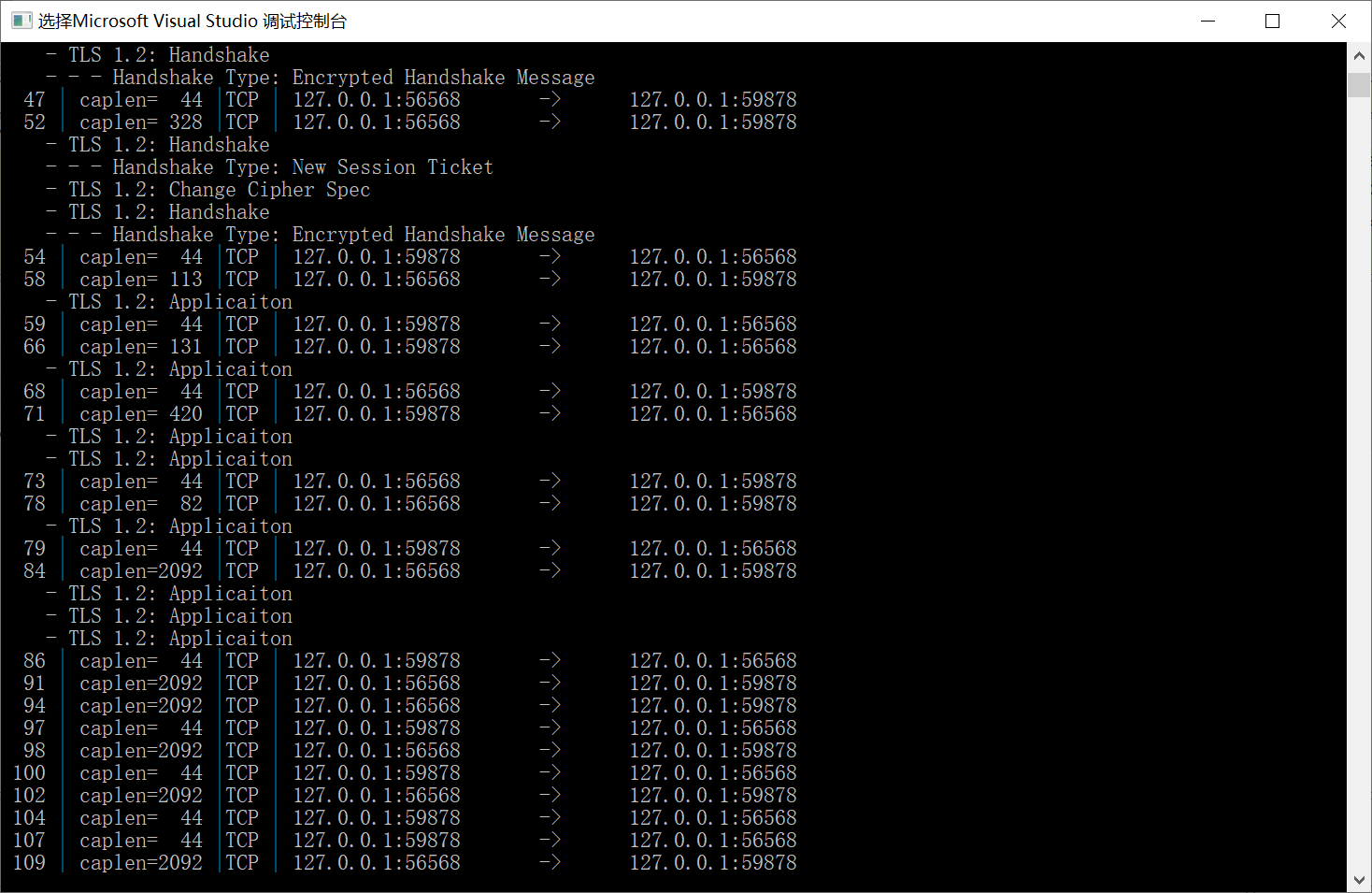
* 证书中的Issuer、Validity、Subject均识别正确。同时Certificate、Server Key Exchange、Server Hello Done 均识别到。



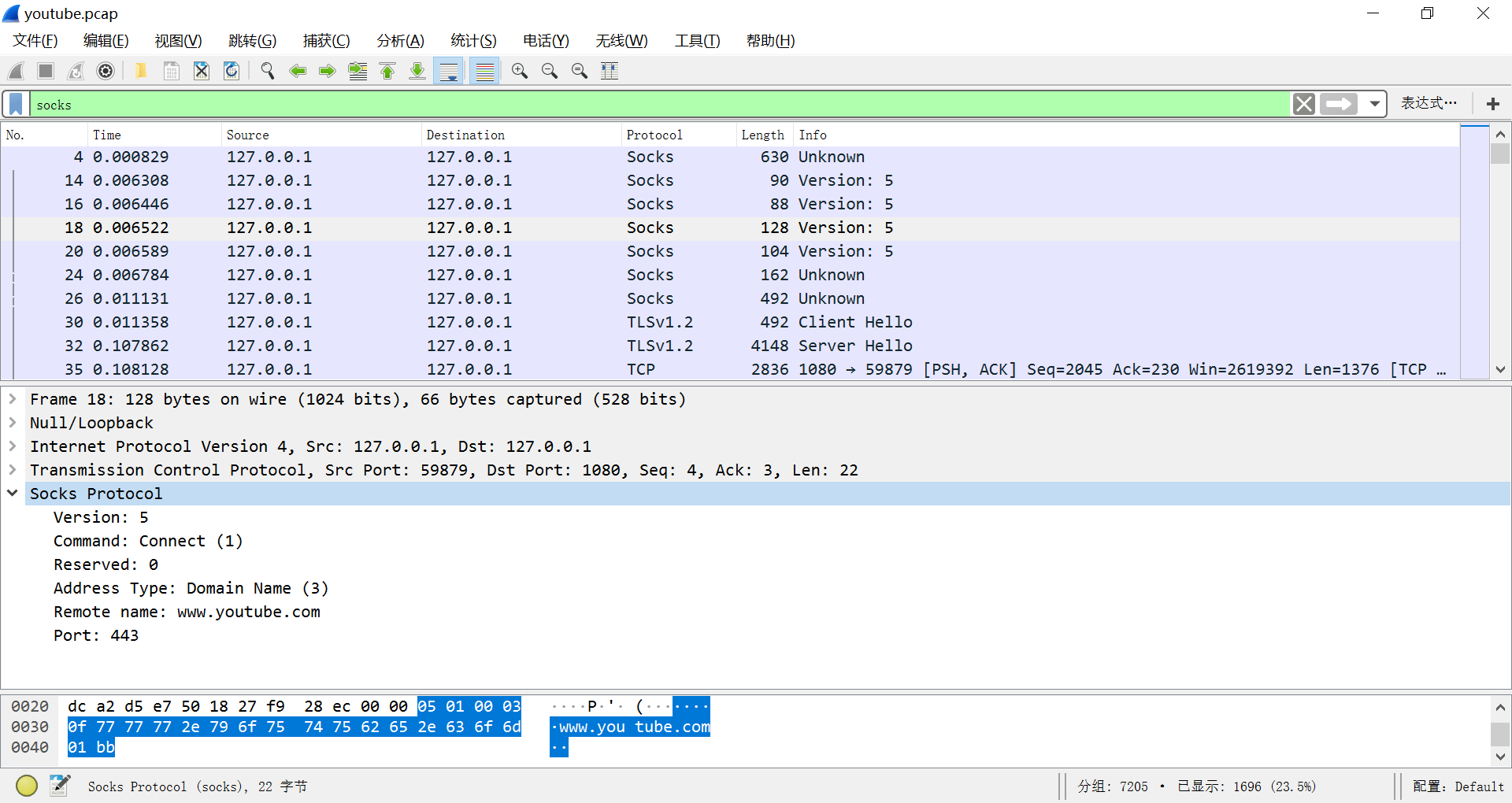
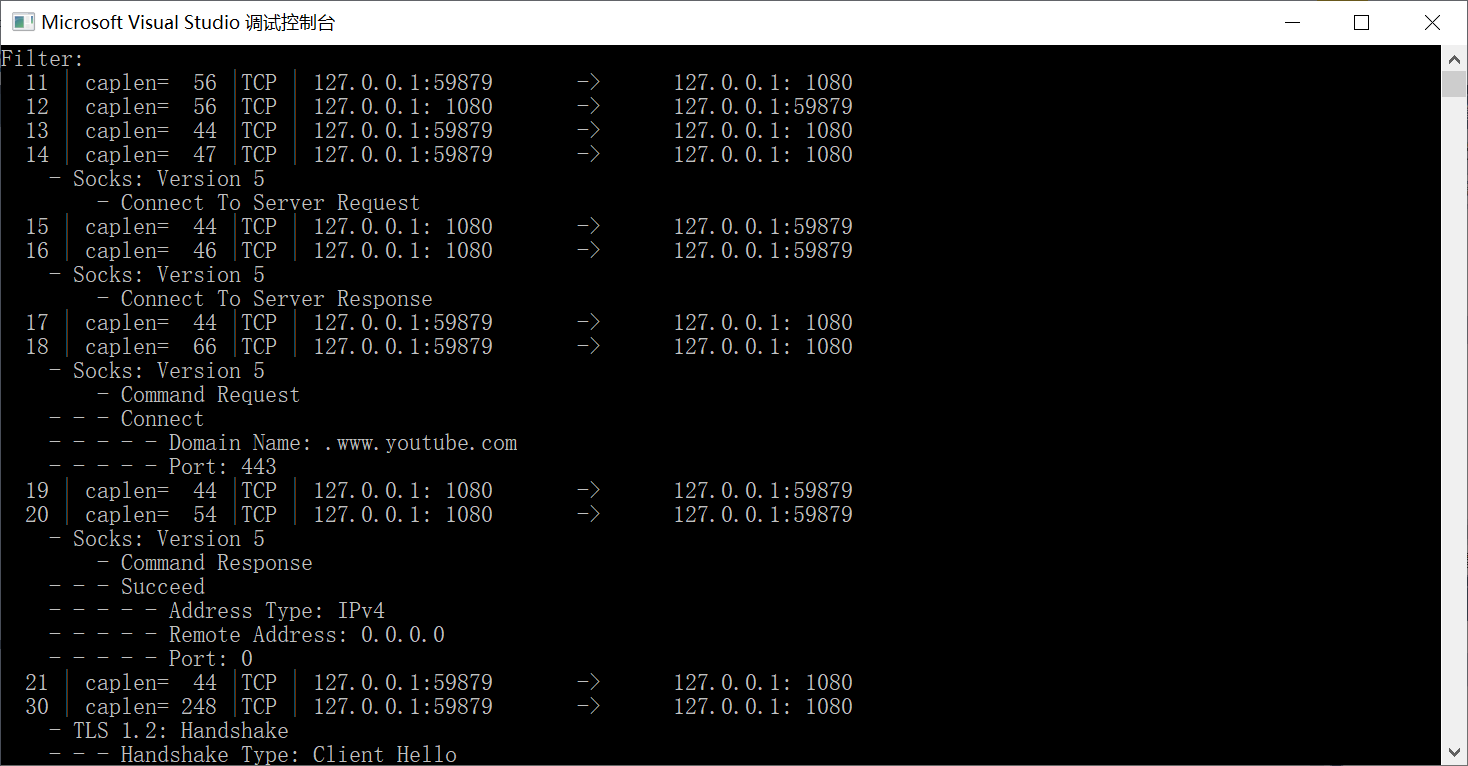
* Client Key Exchange、Change Cipher Spec、Encrypted Handshake Message都正确识别到。



* New Session Ticket、Change Cipher Spec、Encrypted Handshake Message均识别正确。同时Application也正确识别。



* 从上面可以TLS1.2的握手和传输均能有效的完整的识别出。
* 有效的实现了Socks协议的识别和具体内容的分析。



# 密文解析

访问Youtube主页，用Fiddler解析TLS密文。

## 请求报文

GET https://www.youtube.com/ HTTP/1.1

Host: www.youtube.com

Connection: keep-alive

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3

X-Client-Data: CIS2yQEIpLbJAQjEtskBCKmdygEIqKPKAQixp8oBCOKoygEI8KnKAQ==

Accept-Encoding: gzip, deflate, br

Accept-Language: zh-CN,zh;q=0.9

Cookie: VISITOR\_INFO1\_LIVE=lmUgFEp33ag; PREF=f1=50000000&f4=4000000; GPS=1

## 响应报文（部分）

具体见Youtube响应报文.txt

HTTP/1.1 200 OK

X-Frame-Options: SAMEORIGIN

Strict-Transport-Security: max-age=31536000

X-Content-Type-Options: nosniff

Expires: Tue, 27 Apr 1971 19:44:06 EST

P3P: CP="This is not a P3P policy! See http://support.google.com/accounts/answer/151657?hl=en for more info."

Content-Type: text/html; charset=utf-8

Cache-Control: no-cache

Date: Fri, 05 Apr 2019 11:59:00 GMT

Server: YouTube Frontend Proxy

X-XSS-Protection: 0

Set-Cookie: YSC=z-pnazJhcuo; path=/; domain=.youtube.com; httponly

Alt-Svc: quic=":443"; ma=2592000; v="46,44,43,39"

Content-Length: 235542

<!doctype html><html style="font-size: 10px;font-family: Roboto, Arial, sans-serif; " lang="zh-CN"><head><meta http-equiv="origin-trial" data-feature="Media Capabilities" data-expires="2018-04-12" content="AjLq5uNF7MpG/eM34tWcJD3h8yZY1Q72ckfwdKbUNKGtUNaZkrw55eq2tI60vG0Ils

……

# 总结

* Youtube访问需要开启VPN
* 通过Wireshark抓包，并保存Pcap文件。
* 由于传输采用TLS协议，本实验旨在对TLS协议如握手、证书等可提取的信息进行分析并编程显示。此外，也对Socks协议进行了提取、分析和显示。
* 本文对编程细节进行了一定阐述。
* 对于具体的密文载荷，通过Fiddler进行分析便可识别。
* 展示了所开发的软件的显示结果。