HW4: Wireshark

Name: Wang Haoyuan

StuID: 3220105114

> 考虑到报告描述等问题，本报告采用中文书写

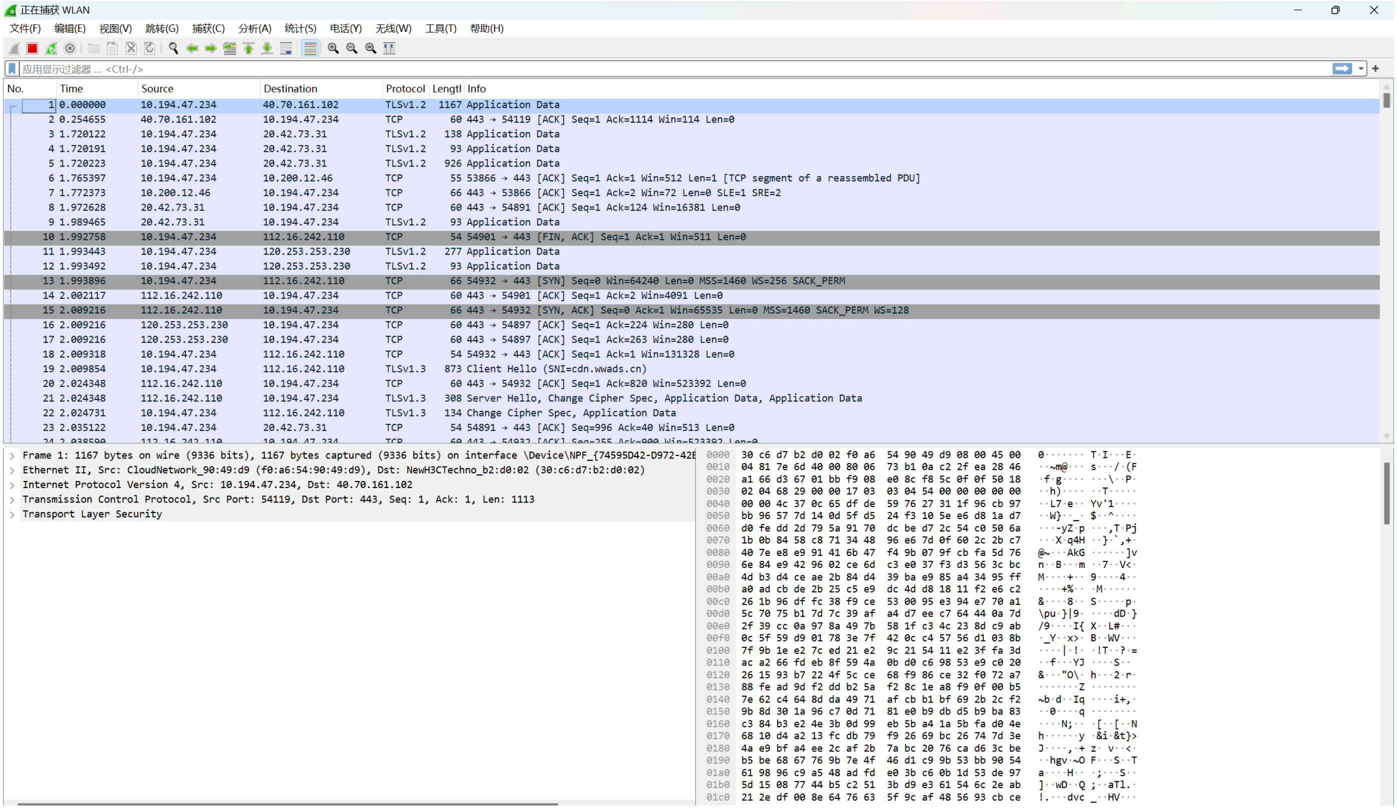
实验目的

本实验的目的为：通过WireShark对网址“https://www.zju.edu.cn” 在访问时传输的数据包进行抓取，并且做出分析。

在本报告中，成功抓取到其作为https加密后的数据包，并根据服务器密钥对其解密，最终获取到HTTP属性的包。

实验流程

step 1: 下载WireShark，选择"WLAN"(WIFI路由)作为线路，并尝试第一次抓取：



```
No.  Time      Source           Destination      Protocol Lengtl Info
   1 0.000000  10.194.47.234    40.70.161.102    TLSv1.2  1167 Application Data
   2 0.254655  40.70.161.102    10.194.47.234    TCP        60 443 → 54119 [ACK] Seq=1 Ack=1114 Win=114 Len=0
   3 1.720122  10.194.47.234    20.42.73.31      TLSv1.2   138 Application Data
   4 1.720191  10.194.47.234    20.42.73.31      TLSv1.2    93 Application Data
   5 1.720223  10.194.47.234    20.42.73.31      TLSv1.2   926 Application Data
   6 1.765397  10.194.47.234    10.200.12.46     TCP        55 53866 → 443 [ACK] Seq=1 Ack=1 Win=512 Len=1 [TCP segment of a reassembled PDU]
   7 1.772373  10.200.12.46     10.194.47.234    TCP        66 443 → 53866 [ACK] Seq=1 Ack=2 Win=72 Len=0 SLE=1 SRE=2
   8 1.972628  20.42.73.31      10.194.47.234    TCP        60 443 → 54891 [ACK] Seq=1 Ack=124 Win=16381 Len=0
   9 1.989465  20.42.73.31      10.194.47.234    TLSv1.2    93 Application Data
  10 1.992758  10.194.47.234    112.16.242.110   TCP        54 54901 → 443 [FIN, ACK] Seq=1 Ack=1 Win=511 Len=0
  11 1.993443  10.194.47.234    120.253.253.230  TLSv1.2   277 Application Data
  12 1.993492  10.194.47.234    120.253.253.230  TLSv1.2    93 Application Data
  13 1.993896  10.194.47.234    112.16.242.110   TCP        66 54932 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
  14 2.002117  112.16.242.110   10.194.47.234    TCP        60 443 → 54901 [ACK] Seq=1 Ack=2 Win=4091 Len=0
  15 2.009216  112.16.242.110   10.194.47.234    TCP        66 443 → 54932 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM WS=128
  16 2.009216  120.253.253.230  10.194.47.234    TCP        60 443 → 54897 [ACK] Seq=1 Ack=224 Win=280 Len=0
  17 2.009216  120.253.253.230  10.194.47.234    TCP        60 443 → 54897 [ACK] Seq=1 Ack=263 Win=280 Len=0
  18 2.009318  10.194.47.234    112.16.242.110   TCP        54 54932 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
  19 2.009854  10.194.47.234    112.16.242.110   TLSv1.3   873 Client Hello (SNI=cdn.wwads.cn)
  20 2.024348  112.16.242.110   10.194.47.234    TCP        60 443 → 54932 [ACK] Seq=1 Ack=820 Win=523392 Len=0
  21 2.024348  112.16.242.110   10.194.47.234    TLSv1.3   308 Server Hello, Change Cipher Spec, Application Data, Application Data
  22 2.024731  10.194.47.234    112.16.242.110   TLSv1.3   134 Change Cipher Spec, Application Data
  23 2.035122  10.194.47.234    20.42.73.31      TCP        54 54891 → 443 [ACK] Seq=996 Ack=40 Win=513 Len=0
  24 2.038590  112.16.242.110   10.194.47.234    TCP        60 443 → 54932 [ACK] Seq=255 Ack=900 Win=523392 Len=0
```

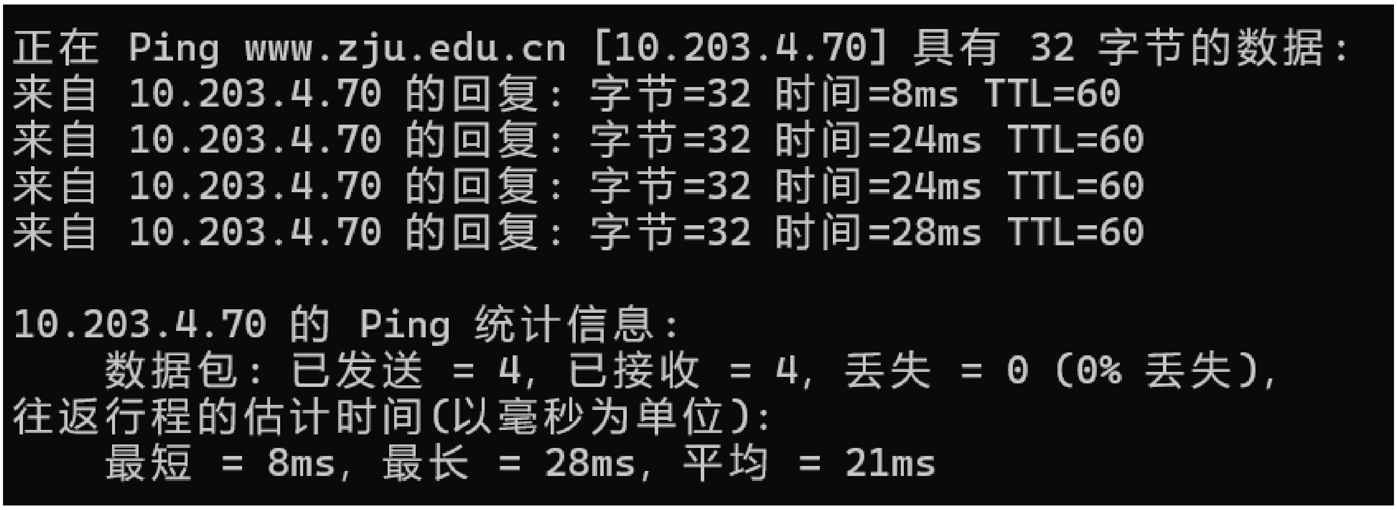
```
> Frame 1: 1167 bytes on wire (9336 bits), 1167 bytes captured (9336 bits) on interface \Device\NPF_{74595D42-D972-42E
> Ethernet II, Src: CloudNetwork_90:49:d9 (f0:a6:54:90:49:d9), Dst: NewH3CTechno_b2:d0:02 (30:c6:d7:b2:d0:02)
> Internet Protocol Version 4, Src: 10.194.47.234, Dst: 40.70.161.102
> Transmission Control Protocol, Src Port: 54119, Dst Port: 443, Seq: 1, Ack: 1, Len: 1113
> Transport Layer Security
```

```
0000  30 c6 d7 b2 d0 02 f0 a6  54 90 49 d9 08 00 45 00   0······· T·I···E·
0010  04 81 7e 6d 40 00 80 06  73 b1 0a c2 2f ea 28 46   ··~m@··· s···/·(F
0020  a1 66 d3 67 01 bb f9 08  e0 8c f8 5c 0f 0f 50 18   ·f·g···· ···\··P·
0030  02 04 68 29 00 00 17 03  03 04 54 00 00 00 00 00   ··h)···· ··T·····
0040  00 00 4c 37 0c 65 df de  59 76 27 31 1f 96 cb 97   ··L7·e·· Yv'1····
0050  bb 96 57 7d 14 0d 5f d5  24 f3 10 5e e6 d8 1a d7   ··W}··_· $··^····
0060  d0 fe dd 2d 79 5a 91 70  dc be d7 2c 54 c0 50 6a   ···-yZ·p ···,T·Pj
0070  1b 0b 84 58 c8 71 34 48  96 e6 7d 0f 60 2c 2b c7   ···X·q4H ··}·`,+·
0080  40 7e e8 e9 91 41 6b 47  f4 9b 07 9f cb fa 5d 76   @~···AkG ······]v
0090  6e 84 e9 42 96 02 ce 6d  c3 e0 37 f3 d3 56 3c bc   n··B···m ··7··V<·
00a0  4d b3 d4 ce ae 2b 84 d4  39 ba e9 85 a4 34 95 ff   M····+·· 9····4··
00b0  a0 ad cb de 2b 25 c5 e9  dc 4d d8 18 11 f2 e6 c2   ····+%·· ·M······
00c0  26 1b 96 df fc 38 f9 ce  53 00 95 e3 94 e7 70 a1   &····8·· S·····p·
00d0  5c 70 75 b1 7d 7c 39 af  a4 d7 ee c7 64 44 0a 7d   \pu·}|9· ····dD·}
00e0  2f 39 cc 0a 97 8a 49 7b  58 1f c3 4c 23 8d c9 ab   /9····I{ X··L#···
00f0  0c 5f 59 d9 01 78 3e 7f  42 0c c4 57 56 d1 03 8b   ·_Y··x>· B··WV···
0100  7f 9b 1e e2 7c ed 21 e2  9c 21 54 11 e2 3f fa 3d   ····|·!· ·!T··?·=
0110  ac a2 66 fd eb 8f 59 4a  0b d0 c6 98 53 e9 c0 20   ··f···YJ ····S·· 
0120  26 15 93 b7 22 4f 5c ce  68 f9 86 ce 32 f0 72 a7   &···"O\· h···2·r·
0130  88 fe ad 9d f2 dd b2 5a  f2 8c 1e a8 f9 0f 00 b5   ·······Z ········
0140  7e 62 c4 64 8d da 49 71  af cb b1 bf 69 2b 2c f2   ~b·d··Iq ····i+,·
0150  9b 8d 30 1a 96 c7 0d 71  81 e0 b9 db d5 b9 ba 83   ··0····q ········
0160  c3 84 b3 e2 4e 3b 0d 99  eb 5b a4 1a 5b fa d0 4e   ····N;·· ·[··[··N
0170  68 10 d4 a2 13 fc db 79  f9 26 69 bc 26 74 7d 3e   h······y ·&i·&t}>
0180  4a e9 bf a4 ee 2c af 2b  7a bc 20 76 ca d6 3c be   J····,·+ z· v··<·
0190  b5 be 68 67 76 9b 7e 4f  46 d1 c9 9b 53 bb 90 54   ··hgv·~O F···S··T
01a0  61 98 96 c9 a5 48 ad fd  e0 3b c6 0b 1d 53 de 97   a····H·· ·;···S··
01b0  5d 15 08 77 44 b5 c2 51  3b d9 e3 61 54 6c 2e ab   ]··wD··Q ;··aTl.·
01c0  21 2e df 00 8e 64 76 63  5f 9c af 48 56 93 cb ce   !.···dvc _··HV···
```

此时抓取的是在线路中的所有数据包，包括其他网址发送的包，因此数量较大且难以整理。

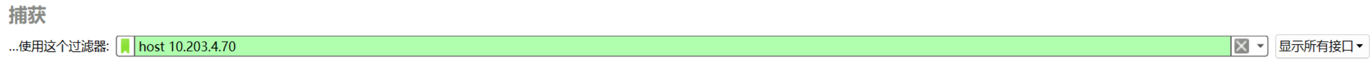
step 2: 采用显示过滤器，对获取的数据包进行筛选：

通过ping目标网址，可以获取到对应网址的IP地址为：10.203.4.70



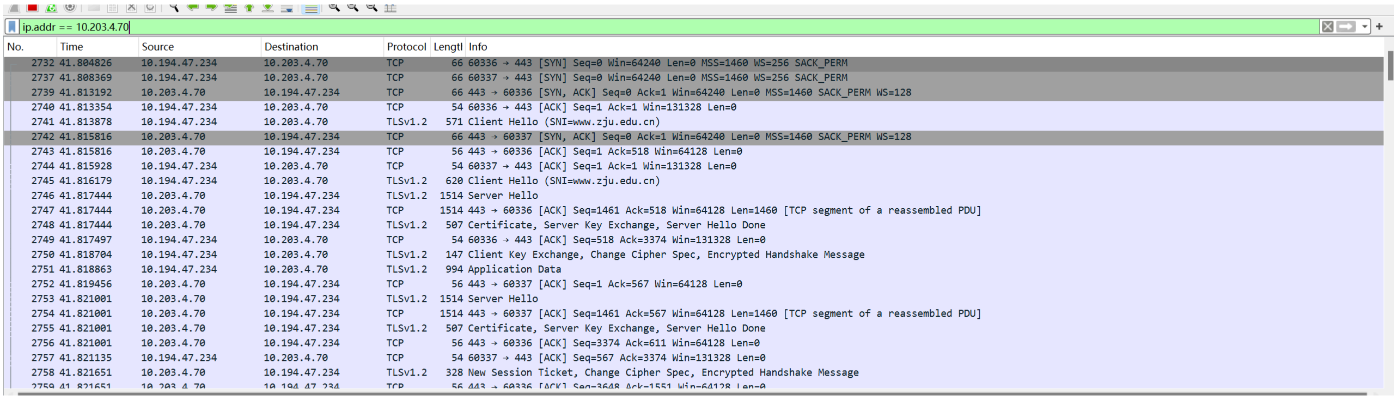
```
正在 Ping www.zju.edu.cn [10.203.4.70] 具有 32 字节的数据：
来自 10.203.4.70 的回复: 字节=32 时间=8ms TTL=60
来自 10.203.4.70 的回复: 字节=32 时间=24ms TTL=60
来自 10.203.4.70 的回复: 字节=32 时间=24ms TTL=60
来自 10.203.4.70 的回复: 字节=32 时间=28ms TTL=60

10.203.4.70 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 8ms, 最长 = 28ms, 平均 = 21ms
```

在抓取数据包前，使用捕获过滤器对抓取范围进行限制（只抓取src或dst为这个IP地址的包）



...使用这个过滤器: host 10.203.4.70

或在WireShark页面抓取数据包时，在显示过滤器中输入"ip.addr = 10.203.4.70"，即可筛选出对应网站与主机之间的数据包传输情况：



ip.addr == 10.203.4.70

```
No.    Time       Source         Destination    Protocol Lengtl Info
  2732 41.804826  10.194.47.234  10.203.4.70    TCP        66 60336 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
  2737 41.808369  10.194.47.234  10.203.4.70    TCP        66 60337 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
  2739 41.813192  10.203.4.70    10.194.47.234  TCP        66 443 → 60336 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
  2740 41.813354  10.194.47.234  10.203.4.70    TCP        54 60336 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
  2741 41.813878  10.194.47.234  10.203.4.70    TLSv1.2   571 Client Hello (SNI=www.zju.edu.cn)
  2742 41.815816  10.203.4.70    10.194.47.234  TCP        66 443 → 60337 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
  2743 41.815816  10.203.4.70    10.194.47.234  TCP        56 443 → 60336 [ACK] Seq=1 Ack=518 Win=64128 Len=0
  2744 41.815928  10.194.47.234  10.203.4.70    TCP        54 60337 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
  2745 41.816179  10.194.47.234  10.203.4.70    TLSv1.2   620 Client Hello (SNI=www.zju.edu.cn)
  2746 41.817444  10.203.4.70    10.194.47.234  TLSv1.2  1514 Server Hello
  2747 41.817444  10.203.4.70    10.194.47.234  TCP      1514 443 → 60336 [ACK] Seq=1461 Ack=518 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
  2748 41.817444  10.203.4.70    10.194.47.234  TLSv1.2   507 Certificate, Server Key Exchange, Server Hello Done
  2749 41.817497  10.194.47.234  10.203.4.70    TCP        54 60336 → 443 [ACK] Seq=518 Ack=3374 Win=131328 Len=0
  2750 41.818704  10.194.47.234  10.203.4.70    TLSv1.2   147 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
  2751 41.818863  10.194.47.234  10.203.4.70    TLSv1.2   994 Application Data
  2752 41.819456  10.203.4.70    10.194.47.234  TCP        56 443 → 60337 [ACK] Seq=1 Ack=567 Win=64128 Len=0
  2753 41.821001  10.203.4.70    10.194.47.234  TLSv1.2  1514 Server Hello
  2754 41.821001  10.203.4.70    10.194.47.234  TCP      1514 443 → 60337 [ACK] Seq=1461 Ack=567 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
  2755 41.821001  10.203.4.70    10.194.47.234  TLSv1.2   507 Certificate, Server Key Exchange, Server Hello Done
  2756 41.821001  10.203.4.70    10.194.47.234  TCP        56 443 → 60336 [ACK] Seq=3374 Ack=611 Win=64128 Len=0
  2757 41.821135  10.194.47.234  10.203.4.70    TCP        54 60337 → 443 [ACK] Seq=567 Ack=3374 Win=131328 Len=0
  2758 41.821651  10.203.4.70    10.194.47.234  TLSv1.2   328 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
  2759 41.821651  10.203.4.70    10.194.47.234  TCP        56 443 → 60336 [ACK] Seq=3648 Ack=1551 Win=64128 Len=0
```

此时能够看到server与client之间建立联系的TCP数据包，包括数据传输请求的数据包。

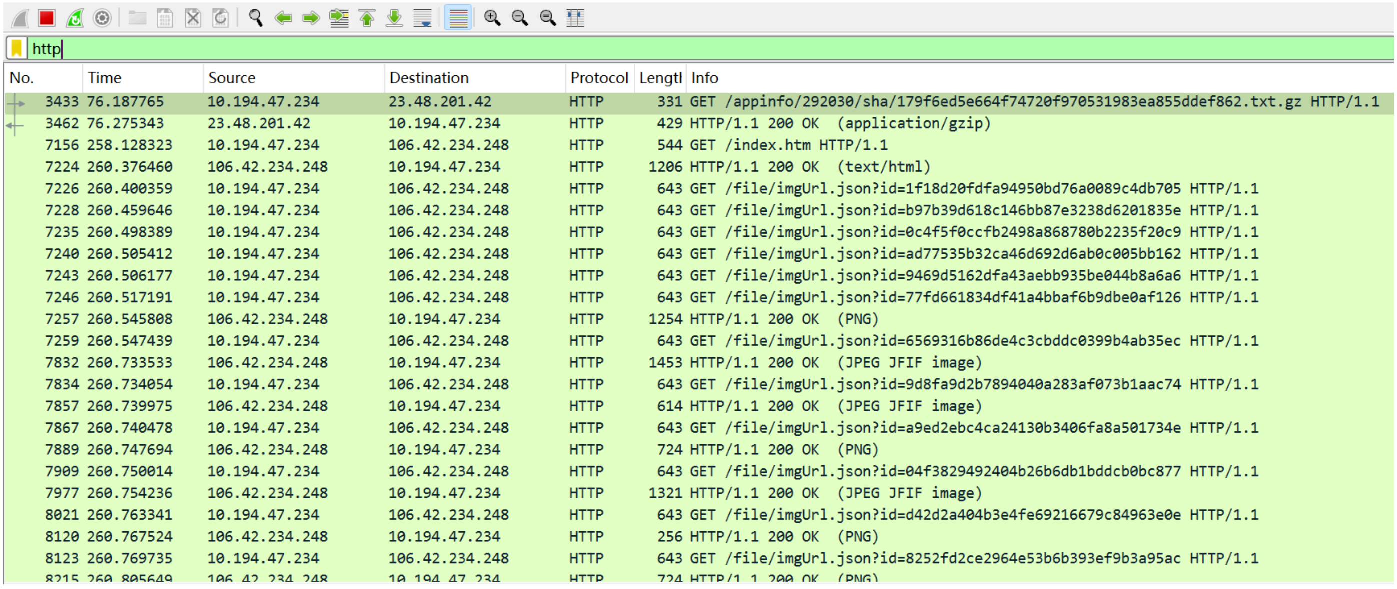
但是发现此时获得的包只有TCP/TLS类型，并不能够获取到传输过来的HTTP包，换言之并没有真实数据的数据包。

step 3: 对比HTTP与HTTPS，对发送来的HTTPS进行解密并获取数据包：

通过检验发现，WireShark需要提供一些密钥数据，才能够对HTTPS包进行解密。

此处，以一个HTTP网站为例（http://www.jiujiezixun.com/index.htm）：（这个网站是在百度上随便找的，现在绝大多数网站使用的都是https）

对该网站进行step 1/step 2的操作，结果如下：

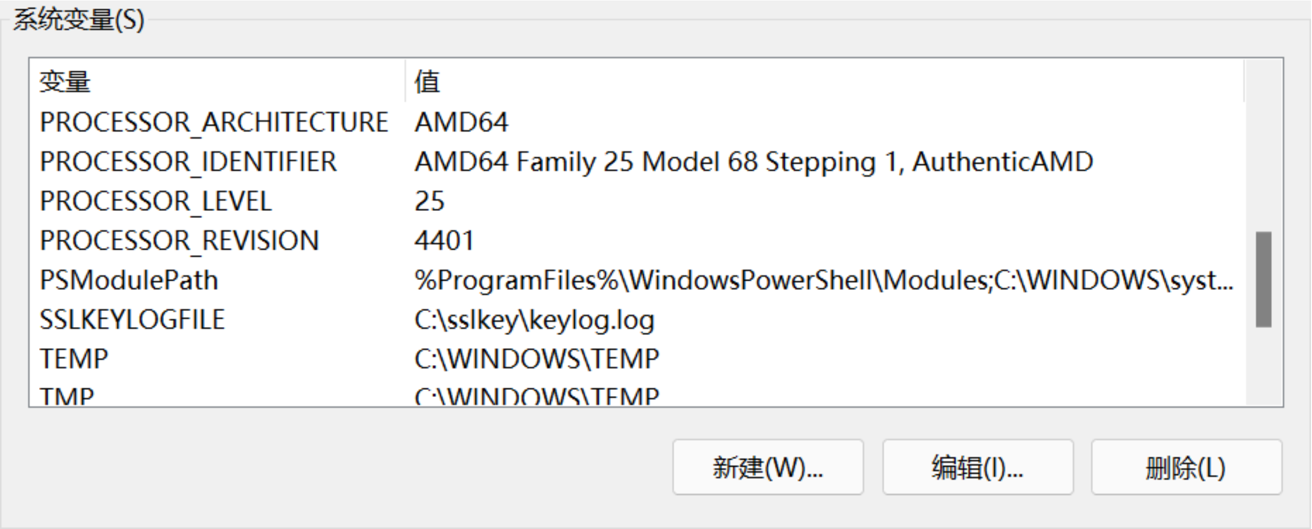


```
http
No.  Time        Source          Destination     Protocol Lengtl Info
3433 76.187765   10.194.47.234   23.48.201.42    HTTP      331 GET /appinfo/292030/sha/179f6ed5e664f74720f970531983ea855ddef862.txt.gz HTTP/1.1
3462 76.275343   23.48.201.42    10.194.47.234   HTTP      429 HTTP/1.1 200 OK  (application/gzip)
7156 258.128323  10.194.47.234   106.42.234.248  HTTP      544 GET /index.htm HTTP/1.1
7224 260.376460  106.42.234.248  10.194.47.234   HTTP     1206 HTTP/1.1 200 OK  (text/html)
7226 260.400359  10.194.47.234   106.42.234.248  HTTP      643 GET /file/imgUrl.json?id=1f18d20fdfa94950bd76a0089c4db705 HTTP/1.1
7228 260.459646  10.194.47.234   106.42.234.248  HTTP      643 GET /file/imgUrl.json?id=b97b39d618c146bb87e3238d6201835e HTTP/1.1
7235 260.498389  10.194.47.234   106.42.234.248  HTTP      643 GET /file/imgUrl.json?id=0c4f5f0ccfb2498a868780b2235f20c9 HTTP/1.1
7240 260.505412  10.194.47.234   106.42.234.248  HTTP      643 GET /file/imgUrl.json?id=ad77535b32ca46d692d6ab0c005bb162 HTTP/1.1
7243 260.506177  10.194.47.234   106.42.234.248  HTTP      643 GET /file/imgUrl.json?id=9469d5162dfa43aebb935be044b8a6a6 HTTP/1.1
7246 260.517191  10.194.47.234   106.42.234.248  HTTP      643 GET /file/imgUrl.json?id=77fd661834df41a4bbaf6b9dbe0af126 HTTP/1.1
7257 260.545808  106.42.234.248  10.194.47.234   HTTP     1254 HTTP/1.1 200 OK  (PNG)
7259 260.547439  10.194.47.234   106.42.234.248  HTTP      643 GET /file/imgUrl.json?id=6569316b86de4c3cbddc0399b4ab35ec HTTP/1.1
7832 260.733533  106.42.234.248  10.194.47.234   HTTP     1453 HTTP/1.1 200 OK  (JPEG JFIF image)
7834 260.734054  10.194.47.234   106.42.234.248  HTTP      643 GET /file/imgUrl.json?id=9d8fa9d2b7894040a283af073b1aac74 HTTP/1.1
7857 260.739975  106.42.234.248  10.194.47.234   HTTP      614 HTTP/1.1 200 OK  (JPEG JFIF image)
7867 260.740478  10.194.47.234   106.42.234.248  HTTP      643 GET /file/imgUrl.json?id=a9ed2ebc4ca24130b3406fa8a501734e HTTP/1.1
7889 260.747694  106.42.234.248  10.194.47.234   HTTP      724 HTTP/1.1 200 OK  (PNG)
7909 260.750014  10.194.47.234   106.42.234.248  HTTP      643 GET /file/imgUrl.json?id=04f3829492404b26b6db1bddcb0bc877 HTTP/1.1
7977 260.754236  106.42.234.248  10.194.47.234   HTTP     1321 HTTP/1.1 200 OK  (JPEG JFIF image)
8021 260.763341  10.194.47.234   106.42.234.248  HTTP      643 GET /file/imgUrl.json?id=d42d2a404b3e4fe69216679c84963e0e HTTP/1.1
8120 260.767524  106.42.234.248  10.194.47.234   HTTP      256 HTTP/1.1 200 OK  (PNG)
8123 260.769735  10.194.47.234   106.42.234.248  HTTP      643 GET /file/imgUrl.json?id=8252fd2ce2964e53b6b393ef9b3a95ac HTTP/1.1
8215 260.805649  106.42.234.248  10.194.47.234   HTTP      724 HTTP/1.1 200 OK  (PNG)
```

发现此时获取到了很多HTTP文件，包括一些图片与文字数据。

现在则需通过配置SSLKEYLOGFILE，人为记录浏览器交互时的解密过程，将这些信息传送给WireShark使其能够解析HTTPS包。

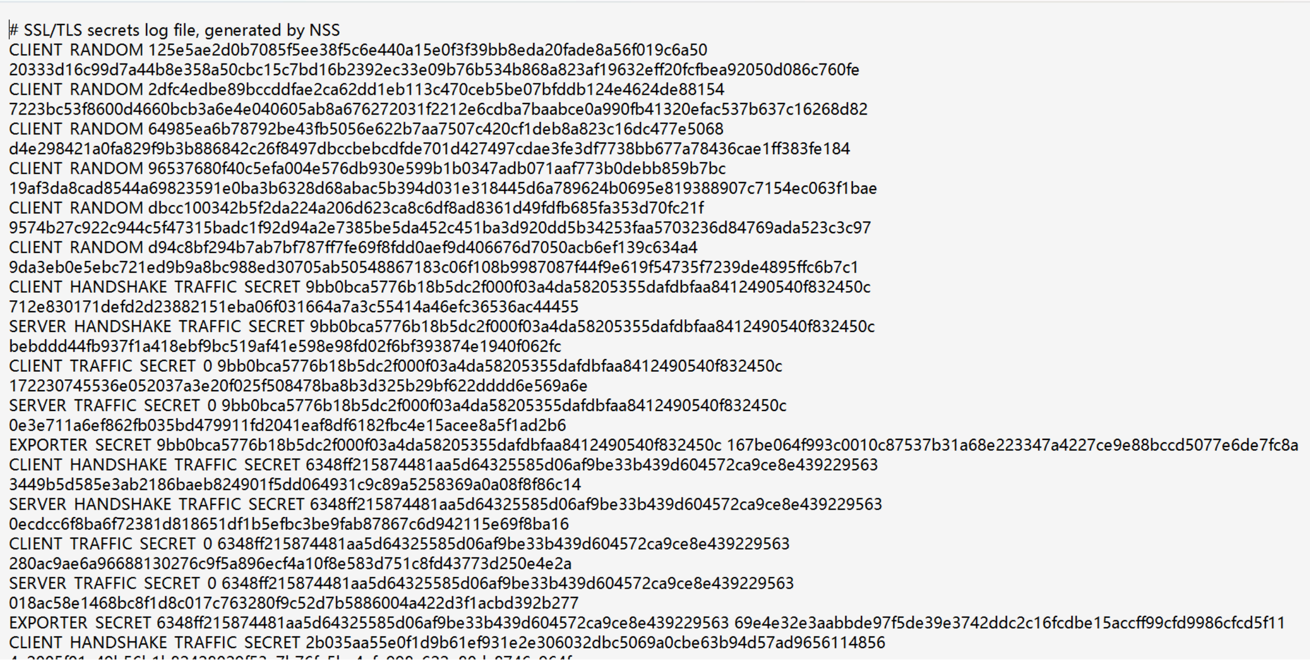
1. 在环境变量中设置：SSLKEYLOGFILE



系统变量(S)

变量	值
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	AMD64 Family 25 Model 68 Stepping 1, AuthenticAMD
PROCESSOR_LEVEL	25
PROCESSOR_REVISION	4401
PSModulePath	%ProgramFiles%\WindowsPowerShell\Modules;C:\WINDOWS\syst...
SSLKEYLOGFILE	C:\sslkey\keylog.log
TEMP	C:\WINDOWS\TEMP
TMP	C:\WINDOWS\TEMP

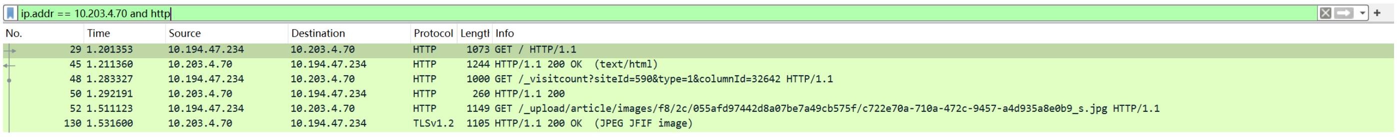
2. 重启浏览器，访问目标网址以记录相应的信息



```
# SSL/TLS secrets log file, generated by NSS
CLIENT_RANDOM 125e5ae2d0b7085f5ee38f5c6e440a15e0f3f39bb8eda20fade8a56f019c6a50
20333d16c99d7a44b8e358a50cbc15c7bd16b2392ec33e09b76b534b868a823af19632eff20fcfbea92050d086c760fe
CLIENT_RANDOM 2dfc4edbe89bccddfae2ca62dd1eb113c470ceb5be07bfddb124e4624de88154
7223bc53f8600d4660bcb3a6e4e040605ab8a676272031f2212e6cdba7baabce0a990fb41320efac537b637c16268d82
CLIENT_RANDOM 64985ea6b78792be43fb5056e622b7aa7507c420cf1deb8a823c16dc477e5068
d4e298421a0fa829f9b3b886842c26f8497dbccbebcdfde701d427497cdae3fe3df7738bb677a78436cae1ff383fe184
CLIENT_RANDOM 96537680f40c5efa004e576db930e599b1b0347adb071aaf773b0debb859b7bc
19af3da8cad8544a69823591e0ba3b6328d68abac5b394d031e318445d6a789624b0695e819388907c7154ec063f1bae
CLIENT_RANDOM dbcc100342b5f2da224a206d623ca8c6df8ad8361d49fdfb685fa353d70fc21f
9574b27c922c944c5f47315badc1f92d94a2e7385be5da452c451ba3d920dd5b34253faa5703236d84769ada523c3c97
CLIENT_RANDOM d94c8bf294b7ab7bf787ff7fe69f8fdd0aef9d406676d7050acb6ef139c634a4
9da3eb0e5ebc721ed9b9a8bc988ed30705ab50548867183c06f108b9987087f44f9e619f54735f7239de4895ffc6b7c1
CLIENT_HANDSHAKE_TRAFFIC_SECRET 9bb0bca5776b18b5dc2f000f03a4da58205355dafdbfaa8412490540f832450c
712e830171defd2d23882151eba06f031664a7a3c55414a46efc36536ac44455
SERVER_HANDSHAKE_TRAFFIC_SECRET 9bb0bca5776b18b5dc2f000f03a4da58205355dafdbfaa8412490540f832450c
bebddd44fb937f1a418ebf9bc519af41e598e98fd02f6bf393874e1940f062fc
CLIENT_TRAFFIC_SECRET_0 9bb0bca5776b18b5dc2f000f03a4da58205355dafdbfaa8412490540f832450c
172230745536e052037a3e20f025f508478ba8b3d325b29bf622dddd6e569a6e
SERVER_TRAFFIC_SECRET_0 9bb0bca5776b18b5dc2f000f03a4da58205355dafdbfaa8412490540f832450c
0e3e711a6ef862fb035bd479911fd2041eaf8df6182fbc4e15acee8a5f1ad2b6
EXPORTER_SECRET 9bb0bca5776b18b5dc2f000f03a4da58205355dafdbfaa8412490540f832450c 167be064f993c0010c87537b31a68e223347a4227ce9e88bccd5077e6de7fc8a
CLIENT_HANDSHAKE_TRAFFIC_SECRET 6348ff215874481aa5d64325585d06af9be33b439d604572ca9ce8e439229563
3449b5d585e3ab2186baeb824901f5dd064931c9c89a5258369a0a08f8f86c14
SERVER_HANDSHAKE_TRAFFIC_SECRET 6348ff215874481aa5d64325585d06af9be33b439d604572ca9ce8e439229563
0ecdcc6f8ba6f72381d818651df1b5efbc3be9fab87867c6d942115e69f8ba16
CLIENT_TRAFFIC_SECRET_0 6348ff215874481aa5d64325585d06af9be33b439d604572ca9ce8e439229563
280ac9ae6a96688130276c9f5a896ecf4a10f8e583d751c8fd43773d250e4e2a
SERVER_TRAFFIC_SECRET_0 6348ff215874481aa5d64325585d06af9be33b439d604572ca9ce8e439229563
018ac58e1468bc8f1d8c017c763280f9c52d7b5886004a422d3f1acbd392b277
EXPORTER_SECRET 6348ff215874481aa5d64325585d06af9be33b439d604572ca9ce8e439229563 69e4e32e3aabbde97f5de39e3742ddc2c16fcdbe15accff99cfd9986cfcd5f11
CLIENT_HANDSHAKE_TRAFFIC_SECRET 2b035aa55e0f1d9b61ef931e2e306032dbc5069a0cbe63b94d57ad9656114856
```

3. 启动WireShark，在protocol-TSL中加入这个文件，并再次抓包：

效果如下：（对HTTP）



ip.addr == 10.203.4.70 and http

```
No.     Time       Source          Destination     Protocol Lengtl Info
     29 1.201353   10.194.47.234   10.203.4.70     HTTP     1073 GET / HTTP/1.1
     45 1.211360   10.203.4.70     10.194.47.234   HTTP     1244 HTTP/1.1 200 OK  (text/html)
     48 1.283327   10.194.47.234   10.203.4.70     HTTP     1000 GET /_visitcount?siteId=590&type=1&columnId=32642 HTTP/1.1
     50 1.292191   10.203.4.70     10.194.47.234   HTTP      260 HTTP/1.1 200
     52 1.511123   10.194.47.234   10.203.4.70     HTTP     1149 GET /_upload/article/images/f8/2c/055afd97442d8a07be7a49cb575f/c722e70a-710a-472c-9457-a4d935a8e0b9_s.jpg HTTP/1.1
    130 1.531600   10.203.4.70     10.194.47.234   TLSv1.2  1105 HTTP/1.1 200 OK  (JPEG JFIF image)
```

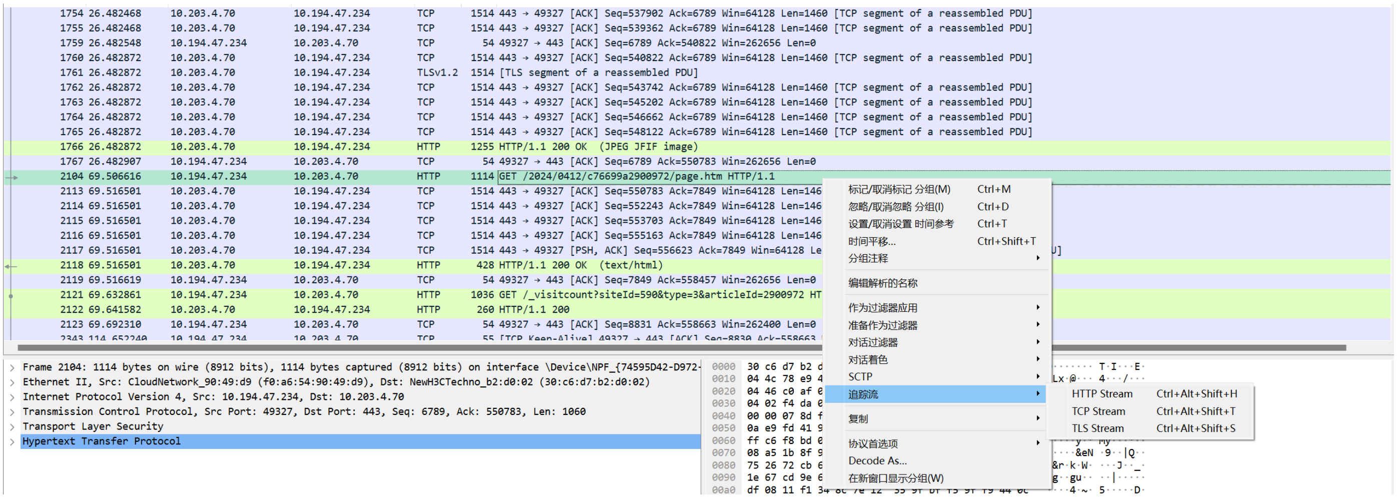
现在可以清楚地看到GET请求的传输与响应过程，以及传输的图片数据包等等信息了。

step 4: 对数据包进行分析：

HTTP网页分析：

我们在网站上随机打开一篇文章（https://www.zju.edu.cn/2024/0412/c76699a2900972/page.htm）

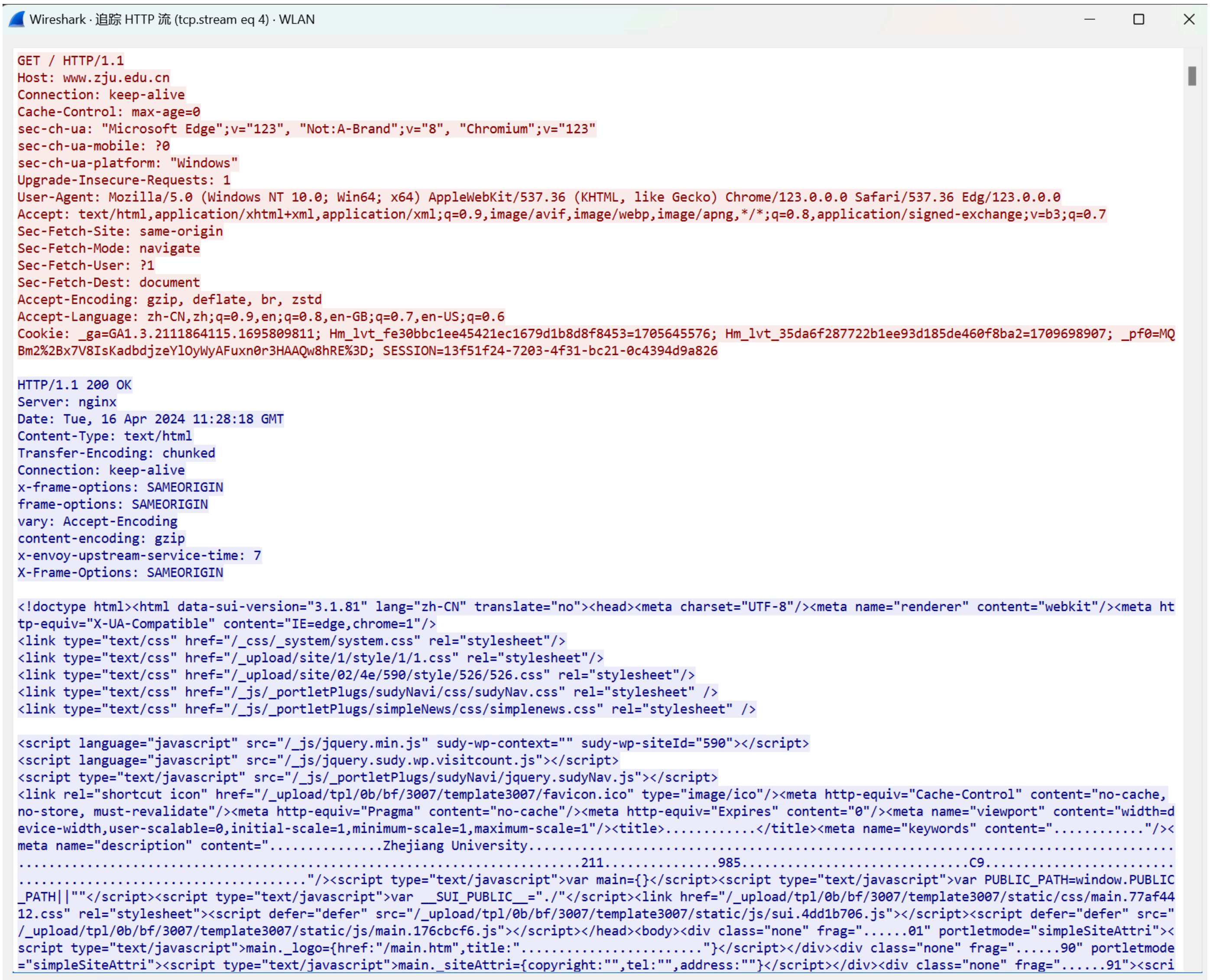
然后对对应的GET请求进行追踪：



```
1754 26.482468   10.203.4.70     10.194.47.234   TCP      1514 443 → 49327 [ACK] Seq=537902 Ack=6789 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
1755 26.482468   10.203.4.70     10.194.47.234   TCP      1514 443 → 49327 [ACK] Seq=539362 Ack=6789 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
1759 26.482548   10.194.47.234   10.203.4.70     TCP        54 49327 → 443 [ACK] Seq=6789 Ack=540822 Win=262656 Len=0
1760 26.482872   10.203.4.70     10.194.47.234   TCP      1514 443 → 49327 [ACK] Seq=540822 Ack=6789 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
1761 26.482872   10.203.4.70     10.194.47.234   TLSv1.2  1514 [TLS segment of a reassembled PDU]
1762 26.482872   10.203.4.70     10.194.47.234   TCP      1514 443 → 49327 [ACK] Seq=543742 Ack=6789 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
1763 26.482872   10.203.4.70     10.194.47.234   TCP      1514 443 → 49327 [ACK] Seq=545202 Ack=6789 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
1764 26.482872   10.203.4.70     10.194.47.234   TCP      1514 443 → 49327 [ACK] Seq=546662 Ack=6789 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
1765 26.482872   10.203.4.70     10.194.47.234   TCP      1514 443 → 49327 [ACK] Seq=548122 Ack=6789 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
1766 26.482872   10.203.4.70     10.194.47.234   HTTP     1255 HTTP/1.1 200 OK  (JPEG JFIF image)
1767 26.482907   10.194.47.234   10.203.4.70     TCP        54 49327 → 443 [ACK] Seq=6789 Ack=550783 Win=262656 Len=0
2104 69.506616   10.194.47.234   10.203.4.70     HTTP     1114 GET /2024/0412/c76699a2900972/page.htm HTTP/1.1
2113 69.516501   10.203.4.70     10.194.47.234   TCP      1514 443 → 49327 [ACK] Seq=550783 Ack=7849 Win=64128 Len=146
2114 69.516501   10.203.4.70     10.194.47.234   TCP      1514 443 → 49327 [ACK] Seq=552243 Ack=7849 Win=64128 Len=146
2115 69.516501   10.203.4.70     10.194.47.234   TCP      1514 443 → 49327 [ACK] Seq=553703 Ack=7849 Win=64128 Len=146
2116 69.516501   10.203.4.70     10.194.47.234   TCP      1514 443 → 49327 [ACK] Seq=555163 Ack=7849 Win=64128 Len=146
2117 69.516501   10.203.4.70     10.194.47.234   TCP      1514 443 → 49327 [PSH, ACK] Seq=556623 Ack=7849 Win=64128 Le                    U]
2118 69.516501   10.203.4.70     10.194.47.234   HTTP      428 HTTP/1.1 200 OK  (text/html)
2119 69.516619   10.194.47.234   10.203.4.70     TCP        54 49327 → 443 [ACK] Seq=7849 Ack=558457 Win=262656 Len=0
2121 69.632861   10.194.47.234   10.203.4.70     HTTP     1036 GET /_visitcount?siteId=590&type=3&articleId=2900972 HT
2122 69.641582   10.203.4.70     10.194.47.234   HTTP      260 HTTP/1.1 200
2123 69.692310   10.194.47.234   10.203.4.70     TCP        54 49327 → 443 [ACK] Seq=8831 Ack=558663 Win=262400 Len=0
2343 114.652240  10.194.47.234   10.203.4.70     TCP        55 [TCP Keep-Alive] 49327 → 443 [ACK] Seq=8830 Ack=558663

> Frame 2104: 1114 bytes on wire (8912 bits), 1114 bytes captured (8912 bits) on interface \Device\NPF_{74595D42-D972-
> Ethernet II, Src: CloudNetwork_90:49:d9 (f0:a6:54:90:49:d9), Dst: NewH3CTechno_b2:d0:02 (30:c6:d7:b2:d0:02)
> Internet Protocol Version 4, Src: 10.194.47.234, Dst: 10.203.4.70
> Transmission Control Protocol, Src Port: 49327, Dst Port: 443, Seq: 6789, Ack: 550783, Len: 1060
> Transport Layer Security
> Hypertext Transfer Protocol

0000  30 c6 d7 b2 d                          ········ T·I···E·
0010  04 4c 78 e9 4                          Lx·@··· 4···/···
0020  04 46 c0 af 0
0030  04 02 f4 da 0
0040  00 00 07 8d f
0050  0a e9 fd 41 9
0060  ff c6 f8 bd 0                          ···y·· My······
0070  08 a5 1b 8f 9                          ····&eN ·9··|Q··
0080  75 26 72 cb 6                          &r·k·W· ···J··_·
0090  1e 67 cd 9e 6                          g··gu·· ··|·····
00a0  df 08 11 f1 34 8c 7e 12  35 9f bf f5 9f f9 44 0c   ····4·~· 5·····D·
```

标记/取消标记 分组(M)　Ctrl+M
忽略/取消忽略 分组(I)　Ctrl+D
设置/取消设置 时间参考　Ctrl+T
时间平移...　Ctrl+Shift+T
分组注释
编辑解析的名称
作为过滤器应用
准备作为过滤器
对话过滤器
对话着色
SCTP
追踪流
复制
协议首选项
Decode As...
在新窗口显示分组(W)
HTTP Stream　Ctrl+Alt+Shift+H
TCP Stream　Ctrl+Alt+Shift+T
TLS Stream　Ctrl+Alt+Shift+S

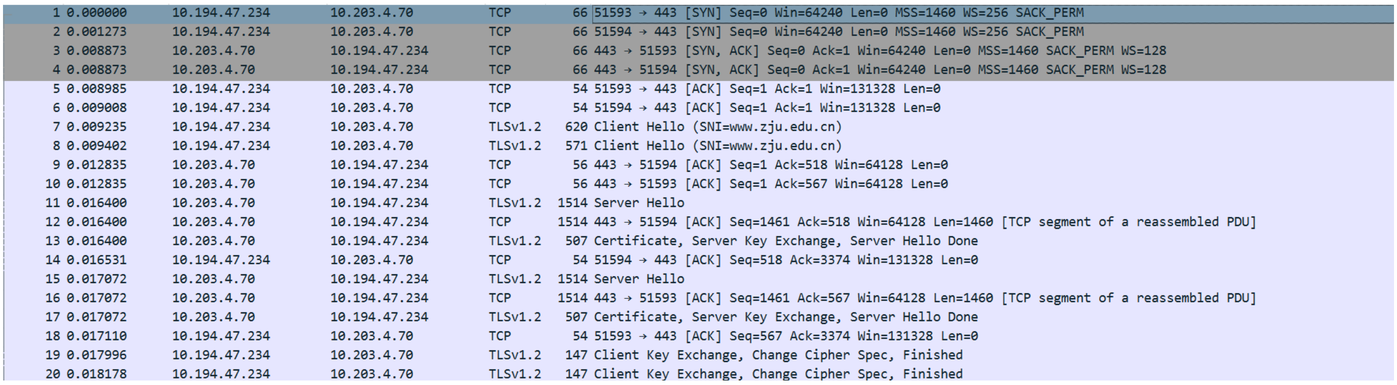
HTTP追踪：可以看到GET请求与响应的具体信息（包括整个页面的html文件）：



```
GET / HTTP/1.1
Host: www.zju.edu.cn
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Microsoft Edge";v="123", "Not:A-Brand";v="8", "Chromium";v="123"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 Edg/123.0.0.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Cookie: _ga=GA1.3.2111864115.1695809811; Hm_lvt_fe30bbc1ee45421ec1679d1b8d8f8453=1705645576; Hm_lvt_35da6f287722b1ee93d185de460f8ba2=1709698907; _pf0=MQ
Bm2%2Bx7V8IsKadbdjzeYlOyWyAFuxn0r3HAAQw8hRE%3D; SESSION=13f51f24-7203-4f31-bc21-0c4394d9a826

HTTP/1.1 200 OK
Server: nginx
Date: Tue, 16 Apr 2024 11:28:18 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
x-frame-options: SAMEORIGIN
frame-options: SAMEORIGIN
vary: Accept-Encoding
content-encoding: gzip
x-envoy-upstream-service-time: 7
X-Frame-Options: SAMEORIGIN

<!doctype html><html data-sui-version="3.1.81" lang="zh-CN" translate="no"><head><meta charset="UTF-8"/><meta name="renderer" content="webkit"/><meta ht
tp-equiv="X-UA-Compatible" content="IE=edge,chrome=1"/>
<link type="text/css" href="/_css/_system/system.css" rel="stylesheet"/>
<link type="text/css" href="/_upload/site/1/style/1/1.css" rel="stylesheet"/>
<link type="text/css" href="/_upload/site/02/4e/590/style/526/526.css" rel="stylesheet"/>
<link type="text/css" href="/_js/_portletPlugs/sudyNavi/css/sudyNav.css" rel="stylesheet" />
<link type="text/css" href="/_js/_portletPlugs/simpleNews/css/simplenews.css" rel="stylesheet" />

<script language="javascript" src="/_js/jquery.min.js" sudy-wp-context="" sudy-wp-siteId="590"></script>
<script language="javascript" src="/_js/jquery.sudy.wp.visitcount.js"></script>
<script type="text/javascript" src="/_js/_portletPlugs/sudyNavi/jquery.sudyNav.js"></script>
<link rel="shortcut icon" href="/_upload/tpl/0b/bf/3007/template3007/favicon.ico" type="image/ico"/><meta http-equiv="Cache-Control" content="no-cache,
no-store, must-revalidate"/><meta http-equiv="Pragma" content="no-cache"/><meta http-equiv="Expires" content="0"/><meta name="viewport" content="width=d
evice-width,user-scalable=0,initial-scale=1,minimum-scale=1,maximum-scale=1"/><title>...........</title><meta name="keywords" content="..........."/><
meta name="description" content="..............Zhejiang University.................................................................................
.......................................................................211..............985..............................C9.........................
....................................."/><script type="text/javascript">var main={}</script><script type="text/javascript">var PUBLIC_PATH=window.PUBLIC
_PATH||""</script><script type="text/javascript">var __SUI_PUBLIC__="./"</script><link href="/_upload/tpl/0b/bf/3007/template3007/static/css/main.77af44
12.css" rel="stylesheet"><script defer="defer" src="/_upload/tpl/0b/bf/3007/template3007/static/js/sui.4dd1b706.js"></script><script defer="defer" src="
/_upload/tpl/0b/bf/3007/template3007/static/js/main.176cbcf6.js"></script></head><body><div class="none" frag="......01" portletmode="simpleSiteAttri"><
script type="text/javascript">main._logo={href:"/main.htm",title:"......................."}</script></div><div class="none" frag="......90" portletmode
="simpleSiteAttri"><script type="text/javascript">main._siteAttri={copyright:"",tel:"",address:""}</script></div><div class="none" frag="......91"><scri
```

TCP协议分析：

我们看在主机发送GET请求前，双方互相发送的20条TCP/TLS请求：



```
 1 0.000000   10.194.47.234   10.203.4.70     TCP       66 51593 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
 2 0.001273   10.194.47.234   10.203.4.70     TCP       66 51594 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
 3 0.008873   10.203.4.70     10.194.47.234   TCP       66 443 → 51593 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
 4 0.008873   10.203.4.70     10.194.47.234   TCP       66 443 → 51594 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
 5 0.008985   10.194.47.234   10.203.4.70     TCP       54 51593 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
 6 0.009008   10.194.47.234   10.203.4.70     TCP       54 51594 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
 7 0.009235   10.194.47.234   10.203.4.70     TLSv1.2  620 Client Hello (SNI=www.zju.edu.cn)
 8 0.009402   10.194.47.234   10.203.4.70     TLSv1.2  571 Client Hello (SNI=www.zju.edu.cn)
 9 0.012835   10.203.4.70     10.194.47.234   TCP       56 443 → 51594 [ACK] Seq=1 Ack=518 Win=64128 Len=0
10 0.012835   10.203.4.70     10.194.47.234   TCP       56 443 → 51593 [ACK] Seq=1 Ack=567 Win=64128 Len=0
11 0.016400   10.203.4.70     10.194.47.234   TLSv1.2 1514 Server Hello
12 0.016400   10.203.4.70     10.194.47.234   TCP     1514 443 → 51594 [ACK] Seq=1461 Ack=518 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
13 0.016400   10.203.4.70     10.194.47.234   TLSv1.2  507 Certificate, Server Key Exchange, Server Hello Done
14 0.016531   10.194.47.234   10.203.4.70     TCP       54 51594 → 443 [ACK] Seq=518 Ack=3374 Win=131328 Len=0
15 0.017072   10.203.4.70     10.194.47.234   TLSv1.2 1514 Server Hello
16 0.017072   10.203.4.70     10.194.47.234   TCP     1514 443 → 51593 [ACK] Seq=1461 Ack=567 Win=64128 Len=1460 [TCP segment of a reassembled PDU]
17 0.017072   10.203.4.70     10.194.47.234   TLSv1.2  507 Certificate, Server Key Exchange, Server Hello Done
18 0.017110   10.194.47.234   10.203.4.70     TCP       54 51593 → 443 [ACK] Seq=567 Ack=3374 Win=131328 Len=0
19 0.017996   10.194.47.234   10.203.4.70     TLSv1.2  147 Client Key Exchange, Change Cipher Spec, Finished
20 0.018178   10.194.47.234   10.203.4.70     TLSv1.2  147 Client Key Exchange, Change Cipher Spec, Finished
```

很明显地看到，主机端口51593, 51594分别与目标端口443进行了三次握手（即1-6条）。

但具体为何主机端口会有两个，并不太清楚，就搜索到的信息而言可能是网络延时或是出现了并行访问。

之后的一些条目即规范化的“互相打招呼”，并且确认数据包传输的加密方式（这也是https的特征）

在这之后，二者就开始传输数据了。