Task 5:

For HTTPS based website access

7. What is the name of website?

- Update.googleapis.com

8. Find the packet that contains the ClientHello message for the website you are accessing.

- 85

9. List all the TLS extensions included in the ClientHello.

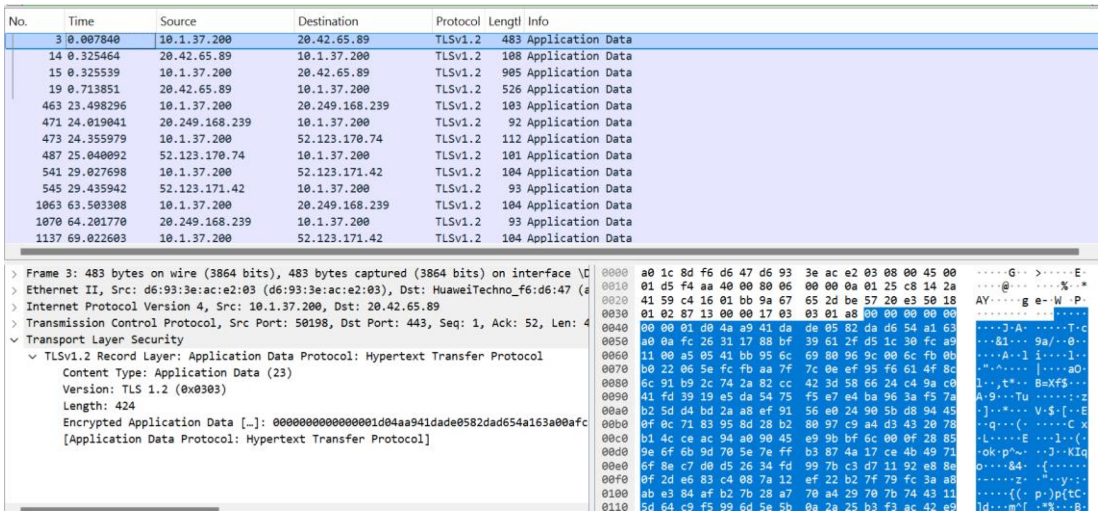
- Length
- Reserved
- extended_master_secret
- ec_point_formats
- server_name
- signature_algorithm
- encrypted_client_hello
- compress_certificate
- psk_key_exchange_modes
- signed_certificate_timestamp
- session_ticket
- renegotiation_info
- application_layer_protection_renegotoation
- supported_groups
- unknown type 17613
- supported_versions
- key_share
- status_request

- reserved

10. Identify the ServerHello message. What cipher suite is chosen by the server?

- Server hello packet:1198
- Cipher suite: TLS_AES_256_GCM_SHA384

12. After the TLS handshake, identify the first encrypted application data packet. Why can't you directly see the HTTP headers in this packet?



```
No.   Time       Source          Destination     Protocol Lengtl Info
    3 0.007840   10.1.37.200     20.42.65.89     TLSv1.2  483 Application Data
   14 0.325464   20.42.65.89     10.1.37.200     TLSv1.2  108 Application Data
   15 0.325539   10.1.37.200     20.42.65.89     TLSv1.2  905 Application Data
   19 0.713851   20.42.65.89     10.1.37.200     TLSv1.2  526 Application Data
  463 23.498296  10.1.37.200     20.249.168.239  TLSv1.2  103 Application Data
  471 24.019041  20.249.168.239  10.1.37.200     TLSv1.2   92 Application Data
  473 24.355979  10.1.37.200     52.123.170.74   TLSv1.2  112 Application Data
  487 25.040092  52.123.170.74   10.1.37.200     TLSv1.2  101 Application Data
  541 29.027698  10.1.37.200     52.123.171.42   TLSv1.2  104 Application Data
  545 29.435942  52.123.171.42   10.1.37.200     TLSv1.2   93 Application Data
 1063 63.503308  10.1.37.200     20.249.168.239  TLSv1.2  104 Application Data
 1070 64.201770  20.249.168.239  10.1.37.200     TLSv1.2   93 Application Data
 1137 69.022603  10.1.37.200     52.123.171.42   TLSv1.2  104 Application Data
```

```
> Frame 3: 483 bytes on wire (3864 bits), 483 bytes captured (3864 bits) on interface \D
> Ethernet II, Src: d6:93:3e:ac:e2:03 (d6:93:3e:ac:e2:03), Dst: HuaweiTechno_f6:d6:47 (a
> Internet Protocol Version 4, Src: 10.1.37.200, Dst: 20.42.65.89
> Transmission Control Protocol, Src Port: 50198, Dst Port: 443, Seq: 1, Ack: 52, Len: 4
v Transport Layer Security
  v TLSv1.2 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 424
      Encrypted Application Data […]: 0000000000000001d04aa941dade0582dad654a163a00afc
      [Application Data Protocol: Hypertext Transfer Protocol]
```

```
0000  a0 1c 8d f6 d6 47 d6 93  3e ac e2 03 08 00 45 00   ·····G·· >·····E·
0010  01 d5 f4 aa 40 00 80 06  00 00 0a 01 25 c8 14 2a   ····@··· ····%··*
0020  41 59 c4 16 01 bb 9a 67  65 2d be 57 20 e3 50 18   AY·····g e-·W ·P·
0030  01 02 87 13 00 00 17 03  03 01 a8 00 00 00 00 00   ········ ········
0040  00 00 01 d0 4a a9 41 da  de 05 82 da d6 54 a1 63   ····J·A· ·····T·c
0050  a0 0a fc 26 31 17 88 bf  39 61 2f d5 1c 30 fc a9   ···&1··· 9a/··0··
0060  11 00 a5 05 41 bb 95 6c  69 80 96 9c 00 6c fb 0b   ····A··l i····l··
0070  b0 22 06 5e fc fb aa 7f  7c 0e ef 95 f6 61 4f 8c   ·"·^···· |····aO·
0080  6c 91 b9 2c 74 2a 82 cc  42 3d 58 66 24 c4 9a c0   l··,t*·· B=Xf$···
0090  41 fd 39 19 e5 da 54 75  f5 e7 e4 ba 96 3a f5 7a   A·9···Tu ·····:·z
00a0  b2 5d d4 bd 2a a8 ef 91  56 e0 24 90 5b d8 94 45   ·]··*··· V·$·[··E
00b0  0f 0c 71 83 95 8d 28 b2  80 97 c9 a4 d3 43 20 78   ··q···(· ·····C x
00c0  b1 4c ce ac 94 a0 90 45  e9 9b bf 6c 00 0f 28 85   ·L·····E ···l··(·
00d0  9e 6f 6b 9d 70 5e 7e ff  b3 87 4a 17 ce 4b 49 71   ·ok·p^~· ··J··KIq
00e0  6f 8e c7 d0 d5 26 34 fd  99 7b c3 d7 11 92 e8 8e   o····&4· ·{······
00f0  0f 2d e6 83 c4 08 7a 12  ef 22 b2 7f 79 fc 3a a8   ·-····z· ·"··y·:·
0100  ab e3 84 af b2 7b 28 a7  70 a4 29 70 7b 74 43 11   ·····{(· p·)p{tC·
0110  5d 64 c9 f5 99 6d 5e 5b  0a 2a 25 b3 f3 ac 42 e9   ]d···m^[ ·*%···B·
```

Why can't directly see HTTP header?

- In the packet details pane, we will see something like "Encrypted Application Data" instead of readable text.
- That's because HTTPS encrypts all HTTP data (headers + body) using the session keys negotiated during the TLS handshake.
- Only the browser (or client) can decrypt it because it has the session key.
- Wireshark does not have the keys by default, so it cannot show the HTTP headers or content.