

PLAGIARISM STATEMENT

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.

Name: Akash Tadwai

Date: 13/02/2022

Signature: Akash Tadwai

Assignment 4: Decrypting TLS and HTTP(S) using Wireshark++

Individual Assignment

PART-A: Decrypt TLS handshake and HTTPS messages between your browser and the web server of Bank X

- Set SSLKEYLOGFILE environment variable in your host OS by following the instructions given in References 1-3 and then launch Chrome browser with a blank tab (for surfing the website of bank x) and wireshark (for capturing all the messages exchanged between your browser and bank website/DNS resolvers/CAs).
- Start packet capture in wireshark
- Type in the hostname of the bank X in the address bar of the browser. Let N be your (RollNo % 4 + 1). If N==1, X=ICICI. If N==2, X=HDFC. If N==3, X=SBI. If N==4, X=Bank of America
 - a. Click on the link that takes you to the online net banking page of the bank.
 - b. Enter some arbitrary values against Username and Password so that the login process fails:-)**
 - c. Stop the packet capture in wireshark and save it as <RollNo-BankName.pcapng>. And also close your browser tab.
 - d. Follow the steps in References 1-3 to specify the complete path of SSLkeyLog file in your computer for wireshark to decrypt TLS and HTTPS messages present in <RollNo-BankName.pcapng>.
 - i. Note that <RollNo-BankName.pcapng> should only contain the messages exchanged between your browser and bank website/DNS resolvers/CAs

including sub-domains/redirections and 3rd party tracking/resource fetching sessions triggered by your visit to the bank's site. So, close all background Apps running on your computer to avoid capturing their messages in your wireshark capture or use appropriate display/capture filters to exclude other messages in your packet trace. **This is your Deliverable-1.**

- ii. Before providing session keys which are present inside SSLkeyLog file to wireshark, you should find that all of the application traffic and most of the handshake (HTTPS) is encrypted and shown as TLS traffic with encrypted application data. Get a snapshot of it. **This is your Deliverable-2.**
- iii. After providing session keys in SSLkeyLog file to wireshark, you should find that all of the application traffic along with handshake traffic is decrypted and shown as HTTP traffic along with TLS handshake messages in plain-text. Get a snapshot of it. **This is Deliverable-3.**

Answer the following queries by referring to the (decrypted) messages in your browsing session with the banking site using wireshark GUI. It is important to keep in mind that an Ethernet frame may contain either a partial, one or more TLS records. This is very different from HTTP(S), for which each Ethernet frame contains either one complete HTTP message or a portion of a HTTP message.

Whenever possible, when answering the questions given below, you should produce a screenshot of the packet(s) within the trace that you used to answer the question asked. Highlight portions of the snapshot to explain your answer. To print a packet in wireshark GUI, use *File->Print Option*, choose *Selected packet only*, choose *Packet summary line*, and select the minimum amount of packet detail that you need to answer the question.

1. What browser did you use, what's the version number?

Google Chrome 97.0.4692.99

2. List out various protocols that you noticed in the column named "Protocol" in the wireshark GUI from the time you keyed in the hostname of the bank in the browser till you start viewing application data. For each such protocol, mention its purpose in brief.

ARP: It maps dynamic IP addresses to the MAC address of an interface

DNS: It resolves hostnames to ip addresses on the internet

MDNS: It is a protocol aimed at helping for name resolution in smaller networks

STUN: It is a client side protocol used by VoIP utilities for communication between the machines hidden behind NAT gateways.

TCP: It is a connection based reliable transport protocol to reliably share data between two applications.

UDP - It is a connectionless datagram based transport protocol which focuses on lower response time rather than reliability.

HTTP2: - It provides a way for users to interact with web resources such as HTML files by transmitting hypertext messages between clients and servers

TLS: The Transport Layer Security is a security protocol that provides confidentiality and data integrity for Internet communications. Implementing TLS is standard practice for building secure web apps.

3. Each of the TLS records begins with the same three fields (with possibly different values). One of these fields is “content type” and has a length of one byte. List all three fields and their lengths for the first 10 records in the trace.

4253	24.6630671...	akash.iith.ac.in	wwwui.ecglb.bac...	TCP	66	39008 → 443 [ACK] Seq=8081 Ack=6213 Win=63712 Len=0 TSval=1297218006
412	2.188457834	akash.iith.ac.in	cloudsearch.goo...	TLSv1.2	583	Client Hello
424	2.303112083	cloudsearch.goo...	akash.iith.ac.in	TLSv1.2	1484	[TCP Previous segment not captured] , Ignored Unknown Record
432	2.333310970	akash.iith.ac.in	cloudsearch.goo...	TLSv1.2	130	Change Cipher Spec, Application Data
433	2.333658632	akash.iith.ac.in	cloudsearch.goo...	TLSv1.2	158	Application Data
434	2.333938793	akash.iith.ac.in	cloudsearch.goo...	TLSv1.2	613	Application Data
435	2.333992107	akash.iith.ac.in	cloudsearch.goo...	TLSv1.2	301	Application Data
440	2.353792550	akash.iith.ac.in	cloudsearch.goo...	TLSv1.2	97	Application Data
457	2.412077610	akash.iith.ac.in	cloudsearch.goo...	TLSv1.2	105	Application Data
962	6.607142835	akash.iith.ac.in	wwwui.ecglb.bac...	TLSv1.2	583	Client Hello
971	6.795112380	akash.iith.ac.in	wwwui.ecglb.bac...	TLSv1.2	583	Client Hello
984	6.929415355	wwwui.ecglb.bac...	akash.iith.ac.in	TLSv1.2	1514	Server Hello
990	6.930361775	wwwui.ecglb.bac...	akash.iith.ac.in	TLSv1.2	627	Certificate, Server Key Exchange, Server Hello Done
993	6.937852084	akash.iith.ac.in	wwwui.ecglb.bac...	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
1081	7.117891818	wwwui.ecglb.bac...	akash.iith.ac.in	TLSv1.2	2962	Server Hello
1083	7.118946715	wwwui.ecglb.bac...	akash.iith.ac.in	TLSv1.2	2075	Certificate, Server Key Exchange, Server Hello Done
1085	7.120404047	akash.iith.ac.in	wwwui.ecglb.bac...	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
1126	7.258321938	wwwui.ecglb.bac...	akash.iith.ac.in	TLSv1.2	72	Change Cipher Spec
1127	7.258323566	wwwui.ecglb.bac...	akash.iith.ac.in	TLSv1.2	111	Encrypted Handshake Message
1133	7.261654573	akash.iith.ac.in	wwwui.ecglb.bac...	TLSv1.2	296	Application Data
1207	7.439772275	wwwui.ecglb.bac...	akash.iith.ac.in	TLSv1.2	72	Change Cipher Spec

Urgent pointer: 0

Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

SEQ/ACK analysis

Timestamps

TCP payload (517 bytes)

Secure Sockets Layer

- TLSv1.2 Record Layer: Handshake Protocol: Client Hello
 - Content Type: Handshake (22)
 - Version: TLS 1.0 (0x0301)
 - Length: 512
 - Handshake Protocol: Client Hello

- Content Type: Handshake (22) (1byte)
Version: TLS 1.0 (0x0301) (2bytes)
Length: 512 (2bytes)
- Content Type: Change Cipher Spec (20) (1byte)
Version: TLS 1.2 (0x0303) (2bytes)
Length: 1 (2bytes)
- Content Type: Application Data (23) (1byte)
Version: TLS 1.2 (0x0303) (2bytes)
Length: 4541 (2bytes)
- Content Type: Application Data (23) (1byte)
Version: TLS 1.2 (0x0303) (2bytes)

Length: 4541 **(2bytes)**

5. Content Type: Application Data (23) **(1byte)**

Version: TLS 1.2 (0x0303) **(2bytes)**

Length: 53 **(2bytes)**

6. Content Type: Application Data (23) **(1byte)**

Version: TLS 1.2 (0x0303) **(2bytes)**

Length: 87 **(2bytes)**

7. Content Type: Application Data (23) **(1byte)**

Version: TLS 1.2 (0x0303) **(2bytes)**

Length: 309 **(2bytes)**

8. Content Type: Application Data (23) **(1byte)**

Version: TLS 1.2 (0x0303) **(2bytes)**

Length: 541 **(2bytes)**

9. Content Type: Handshake (22) **(1byte)**

Version: TLS 1.0 (0x0301) **(2bytes)**

Length: 512 **(2bytes)**

10. Content Type: Handshake (22) **(1byte)**

Version: TLS 1.0 (0x0301) **(2bytes)**

Length: 512 **(2bytes)**

4. **Cipher Suites in ClientHello Record: Look at the first two and the last cipher suites offered by the client and compare them. What cipher suite the server selected?**

Cipher Suites offered by client:

960	6.606518000	wwwui.ecglb.bac...	akash.iith.ac.in	TCP	74 443 → 39006	[SYN, ACK]
961	6.606610546	akash.iith.ac.in	wwwui.ecglb.bac...	TCP	66 39006 → 443	[ACK] Seq
962	6.607142835	akash.iith.ac.in	wwwui.ecglb.bac...	TLSv1.2	583	Client Hello
963	6.631159332	www.google.com	akash.iith.ac.in	TLSv1.3	132	Application Data
964	6.631159679	www.google.com	akash.iith.ac.in	TLSv1.3	97	Application Data
965	6.631159764	www.google.com	akash.iith.ac.in	TLSv1.3	105	Application Data

▼ Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 508
Version: TLS 1.2 (0x0303)
▶ Random: 616eb228c13648b05ee6ca5e05883a9ece7ad3857e1329f4...
Session ID Length: 32
Session ID: 70ffe3160b347a57df75cc1f667a830540f29247188cb82d...
Cipher Suites Length: 32
▼ Cipher Suites (16 suites)
Cipher Suite: Reserved (GREASE) (0x2a2a)
Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
Compression Methods Length: 1
▶ Compression Methods (1 method)

The first two cipher suites are :

- Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
- Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)

TLS_AES_256_GCM_SHA384 (0x1302) is the strongest cipher suite since it is using AES_GCM_SHA384.

The server selected TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) cipher suite as shown in the below figure.

983	6.926961313	wwwui.ecglb.bac...	akash.iith.ac.in	TCP	66 443 → 39006 [ACK] Seq=1 Ack=518 Win=4897 Len=0 TSval=
984	6.929415355	wwwui.ecglb.bac...	akash.iith.ac.in	TLSv1.2	1514 Server Hello
985	6.929432610	akash.iith.ac.in	wwwui.ecglb.bac...	TCP	66 39006 → 443 [ACK] Seq=518 Ack=1449 Win=63712 Len=0 TS
986	6.929606395	wwwui.ecglb.bac...	akash.iith.ac.in	TCP	1514 443 → 39006 [ACK] Seq=1449 Ack=518 Win=4897 Len=1448
987	6.929615107	akash.iith.ac.in	wwwui.ecglb.bac...	TCP	66 39006 → 443 [ACK] Seq=518 Ack=2897 Win=63712 Len=0 TS
988	6.929801581	wwwui.ecglb.bac...	akash.iith.ac.in	TCP	1514 443 → 39006 [ACK] Seq=2897 Ack=518 Win=4897 Len=1448
989	6.929808769	akash.iith.ac.in	wwwui.ecglb.bac...	TCP	66 39006 → 443 [ACK] Seq=518 Ack=4345 Win=63712 Len=0 TS
990	6.930361775	wwwui.ecglb.bac...	akash.iith.ac.in	TLSv1.2	627 Certificate, Server Key Exchange, Server Hello Done
991	6.930371438	akash.iith.ac.in	wwwui.ecglb.bac...	TCP	66 39006 → 443 [ACK] Seq=518 Ack=4906 Win=63712 Len=0 TS
992	6.936472657	www.google.com	akash.iith.ac.in	TCP	66 443 → 54922 [ACK] Seq=306778 Ack=9879 Win=106496 Len=
993	6.937852984	akash.iith.ac.in	wwwui.ecglb.bac...	TLSv1.2	192 Client Key Exchange, Change Cipher Spec, Encrypted Ha
994	6.945713651	www.google.com	akash.iith.ac.in	TLSv1.3	511 Application Data
995	6.945758579	www.google.com	akash.iith.ac.in	TLSv1.3	1484 Application Data
996	6.945769641	akash.iith.ac.in	www.google.com	TCP	66 54922 → 443 [ACK] Seq=11364 Ack=308641 Win=633472 Len=
997	6.945979350	www.google.com	akash.iith.ac.in	TLSv1.3	1484 Application Data
998	6.946105481	www.google.com	akash.iith.ac.in	TLSv1.3	2902 Application Data, Application Data
999	6.946119160	akash.iith.ac.in	www.google.com	TCP	66 54922 → 443 [ACK] Seq=11364 Ack=312895 Win=642048 Len=

Handshake Type: Server Hello (2)
Length: 87
Version: TLS 1.2 (0x0303)
Random: 70f4f2aa33180292760617e2e47db97dd99d5c3c131bd452...
Session ID Length: 32
Session ID: 93d63c79776bc12d9b3fd8eafd528954ee48d71d6b1911a8...
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
Compression Method: null (0)
Extensions Length: 15
Extension: renegotiation_info (len=1)
Extension: ec_point_formats (len=2)
Extension: extended_master_secret (len=0)

5. What is the SNI value in ClientHello Record? What's its purpose? In other words, why is the client advertising it to the server?

▼ Extension: server_name (len=26)
Type: server_name (0)
Length: 26

▼ Server Name Indication extension

Server Name list length: 24
Server Name Type: host_name (0)
Server Name length: 21
Server Name: www.bankofamerica.com

► Extension: extended_master_secret (len=0)

The fields in Server Name Indication extension (SNI) are as in the above snapshot. With the help of SNI extension, the server can host multiple TLS certificates for multiple websites under a single IP address. SNI extension contains the host name in it so as the websites can be uniquely identified.

6. What is the ALPN value(s) in ClientHello Record? What's its purpose? Which one the server selected?

970	6.794849819	akash.iith.ac.in	wwwui.ecglb.bac...	TCP	66 39008 → 443 [ACK]
971	6.795112380	akash.iith.ac.in	wwwui.ecglb.bac...	TLSv1.2	583 Client Hello
972	6.838538867	akash.iith.ac.in	www.google.com	TLSv1.3	522 Application Data
973	6.857823707	www.google.com	akash.iith.ac.in	TCP	66 443 → 54922 [ACK]
974	6.898877808	www.google.com	akash.iith.ac.in	TLSv1.3	130 Application Data
975	6.898877968	www.google.com	akash.iith.ac.in	TLSv1.3	97 Application Data
976	6.898969617	akash.iith.ac.in	www.google.com	TCP	66 54922 → 443 [ACK]
977	6.898986073	www.google.com	akash.iith.ac.in	TLSv1.3	105 Application Data
978	6.900178777	akash.iith.ac.in	www.google.com	TLSv1.3	105 Application Data

<ul style="list-style-type: none"> ▶ Extension: extended_master_secret (len=0) ▶ Extension: renegotiation_info (len=1) ▶ Extension: supported_groups (len=10) ▶ Extension: ec_point_formats (len=2) ▶ Extension: SessionTicket TLS (len=0) ▼ Extension: application_layer_protocol_negotiation (len=14) <ul style="list-style-type: none"> Type: application_layer_protocol_negotiation (16) Length: 14 ALPN Extension Length: 12
<ul style="list-style-type: none"> ▼ ALPN Protocol <ul style="list-style-type: none"> ALPN string length: 2 ALPN Next Protocol: h2 ALPN string length: 8 ALPN Next Protocol: http/1.1 ▶ Extension: status_request (len=5)

ALPN values in Client Hello are h2 and http/1.1.

i9	1.683461	172.217.163.170	192.168.0.176	TLSv1.3	2902 Server Hello, Change Cipher Spec
i0	1.683505	192.168.0.176	172.217.163.170	TCP	66 39184 → 443 [ACK] Seq=518 Ack=2837 Win=63488 Len=0 TSval=1136520
i1	1.683597	172.217.163.170	192.168.0.176	TLSv1.3	1910 Encrypted Extensions, Certificate, Certificate Verify, Finished
i2	1.683613	192.168.0.176	172.217.163.170	TCP	66 39184 → 443 [ACK] Seq=518 Ack=4681 Win=63872 Len=0 TSval=1136520
i3	1.689066	192.168.0.176	172.217.163.170	TLSv1.3	130 Change Cipher Spec, Finished
i4	1.709235	172.217.163.170	192.168.0.176	HTTP2	674 SETTINGS[0], WINDOW_UPDATE[0]
i5	1.709354	192.168.0.176	172.217.163.170	TCP	66 39184 → 443 [ACK] Seq=582 Ack=5289 Win=64128 Len=0 TSval=1136520


```

Length: 4542
[Content Type: Handshake (22)]
Handshake Protocol: Encrypted Extensions
Handshake Type: Encrypted Extensions (8)
Length: 11
Extensions Length: 9
  Extension: application_layer_protocol_negotiation (len=5)
    Type: application_layer_protocol_negotiation (16)
    Length: 5
    ALPN Extension Length: 3
      ALPN Protocol
        ALPN string length: 2
        ALPN Next Protocol: h2
Handshake Protocol: Certificate

```

Application-Layer Protocol Negotiation (ALPN) is a Transport Layer Security (TLS) extension that allows the application layer to negotiate which protocol should be performed over a secure connection in a manner that avoids additional round trips and which is independent of the application-layer protocols.

From the above figure we can see that the server had selected h2.

7. Does the ClientHello contain status_request, supported_versions, psk_key_exchange_modes extensions? If so, what do they convey to the server?

970	6.794849819	akash.iith.ac.in	wwwui.ecglb.bac...	TCP	66 39008 → 443 [AC
971	6.795112380	akash.iith.ac.in	wwwui.ecglb.bac...	TLSv1.2	583 Client Hello
972	6.838538867	akash.iith.ac.in	www.google.com	TLSv1.3	522 Application Dat
973	6.857823707	www.google.com	akash.iith.ac.in	TCP	66 443 → 54922 [AC
974	6.898877808	www.google.com	akash.iith.ac.in	TLSv1.3	130 Application Dat
975	6.898877968	www.google.com	akash.iith.ac.in	TLSv1.3	97 Application Dat
976	6.898969617	akash.iith.ac.in	www.google.com	TCP	66 54922 → 443 [AC


```

  Random: 738cadee0af6aa34a40a33ffb0ccb1c799c09fa90902ef22...
  Session ID Length: 32
  Session ID: c7e6cb6e9b30506b2a472df800e91d423b4c5df5c509fae3...
  Cipher Suites Length: 32
  Cipher Suites (16 suites)
  Compression Methods Length: 1
  Compression Methods (1 method)
  Extensions Length: 403
  Extension: Reserved (GREASE) (len=0)
  Extension: server_name (len=26)
  Extension: extended_master_secret (len=0)
  Extension: renegotiation_info (len=1)
  Extension: supported_groups (len=10)
  Extension: ec_point_formats (len=2)
  Extension: SessionTicket TLS (len=0)
  Extension: application_layer_protocol_negotiation (len=14)
  Extension: status_request (len=5)
  Extension: signature_algorithms (len=18)
  Extension: signed_certificate_timestamp (len=0)
  Extension: key_share (len=43)
  Extension: psk_key_exchange_modes (len=2)
  Extension: supported_versions (len=11)
  Extension: Unknown type 27 (len=3)
  Extension: Unknown type 17513 (len=5)
  Extension: Reserved (GREASE) (len=1)
  Extension: padding (len=190)

```

Yes, the ClientHello contains status_request, supported_versions, psk_key_exchange_modes extensions.

Status_request: Constrained clients may wish to use a certificate-status protocol such as OCSP [RFC2560] to check the validity of server certificates, in order to avoid transmission of CRLs and therefore save bandwidth on constrained networks. This extension allows for such information to be sent in the TLS handshake, saving round trips and resources.

Supported Versions: Supported Versions tell the server that the client supports a variety of TLS protocols offered in supported_versions extension.

psk_key_exchange_modes: The semantics of this extension are that the client only supports the use of PSKs with these modes, which restricts both the use of PSKs offered in this ClientHello and those which the server might supply via NewSessionTicket.

8. Does ClientHello Record contain the Signature_algorithms extension? What's its purpose?

970	6.794849819	akash.iith.ac.in	wwwui.ecglb.bac...	TCP	66 39008 → 443 [ACK]
971	6.795112380	akash.iith.ac.in	wwwui.ecglb.bac...	TLSv1.2	583 Client Hello
972	6.838538867	akash.iith.ac.in	www.google.com	TLSv1.3	522 Application Data
973	6.857823707	www.google.com	akash.iith.ac.in	TCP	66 443 → 54922 [ACK]
974	6.898877808	www.google.com	akash.iith.ac.in	TLSv1.3	130 Application Data
975	6.898877968	www.google.com	akash.iith.ac.in	TLSv1.3	97 Application Data
976	6.898969617	akash.iith.ac.in	www.google.com	TCP	66 54922 → 443 [ACK]
977	6.898986073	www.google.com	akash.iith.ac.in	TLSv1.3	105 Application Data
978	6.900178777	akash.iith.ac.in	www.google.com	TLSv1.3	105 Application Data
979	6.917031228	akash.iith.ac.in	www.google.com	TLSv1.3	1368 Application Data
980	6.918378983	akash.iith.ac.in	www.google.com	TLSv1.3	243 Application Data
981	6.919072311	www.google.com	akash.iith.ac.in	TCP	66 443 → 54922 [ACK]
982	6.923285559	akash.iith.ac.in	www.google.com	TLSv1.3	1374 Application Data
983	6.926961313	wwwui.ecglb.bac...	akash.iith.ac.in	TCP	66 443 → 39006 [ACK]
984	6.929415355	wwwui.ecglb.bac...	akash.iith.ac.in	TLSv1.2	1514 Server Hello
985	6.929432610	akash.iith.ac.in	wwwui.ecglb.bac...	TCP	66 39006 → 443 [ACK]
986	6.929606395	wwwui.ecglb.bac...	akash.iith.ac.in	TCP	1514 443 → 39006 [ACK]

- ▶ Extension: SessionTicket TLS (len=0)
- ▶ Extension: application_layer_protocol_negotiation (len=14)
- ▶ Extension: status_request (len=5)
- ▶ Extension: signature_algorithms (len=18)
 - Type: signature_algorithms (13)
 - Length: 18
 - Signature Hash Algorithms Length: 16
 - ▶ Signature Hash Algorithms (8 algorithms)
 - ▶ Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
 - ▶ Signature Algorithm: rsa_pss_rsae_sha256 (0x0804)
 - ▶ Signature Algorithm: rsa_pkcs1_sha256 (0x0401)
 - ▶ Signature Algorithm: ecdsa_secp384r1_sha384 (0x0503)
 - ▶ Signature Algorithm: rsa_pss_rsae_sha384 (0x0805)
 - ▶ Signature Algorithm: rsa_pkcs1_sha384 (0x0501)
 - ▶ Signature Algorithm: rsa_pss_rsae_sha512 (0x0806)
 - ▶ Signature Algorithm: rsa_pkcs1_sha512 (0x0601)

Yes, ClientHello Record contains the signature_algorithms extension. This extension provides the signature algorithms supported by clients out of which one algorithm is accepted and is used to sign all the TLS handshake messages.

9. Does the client offer any Random number, key share, Supported Groups and PSK in ClientHello Record? How will be these used by the Server?

Yes, the below image shows that the client offers all of the Random number, key share, supported groups and PSK in ClientHello Record.

Random: This is a 32-byte random number. The client random and the server random are later used to generate the key for encryption.

key share: The "key_share" extension contains the endpoint's cryptographic parameters. Clients MAY send an empty client_shares vector in order to request group selection from the server, at the cost of an additional round trip.

Supported groups: When sent by the client, the "supported_groups" extension indicates the named groups which the client supports for key exchange, ordered from most preferred to least preferred.

PSK: These pre-shared keys are symmetric keys shared in advance among the client and server which can be established through out of band or session resumption.

333	1.620294	192.168.0.176	172.217.163.170	TCP	66 39184 → 443 [AC
334	1.621514	192.168.0.176	172.217.163.170	TLSv1.3	583 Client Hello
335	1.621891	192.168.0.176	172.217.166.100	HTTP2	105 PING[0]
336	1.632868	142.250.182.46	192.168.0.176	HTTP2	135 HEADERS[3]: 200
337	1.632933	142.250.182.46	192.168.0.176	HTTP2	744 DATA[3]
338	1.633030	192.168.0.176	142.250.182.46	TCP	66 41834 → 443 [AC
339	1.642755	142.250.182.46	192.168.0.176	HTTP2	179 DATA[3]
340	1.642759	142.250.182.46	192.168.0.176	HTTP2	158 DATA[3]


```

Length: 512
▼ Handshake Protocol: Client Hello
  Handshake Type: Client Hello (1)
  Length: 508
  Version: TLS 1.2 (0x0303)
  Random: f0e751b4c3630848a833f899da6b325f2aca8b9d25051b7c...
  Session ID Length: 32
  Session ID: 035da006346aaa820fba2b13ffda45304f39a7aa56f45f0b...
  Cipher Suites Length: 32
  ▶ Cipher Suites (16 suites)
  Compression Methods Length: 1
  ▶ Compression Methods (1 method)
  Extensions Length: 403
  ▶ Extension: Reserved (GREASE) (len=0)
  ▶ Extension: server_name (len=31)
  ▶ Extension: extended_master_secret (len=0)
  ▶ Extension: renegotiation_info (len=1)
  ▶ Extension: supported_groups (len=10)
  ▶ Extension: ec_point_formats (len=2)
  ▶ Extension: SessionTicket TLS (len=0)
  ▶ Extension: application_layer_protocol_negotiation (len=14)
  ▶ Extension: status_request (len=5)
  ▶ Extension: signature_algorithms (len=18)
  ▶ Extension: signed_certificate_timestamp (len=0)
  ▶ Extension: key_share (len=43)
  ▶ Extension: psk_key_exchange_modes (len=2)
  ▶ Extension: supported_versions (len=11)
  ▶ Extension: Unknown type 27 (len=3)
  ▶ Extension: Unknown type 17513 (len=5)
  ▶ Extension: Reserved (GREASE) (len=1)
  ▶ Extension: padding (len=185)

```

10. What TLS versions your browser/client is supporting? Which one the server selected?

333	1.620294	192.168.0.176	172.217.163.170	TCP	66 39184 → 443 [AC
334	1.621514	192.168.0.176	172.217.163.170	TLSv1.3	583 Client Hello
335	1.621891	192.168.0.176	172.217.166.100	HTTP2	105 PING[0]
336	1.632868	142.250.182.46	192.168.0.176	HTTP2	135 HEADERS[3]: 200
337	1.632933	142.250.182.46	192.168.0.176	HTTP2	744 DATA[3]
338	1.633030	192.168.0.176	142.250.182.46	TCP	66 41834 → 443 [AC
339	1.642755	142.250.182.46	192.168.0.176	HTTP2	179 DATA[3]
340	1.642759	142.250.182.46	192.168.0.176	HTTP2	158 DATA[3]


```

▼ Extension: supported_versions (len=11)
  Type: supported_versions (43)
  Length: 11
  Supported Versions length: 10
  Supported Version: Unknown (0xbaba)
  Supported Version: TLS 1.3 (0x0304)
  Supported Version: TLS 1.2 (0x0303)
  Supported Version: TLS 1.1 (0x0302)
  Supported Version: TLS 1.0 (0x0301)
  ▶ Extension: Unknown type 27 (len=3)
  ▶ Extension: Unknown type 17513 (len=5)

```

As we can see above, browser/client supports TLS 1.0, 1.1, 1.2, 1.3 and the server selected, TLS 1.2 (figure is shown below)

348	1.678564	172.217.166.100	192.168.0.176	TCP	66 443 → 40978 [ACK] Seq=50955 Ack=2588 W
349	1.683461	172.217.163.170	192.168.0.176	TLSv1.3	2902 Server Hello, Change Cipher Spec

```

Handshake Type: Server Hello (2)
Length: 118
Version: TLS 1.2 (0x0303)
Random: bc52d0a5575ac89c45af0c712a866341249ad1ea75d6a912...
Session ID Length: 32
Session ID: 035da006346aaa820fba2b13ffda45304f39a7aa56f45f0b...
Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
Compression Method: null (0)
Extensions Length: 46
  ▶ Extension: key_share (len=36)
  ▶ Extension: supported_versions (len=2)
▼ TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
Content Type: Change Cipher Spec (20)
Version: TLS 1.2 (0x0303)
Length: 1
Change Cipher Spec Message

```

11. Look at Certificate Record from the server to the client: How many certificates did the server return and how are they related? Who is the issuer of the Bank's certificate? What type of public key the bank is using?

973	00.421867388	192.168.0.176	192.229.237.104	TCP	66 52342 → 443 [ACK] Seq=873 Ack=4196 Win=64128 Len=0 TSval=358391429 TSecr=2410830455
974	00.422768208	192.229.237.104	192.168.0.176	TLSv1.3	2426 Certificate, Certificate Verify, Finished
975	00.422862753	192.168.0.176	192.229.237.104	TCP	66 52342 → 443 [ACK] Seq=873 Ack=6556 Win=63744 Len=0 TSval=358391430 TSecr=2410830456
976	00.430611381	192.168.0.176	192.229.237.104	TLSv1.3	140 Finished
977	00.433370769	142.250.182.78	192.168.0.176	TCP	66 443 → 41350 [ACK] Seq=12709 Ack=4972 Win=76544 Len=0 TSval=4568074 TSecr=3815065330
978	00.433371275	142.250.182.78	192.168.0.176	TCP	66 443 → 41350 [ACK] Seq=12709 Ack=5011 Win=76544 Len=0 TSval=4568074 TSecr=3815065330
979	00.433371309	142.250.182.78	192.168.0.176	HTTP2	105 PING[0]
980	00.433441359	192.168.0.176	142.250.182.78	TCP	66 41350 → 443 [ACK] Seq=6408 Ack=12748 Win=64128 Len=0 TSval=3815065351 TSecr=4568074
981	00.439630509	142.250.182.78	192.168.0.176	TCP	66 443 → 41350 [ACK] Seq=12748 Ack=6408 Win=79360 Len=0 TSval=4568080 TSecr=3815065330
982	00.451081881	192.229.237.141	192.168.0.176	TCP	66 443 → 33124 [ACK] Seq=100 Ack=872 Win=68896 Len=0 TSval=2594503613 TSecr=743320140
983	00.451979744	192.229.237.141	192.168.0.176	TLSv1.3	1514 Server Hello, Encrypted Extensions
984	00.451996624	192.168.0.176	192.229.237.141	TCP	66 33124 → 443 [ACK] Seq=872 Ack=1548 Win=64128 Len=0 TSval=743320206 TSecr=2594503614
985	00.452825777	192.229.237.141	192.168.0.176	TCP	1514 443 → 33124 [PSH, ACK] Seq=1548 Ack=872 Win=68896 Len=1448 TSval=2594503614 TSecr=743320140 [TCP ...

```

▶ Transmission Control Protocol, Src Port: 443, Dst Port: 52342, Seq: 4196, Ack: 873, Len: 2360
▶ [4 Reassembled TCP Segments (5899 bytes): #968(1251), #970(1448), #972(1200), #974(2000)]
▼ Secure Sockets Layer
  ▼ TLSv1.3 Record Layer: Handshake Protocol: Certificate
    Opaque Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 5894
    [Content Type: Handshake (22)]
    ▼ Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 5873
      Certificate Request Context Length: 0
      Certificates Length: 5869
      ▼ Certificates (5869 bytes)
        Certificate Length: 1842
        ▶ Certificate: 3082072e30820616a00302010202100363ad6c6090c8aa8c... (id-at-commonName=www1.bac-assets.com,id-at-serialNumber=2927442,id-at-businessCategory=Private Organization,id-at-organizationName=
          Extensions Length: 1593
          ▶ Extension: status_request (len=1589)
          Certificate Length: 1329
        ▶ Certificate: 3082052d30820415a003020102020c61ae7d200000000051... (id-at-commonName=Entrust Certification Authority - L1M,id-at-organizationalUnitName=(c) 2014 Entrust, Inc. - for autho,id-at-organi
          Extensions Length: 0
          Certificate Length: 1090
        ▶ Certificate: 3082043e30820326a00302010202044a538c28300d06092a... (id-at-commonName=Entrust Root Certification Authority - G2,id-at-organizationalUnitName=(c) 2009 Entrust, Inc. - for autho,id-at-or
          Extensions Length: 0
      ▼ Secure Sockets Layer

```

Server has returned 3 certificates in a chain of trust.

1. User- bankofamerica.com
2. Intermediate CA - Entrust Certification Authority
3. Root CA - Entrust Root Certification Authority

Issuer of the bank's certificate is Entrust Certification Authority. Bank is Using RSA public key.

12. Comment on the key exchange algorithm agreed upon, what are the parameters that got exchanged between client and server to derive the session keys.

958	60.385977955	192.229.237.141	192.168.0.176	TLSv1.3	165 Hello Retry Request, Change Cipher Spec
960	60.386259183	192.168.0.176	192.229.237.141	TLSv1.3	420 Change Cipher Spec, Client Hello
968	60.421393217	192.229.237.104	192.168.0.176	TLSv1.3	1514 Server Hello, Encrypted Extensions
974	60.422788286	192.229.237.104	192.168.0.176	TLSv1.3	2426 Certificate, Certificate Verify, Finished
976	60.430811381	192.168.0.176	192.229.237.104	TLSv1.3	140 Finished
983	60.451979744	192.229.237.141	192.168.0.176	TLSv1.3	1514 Server Hello, Encrypted Extensions
991	60.453894027	192.229.237.141	192.168.0.176	TLSv1.3	976 Certificate, Certificate Verify, Finished
993	60.461176529	192.168.0.176	192.229.237.141	TLSv1.3	140 Finished
998	60.484525719	192.229.237.104	192.168.0.176	TLSv1.3	321 New Session Ticket
1000	60.485343085	192.229.237.104	192.168.0.176	TLSv1.3	321 New Session Ticket

```

    ▶ Extension: status_request (len=5)
    ▶ Extension: signature_algorithms (len=18)
    ▶ Extension: signed_certificate_timestamp (len=0)
    ▶ Extension: key_share (len=71)
      Type: key_share (51)
      Length: 71
      ▶ Key Share extension
        Client Key Share Length: 69
        ▶ Key Share Entry: Group: secp256r1, Key Exchange length: 65
          Group: secp256r1 (23)
          Key Exchange Length: 65
          Key Exchange: 04f2312b7df82bf97f91ab144e1d8dc837eb87ae48efef0e...
        ▶ Extension: psk_key_exchange_modes (len=2)
          Type: psk_key_exchange_modes (45)
          Length: 2
          PSK Key Exchange Modes Length: 1
      PSK Key Exchange Mode: PSK with (EC)DHE key establishment (psk_dhe_ke) (1)
    ▶ Extension: supported_versions (len=11)

```

968	60.421393217	192.229.237.104	192.168.0.176	TLSv1.3	1514 Server Hello, Encrypted Extensions
974	60.422788286	192.229.237.104	192.168.0.176	TLSv1.3	2426 Certificate, Certificate Verify, Finished
976	60.430811381	192.168.0.176	192.229.237.104	TLSv1.3	140 Finished
983	60.451979744	192.229.237.141	192.168.0.176	TLSv1.3	1514 Server Hello, Encrypted Extensions
991	60.453894027	192.229.237.141	192.168.0.176	TLSv1.3	976 Certificate, Certificate Verify, Finished
993	60.461176529	192.168.0.176	192.229.237.141	TLSv1.3	140 Finished
998	60.484525719	192.229.237.104	192.168.0.176	TLSv1.3	321 New Session Ticket
1000	60.485343085	192.229.237.104	192.168.0.176	TLSv1.3	321 New Session Ticket

```

    ▶ TLSv1.3 Record Layer: Handshake Protocol: Certificate Verify
      Opaque Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 281
      [Content Type: Handshake (22)]
      ▶ Handshake Protocol: Certificate Verify
        Handshake Type: Certificate Verify (15)
        Length: 260
        ▶ Signature Algorithm: rsa_pss_rsae_sha256 (0x0804)
          Signature length: 256
          Signature: 5cbc8987e7a99ad4263feb88f652cb28aee9fd7e042a3b9...
      ▶ TLSv1.3 Record Layer: Handshake Protocol: Finished
        Opaque Type: Application Data (23)
        Version: TLS 1.2 (0x0303)
        Length: 69
        [Content Type: Handshake (22)]
        ▶ Handshake Protocol: Finished
          Handshake Type: Finished (20)
          Length: 48
          Verify Data

```

We observe that Client and Server exchange ECDHE key parameters i.e. Curve type, named curve, pubkey length and pubkey.

13. Which certificate type (DV/OV/EV) the bank is using?

990	6.930361775	wwwui.ecglib.bac...	akash.iith.ac.in	TLSv1.2	627 Certificate, Server Key Exchange, Server Hello Done
991	6.930371438	akash.iith.ac.in	wwwui.ecglib.bac...	TCP	66 39006 → 443 [ACK] Seq=518 Ack=4906 Win=63712 Len=0 TSV:
992	6.936472657	www.google.com	akash.iith.ac.in	TCP	66 443 → 54922 [ACK] Seq=306778 Ack=9879 Win=106496 Len=0
993	6.937852984	akash.iith.ac.in	wwwui.ecglib.bac...	TLSv1.2	192 Client Key Exchange, Change Cipher Spec, Encrypted Hand
994	6.945713651	www.google.com	akash.iith.ac.in	TLSv1.3	511 Application Data
995	6.945758579	www.google.com	akash.iith.ac.in	TLSv1.3	1484 Application Data

```

    ▶ Certificates (4450 bytes)
      Certificate Length: 2022
      ▶ Certificate: 308207e2308206caa003020102021036f36b65b810abdacb... (id-at-commonName=www.bankofamerica.com,id-at-serialNumber=29274
      Certificate Length: 1329
      ▶ Certificate: 3082052d30820415a003020102020c61a1e7d20000000051... (id-at-commonName=Entrust Certification Authority - L1M,id-at-or
      Certificate Length: 1090
      ▶ Certificate: 3082043e30820326a00302010202044a538c2830cd06092a... (id-at-commonName=Entrust Root Certification Authority - G2,id-a
    ▶ Secure Sockets Layer
      ▶ TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
        Content Type: Handshake (22)
        Version: TLS 1.2 (0x0303)

```

As the subject field contains jurisdictionCountryname, serial number and much more information, the certificate type bank is using is EV certificate.

14. Which certificate type (single or multi-domain or wild-card) the bank is using?

989	6.929808769	akash.iith.ac.in	wwwui.ecglb.bac...	TCP	66	39006 → 443 [ACK]	Seq=518 Ack=4345 Win=63712 Len=0
990	6.930361775	wwwui.ecglb.bac...	akash.iith.ac.in	TLSv1.2	627	Certificate, Server Key Exchange, Server Hello Done	
991	6.930371438	akash.iith.ac.in	wwwui.ecglb.bac...	TCP	66	39006 → 443 [ACK]	Seq=518 Ack=4906 Win=63712 Len=0
992	6.936472657	www.google.com	akash.iith.ac.in	TCP	66	443 → 54922 [ACK]	Seq=306778 Ack=9879 Win=106496 Len=0
993	6.937852984	akash.iith.ac.in	wwwui.ecglb.bac...	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted	
994	6.945713651	www.google.com	akash.iith.ac.in	TLSv1.3	511	Application Data	
995	6.945758579	www.google.com	akash.iith.ac.in	TLSv1.3	1484	Application Data	
996	6.945769641	akash.iith.ac.in	www.google.com	TCP	66	54922 → 443 [ACK]	Seq=11364 Ack=308641 Win=633472 Len=0
997	6.945979350	www.google.com	akash.iith.ac.in	TLSv1.3	1484	Application Data	
998	6.946105481	www.google.com	akash.iith.ac.in	TLSv1.3	2902	Application Data, Application Data	
999	6.946119160	akash.iith.ac.in	www.google.com	TCP	66	54922 → 443 [ACK]	Seq=11364 Ack=312895 Win=642048 Len=0
1000	6.946286675	www.google.com	akash.iith.ac.in	TLSv1.3	1484	Application Data	
1001	6.946621096	www.google.com	akash.iith.ac.in	TLSv1.3	5738	Application Data, Application Data, Application Data	

- Extension (id-ce-subjectKeyIdentifier)
- Extension (id-ce-authorityKeyIdentifier)
- Extension (id-pe-authorityInfoAccessSyntax)
- Extension (id-ce-cRLDistributionPoints)
- Extension (id-ce-subjectAltName)
 - Extension Id: 2.5.29.17 (id-ce-subjectAltName)
 - GeneralNames: 6 items
 - GeneralName: dNSName (2)
 - dNSName: www.bankofamerica.com
 - GeneralName: dNSName (2)

We observe from the id-ce-subjectAltName extension that the certificate is a single-domain certificate.

15. How can the client check whether the certificate is revoked or not: OCSP/CRL?

Does the server support OCSP stapling?

<ul style="list-style-type: none"> Extension (id-pe-authorityInfoAccessSyntax) Extension (id-ce-cRLDistributionPoints) <ul style="list-style-type: none"> Extension Id: 2.5.29.31 (id-ce-cRLDistributionPoints) CRLDistPointsSyntax: 1 item <ul style="list-style-type: none"> DistributionPoint <ul style="list-style-type: none"> distributionPoint: fullName (0) <ul style="list-style-type: none"> fullName: 1 item <ul style="list-style-type: none"> GeneralName: uniformResourceIdentifier (6) <ul style="list-style-type: none"> uniformResourceIdentifier: http://crl.entrust.net/level1m.crl Extension (id-ce-subjectAltName)
--

We can open the link provided in cRLDistributionPoints extension to check the OCSP status to see if the certificate is revoked or not.

No, the server doesn't support ocsp stapling as shown in the below screenshot.

SSL Certificate Analysis	
RSA CERTIFICATE INFORMATION	
Issuer	Entrust Certification Authority - L1M
Trusted	Yes
Common Name	www.bankofamerica.com
Key Type/Size	RSA 2048 bits
Signature Algorithm	sha256WithRSAEncryption
Subject Alternative Names	DNS:www.bankofamerica.com, DNS:mobile.bankofamerica.com, DNS:smallbusinessonlinecommunity.bankofamerica.com, DNS:chatui.ml.com, DNS:chatui.merrill.com, DNS:chatui.merrilledge.com
Transparency	Yes
Validation Level	EV
CRL	http://crl.entrust.net/level1m.crl
OCSP	http://ocsp.entrust.net
OCSP Must-Staple	No
Supports OCSP Stapling	No
Valid From	November 3rd 2021, 14:42 CET
Valid To	November 3rd 2022, 14:42 CET

16. How many log servers logged the certificate of the bank? What role does the log server play in the Web PKI ecosystem? Refer: SCT extension.

```
▼ Extension (SignedCertificateTimestampList)
  Extension Id: 1.3.6.1.4.1.11129.2.4.2 (SignedCertificateTimestampList)
  Serialized SCT List Length: 360
  ▶ Signed Certificate Timestamp (DigiCert Log Server)
  ▶ Signed Certificate Timestamp (Unknown Log)
  ▶ Signed Certificate Timestamp (Unknown Log)
▶ algorithmIdentifier (sha256WithRSAEncryption)
  Padding: 0
```

As we can see, there are 3 log servers which log the certificate of the bank. The role of log server in the Web PKI ecosystem is to maintain certificate transparency

17. How is the application data being encrypted? Do the records containing application data include a separate MAC? Does Wireshark distinguish between the encrypted application data and the MAC?

- Application data is encrypted using “Symmetric encryption algorithm” (ECHDE in this case)
- MAC is included in the records containing application data.
- Wireshark cannot distinguish between encrypted applications data and MAC.

18. Look at various keys logged in the file pointed to by the SSLKEYLOGFILE environment variable in your host OS and describe their usage. Also comment on how they are derived from nonces and other parameters using HKDF. Which entity in your system does this job on-the-fly?

The file contains 6 keys:

1. **CLIENT_HANDSHAKE_TRAFFIC_SECRET**: This key is hex-encoded and used by the client for handshake.
2. **SERVER_HANDSHAKE_TRAFFIC_SECRET**: This key is hex-encoded and used by the server for handshake.
3. **CLIENT_TRAFFIC_SECRET_0**: It's the client side's first hex-encoded application traffic secret (for TLS 1.3).
4. **SERVER_TRAFFIC_SECRET_0**: It's the server side's first hex-encoded application traffic secret (for TLS 1.3).
5. **EXPORTER_SECRET**: The exporter secret encoded by the hex (For TLS 1.3).
6. **CLIENT_RANDOM**: It is encoded as 96 hexadecimal characters, 48 bytes for the master secret.

19. Do you see any support for session resumption in the trace? What do you find inside the session ticket, if it is used? Is it based on Session ID/Session ticket or

PSK based Session ticket? What role do the session IDs play in TLS 1.3?

Yes, there is support for the session resumption, as there is a New Session ticket issued by the server after the client sends the finished message. Session resumption here is based on Session Ticket. TLS Session ticket contains Lifetime hint, Nonce, Length and Session Ticket fields. TLS 1.3 replaces session IDs and session tickets with the concept of session resumption via pre-shared keys (PSK), hence session IDs doesn't play any role in TLS 1.3.

```
▼ TLS Session Ticket
  Session Ticket Lifetime Hint: 7200 seconds (2 hours)
  Session Ticket Age Add: 2681851543
  Session Ticket Nonce Length: 8
  Session Ticket Nonce: 0000000000000000
  Session Ticket Length: 208
  Session Ticket: bd0f624da443791b9faa9a67ab82f555589d23835d6f83c4...
  Extensions Length: 0
```

20. How long does it take for TLS to establish a secure pipe? How much of it could be reduced when session resumption is used?

In TLS 1.2 it takes around 2 RTTs (approx 112ms) for establishing a secure connection, whereas TLS 1.3 takes 1 RTT. If resumption is used, TLS 1.2 takes 1 RTT whereas TLS 1.3 takes 0RTTs.

21. What is the duration of the HTTPS session, how many IP packets are exchanged in the browsing session (starting from the first TCP SYN packet till TCP FIN packet)?

1583	8.845535689	akash.iith.ac.in	secure.ecglb.bac.com	TCP	74 44676 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T
1584	8.847427439	akash.iith.ac.in	192.229.237.105	TCP	74 51956 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T
1585	8.853702913	dcsc-edge-ndi-15...	akash.iith.ac.in	TCP	74 443 → 33530 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1460 SA
1586	8.853787444	akash.iith.ac.in	dcsc-edge-ndi-1562177...	TCP	66 33530 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=269254648
1587	8.854351749	akash.iith.ac.in	dcsc-edge-ndi-1562177...	TLSv1.2	583 Client Hello
1588	8.854785363	dlinkrouter.iith...	akash.iith.ac.in	DNS	115 Standard query response 0x54a8 A careers2019.egloba12.bac.com
1589	8.883538266	dcsc-edge-ndi-15...	akash.iith.ac.in	TCP	66 443 → 33530 [ACK] Seq=1 Ack=518 Win=28160 Len=0 TSval=25274533
1590	8.883712433	dcsc-edge-ndi-15...	akash.iith.ac.in	TLSv1.2	4430 Server Hello, Certificate, Server Key Exchange, Server Hello D
1591	8.883756637	akash.iith.ac.in	dcsc-edge-ndi-1562177...	TCP	66 33530 → 443 [ACK] Seq=518 Ack=4365 Win=61824 Len=0 TSval=26925
1592	8.891536331	192.229.237.105	akash.iith.ac.in	TCP	74 443 → 51956 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SA
1593	8.891673537	akash.iith.ac.in	192.229.237.105	TCP	66 51956 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1918961705

4219	23.657830279	secure.ecglb.bac...	akash.iith.ac.in	TCP	66 443 → 44676 [FIN, ACK] Seq=
4220	23.657880999	akash.iith.ac.in	secure.ecglb.bac.com	TCP	66 44676 → 443 [ACK] Seq=1265
4221	23.661177459	wwwui.ecglb.bac...	akash.iith.ac.in	TCP	66 443 → 39008 [ACK] Seq=6212
4222	23.661177682	wwwui.ecglb.bac...	akash.iith.ac.in	TCP	66 443 → 39008 [FIN, ACK] Seq=
4223	23.661177780	wwwui.ecglb.bac...	akash.iith.ac.in	TCP	66 443 → 39006 [ACK] Seq=6944
4224	23.661177885	wwwui.ecglb.bac...	akash.iith.ac.in	TCP	66 443 → 39006 [FIN, ACK] Seq=
4225	23.662878202	secure.ecglb.bac...	akash.iith.ac.in	TCP	66 443 → 44706 [ACK] Seq=148 A
4226	23.662979914	secure.ecglb.bac...	akash.iith.ac.in	TCP	66 443 → 44706 [FIN, ACK] Seq=

Total time of the browsing session = 23.662979914 - 8.845535689 = 14.817444225 sec.

Total number of IP packets exchanged = 4226-1592 = 2634

22. How many TLS connections are established?

For calculating the number of connections established I have counted the number of “Finish” messages because it marks the end of session establishment in TLS.

Total count: 11

Count with bank of america: 5

23. How many HTTP request/response packets are exchanged in the browsing session? Identify the packet(s) that carried the response that included the Netbanking LOG-IN page of the bank. Do these response messages carry any security related directives like XSS, sameorigin, HSTS?

Total of 4158 packets are exchanged in the browsing session.

16 0.165694235	akash.iith.ac.in	clientservices.google...	HTTP2	158 Magic, SETTINGS[0], WINDOW_UPDATE[0]
17 0.166098948	akash.iith.ac.in	clientservices.google...	HTTP2	380 HEADERS[1]: GET /chrome-variations/seed?
18 0.185281826	clientservices.g...	akash.iith.ac.in	HTTP2	674 SETTINGS[0], WINDOW_UPDATE[0]
20 0.185570021	akash.iith.ac.in	clientservices.google...	HTTP2	97 SETTINGS[0]
21 0.186703589	clientservices.g...	akash.iith.ac.in	HTTP2	97 SETTINGS[0]
26 0.374546251	clientservices.g...	akash.iith.ac.in	HTTP2	326 HEADERS[1]: 304 Not Modified
27 0.374546425	clientservices.g...	akash.iith.ac.in	HTTP2	97 DATA[1]
28 0.374546493	clientservices.g...	akash.iith.ac.in	HTTP2	105 PING[0]
34 0.377954921	akash.iith.ac.in	clientservices.google...	HTTP2	105 PING[0]
55 0.470002963	akash.iith.ac.in	accounts.google.com	HTTP2	158 Magic, SETTINGS[0], WINDOW_UPDATE[0]
57 0.470218244	akash.iith.ac.in	accounts.google.com	HTTP2	784 HEADERS[1]: POST /ListAccounts?gpsia=1&s
58 0.470279328	akash.iith.ac.in	accounts.google.com	HTTP2	98 DATA[1] (application/x-www-form-urlencoded
4044 23.270137622	akash.iith.ac.in	prod-lb-8-1772099769...	HTTP2	108 RST_STREAM[29]
4112 23.351013522	prod-lb-8-177209...	akash.iith.ac.in	HTTP2	654 HEADERS[29]: 200 OK, DATA[29]
4113 23.351013745	prod-lb-8-177209...	akash.iith.ac.in	HTTP2	104 DATA[29] (text/javascript)
4173 23.403313246	prod-lb-8-177209...	akash.iith.ac.in	HTTP2	654 HEADERS[31]: 200 OK, DATA[31]
4174 23.403313647	prod-lb-8-177209...	akash.iith.ac.in	HTTP2	104 DATA[31] (text/javascript)

Below is the packet that carries the response that included the Net Banking LOG-IN page of the bank.

1987	10.337344625	akash.iith.ac.in	secure.ecglib.bac.com	HTTP	295	GET /login/sign-in/entry/cc.go HTTP/1.1
2005	10.410726069	adservice.google...	akash.iith.ac.in	HTTP2	290	HEADERS[3]: 200 OK
2006	10.410726270	adservice.google...	akash.iith.ac.in	HTTP2	139	DATA[3]
2008	10.412166054	adservice.google...	akash.iith.ac.in	HTTP2	97	DATA[3] (GIF89a)
2009	10.412166123	adservice.google...	akash.iith.ac.in	HTTP2	105	PING[0]
2011	10.412732150	akash.iith.ac.in	adservice.google.com	HTTP2	105	PING[0]

Hypertext Transfer Protocol

GET /login/sign-in/entry/cc.go HTTP/1.1\r\n

[Expert Info (Chat/Sequence): GET /login/sign-in/entry/cc.go HTTP/1.1\r\n]
Request Method: GET
Request URI: /login/sign-in/entry/cc.go
Request Version: HTTP/1.1
Host: secure.bankofamerica.com\r\n
Connection: keep-alive\r\n
sec-ch-ua: " Not;A Brand";v="99", "Google Chrome";v="97", "Chromium";v="97"\r\n
sec-ch-ua-mobile: ?0\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36\r\n
sec-ch-ua-platform: "Linux"\r\n
Accept: */*\r\n
Sec-Fetch-Site: same-site\r\n
Sec-Fetch-Mode: no-cors\r\n

Fields under Hypertext Transfer Protocol carry security related directives.

24. Identify the HTTP packet(s) that carried LOG-IN credentials supplied by you. Look at the raw bytes displayed in the wireshark GUI and identify strings that carried your LOG-IN credentials. Are you able to find both user id and password in the raw packet capture?1

- It's important that you only keyed in some arbitrary user id and password as part of this assignment for more safety!

3123	15.84731133...	akash.iith.ac.in	wwwui.ecglib.bac...	HTTP	1270
3025	15.0982305...	akash.iith.ac.in	secure.ecglib.ba...	HTTP	650
344	1.842778503	akash.iith.ac.in	www.google.com	HTTP2	101
366	1.878007480	akash.iith.ac.in	www.google.com	HTTP2	101
462	2.578735921	akash.iith.ac.in	www.google.com	HTTP2	101

File Data: 27582 bytes

HTML Form URL Encoded: application/x-www-form-urlencoded

Form item: "f_variable" = "TF1;015;;;;;;;;;;;;;Mozilla;Netscape;5.0%20%
Form item: "locale" = "en-US"
Form item: "anotherOnlineIDFlag" = "N"
Form item: "dltoken" = ""
Form item: "Access_ID_1" = ""
Form item: "reason" = ""
Form item: "passcode" = "akash"
Form item: "onlineId" = "akash"
Form item: "multiID" = ""
Form item: "saveMyID" = "N"
Form item: "webAuthAPI" = "true"
Form item: "origin" = "sparta_homepage"
Form item: "_ia" = "Is0FwpJepC0Fw5sVYMOUw5IOa80fwocFbM0Sw5A00M0Ww5FSe80Lw4NDMMKD

Yes, the userid and password are both visible in textual format after decrypting using sslkeyfile.

25. Generate an SSL report of the bank using [SSL Server Test \(Powered by Qualys](#)

[SSL Labs](#)) and summarise what security features are implemented by the bank's web server for improved online banking by its customers. Does the report flag any issues with the security of the bank?

From the report shown below, some security features implemented by the bank are server supported secure Renegotiation, TLS_FALLBACK_SCSV usage to prevent downgrade attacks, heartbleed extension disabled and prevention from BEAST and POODLE attacks. It doesn't support OCSP stapling which is non-compliant with NIST guidelines. The server supports a client-initiated secure renegotiation that may be unsafe and allow Denial of Service attacks.

P-384 (secp384r1) (384 bits)	Good configuration
POODLE OVER TLS ⓘ	
The server is not vulnerable to POODLE over TLS.	Not vulnerable
GOLDENDOODLE ⓘ	
The server is not vulnerable to GOLDENDOODLE.	Not vulnerable
Zombie POODLE ⓘ	
The server is not vulnerable to Zombie POODLE.	Not vulnerable
Sleeping POODLE ⓘ	
The server is not vulnerable to Sleeping POODLE.	Not vulnerable
0-Length OpenSSL ⓘ	
The server is not vulnerable 0-Length OpenSSL.	Not vulnerable
CVE-2016-2107 ⓘ	
The server is not vulnerable to CVE-2016-2107.	Not vulnerable
SERVER DOES NOT SUPPORT CLIENT-INITIATED INSECURE RENEGOTIATION ⓘ	
The server does not support client-initiated insecure renegotiation.	Good configuration
ROBOT ⓘ	
The server is not vulnerable to ROBOT vulnerability.	Not vulnerable
HEARTBLEED ⓘ	
The server version of OpenSSL is not vulnerable to Heartbleed attack.	Not vulnerable
CVE-2014-0224 ⓘ	
The server is not vulnerable to CCS Injection.	Not vulnerable
CVE-2021-3449 ⓘ	
The server is not vulnerable to CVE-2021-3449 (OpenSSL Maliciously Crafted Renegotiation Vulnerability).	Not vulnerable

SSL Certificate Analysis

RSA CERTIFICATE INFORMATION

Issuer	Entrust Certification Authority - L1M
Trusted	Yes
Common Name	www.bankofamerica.com
Key Type/Size	RSA 2048 bits
Signature Algorithm	sha256WithRSAEncryption
Subject Alternative Names	DNS:www.bankofamerica.com, DNS:mobile.bankofamerica.com, DNS:smallbusinessonlinecommunity.bankofamerica.com, DNS:chatui.ml.com, DNS:chatui.merrill.com, DNS:chatui.merrilledge.com
Transparency	Yes
Validation Level	EV
CRL	http://crl.entrust.net/level1m.crl
OCSP	http://ocsp.entrust.net
OCSP Must-Staple	No
Supports OCSP Stapling	No
Valid From	November 3rd 2021, 14:42 CET
Valid To	November 3rd 2022, 14:42 CET

26. Comment on and explain anything else that you found interesting in the trace.

The server doesn't provide OCSP stapling but it provides Session resumption as it issued a new ticket.

Note 1: Bonus 30 marks if you complete TLS and HTTPS decrypting using openssl and shell/python scripting. Make sure your code is well documented.

Note 2: Add screenshots of relevant in your report in order to prove that the capture trace used for analysis is indeed of your own and makes the evaluation easy for TAs!!

PS: What's Wireshark++? Wireshark + Key log file!

Deliverables in GC as a tar ball:

- A readable PDF Report with name "TLSAsg-<RollNo>.PDF"
- Deliverables 1-3 (refer page-1 of the assignment)
- SSL Key Log File

- Shell scripts written for openssl based decrypting of TLS and HTTPS session (optional, to get bonus marks)

References:

1. [Article: K50557518 - Decrypt SSL traffic with the SSLKEYLOGFILE environment variable on Firefox or Google Chrome using Wireshark \(f5.com\)](#)
2. [Wireshark Tutorial: Decrypting HTTPS Traffic \(Includes SSL and TLS\) \(paloaltonetworks.com\)](#)
3. [Decrypting TLS Streams With Wireshark: Part 1 | Didier Stevens](#)
4. <http://www.motobit.com/util/base64-decoder-encoder.asp>
5. [Dissecting TLS Using Wireshark \(catchpoint.com\)](#)
6. <https://tls13.ulfheim.net/>
7. <https://www.davidwong.fr/tls13/>
8. [SSL Server Test \(Powered by Qualys SSL Labs\)](#)
9. [Application-Layer Protocol Negotiation - Wikipedia](#)
10. [RFC 6066 - Transport Layer Security \(TLS\) Extensions](#)
11. [RFC 8446 - The Transport Layer Security \(TLS\) Protocol Version 1.3](#)