Network security - Assignment 4

Nisha M
CS19BTECH11012

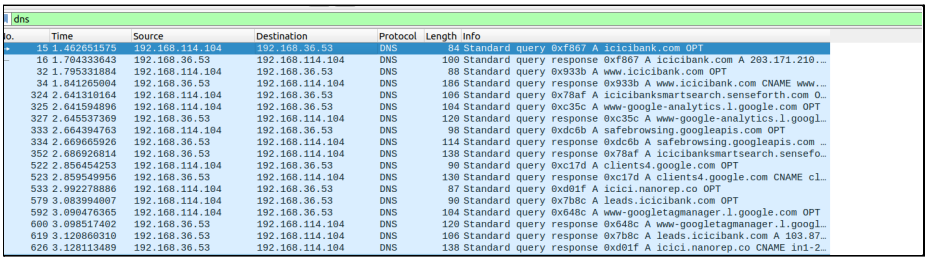
Question 1

The browser used was google chrome and the version 98.0.4758.102 (Official Build) (64-bit).

Question 2

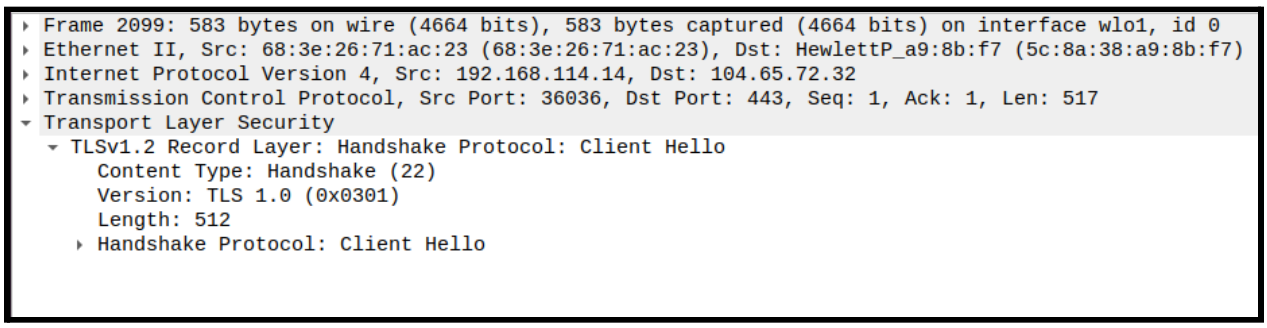
The various protocols noticed under “Protocol” include :

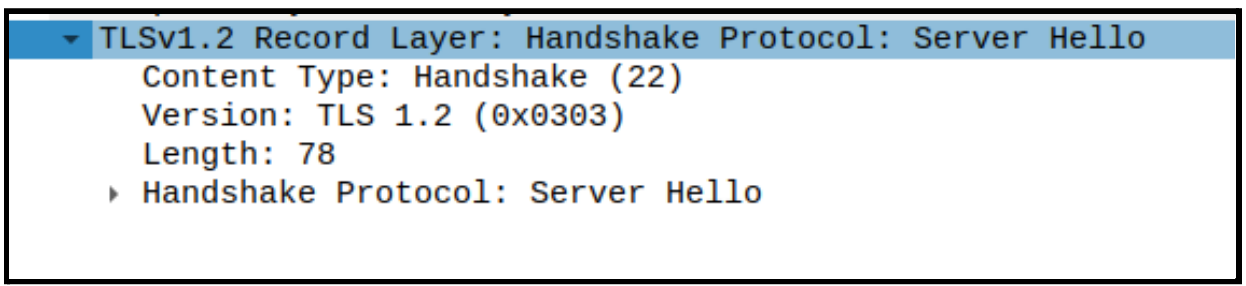
1. DNS protocol - The IP address of the hostname is queried (hostname - www.icicibank.com)

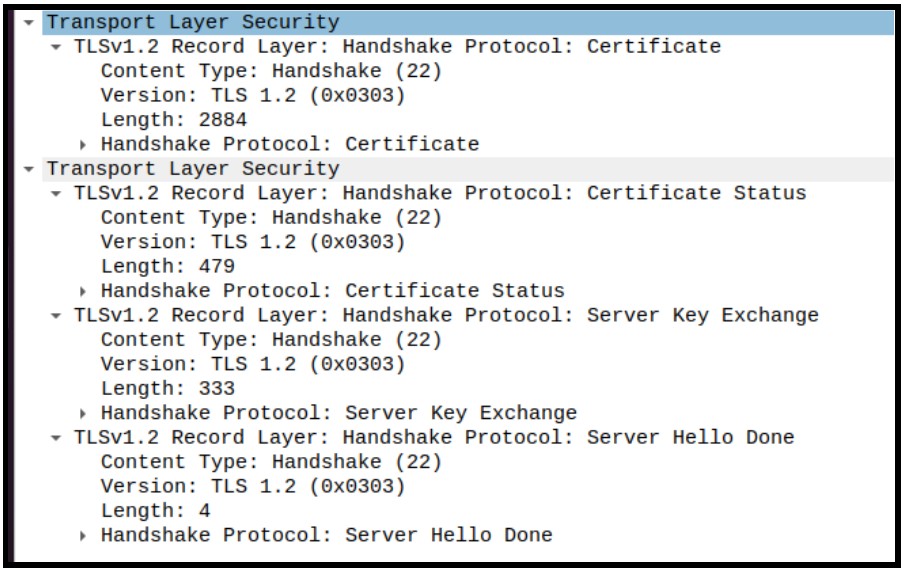


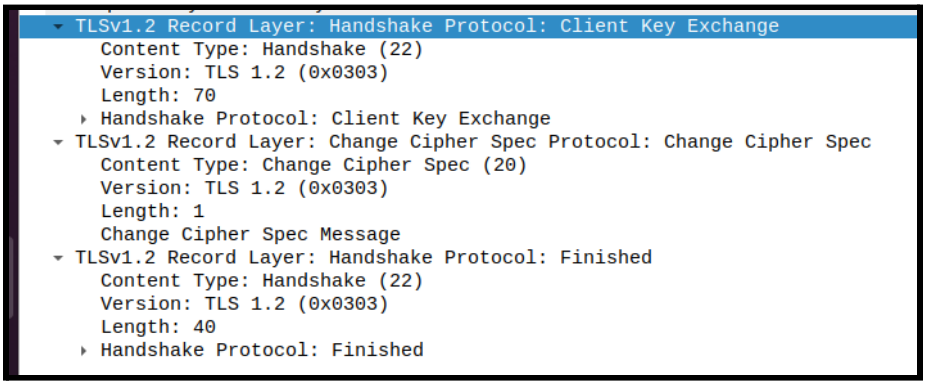
2. TCP protocol - This is the TCP handshake that includes the SYN, ACKand integrity check as well (by checking for the checksum)
3. TLSv1.2/TLSv1.3 - This is the TLS handshake that includes the client hello , server hello messages etc.
4. HTTP - Get requests with the url : https://www.icicibank.com/

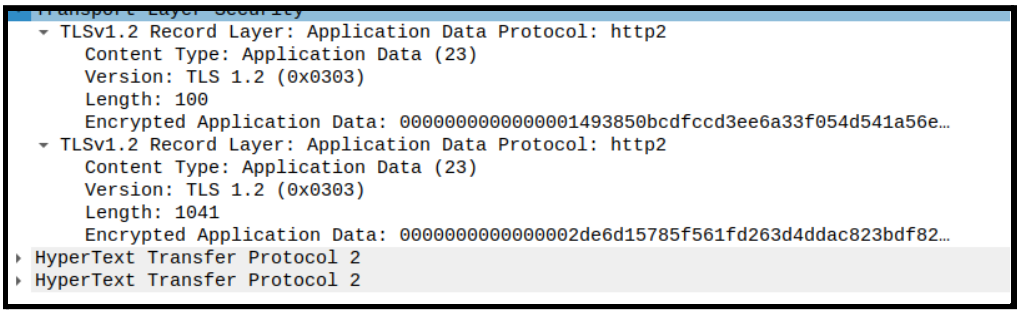
Question 3

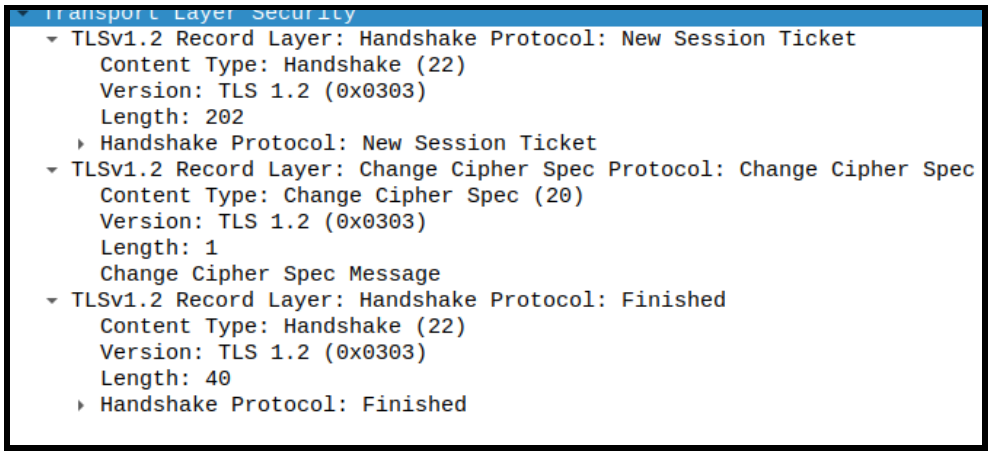


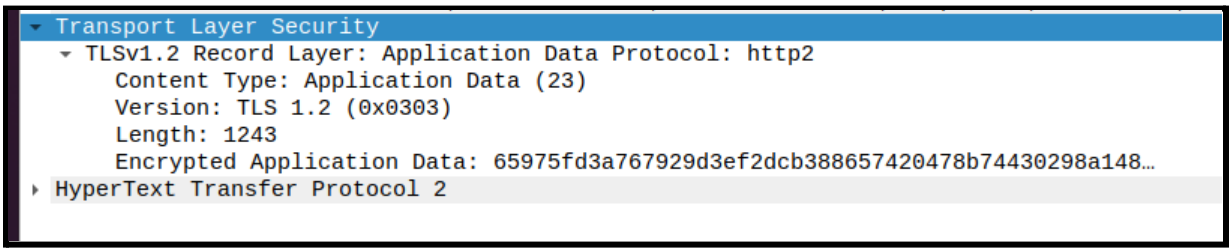


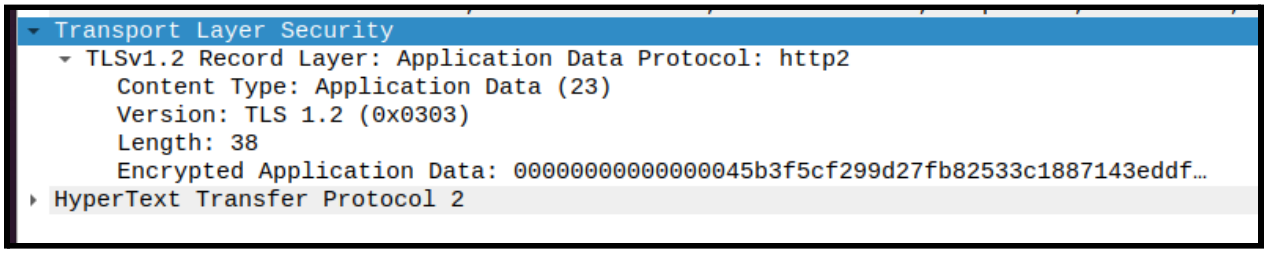

Transport Layer Security
TLSv1.2 Record Layer: Handshake Protocol: Certificate
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 2884
Handshake Protocol: Certificate
Transport Layer Security
TLSv1.2 Record Layer: Handshake Protocol: Certificate Status
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 479
Handshake Protocol: Certificate Status
TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 333
Handshake Protocol: Server Key Exchange
TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 4
Handshake Protocol: Server Hello Done

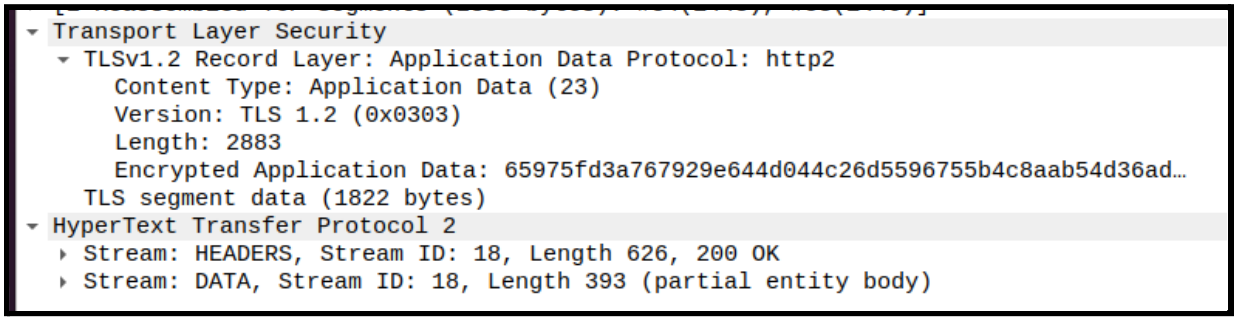

TLSv1.2 Record Layer: Handshake Protocol: Client Key Exchange
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 70
Handshake Protocol: Client Key Exchange
TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
Content Type: Change Cipher Spec (20)
Version: TLS 1.2 (0x0303)
Length: 1
Change Cipher Spec Message
TLSv1.2 Record Layer: Handshake Protocol: Finished
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 40
Handshake Protocol: Finished

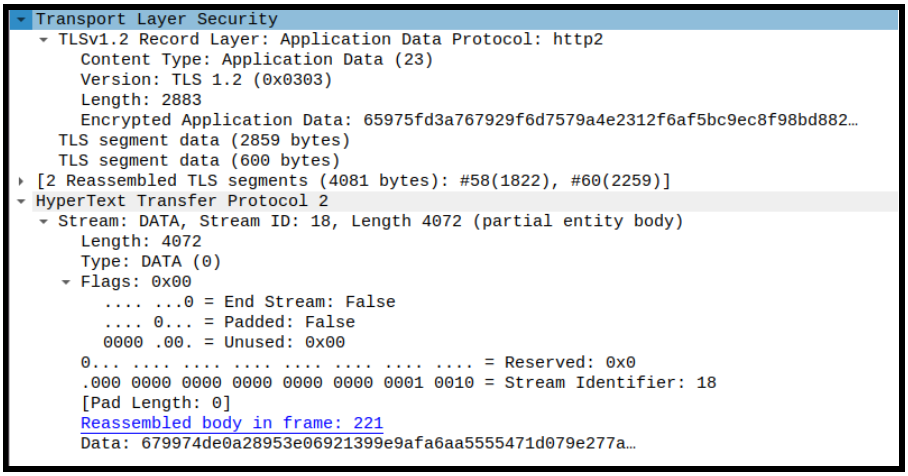

TLSv1.2 Record Layer: Application Data Protocol: http2
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 100
Encrypted Application Data: 00000000000000001493850bcdfccd3ee6a33f054d541a56e…
TLSv1.2 Record Layer: Application Data Protocol: http2
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 1041
Encrypted Application Data: 0000000000000002de6d15785f561fd263d4ddac823bdf82…
HyperText Transfer Protocol 2
HyperText Transfer Protocol 2


Transport Layer Security
TLSv1.2 Record Layer: Handshake Protocol: New Session Ticket
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 202
Handshake Protocol: New Session Ticket
TLSv1.2 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
TLSv1.2 Record Layer: Handshake Protocol: Finished
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 40
Handshake Protocol: Finished


Transport Layer Security
TLSv1.2 Record Layer: Application Data Protocol: http2
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 1243
Encrypted Application Data: 65975fd3a767929d3ef2dcb388657420478b74430298a148…
HyperText Transfer Protocol 2

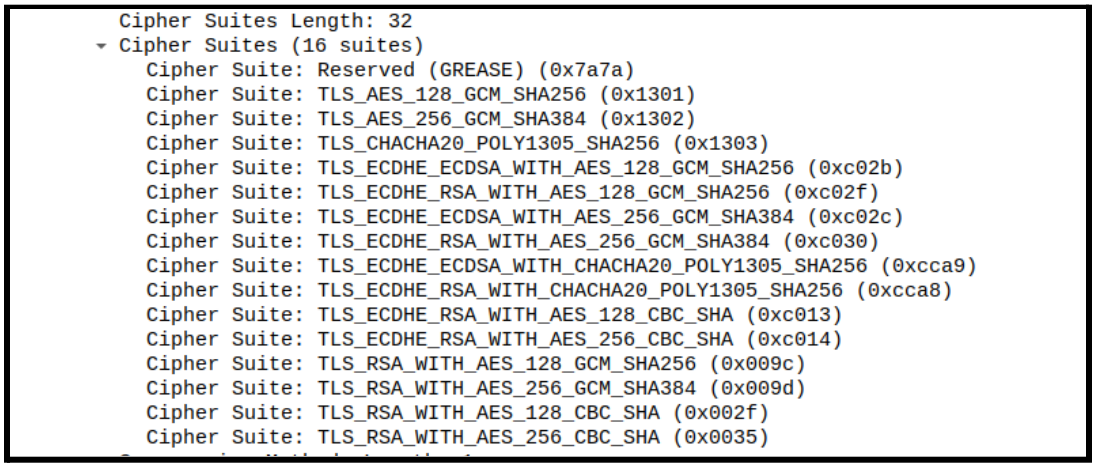

Transport Layer Security
TLSv1.2 Record Layer: Application Data Protocol: http2
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 38
Encrypted Application Data: 00000000000000045b3f5cf299d27fb82533c1887143eddf…
HyperText Transfer Protocol 2


Transport Layer Security
TLSv1.2 Record Layer: Application Data Protocol: http2
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 2883
Encrypted Application Data: 65975fd3a767929e644d044c26d5596755b4c8aab54d36ad…
TLS segment data (1822 bytes)
HyperText Transfer Protocol 2
Stream: HEADERS, Stream ID: 18, Length 626, 200 OK
Stream: DATA, Stream ID: 18, Length 393 (partial entity body)



Question 4

Cipher suites offered by the client are :



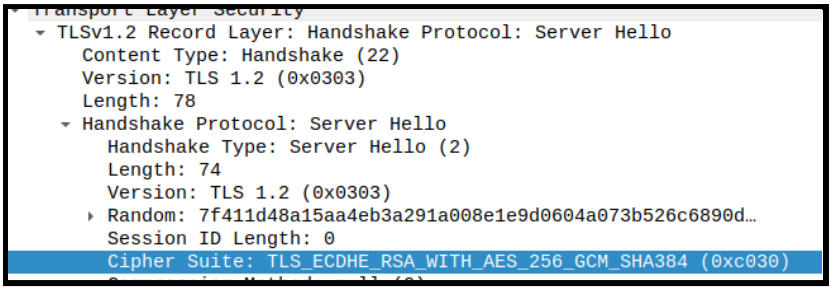
The first two cipher suites offered by client are :

1. Reserved (GREASE) (0x7a7a)
2. TLS_AES_128_GCM_SHA256 (0x1301)

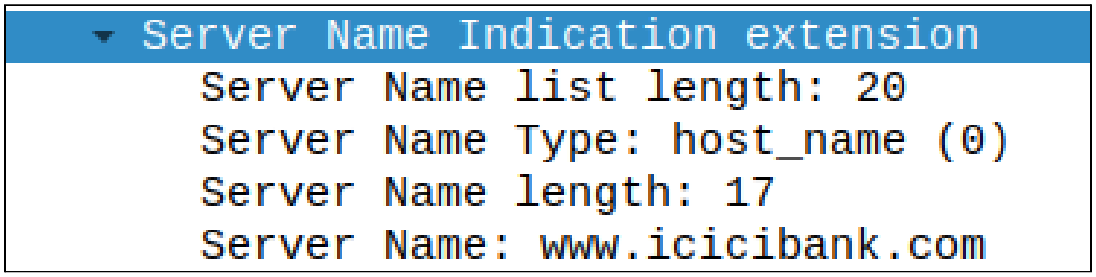
The last one is TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)

The cipher suite selected by the server is :

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)



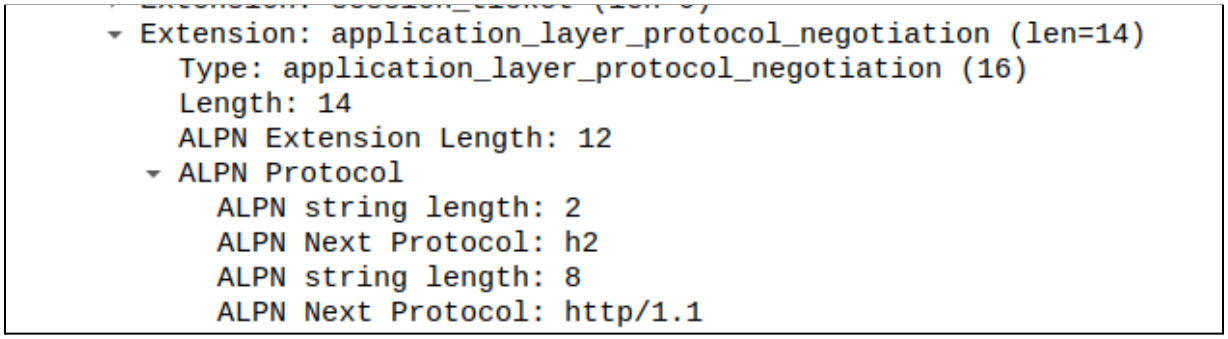
Question5



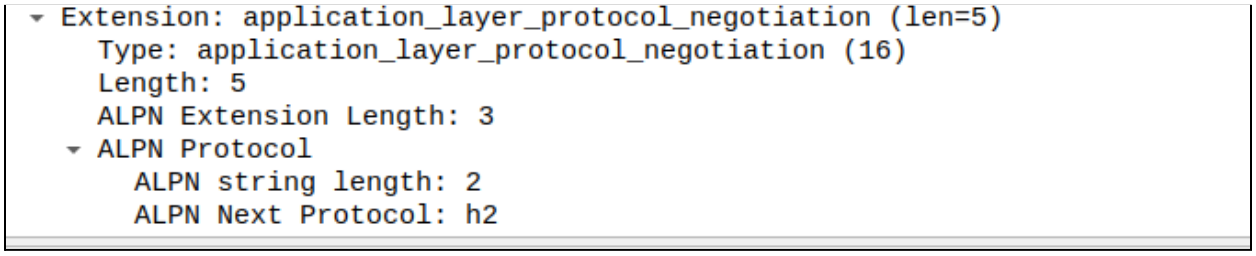
Server name identification is used to host several TLS certificates from different sites on a single IP address.

Question 6

Client Hello



Server Hello

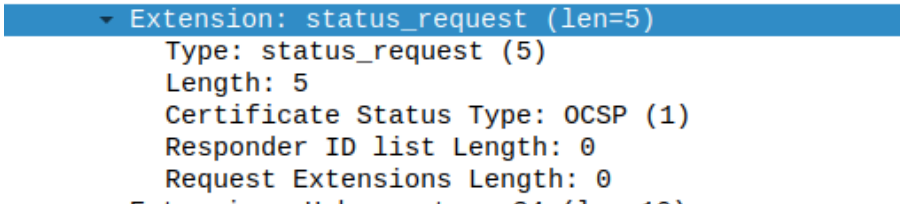


ALPN, which is independent of application-layer protocols, allows the application layer to negotiate which protocol should be implemented through a secure connection while eliminating needless round trips.

The server has selected h2 as shown in the above image .

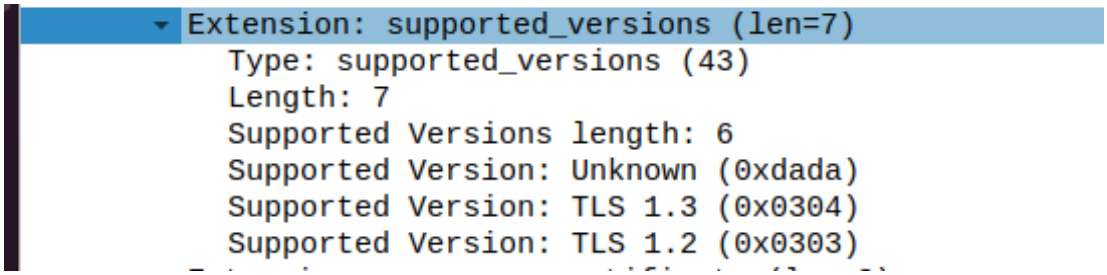
Question 7

status_request- This actually means the certificate status request which the client asks for the server, so that when the server responds back it will send the desired certificate status

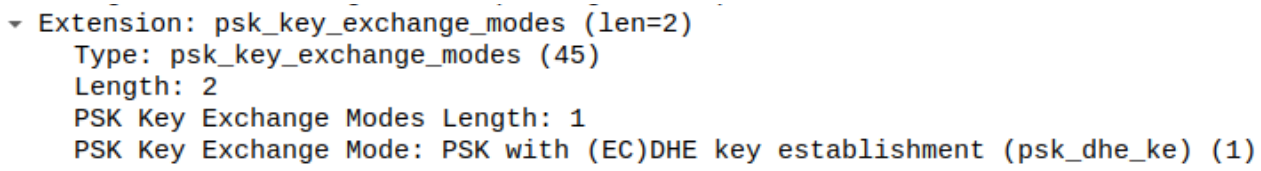


.

supported_versions- The "supported versions" extension is used by the client to specify which TLS versions it supports. Servers MUST NOT utilize the ClientHello.legacy version value for version negotiation and MUST ONLY use the "supported versions" extension to determine client preferences if this extension is present in the ClientHello. Servers MUST ONLY USE the version of TLS that is available in that extension and MUST DISREGARD any unknown versions that are present in that extension.

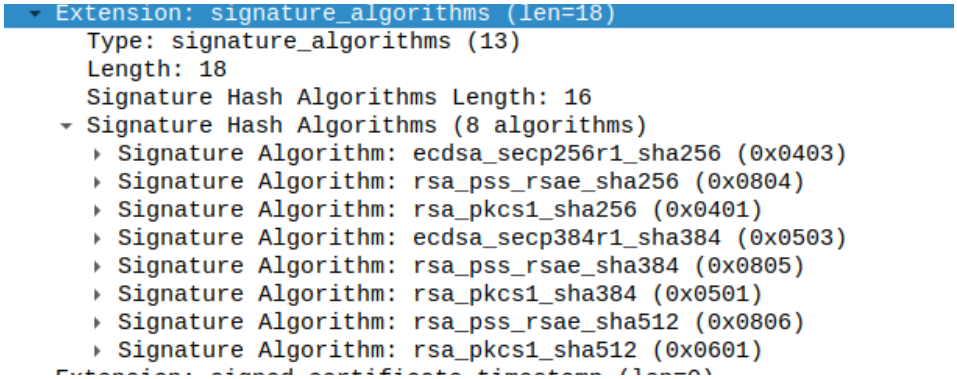


psk_key_exchange_modes - Servers MUST terminate the handshake if clients supply "pre shared key" without a "psk key exchange modes" extension. Servers must not use a key exchange mechanism not supported by the client. This modification also limits the modes that may be used with PSK resumption. Servers should not send new session tickets with tickets that are incompatible with the advertised modes



Question8

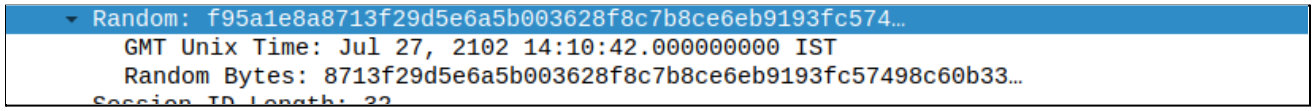
The client hello does contain “signature_algorithms” in its record.The purpose of it is to tell the server which sighash algorithm pairs may be utilized in the SSL handshake messages.



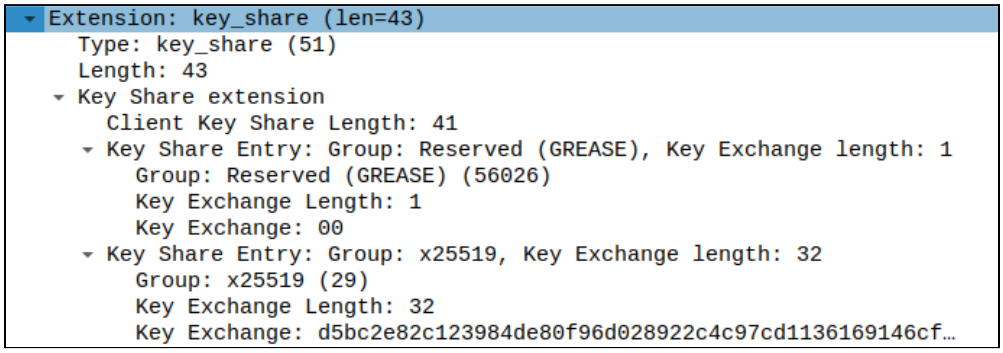
Question 9

Random - 32 bytes generated by a secure random number generator. TLS uses random values:

1. in public protocol fields such as the public Random values in the ClientHello and ServerHello.
2. To generate keying material

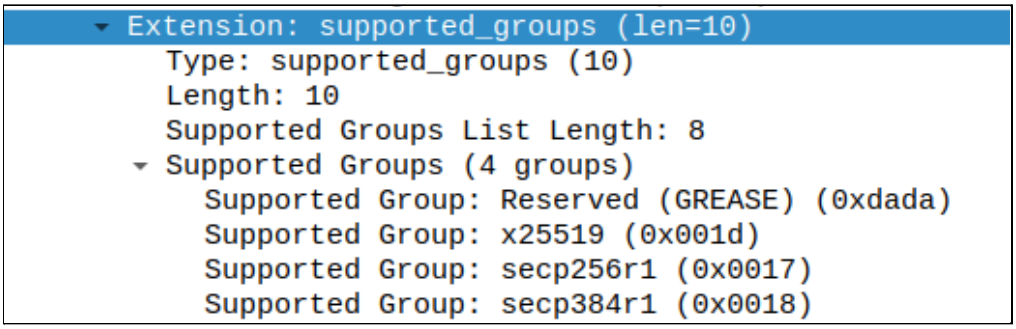


Key share - The cryptographic parameters of the endpoint are stored in the "key share" extension. Clients MAY send an empty client shares vector to the server in order to request group selection, at the penalty of an extra round trip.

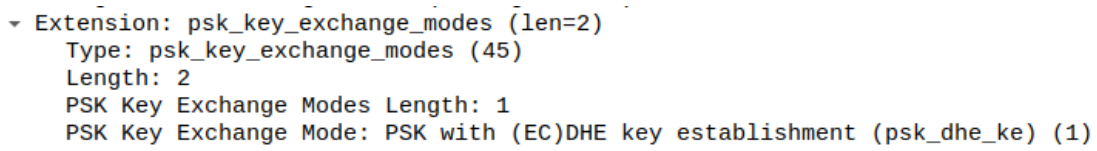


Supported groups - When sent by the client, the "supported_groups" extension indicates

the named groups which the client supports for key exchange.If the server prefers a different group than those in the "key share" extension but is still ready to accept the ClientHello, it SHOULD send "supported groups" to update the client's view of its preferences.

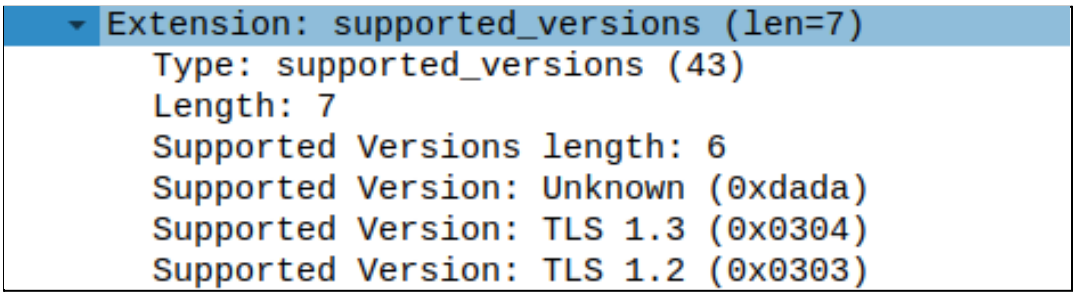


PSK - The semantics of this addition are that the client only permits the usage of PSKs with certain modes, which limits the use of both the PSKs presented in this ClientHello and those supplied by the server via NewSessionTicket.

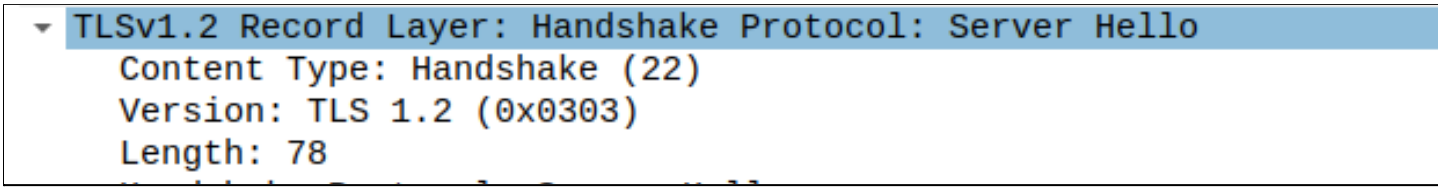


Question 10

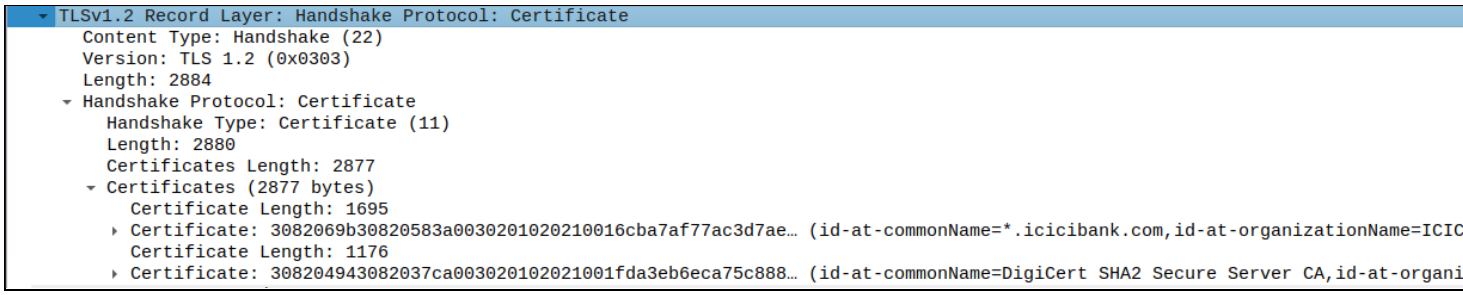
The browser/TLS versions supported by the client are shown in the below snapshot



The version selected by the server is shown in the below snapshot



Question 11



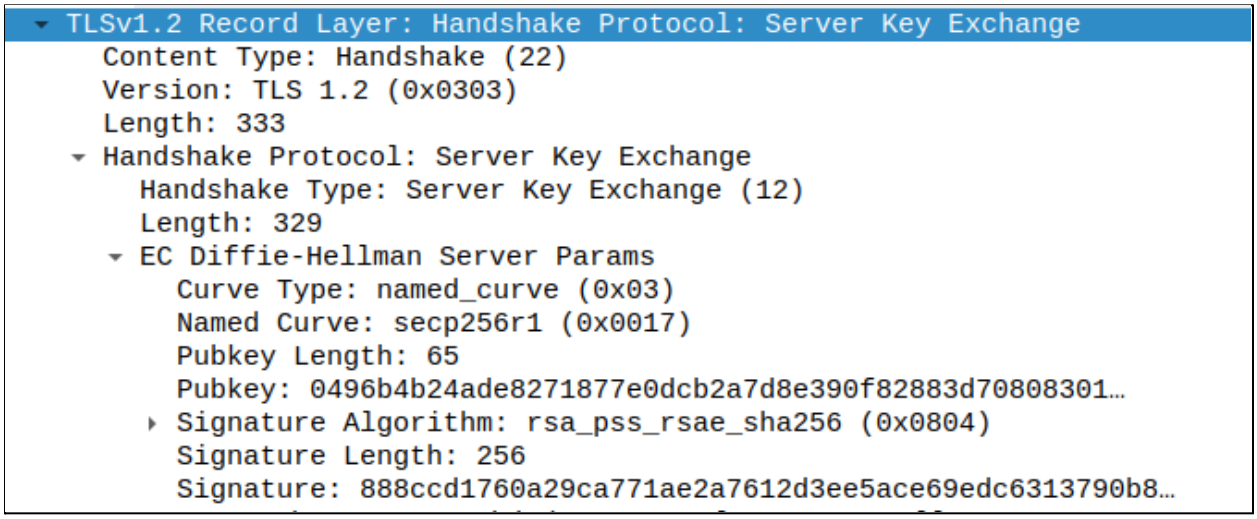
As we can infer from the above snapshot, the server has returned two certificates to the client :

1. The server certificate (*.icicibank.com)
2. The intermediate CA certificate(DigiCert SHA2 Secure Server CA)

Server certificates are used to authenticate the identity of a server whereas the intermediate CA certificates are used to authenticate end user certificates. The relation is that the intermediate certificate connects the server certificate to the root certificate of the CA. They constitute the SSL chain of trust, which is an ordered list of certificates that allows the receiver (a web browser) to verify the sender (your secure server) and the CA are trustworthy.

Question 12

The key exchange algorithm that the client and server agreed upon is “Elliptic Curve Diffie–Hellman Key Exchange”.It is proven to provide perfect forward secrecy. The parameters that got exchanged are shown in the below snapshot:



Question 13

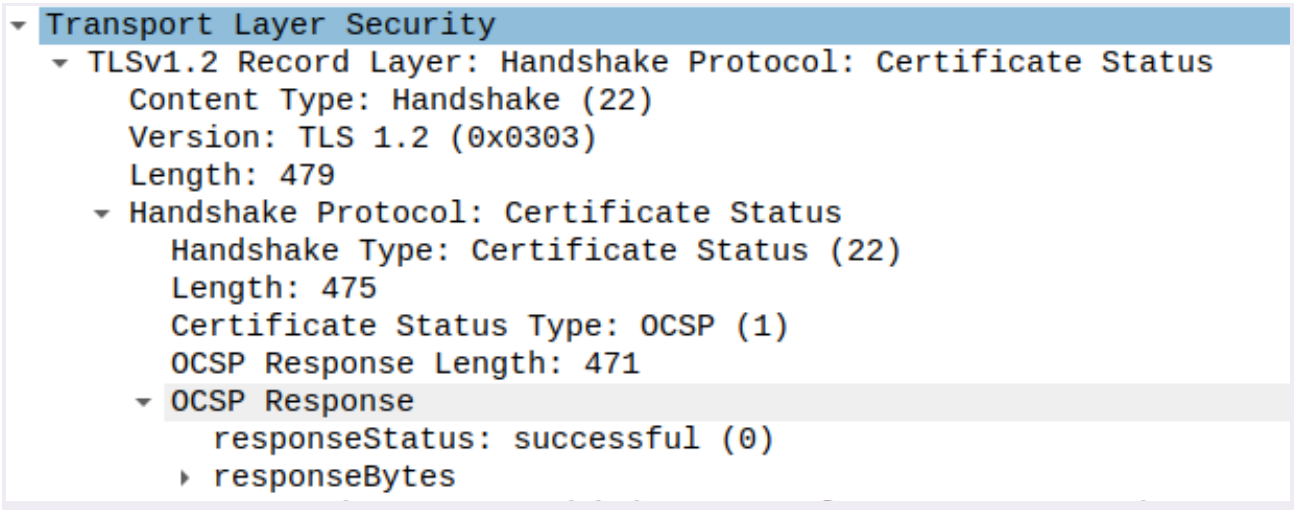
```
Certificate: 3082069b30820583a0030201020210016cba7af77ac3d7ae...
(id-at-commonName=*.icicibank.com,id-at-organizationName=ICICI
Bank
Limited,id-at-localityName=Mumbai,id-at-stateOrProvinceName=Maha
rashtra,id-at-countryName=IN)
```

From the above information we can conclude that the certificate is of the type **OV** since it is providing detailed information .

Question 14

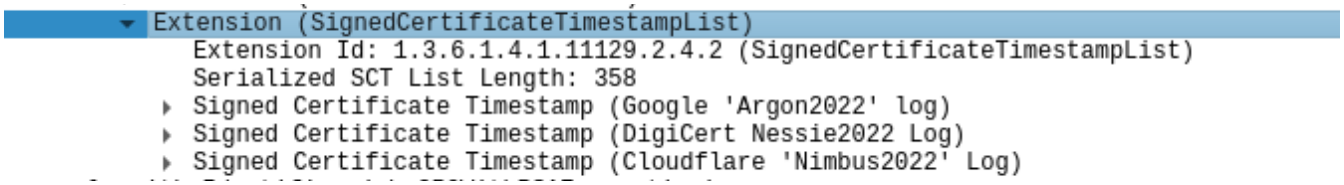
`id-at-commonName=*.icicibank.com` Indicates that the bank is using a wild-card type certificate.

Question 15



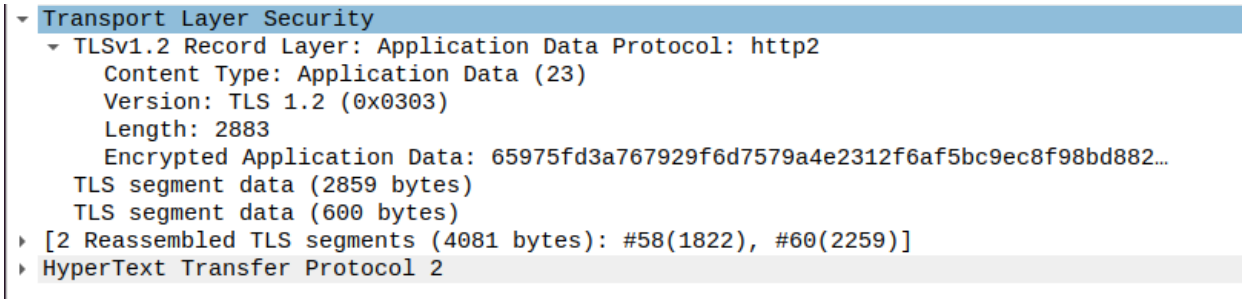
Using the OCSP response status, the client may determine whether or not the certificate has been revoked. From the above snapshot, we can see that the OCSP response status is successful indicating the certificate has been revoked.
The server supports the OCSP stapling.

Question 16



As we can see from the above snapshot, in the extension SCT three servers have logged the certificate of the bank. With the help of these log servers, it gives the freedom to query the servers about the certificates and also the timestamp they were included at.

Question 17



The application data is encrypted using symmetric key encryption, with the encryption key being the master key that can only be retrieved once.
Yes, the records containing application data include a separate MAC.
Wireshark doesn't distinguish between the encrypted application data and the MAC.

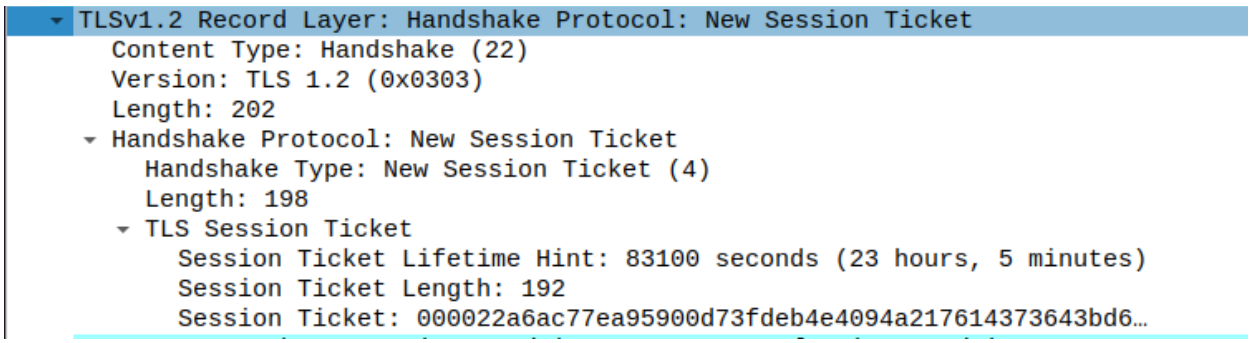
Question 18

1. CLIENT_HANDSHAKE_TRAFFIC_SECRET - The hex-encoded handshake traffic secret for the client side. HANDSHAKE SECRET and ECDH parameters were used to create this value.
2. CLIENT_RANDOM - 48 bytes for the master secret, encoded as 96 hexadecimal characters.A random function was used to choose.
3. CLIENT_TRAFFIC_SECRET_0 - The first hex-encoded application traffic secret for the client side. This is derived from MASTER_SECRET.
4. SERVER_TRAFFIC_SECRET_0 - The first hex-encoded application traffic secret for the server side.This is derived from MASTER_SECRET.

5. EXPORTER_SECRET - The hex-encoded exporter secret.This is derived from MASTER_SECRET.
6. SERVER_HANDSHAKE_TRAFFIC_SECRET - The hex-encoded handshake traffic secret for the server side. HANDSHAKE SECRET and ECDH parameters were used to create this value.

Question 19

There is a scope for session resumption since the server is keeping track of the recently used session ID's



From the above snapshot we can see that it is based on session ID/session tickets and not on psk.
The role of session tickets in TLS 1.3 is that ,when a client reconnects to server with a session ID, the server can quickly look up the session keys and resume the encrypted communication.

Question 20

To establish a secure connection, a full TLS handshake needs two round trips.

We can achieve it with just one round trip with the help of TLS session resumption i.e half of the number of round trips can be reduced.

Question 21

Wireshark · All Addresses · CS19BTECH11012.pcapng

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
All Addresses	3189				0.1380	100%	2.8300	3.262
74.125.68.154	47				0.0020	1.47%	0.2300	4.643
65.2.91.176	210				0.0091	6.59%	1.5800	3.271
52.172.161.242	21				0.0009	0.66%	0.1900	4.192
49.44.178.83	14				0.0006	0.44%	0.1000	5.779
49.44.178.66	14				0.0006	0.44%	0.1000	6.139
49.44.119.163	29				0.0013	0.91%	0.1300	6.018
40.69.200.41	39				0.0017	1.22%	0.1000	7.671
34.96.102.137	165				0.0071	5.17%	0.4500	2.691
255.255.255.255	4				0.0002	0.13%	0.0300	8.667
239.255.255.250	43				0.0019	1.35%	0.0200	1.001
224.0.0.251	24				0.0010	0.75%	0.0400	0.371
224.0.0.22	2				0.0001	0.06%	0.0100	2.419
224.0.0.2	12				0.0005	0.38%	0.0400	4.160
216.58.200.141	6				0.0003	0.19%	0.0600	16.631
216.58.200.132	63				0.0027	1.98%	0.1900	7.825
216.239.36.54	54				0.0023	1.69%	0.2300	4.053
203.189.92.162	134				0.0058	4.20%	0.4000	15.398
203.171.210.25	16				0.0007	0.50%	0.1500	1.705
192.168.36.53	56				0.0024	1.76%	0.0600	2.641
192.168.115.48	9				0.0004	0.28%	0.0200	12.254
192.168.115.255	2				0.0001	0.06%	0.0100	2.830
192.168.115.157	3				0.0001	0.09%	0.0100	7.967
192.168.114.91	2				0.0001	0.06%	0.0100	22.899
192.168.114.86	4				0.0002	0.13%	0.0100	7.438
192.168.114.84	3				0.0001	0.09%	0.0100	7.335
192.168.114.83	3				0.0001	0.09%	0.0100	17.373
192.168.114.75	1				0.0000	0.03%	0.0100	22.185
192.168.114.72	4				0.0002	0.13%	0.0100	0.270
192.168.114.49	4				0.0002	0.13%	0.0400	8.667
192.168.114.44	2				0.0001	0.06%	0.0100	7.232
192.168.114.43	10				0.0004	0.31%	0.0200	0.576
192.168.114.41	2				0.0001	0.06%	0.0100	2.419
192.168.114.23	2				0.0001	0.06%	0.0100	4.161
192.168.114.150	11				0.0005	0.34%	0.0200	3.342
192.168.114.15	2				0.0001	0.06%	0.0100	4.162
192.168.114.146	3				0.0001	0.09%	0.0100	15.938
192.168.114.13	2				0.0001	0.06%	0.0100	7.233

Display filter:

.

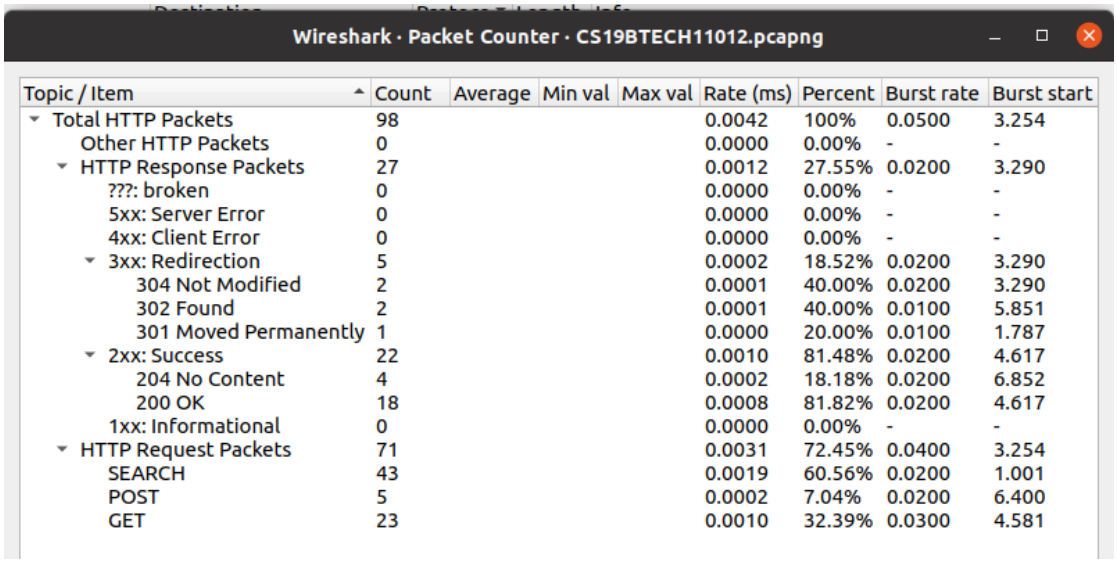
Duration of the HTTPS session is 20 secs and the total number of packets exchanged is 3189 packets.

Question 22

The number of packets exchanged over the TLS pipeline is 1240 .

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End By
Frame	100.0	1184	100.0	2613699	1,257 k	0	0
Ethernet	100.0	1184	0.6	16576	7,976	0	0
Internet Protocol Version 4	100.0	1184	0.9	23680	11 k	0	0
Transmission Control Protocol	100.0	1184	98.5	2573443	1,238 k	0	0
Transport Layer Security	104.7	1240	161.7	4225442	2,033 k	360	61584

Question 23



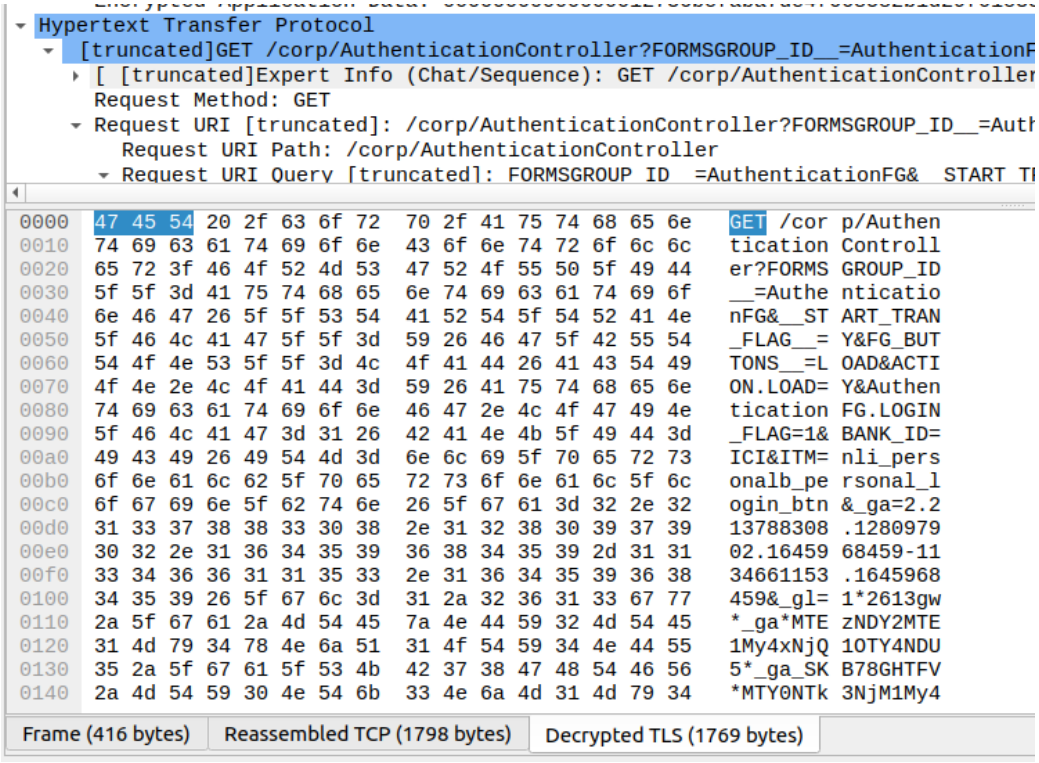
Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Total HTTP Packets	98				0.0042	100%	0.0500	3.254
Other HTTP Packets	0				0.0000	0.00%	-	-
HTTP Response Packets	27				0.0012	27.55%	0.0200	3.290
???: broken	0				0.0000	0.00%	-	-
5xx: Server Error	0				0.0000	0.00%	-	-
4xx: Client Error	0				0.0000	0.00%	-	-
3xx: Redirection	5				0.0002	18.52%	0.0200	3.290
304 Not Modified	2				0.0001	40.00%	0.0200	3.290
302 Found	2				0.0001	40.00%	0.0100	5.851
301 Moved Permanently	1				0.0000	20.00%	0.0100	1.787
2xx: Success	22				0.0010	81.48%	0.0200	4.617
204 No Content	4				0.0002	18.18%	0.0200	6.852
200 OK	18				0.0008	81.82%	0.0200	4.617
1xx: Informational	0				0.0000	0.00%	-	-
HTTP Request Packets	71				0.0031	72.45%	0.0400	3.254
SEARCH	43				0.0019	60.56%	0.0200	1.001
POST	5				0.0002	7.04%	0.0200	6.400
GET	23				0.0010	32.39%	0.0300	4.581

From the above snapshot, we can observe that the total number of HTTP packets is 98 out of which :

1. HTTP Request - 23 packets
2. HTTP response - 5 packets

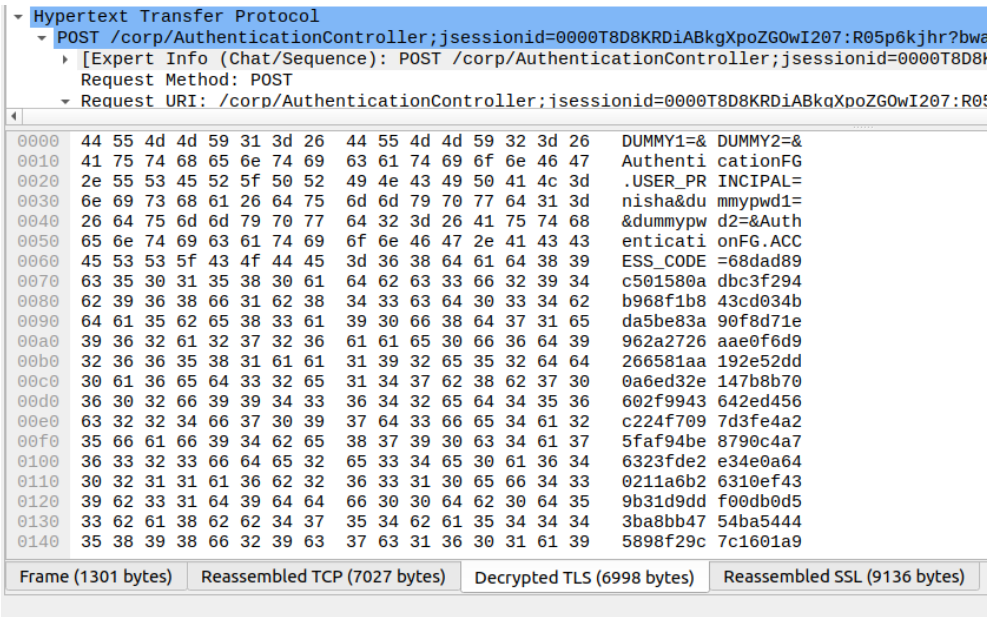
The packet that carried the response that included the Net Banking LOG-IN page of ICICI bank is shown below:

```
HTTP/1.1 200 OK
Date: Sun, 27 Feb 2022 15:39:15 GMT
Strict-Transport-Security: max-age=31536000; includeSubdomains; preload
X-Powered-By:
VIEW_ID: AuthenticationScreen
Cache-Control: no-store
Pragma: no-cache
Expires: 0
Title: Log in to Internet Banking
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Vary: Accept-Encoding
Content-Encoding: gzip
Keep-Alive: timeout=10, max=98
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html;charset=UTF-8
Content-Language: en-US
Set-Cookie: NSC_jogjojuz.jdjdjcbol.dpn_64.224=ffffffffaf181aa745525d5f4f58
GMT;path=/;secure;httponly
```



The response messages do carry security related directives like XSS, sameorigin, HSTS.

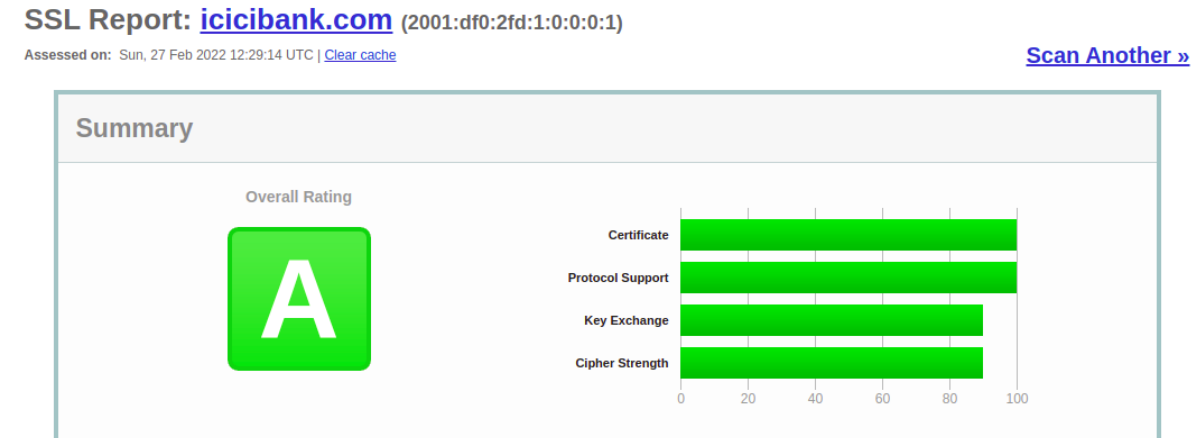
Question 24



Both the user id and password in the raw packet capture are shown in the above and below snapshots.

```
HTML Form URL Encoded: application/x-www-form-urlencoded
  Form item: "DUMMY1" = ""
  Form item: "DUMMY2" = ""
  Form item: "AuthenticationFG.USER_PRINCIPAL" = "nisha"
  Form item: "dummypwd1" = ""
  Form item: "dummypwd2" = ""
  Form item: "AuthenticationFG.ACCESS_CODE" = "68dad89c501580adbc3f294b968f1b843cd034bda5be83a90f8d71e962a2726aae0f6d9266581aa192e52dd0a6ed3
  Form item: "dummy1" = "4c3298a431733563b6db83f2b328a3c65ac52e0829f86c76759e7bb8d3a700430d1c7705fb6f8a482155a360aaaee81accaf1f1e0539693ad48
  Form item: "AuthenticationFG.MENU_ID" = "RDASH"
  Form item: "Action.VALIDATE_CREDENTIALS" = "Login"
  Form item: "AuthenticationFG.RIB_COUNTRY_CODE" = "91"
  Form item: "AuthenticationFG.RIB_LOGIN_MOBILE" = ""
  Form item: "FG_BUTTONS__" = "VALIDATE_CREDENTIALS,VALIDATE_CREDENTIALS_DIG_CERT,BACK,CLEAR_VALUES"
  Form item: "AuthenticationFG.IS_FIRST_AUTHENTICATION" = "Y"
  Form item: "QS" = ""
  Form item: "MD" = ""
  Form item: "PID" = ""
  Form item: "PRN" = ""
  Form item: "ITC" = ""
  Form item: "AMT" = ""
  Form item: "CRN" = ""
  Form item: "RU" = ""
  Form item: "CG" = ""
  Form item: "ES" = ""
  Form item: "RUBACK" = "null"
  Form item: "AuthenticationFG.LOGIN_MODE_FLAG" = "U"
  Form item: "AuthenticationFG.__CALL_MODE__" = "null"
  Form item: "CATEGORY_ID" = ""
  Form item: "AuthenticationFG.PREFERRED_LANGUAGE" = "001"
  Form item: "FORMSGROUP_ID__" = "AuthenticationFG"
```

Question 25



The security features implemented by the bank's web server include :

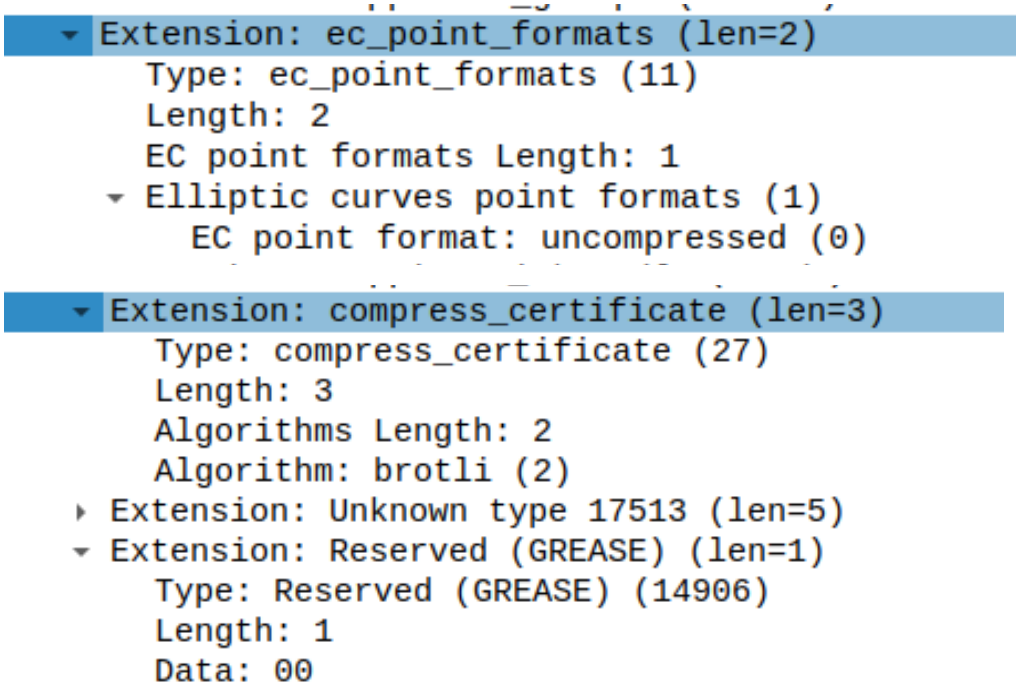
1. Providing perfect forward secrecy by using the below mention cipher suites as per TLSv1.2

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)

2. OCSP stapling
3. Secure Renegotiation
4. No ECDH public server param reuse

Question 26



- The purpose of sending the extension is to notify the peer whether you support compressed points, as support for uncompressed points is required (if you support EC ciphersuites and certs at all).

  If you don't support compression and don't send the extension, there's a chance the peer will send you a compressed-point cert/chain and/or ephemeral key, which you won't be able to process, and the handshake will fail, even if the peer could have used an uncompressed-point cert/chain and/or key or some other negotiable ciphersuite that would have worked.
- The sender communicates to the peer the certificate-compression techniques it is willing to apply for decompression by sending a compress certificate extension. This extension's "extension data" field MUST have the value CertificateCompressionAlgorithms.

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 plagiarized 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 honor violations by other students if I become aware of it.

Name: Nisha M
Date: 27/2/2022
Signature: Nisha M