Networking: TCP and TLS

Question 1 **(min)**

Q1.1 TRUE or FALSE: TLS has end-to-end security, so it is secure against an attacker who steals the private key of the server.

○ TRUE ○ FALSE

Q1.2 TRUE or FALSE: By default, in a TLS connection, both the server and client are authenticated to each other.

○ TRUE ○ FALSE

Q1.3 TRUE or FALSE: If the server's random number a in Diffie-Hellman TLS is the same in every handshake, Diffie-Hellman TLS no longer has forward secrecy. Assume the value a is stored on the server along with its secret key.

○ TRUE ○ FALSE

Q1.4 TRUE or FALSE: Randomizing the client port helps defend TCP against on-path attackers.

○ TRUE ○ FALSE

Q1.5 TRUE or FALSE: TLS provides end-to-end security, so it is secure even if the server has a buffer overflow vulnerability.

○ TRUE ○ FALSE

Q1.6 TRUE or FALSE: Suppose we modified TCP so that the sequence number increases by 2 for every byte sent, but the initial sequence numbers are still randomly chosen. This modified protocol has the same security guarantees as standard TCP.

○ TRUE ○ FALSE

Q1.7 TRUE or FALSE: Consider a modified version of DHCP, where in the server offer step, the server signs its message and sends its public key along with the signed message. This version of DHCP is secure against the DHCP spoofing attack.

○ TRUE ○ FALSE

Q1.8 TRUE or FALSE: TCP is secure against a DoS attack by a man-in-the-middle (MITM) because TCP guarantees delivery and will re-send messages until they are delivered.

- ○ TRUE
- ○ FALSE

Q1.9 TRUE or FALSE: RSA-TLS is still secure if we use publically known lottery numbers as the value of the premaster secret (PS).

- ○ TRUE
- ○ FALSE

Question 2 **(15 min)**

Q2.1 Alice clears all her network settings and broadcasts a DHCP discover message. What information should she expect to receive in the DHCP offer in response?

- ☐ (A) DNS server
- ☐ (B) Source port
- ☐ (C) Lease time
- ☐ (D) Premaster secret
- ☐ (E) Gateway router
- ☐ (F) IP address

Q2.2 After receiving the DHCP offer, Alice tries connecting to `www.cutecats.com`, but instead of pictures of cats, the site she gets is filled with dog photos.

How did the attacker compromise DHCP to accomplish this?

Which of the following could the attacker have replaced?

- ☐ (G) DNS server
- ☐ (H) Source port
- ☐ (I) Lease time
- ☐ (J) Premaster secret
- ☐ (K) Gateway router
- ☐ (L) IP address

Q2.3 Alice clears all her network settings and starts a new connection to `www.cutecats.com` with TCP. Now an off-path attacker wants to send a packet to the server to interfere with Alice's connection. What information do they need to know?

☐ (A) Server sequence number

☐ (B) Source port

☐ (C) Client sequence number

☐ (D) Destination IP address

☐ (E) Destination port

☐ (F) Source IP address

Q2.4 At some point, Alice's connection with `www.cutecats.com` is suddenly terminated. Assuming some information was leaked and the attacker correctly guessed the fields from the previous part, how was the attacker able to execute this attack?

○ (G) — ○ (H) — ○ (I) — ○ (J) — ○ (K) — ○ (L) —

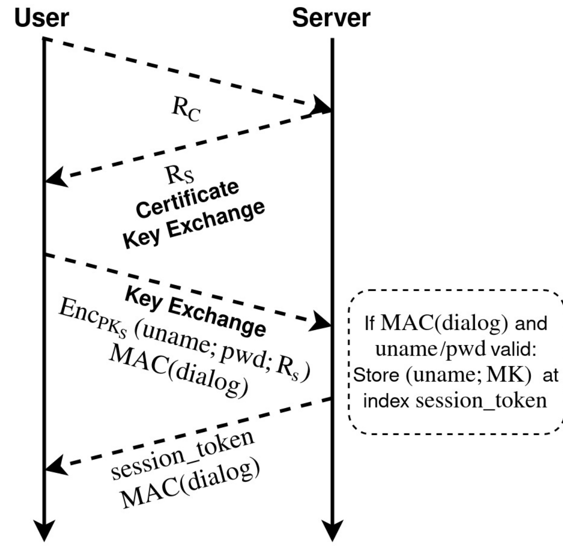
Question 3 **(37 min)**

FastCash is a fast banking service which requires users to log in before making a transfer, and uses TLS with ephemeral Diffie Hellman and RSA certificates to secure all their connections. They implemented a TLS extension called *0-Round Trip (0-RTT)* to speed up the connection process. 0-RTT changes the initial handshake as follows:

- Users authenticate themselves during the second round of the handshake
- If the user authenticates correctly, the server stores a session_token for that user

(Recall that in TLS, PS, R_S*, and* R_C *generate a master key set* MK *which contains all the symmetric keys.* Enc_{PK_S} *denotes RSA encryption using the server's public RSA key.)*

A user only needs to perform the modified TLS handshake once. To send an HTTP request after the initial connection ends, a user encrypts it using the keys derived in the initial handshake and attaches the session_token. The server verifies that the entry session_token : (uname, MK) exists and, if so, decrypts and executes the request as the user uname using the keys derived from MK.



Simplified diagram of modified initial TLS handshake

Assume that any on-path TCP injection attacks are impossible, and that once a user makes the initial modified TLS handshake, they will use the 0-RTT extension for future requests to the server.

Q3.1 An on-path attacker observes an initial TLS handshake between a user and server, as well as a subsequent 0-RTT packet which contains an encrypted HTTP request. What can they do?

☐ (A) Read the user's future communications

☐ (B) Pretend to be the server to the user

☐ (C) Pretend to be the user to the server in a new handshake

☐ (D) Replay the encrypted HTTP request to the server

☐ (E) Learn the master key set

☐ (F) None of the above

Q3.2 Suppose we removed R_S from the user's KeyExchange in the third step of the handshake. After observing an initial handshake between a user and the server, what can an on-path adversary do?

☐ (G) Read the user's future communications

☐ (H) Pretend to be the server to the user

☐ (I) Pretend to be the user to the server in a new handshake

☐ (J) Learn the premaster secret

☐ (K) Learn the master key set

☐ (L) None of the above

Q3.3 Due to a bug, an on-path adversary is able to choose the server's R_S. After observing an initial handshake between a user and the server, what can they do?

☐ (A) Read the user's future communications

☐ (B) Pretend to be the server to the user

☐ (C) Pretend to be the user to the server in a new handshake

☐ (D) Learn the premaster secret

☐ (E) Learn the master key set

☐ (F) None of the above

Q3.4 An on-path adversary observes a user and the server communicating using 0-RTT for some time (without observing the initial handshake). At some point in the future, the adversary manages to learn all of the server's session_token : (uname, MK) entries. What can they do?

☐ (G) Read the user's future communications

☐ (H) Pretend to be the server to the user

☐ (I) Pretend to be the user to the server in a new handshake

☐ (J) Learn the premaster secret

☐ (K) Learn the master key set

☐ (L) None of the above

Q3.5 Consider a MITM adversary during the initial handshake between a user and the server. Describe how this adversary can send a malicious HTTP request that appears to come from the legitimate user (Be specific with what is sent). Disregard any bugs from previous parts.