Q1

- **Brute-Force Cryptanalytic Attack**

SSL uses RC2 or RC4 as stream ciphers, both with keys at 128-bit size, which is computationally infeasible to brute force the whole key space

It also uses various block cipher algorithms with keys at 128,168 or 256 bit which are virtually impossible to crack.

- **Known Plaintext Dictionary Attack**

The same strategy used to neutralize the previous attack is used to defeat this one. Also, SSL uses random numbers to generate keys, so this attack will be neutralized.

- **Replay Attack**

The randomized parameter “Nonces” and sequence number defend against replay attacks.

- **Man-in-the-Middle Attack**

MITM attacks are countered by authentication based on digital certificates and shared secret keys.

- **Password Sniffing**

Sniffing is impossible because of message encryption

- **IP Spoofing**

SSL can’t defend against this attack because peers are not authenticated based on IP.

- **IP Hijacking**

If the attacker hijacks the connection after authentication, he has no way of knowing the encryption key.

- **SYN Flooding**

the SSL/TLS protocol starts only after a successful TCP handshake, Therefore SSL/TLS does not help against SYN flooding, it sits on top of TCP.

Q2

SSL relies on an underlying reliable protocol (TCP) to assure that bytes are not lost or inserted or out of order, so basically receiver will always get the record blocks in correct order, BUT it's not done in SSL itself, but a layer below it.

Q3

- **Server Guarantees:**

Secure channel: Similar to the client, following the handshake based on session keys, the server is assured of having a secure channel for communication.

- **Client Guarantee:**

Communicating with authentic server: The client has the guarantee that it is talking to a legit server and not a malicious third person. This is because the client validated the server's certificate which contains the server's public key signed by a trusted CA.

Secure channel: Following the handshake, the client is guaranteed a secure encrypted channel for communication with the server. Data integrity, confidentiality, and authentication are provided via this channel. This is true because public key cryptography and verified identities are used to generate session keys securely during the handshake for encryption and integrity check.

Example scenario

Although many widely used web applications need secure access via SSL/TLS, in order for clients to access sensitive data or transmit data, the server must first establish trust. Imagine this: as a user, you visit YouTube.com and enter your personal information (username and password, for example) you must be sure beforehand that this is the real YouTube.

Q4

Comparison:

scenario A, client A needs to contact KDC to get K_S (shared key) in 2 encrypted forms, one with A's key and other with B's. then decrypt the one for itself and sends the other for client B, client B decrypts K_S, and the authentication step begins.

But in scenario B, client A contacts B directly, B gets K_S from KDC in encrypted forms (with A's and B's keys) decrypt the first one for itself and sends the other to A.

Merits and Demerits:

In scenario A, the heavy work is on KDC itself but in the second scenario, client B has to do all the dirty work which is not good (KDC is the server not client B!)

Scenario A takes more time and needs more steps to perform authentication, but in the second scenario server includes ID_A with N_A (nonce) and ID_B with N_A which authenticates each peer for the other one.

Q5

a client using a CRL doesn't really know if there's a new entry. All it does is get the new list periodically, if there's a new entry then good, if there isn't then they check again later.

The primary implementation of CRLs is not "go get one when you need it" but "get it ahead of time, and check it when you need to". Which means the primary mechanism of distribution is a series of file downloads and caches

Q6

PKI is used to digitally sign documents transactions, and software to prove the source as well as the integrity of those materials, so the hashes of the certificates are needed to check the integrity of them.