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

Recall the topics covered on role of nonces (one each from Alice and Bob) and sequence number counters (one each for Alice and Bob) and answer the following queries in crisp (5-10 lines). Alice is a web browser whereas Bob is a web server with Digital Certificate signed by a root CA using RSA. Also assume that RSA is used for key exchange between Alice and Bob.

Q1. Assume that TLS does not use any nonces. Explain how Trudy can be successful in launching session/connection replay attacks by capturing all the messages exchanged between Alice and Bob a while ago. You can assume Alice and Bob used TLS for securing their communication related to say ordering an item for e-commerce, online payments, secure file transfers, etc. Can Trudy replay Alice's previous messages with Bob for successfully launching a session/connection relay attack with Bob? Explain your answer. Can Trudy replay Bob's previous messages with Alice for successfully launching a session/connection replay attack? Explain your answer.

A: Yes Trudy can replay Alice’s previous messages with Bob . For example, let's assume Alice is a client who is ordering some product in an e-commerce store. He connects to a server (Bob) on the internet. So Trudy in order to launch a replay attack, should be able to order more products of the same type that Alice has ordered before. Initially Trudy doesn’t launch any attack but keeps reading these packets until the session is completed. As there are no nonces used and PRF is a known function, the master key and the key material are pre-determined. Trudy may simply replay the packets in the same order as the original session with this level of TLS protection. Despite the fact that Trudy is unable to decrypt the Premaster secret (which is encrypted with Bob's public key) or other encrypted packets, she can establish the connection by sending the packet containing the original session's Premaster secret and then replaying all of the packets in the same order to repeat the funds transfer session and hence replaying Alice’s previous messages with Bob.

But Trudy *cannot replay Bob's previous messages* with Alice. Since the PMS is sent by Alice, it would be different for this new session. So, the key material generated for this session would be different from the last session. Hence not allowing Trudy to replay Bob's previous messages with Alice.

Q2. Explain how nonces employed in TLS help in preventing session/connection replay attacks in Q1.
A: If we are generating nonces, i.e, Alice and Bob randomly generate their own nonces. The PRF will utilise the premaster secret and nonces to generate master key, and then nonces and master key will be used by the PRF to construct key material. Trudy now has to generate the exact identical key material in order to launch a replay session attack or replay the original session messages. As a result, Trudy must create the same nonces and pre-master key (since the PRF is known to everyone already). Trudy can send the identical premaster secret and nonce generated by her (client), but she has no control over Bob's nonce (Server). As a result, different key material is generated this time. Even when Trudy delivers the original session packets, the MAC authentication fails because the packets are encrypted with a different key than the one now provided. As a result, session replay attacks are prohibited.

Q3. How does Alice derive the PreMaster Secret (PMS) which she wants to send to Bob? Refer RFC 5246.
A: PMS is a randomly generated number whose method of creation depends upon the cipher suite agreed upon by Alice and Bob. If the RSA suite is used, Alice generates a 48-byte PreMaster Secret (46 bytes generated randomly and 2 bytes for the latest TLS version supported by the client), encrypts it with Bob's public key received from his digital certificate sent in the previous step. In the ECDH suite , $g^{ab} \, mod \, p$ will be utilised as the PMS. Alice gets $g^{b} \, mod \, p$ from bob and then he calculates the PMS, whereas bob calculates the shared secret by getting $g^{a} \, mod \, p$ from Alice.

Q4. Why can't Bob derive PMS and share it with Alice?
A: If Bob derives the PMS, he must carefully share it with Alice. Bob (server) needs Alice's (client) certificate, or more particularly, Alice's public key for RSA or ECDH, in order for Bob to encrypt the PMS with Alice's public key and safely deliver it to Alice. As a result, client authentication is always required in this type of TLS handshake. However, with our current systems, it is unlikely that all clients will have their own digital certificates. Client authentication is not necessary except in a few instances, hence server authentication is more important. Also, because Bob is the server, he will have a lot of connections, so creating a PMS and securely sharing it will be an overhead for him. Hence, Bob doesn't derive PMS in traditional TLS handshakes.

Q5. Think of a scenario in which it's possible for Bob to derive PMS and share it to Alice. Refer TLS 1.2 handshake message protocol and explain how it can be extended (say, by adding new messages) to achieve this behaviour.
A: Following the server hello and the server (Bob) sending its certificate to the client (Alice), Bob makes a request for the client's certificate, but the server key exchange has not yet been sent. Bob exchanges server keys after checking Alice's certificate (sends PMS to Alice). Note that server key exchange is now required in this new TLS handshake, whereas client key exchange is optional, and that server key exchange occurs after phase-3 (client certificate verification). The Master secrets and key material are

then generated by both Alice and Bob, and the next steps are identical to the original TLS 1.2v handshake.

Q6. Note that MS is derived by feeding PMS and nonces of Alice and Bob as inputs to a PRF (that is known to all) by both Alice and Bob independently. Similarly, MS and nonces of Alice and Bob, and key_block size are fed as inputs to a PRF to derive key material which are split into MAC keys, session keys and IVs (IVs for AES-CBC only) by both Alice and Bob independently. To lessen the burden(!) on Bob out of her love for Bob, Alice said that she would generate MS from PMS and nonces of Alice and Bob and directly share the MS to Bob by encrypting it with Bob's public key. Trudy captured messages exchanged between Alice and Bob in this modified handshake protocol. Do you think Trudy can succeed in launching session/connection replay attacks on Bob? Justify your answer.

A: Even if Alice exchanges MS instead of PMS, Trudy will not be able to launch a successful replay attack. Because the final key material is determined by MS and Alice and Bob's nonces, the key generated by Bob will differ from the prior session key that Trudy has recorded. As a result, as part of a replay attack, Bob would simply drop those encrypted messages delivered by Trudy. (Trudy is unable to generate the new key since MS is encrypted using Bob's public key, which Trudy can't decrypt as Bob's private key is not known to her).

Q7. More love from Alice. Extension to Q6. Alice said that she would generate key material from MS and nonces of Alice and Bob, and key_block size and share the key material directly to Bob by encrypting it with Bob's public key. Trudy captured messages exchanged between Alice and Bob in this modified handshake protocol. Do you think Trudy can succeed in launching session/connection replay attacks on Bob? Justify your answer.

A: No Trudy can’t succeed in launching session/connection replay attacks on Bob. Even though he knows nonces, since MS is not known, Bob can’t verify whether KeyMaterial is the new one or not. But when verifying the FINISH message it fails because the client and server digests do not match because the server has generated a new server hello with a new random number that differs from the one used to generate the client's digest, which the attacker is replaying.

Q8. Sequence number counter (initially set to 0) is used by Alice to input the current value of the sequence number counter while calculating MAC for inclusion into TLS records for integrity protection. Assume that Alice has been sending 10 TLS records carrying application data (each of size 500 Bytes) to Bob. Trudy being Woman-in-the-Middle between Alice and Bob, deletes record numbered 7th. She wants to fool TCP's insequence delivery mechanism so that the TCP receiver at Bob thinks everything is perfect and forwards the received TLS records to the TLS layer. How could she get away and pass through TCP checks? Hint: Trudy has to manipulate TCP segments numbered 8th, 9th and 10th. How?

A: The insequence delivery mechanism in TCP verifies that the arriving TCP packets are in the correct order according to the TCP sequence number. TCP's insequence delivery mechanism will detect if a TCP packet with sequence number 7 is deleted. Trudy must collect the packets with TCP sequence numbers 8, 9 and 10 and modify them to 7, 8 and 9 and send them back to Bob in order for the TCP receiver to believe the packets came in order.

Q9. Having successfully fooled the TCP receiver of Bob in Q8, do you think Trudy can fool the TLS receiver of Bob? Explain.

A: No, Trudy can't fool TLS receiver Bob because MAC authentication is done in TLS layer . MAC is created by hashing TLS sequence number, message and the MAC key with HMAC hash function. Even though the message or MAC key may not have changed, because the TLS sequence number has changed, the MAC generated at the TLS receiver does not match the previously calculated MAC, and the MAC authentication fails, resulting in Trudy's erroneous packet being detected. As a result, Trudy will be unable to deceive the TLS layer.

Q10. Assume that Trudy captured application data messages exchanged between Alice and Bob using TLS 1.2. Alice is a web browser whereas Bob is a web server with Digital Certificate signed by a CA using RSA. After a year from this correspondence between Alice and Bob, Trudy hacked into the webserver and stole Bob's private key. Explain how Trudy can decrypt all of the old application data exchanged between Alice and Bob? This means there is no forward secrecy. It's indeed possible when TLS_RSA_WITH_AES_256_CBC_SHA256 is used as the cipher suite.

A: Trudy has Bob's (server) private key, which she can use to decrypt the PMS encrypted with Bob's (server) public key. She can now construct Master secret by feeding nonces and PMS to PRF, and subsequently key material by feeding MS and nonces to PRF, because she knows the nonces of Bob and Alice (which were caught previously because they were visible to everyone). Trudy can simply decrypt the application data packets now that she has key material.

Q11. You are tasked with providing perfect forward secrecy by fixing the issue described in Q10. What tweaks do you make to TLS_RSA_WITH_AES_256_CBC_SHA256 for that? Hint: You can't replace RSA with any other algorithm.

A: Bob should generate a new public private key pair for every session. As a part of server key exchange message, Bob should send the new public key as a message, along with the hash of this message signed by Bob using his old private key.From comparing the digest obtained by hashing the message and decrypting the signed message, Alice may now confirm Bob's new public key. Now, Alice encrypts the session key as part of the client key exchange message using Bob's new public key. Trudy can only obtain the session public key, from which she cannot extract the session private key, preventing her from decrypting the PMS and maintaining forward secrecy.

Q12. Does TLS_ECDH_RSA_WITH_AES_256_CBC_SHA offer perfect forward secrecy? Explain.

A: No, TLS_ECDH_RSA_WITH_AES_256_CBC_SHA doesn't offer perfect forward secrecy. ECDH in static mode uses a long term ECDH key and hence if this key is compromised we can generate the symmetric keys using PRF and MS and all the previous messages can be decrypted.

Q13. Refer RFC 5246 on Cipher Suites of TLS 1.2 and list down the ones that offer perfect forward secrecy

A:

1. TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
2. TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
3. TLS_DHE_DSS_WITH_AES_128_CBC_SHA

4. TLS_DHE_RSA_WITH_AES_128_CBC_SHA
5. TLS_DHE_DSS_WITH_AES_256_CBC_SHA
6. TLS_DHE_RSA_WITH_AES_256_CBC_SHA
7. TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
8. TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
9. TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
10. TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

Q14. Refer RFC 8446 on Cipher Suites of TLS 1.3 and list down the ones that offer perfect forward secrecy

A: In TLS 1.3 it is mandatory for a cipher suite to offer Perfect Forward Secrecy. The following Cipher suites are listed in the RFC.

1. TLS_AES_128_GCM_SHA256
2. TLS_AES_256_GCM_SHA384
3. TLS_CHACHA20_POLY1305_SHA256
4. TLS_AES_128_CCM_SHA256
5. TLS_AES_128_CCM_8_SHA256

A15. Privacy issues with TLS 1.2: Does any 3rd party like ISPs/enterprises profile their users (i.e., browsing patterns) even though their application data is encrypted? Explain!
A: ISP/enterprises can see the stats or traffic flow between two points on the internet but they can't observe the user's application level data as they are completely encrypted. They can just see the websites a particular person is visiting, different sites they are connecting to etc.

Deliverables: A Google Doc listing down Q&As. Write crisp answers (5-10 lines) based on your own understanding of the concepts. Copying from any sources will be dealt with seriously.

References:

- **Slide deck on TLS**
- https://tools.ietf.org/html/rfc5246
- https://www.coursera.org/learn/crypto/lecture/WZUsh/case-study-tls-1-2
- https://stackoverflow.com/questions/61155273/the-difference-between-ecdh-ecdsa-aes128-sha256-and-ecdhe-ecdsa-aes128-sha256