Recall topics covered in today's class 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 plain English sentences in the space provided (no links/attachments). Alice is a web browser where as Bob is a web server with Digital Certificated signed by a CA using RSA.

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 session/connection relay attack with Bob? Explain your answer. Can Trudy replay Bob's previous messages with Alice for successfully launching session/connection replay attack? Explain your answer.

Q2. Explain how nonces employed in TLS help in preventing session/connection replay attacks in Q1.

Q3. How does Alice derive PreMaster Secret (PMS) which she wants to send to Bob? Refer RFC 5246.

Q4. Why can't Bob derive PMS and share it to Alice?

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.

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. I wrongly told in today's class that nonces are used only to generate key material, not MS. 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 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.

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.

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 TCP receiver at Bob thinks everything is perfect and forwards the received TLS records to 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?

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

Q10. Assume that Trudy captured application data messages exchanged between Alice and Bob using TLS 1.2. Alice is a web browser where as Bob is a web server with Digital Certificated signed by a CA using RSA. After a year from this correspondence between Alice and Bob, Trudy hacked into the webserver and stolen 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.

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

PS: Refer attached links, slide deck on TLS and today's class lecture to answer these questions. Feel free to reply this thread if you have any doubts in answering any of these queries.

[Case Study: TLS 1.2 - Authenticated Encryption | Coursera](https://www.coursera.org/learn/crypto/lecture/WZUsh/case-study-tls-1-2)
https://www.coursera.org/learn/crypto/lecture/WZUsh/case-study-tls-1-2

[rfc5246](https://tools.ietf.org/html/rfc5246)
https://tools.ietf.org/html/rfc5246

Your answer
1. Let us consider the scenario that Alice is using TLS communication without nonces with bob. Let PMF1 be the encrypted PMF sent by Alice to Bob. Then the key to be used for decryption and encryption of application data of phase 4 will only depend on PMF1.
i.e., MS1 = PRF(PMF1),
key1 = PRF(MS1) = PRF(PRF(PMF1))

Trudy launching session/connection relay attack with Bob:
After Alice and Bob’s session is completed, let's assume Trudy got hold of this entire session(including PMF1). Now, Trudy launched an attack replaying Alice by sending the PMF1 to Bob(after sending the same certificates as Alice), so the key generated by Bob will be the same as the previous one.
i.e., MS2 = PRF(PMF1) = MS1
Key2 = PRF(MS2) = PRF(MS1) = key1
As the keys are the same, Bob will again be able to perform the decryption of the next recorded data messages. Thus, making the attack successful.

Trudy launching session/connection relay attack with Alice:
Now, Trudy has sent Bob’s certificate and the validation is done by Alice. Now Alice will send different PMF2(encrypted with Bob's public key) to Trudy. Thus a different key will be generated by Alice. As Trudy cannot decrypt the PMF2, it cannot generate the key. If Trudy sends the previous messages encrypted by Bob before, as the key now is different, Alice will not be able to decrypt it. Thus, making the attack unsuccessful.

Conclusion: Trudy can successfully launch a session/connection relay attack with Bob but not with Alice.

2. Let us consider the above scenario using nonces. Let na1, nb1 be the nonces of Alice and Bob respectively and PMF1 be the encrypted PMF sent by Alice to Bob. Then the key material is given by
MS1 = PRF(PMF1),
key1 = PRF(MS1,na1,nb1)
If Trudy wants to launch a session replay attack by sending the same na1, the nonce sent by Bob will be a different random number nb2(i.e., nb1 != nb2). Then the generated now will be different from the before generated key.
i.e., MS2 = PRF(PMF1) = MS1,
key2 = PRF(MS1, na1, nb2) != key1
And thus Bob cannot decrypt the recorded encrypted application data sent by Trudy, and thus fail the attack.
Similarly, even if Trudy replays as Bob, there will be a difference in Alice nonce(i.e., na1 != na2) and thus making a different key. So, it is not possible to attack.

Conclusion: Trudy can launch a session/connection relay attack with neither Bob nor Alice.

3. If RSA is used for the authentication, Alice derives the 48 bytes premaster secret(PMS) as the following structure:
```
struct {
      ProtocolVersion client_version;
      opaque random[46];
 } PreMasterSecret;
```
Where client_version is the latest (newest) version supported by the client and is used to detect version rollback attacks, random 46 securely-generate random bytes.

4. In the present TLS handshake, Bob's certificate verification is done in phase 2 before Alice's certificate verification in phase 3. So if Alice derives the PMS, it can encrypt it with the public key received in phase2 and can send it in phase 3 parallel to Alice's certificate. But, If Bob derives the PMS and wants to share it to Alice, it requires Alice's public key for the encryption which is an optional step in our present handshake protocol. So, Bob can't derive PMS as it may or may not receive Alice's certificate.

5. As discussed above, for Bob to derive PMS it requires an additional step of verifying Alice's certificate. So it should wait until Alice's public key is shared in phase 3 and will send the encrypted PMS in the next phase and thus increasing the handshake time. So, in general the derivation of PMS is done by Alice rather than Bob.

6. It is said that MS are made giving PMS, nonces as inputs and key material is made by giving MS, nonces, key_block size as input
i.e., MS = PRF(PMS,na,nb)

key material = PRF(MS,na,nb,key_block size)
Let us consider the scenario where Alice is sending the encrypted MS instead of PMS. Now after the session between Alice and Bob is ended lets assume trudy got hold of the following parameters - na1, nb1, encrypted MS1, encrypted application Data1. If Trudy is replaying the Alice and sends na1 to Bob, the Bob will send a different nonce nb2(i.e., nb2 != nb1). So, even if the trudy sent the same encrypted MS1 to bob, as nb is a parameter of key material it will be different to previous key. And thus Trudy cannot succeed in launching session/connection replay attacks on Bob.

7. If the key material itself is shared by Alice to Bob, when the trudy is replaying Alice it sends the same encrypted key material to Bob as before, and thus bob will be able to decrypt the next recorded messages sent by Trudy and this succeeding in launching session/connection replay attacks on Bob.

8. The Trudy being Woman-in-the-Middle between Alice and Bob, deletes record numbered 7th but wants to fool TCP's insequence delivery mechanism so that TCP receiver at Bob thinks everything is perfect. To do that Trudy has to modify the 8th, 9th, 10th TCP segments in such a way that the 7th layer is never lost. So it will renumber the sequence number of 8th, 9th, 10th packet to 7th, 8th, 9th packet respectively, so that TCP layer thinks all the packets arrived in order without loss in data and thus forwards the received TLS records to TLS layer.

9. Even though Trudy successfully fooled TCP receiver of Bob, it cannot fool TLS receiver of Bob. This is because the decrypted TLS record contains the message along with the MAC digest. This MAC digest is generated by Alice by passing the message, sequence number, MAC key as inputs to HMAC algorithm. So this decrypted record contains the MAC digest with the original sequence number. But when Bob generates MAC digest with the sequence numbers renumbered by Trudy using HMAC algorithm, it will not match with the decrypted MAC of the original message. Hence indicating loss of integrity. So it cannot pass the TLS receiver of Bob.

10. If Trudy captures the application data messages exchanged between Alice and Bob using TLS 1.2 and get hold of the Bob’s private key after an year, he can able to decrypt encrypted Premaster secret(PMS) sent by Alice and thus able to generate the key material using PRF as he already accompanied with data of nonces. So, Trudy can decrypt all of the old application data exchanged between Alice and Bob using this key.

11. Forward secrecy can be provided for the above issue by using a public-key encryption scheme with an additional per-session key. This is explained in the following mechanism.
Alice - (PrivA, pubA)
Bob - (PrivB, pubB)
Consider the TLS handshake completed upto the client certificate. Now consider the below modification of mechanism.
Bob:
Generate a new RSA key pair (privT, pubT) used for encryption.
Digest the pubT key and sign it using privB giving a signature S.

Send pubT, S to Alice
Alice:
Verify the signature S and confirm the pubT is sent by Bob.
Generate PMS and encrypt with pubT.
Send encrypted PMS to Bob
Bob:
Decrypt the encrypted PMS using privT.
Generate the key material using PMS and nonce.
Once the session is completed the (privT, pubT) will be deleted forever and thus the trudy cannot be able to decrypt the messages as they are encrypted with a key that is generated by the PMS encrypted with a public key for which its private key pair is deleted once the session is completed. By using this mechanism even though the trudy gets hold of (PrivB, pubB) he cannot decrypt the application messages.