Assignment: Cryptographic Scenarios - Noah Lee

1. Alice wants to send Bob a **long** message, and she doesn't want Eve to be able to read it. Assume for this scenario that AITM is impossible.

Since the message is long, we will use **symmetric encryption**. Firstly, Alice and Bob use **Diffie-Helman** to agree on a shared secret from which they derive a shared **AES** key - K. Alice first uses this key to create ciphertext C = AES(K, M) where M is the original message - Eve may see this ciphertext, but can not decrypt it without breaking the AES algorithm to find the key. Alice then sends the ciphertext C to Bob, where Bob is able to decrypt it using M = AES_D(K, C).

This plan works because Eve is not able to decrypt the Cipher Key, as the shared key K is agreed upon with Diffie-Helman, and AITM is not a factor.

2. Alice wants to send Bob a long message. She doesn't want Mal to be able to modify the message without Bob detecting the change.

For this problem, we will use **symmetric encryption** (because the message is long), **Diffie-Helman** (for key transfer) and SHA-256 **cryptographic hash** to create a **digital signature**. First, Alice and Bob use Diffie-Helman to agree on a shared secret from which they derive a shared AES key - K. Alice first uses this key to create a ciphertext C - C = AES(K, M) where M is the original long message. Alice then hashes the message using SHA-256 to get H(M) and then creates a digital signature using her private key S_A. This is denoted as S = E(S_A, H(M)). Alice then sends C along with S to Bob. Mal can not mess with the bits in this message as well, because then the hash of the message would not match the hash in the signature. When Bob receives C and S, Bob decypts is with M = AES_D(K, C) and then calculates the H(M) using the same hash function on the message. Bob then decypts the signature using Alice's public key - H(M) = E(P_A, S) and then compares the this hash with the hash of the contents of M - if they are the same, then the message has not been modified by Mal.

This plan works because Mal is unable to modify the message because Bob will be able to tell if Mal does by comparing H(M).

3. Alice wants to send Bob a long message (in the case, it's a signed contract between AliceCom and BobCom), she doesn't want Eve to be able to read it, and she wants Bob to have confidence that it was Alice who sent the message. Assume for this scenario that AITM is impossible.

For this problem, we will use **asymmetric encryption** (so that Alice can use her public key as a signature) and **digital signatures**. First, Alice will encrypt the message M using her private key - C1 = E(S_A, M). Then Alice will decrypt the message again using Bob's public key - C2 = E(P_B, C1). Alice sends C2 to Bob. Eve can see C2, but can not read M. When Bob gets

C2, he first uses his private key to get C1 - C1 = E(S_B, C2) and then uses Alice's public key on C1 to get M - M - E(S_A, C1). This confirms the message was sent by Alice, because only Alice is in possession of the private key that is connected to her public key.

This plan works because Bob can confirm Alice sent the message by using Alice's public key to verify it was from her and because AITM is not a factor here.

4. Consider a scenario where Alice and Bob have been in contract negotiations and sharing documents electronically along the way. Suppose Bob sues Alice for breach of contract and presents as evidence the digitally signed contract (C || Sig) and Alice's public key P_A. Here, C contains some indication that Alice has agreed to the contract—e.g., if C is a PDF file containing an image of Alice's handwritten signature. Sig, on the other hand is a digital signature, as described at 9:23 or so of the Cryptographic Hash Functions video.
   Suppose Alice says in court "C is not the contract I sent to Bob". (This is known as *repudiation* in cryptographic vocabulary.) Alice will now need to explain to the court what she believes happened that enabled Bob to end up with an erroneous contract. List at least three things Alice could claim happened. For each of Alice's claims, state briefly how plausible you would find the claim if you were the judge. (Assume that you, the judge, studied cryptography in college.)

1) "A malefactor altered the contract in transmission"

Altering the contract is not possible unless the signature has been compromised. This is because the hash of C is embedded into the signature and compared to the original message so if any contents were altered along the way, Bob would immediately know because the hash of C of the message would not be equal to the one in the certificate, so unless a malefactor is able to alter the contents of the signature, this would not be possible.

Especially if a third party was in charge of handling the signatures, this claim is highly implausible because it would be very difficult to compromise the certificate and the malefactor would need to alter both the contract and the certificate before it reaches Bob.

2) "Someone else sent that contract posing as Me (Alice)"

This is also hard to do because the point of a certificate is primarily to verify that the person that sent the data came from the party we think we are talking to. Again, unless the certificate was compromised and a malefactor was able to obtain the exact private key that was used to create the certificate, it would be easy for Alice to tell that someone else sent the contract. This claim is unlikely - and if it were, the certificate authority would be liable.

3) "The contents in the contract were different that to what was signed - some bits were lost or flipped during transit"

Given that the certificate is used to verify that the contract had not altered, unless the bits were somehow flipped in the exact same way for both the encrypted message and the hashed message in the certificate, the odds of this happening are significantly unlikely. This claim is very implausible.

5. For this scenario, suppose the assumption that everybody has everybody else's correct public keys is no longer true. Instead, suppose we now have a certificate authority CA, and that everybody has the correct P_CA (i.e. the certificate authority's key). Suppose further that Bob sent his public key P_B to CA, and that CA then delivered to Bob this certificate:
Cert_B = "bob.com" || P_B || Sig_CA
In terms of P_CA, S_CA, H, E, etc., of what would Sig_CA consist? That is, show the formula CA would use to compute Sig_CA.

To encrypt:
M = “bob.com” || P_B
Sig_CA = E(S_CA, H(M))

Then this is concattenated to make Cert_B, sent to Bob: Cert_B = “bob.com” || P_B || Sig_CA

6. Bob now has the certificate Cert_B from the previous question. During a communication, Bob sends Alice Cert_B. Is that enough for Alice to believe she's talking to Bob? (Hint: no.) What could Alice and Bob do to convince Alice that Bob has the S_B that goes with the P_B in Cert_B?
No, simply sending the Cert_B does not prove that the person sending the message is Bob, this is because anybody could intercept the Cert_B and send it to be Alice pretending to be Bob. Bob would need to prove that he has S_B to validate that he is indeed Bob. To prove that Bob has the corresponding S_B, Alice could send a message back to Bob, using the P_B in the certificate, Bob can then use his S_B to decrypt the message and send the message back to Alice - verifying that he has the S_B, and proving his identity.

7. Finally, list at least two ways the certificate-based trust system from the previous two questions could be subverted, allowing Mal to convince Alice that Mal is Bob.
1) If Mal get’s Bob’s private key or manages to calculate it.
2) If Mal makes a similar “bob.net” or creates their own signature and contacts Alice posing as bob with a different signature.