Nathan Parker and Alden Harcourt

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.
    - Alice and Bob use Diffie-Helman to agree on a shared secret which is used as key K for the AES encryption algorithm.
    - Alice sends AES(K, M) to Bob.
    - Bob uses AES_D(K, C) to decrypt the ciphertext back into the message.

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.
    - Alice uses SHA-256 to create the digest H(M)
    - Alice uses public key encryption to create a ciphertext of H(M), C. C = E(P_B, H(M))
    - Alice sends Bob M, along with ciphertext C
    - Bob receives C and decodes it using his private key S_B. E(S_B, C) = E(S_B, E(P_B, H(M)) = H(M)
    - Bob then uses SHA-256 to hash M, and checks it against H(M) to see if they are the same. If they are not the same, then he knows that some AITM intercepted and modified the message. However, the AITM would not be able to modify C because they would not be able to decode it.

3. Alice wants to send Bob a long message (in this 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.
    - Alice and Bob use Diffie-Helman to agree on a shared secret which is used as key K for the AES encryption algorithm
    - Alice encrypts M using AES to create ciphertext C
    - Alice hashes M using SHA-256, making H(M)
    - Alice sends H(M) using public key encryption, along with C to bob
    - Bob decrypts C using the pre-shared key to create M’
    - Bob hashes M’, creating H(M’) and checks to see if H(M’) matches H(M)

- Bob can be sure that Alice sent him the message because if Mal had done something malicious, the H(M') wouldn't match the H(M) that Alice sent via public key encryption

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.)
   - Alice's private key was compromised
     i. Alice could say her secret key was compromised and Mal used it to sign a document that she didn't want to sign. In this situation, the document would still be able to be verified using the public key without Alice having signed it. This is definitely plausible if Alice was insecure with the use of her secret key.
   - Bob changed the document after receiving it
     i. Alice could argue that the contract she signed and sent to Bob is not the contract that Bob is displaying, and that he modified it after the fact. This is an easily provable/disprovable claim, as someone could just see if the hashes match or not.
   - Someone else logged into her computer and signed it for her
     i. Alice could argue that someone simply logged into her computer, and signed the document for her without her knowledge. This is also fairly plausible if Alice has bad security practices, like maybe leaving her laptop open unattended.

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:

   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`.
   - Sig_CA = E(S_CA, H("bob.com" || P_B))

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`?
   - Alice sends Bob a short message M, encoded using E(P_B, M). Bob sends Alice back M using E(S_B, E(P_B, M) = M. Both sides now are sure that each other have the correct public/secret keys.

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.
   - The certificate authority is compromised
     i. If Mal compromised the certificate authority she could link her public key to bob.com. Then the CA would issue the fraudulent certificate that looks valid because it is signed with the CA's private key
   - AITM attack
     - If Mal is able to intercept Alice's communications such that Alice believes she is connecting to Bob's server but is in fact connecting to Mal's server. Mal then forges a certificate and presents it to Alice, possibly convincing Alice that Mal is Bob.