Mary Blanchard and Kaitlyn Peterson
Encryption Scenarios

1. First, Bob and Alice use the Diffie-Hellman Key Exchange to agree on a shared secret key (K). Then, Alices uses the symmetric algorithm AES to encrypt her message using C = AES(K, M). She sends C to Bob, who decrypts using M = AES_D(K, C)
    a. This works because an eavesdropper can see all parts of a Diffie-Hellman exchange without finding out the secret key. Since the message Alice wants to send is long, she does not want to use a public key encryption technique, so AES is the best choice.

2. Alice first computes D = H(M). Then she computes the digital signature Sig = E(S_A, D). She then sends Bob the original message M concatenated with the digital signature. Bob can recompute the digest with D = H(M) and decrypt the signature using E(P_A, Sig). If the decrypted signature E(P_A, Sig) matches the hash of the message he received D, he knows the message sent by Alice was not corrupted by Mal.
    a. Since we are assuming that Alice's public key does belong to her and that she has successfully kept her private key a secret, Bob knows that when he decrypts the signature using Alice's key, only Alice could have encrypted it. Additionally, Mal could not corrupt the message without the message's hash changing and Bob noticing, or without recomputing the hash of the corrupted message and encrypting it using Alice's secret key, which Mal does not have.

3. Alice and Bob use the Diffie-Hellman Key Exchange to agree on a shared secret key (K). Alice then can use the symmetric algorithm AES to encrypt her message using C = AES(K, M). To ensure Bob knows the message is from her, she adds a signature. To do this, she first computes the digest of the ciphertext D = H(C). To compute the signature, she encrypts the digest with her secret key Sig = E(S_A, D). Then she sends the encrypted ciphertext concatenated with the signature C||Sig to Bob. Bob can recompute the hash of the ciphertext D = H(C). Then, he will check that the digest equals the signature, which he decrypts using Alice's public key E(P_A, Sig). If these are equal, he knows that Alice sent the message. Now he can decrypt the ciphertext using their shared secret key AES_D(K,C) to read the message.
    a. Since PitM is not a concern here, the Diffie-Hellman exchange means that the message encrypted with AES is not readable to Eve since she can not know the secret key K. As described above, the signature process ensures that Alice sent the message to Bob since she is the only one with access to her private key.

4. The first scenario Alice might describe is that they were duped by a person in the middle attack. Say the person in the middle is named Mal. During the Diffie-Hellman exchange, Mal could have established his own secret keys K_1 that he shared with Alice and K_2 that he shared with Bob. After Alice sends the ciphertext and signature, Mal can decrypt the contract using AES_D(K_1, C), change it, and re-encrypt it using AES(K_2, M) before passing it along to Bob. Then any of the three following scenarios could have happened.

a. First, it's possible that Bob simply did not check the signature when he received the contract. It is possible that Mal altered the contract and did not change the signature. If Bob had checked the signature, he would have noticed that the hash of the corrupted contract did not match the hash in the decrypted signature and could have prevented the whole debacle. In this world where Alice, Bob, and Mal are doing the cryptography themselves, it is fairly plausible that Bob got tired of checking signatures all the time and neglected it. However, in the real world where these kinds of security measures are automatic, this type of security neglect is less likely.
b. Another possibility is that Mal used brute force to figure out Alice's secret key. He could compute the hash digest since he knows the ciphertext. He also has the encrypted version of the digest that he can check against. Using brute force, Mal could have figured out Alice's secret key S_A. Then, when Mal corrupts the contract, he can re-encrypt the ciphertext using his shared key with Bob and redo the hash and signature using his newfound knowledge of Alice's secret key. When Bob receives the corrupted contract and matching signature, he would have no reason to believe the contract did not come from Alice. The likelihood of this scenario depends on Mal's resources and the size of the integers that Alice and Bob are using in their asymmetric encryption key pairs. Further questioning of the encryption method would be needed.
c. Finally, Alice might claim that Mal altered the contract to ciphertext that hashes to the same value as the original contract. This would mean that Mal was able to change the contract and keep the signature the same, so he doesn't need Alice's secret key to re-encrypt a new signature. This scenario is extremely unlikely because the hash function SHA256 is collision-resistant. However it isn't entirely impossible because there are a finite number of 256 bit hashes and an infinite number of ways to corrupt the contract. This scenario is easily proven in court by having Alice present the original contract, computing the hash of the original and the corrupted contract, and comparing them. If they do work out to be the same, this scenario could be the explanation.

5. First the CA will hash the data contained in the TBS (to be signed) portion of the certificate. This includes the domain bob.com, Bob's public key P_b, as well as any other data the CA deems fit, such as the date or the expiration. Once this data has been hashed into a digest, the CA will encrypt the digest using its secret key S_{CA}, as Sig = $E(S_{CA}, H(TBS))$. The signature is appended to the TBS, and that's the whole certificate.

6. Bob's certificate is public, so anyone could send Bob's certificate to Alice and say they were him. In order to prove his identity, Bob could encrypt something that Alice picks with his private key, send it to Alice to decrypt with his public key and she would see that he does in fact have the matching private key to the public key in his certificate. However, doing this alone is still susceptible to situations with a person in the middle, such as Mal. In order to create an entirely secure and trustworthy connection between Alice and Bob, they can do the following. Alice and Bob will first use the Diffie-Hellman exchange. Then, Alice will send Bob an unencrypted random integer R. To prove his

identity, Bob will send Alice $E(S_B, H(K||R))$. Alice can then decrypt the message using Bob's public key and recompute the hash to validate the signature. If Mal were in the middle, he could establish separate keys K_1 and K_2 with Alice and Bob and pass the random number along. However, when it came time to encrypt the hash, Mal would not be able to use Bob's secret key because he doesn't know it. When Alice goes to decrypt it using Bob's public key, it would not work and she would realize that it was not Bob who encrypted it.

7. Subverting the certificate trust system:
    a. One way Mal could subvert the certificate-based trust system is by fraudulently convincing the certificate authority that he is Bob. This could be done by stealing Bob's drivers license or other form of ID that the CA wants. Then Mal can be issued a certificate with his own public key under Bob's name. Using this certificate, he can convince Alice that he is Bob.
    b. Mal could pretend to be a certificate authority and issue himself a certificate pretending to be Bob, signing with S_{CA}, which is actually his own secret key. If Mal is able to successfully convince browsers to publish his public key as the fake CA, then he can act as a CA and create a forged but trustworthy-seeming certificate for bob.com that actually uses another one of Mal's own key pairs. With this certificate, Mal can easily convince Alice that he is Bob.