Cryptography Scenarios
Vanessa Heynes

1.
- Alice and Bob first agree on a key, K, using Diffie-Hellman.
- Alice encrypts her message with the function AES(K, M) = C, where C is the newly created ciphertext.
- Alice sends C to Bob.
- Bob can decrypt C to reveal M by doing AES_D(K, C).
- Eve will not be able to decrypt the message without having access to K.

2.
- Alice and Bob can use a data integrity check by completing the following:
- Alice and Bob use Diffie-Hellman to get a shared key, K.
- Alice sends Bob AES(K, M || E(S_A, H(M))).
- Bob decrypts E(S_A, H(M)) using P_A.
- Bob recomputes H(M) using the M he received, as well as the corresponding hash function Alice used.
- If the value of H(M) that Bob computed is equal to the digest Alice sent him, then he can be certain that M did not get corrupted. Otherwise, he will be able to detect that Mal modified Alice's message.

3.
- Alice and Bob can use a digital signature by completing the following:
- Alice encrypts her message using a hash function which is denoted as D = H(M), where M is her message.
- Alice creates her digital signature by applying her secret key and D to a public key encryption function which can be denoted by Sig = E(S_A, D)
- A concatenated version of M and her signature are sent to Bob by doing (AES(K, M || Sig)).
- Bob recomputes the hash of M.
- Bob decrypts Alice's signature using her public key as denoted by E(Pa, Sig).
- If the decrypted value is equal to his computation of H(M), then he can be certain that Alice is the one who sent M.

4.
1. Alice could claim that an AITM corrupted C in the process of it being sent to Bob. This is not plausible because since they used a digital signature, Bob would have known right away if Alice wasn't the one who sent C. After Bob receives C, he would need to compute the hash of C, decrypt Alice's signature using her public key, and compare if these two values are equal or not. If they were equal, Bob wouldn't have reason to believe that C was corrupted.
2. Alice could claim that she used a different hash function to encrypt her signature. This is plausible because if Bob is unaware of the function he used, then he could have very

well been communicating with an AITM instead of Alice. If he had used the correct hash function / the one that Alice used, then he would've gotten an unequivalent value for the digest and would know that the message was corrupted.

3. Alice could claim that Bob doesn't have her actual public key. This is not plausible because of the assumption that everyone has everyone else's correct public keys. Since Alice encrypted her signature with her private key, and Bob was able to decrypt it plus see that it is equivalent to his computation of H(C), then Alice's claim is not true.

5.

Sig_CA = E(P_CA, H(TBS) where TBS refers to the subject (bob.com) and P_B.

6.

- Alice and Bob can do the following to convince Alice that Bob has the secret key that goes with the listed public key in Cert_B.
- Alice and Bob can complete a challenge alongside Diffie-Hellman.
- First, Alice sends Bob a random number, R.
- Then, Bob sends Alice E(S_B, R || g^b mod p).
- If Alice receives the expected value of R || g^b mod p, then she can be certain that an AITM did not change the contents associated with the certificate.
- This adversary will also not be able to change the contents without knowing S_B, so Alice would be able to detect any changes.

7.

1. Mal can forge a certificate by exploiting weaknesses in the hash function used to create the digital signature. In this case, the hash of one set of malicious data will be equivalent to the hash of the data Alice or Bob is sending so from their perspective it seems as if the certificate's contents can be trusted.
2. If the certificate is expired or self signed, then Alice and Bob are even more prone to Mal's attacks. In this case, it is harder for Alice or Bob to verify a certificate's authenticity and if they're actually receiving content from one another instead of Mal.