1. Alice and Bob should use the Diffie-Hellman key exchange to agree on a secret key. Using this key K, Alice can then encrypt her message M using AES(K, M), producing ciphertext C. Since Bob is the only other person with the secret key K, he and he alone can decrypt the C that Alice sends over using AES_D(K, C), resulting in Alice's original message M.

2. This time, Alice is going to want to add a digital signature to her message to demonstrate its authenticity to Bob. She can do this by first using H(M) to create a SHA-256 hash of her message M, resulting in a digest D (this compacts the data so it can be RSA encrypted later on, and also provides an integrity check). Then, she can RSA encrypt D using E(S_A, D) to get the signature Sig, concatenate Sig to M as M || Sig, and send the concatenation over to Bob. Bob in turn can then hash the unencrypted M he receives using the same hash function as Alice, decrypt Sig using Alice's public key P_A, and then compare his D' to E(P_A, Sig). If they are the same, Bob can be sure the message came from Alice because only she could have encrypted Sig using her private key.

3. Alice and Bob should first use Diffie-Hellman to agree on a shared key K, to be used later on. Then, Alice should proceed in a manner almost identical to how she did in scenario 2. That is, she should use H(M) to create a digest D of her M, encrypt D using E(S_A, D) resulting in a signature Sig. But now instead of concatenating Sig to M, she should first encrypt M using AES(K, M) based on the K that she agreed on with Bob at the start, and then create a concatenation C || Sig to send over. When Bob receives C || Sig, he can decrypt the ciphertext C using AES_D(K, C) to read the secret message. To ensure confidence that it was Alice that sent the message, he can once again decrypt Sig using E(P_A, Sig), take the M he decrypted using AES_D(K, C), hash it with H(M), and compare this D' to E(P_A, Sig). If they are the same, he can be sure the message also came from Alice.

4. One explanation Alice could give is that a hacker used PITM to intercept her Diffie-Hellman exchange with Bob, stealing their key K. Then the hacker used that K to encrypt a new, altered contract and concatenate it to the original signature on the end of the authentic message. This would be a terrible excuse, because the signature sent over by Alice would've revealed a digest that was different from the one produced by the actual message, and thus Bob would not have accepted the contract as being from Alice. I would probably be suspicious of Alice for this reason. Another explanation that Alice could claim is that she accidentally exposed her private key. A hacker got ahold of this private key S_A, and then used a PITM attack to intercept her communications with Bob as they conducted a Diffie-Hellman exchange. Equipped with the shared AES key K and Alice's private key S_A, the hacker had all they needed to send an encrypted message using K that looked like it came from Alice. If I were the judge I would find this

relatively plausible, since humans tend to be lackluster about privacy around extremely private things like passwords or secret keys. Finally, Alice could claim that Bob incorrectly identified Alice's public key P_A due to the hacker posing as a certificate authority, when it was really P_H (the hacker's public key). Once again, the hacker stole the shared key K that they agreed upon during Diffie-Hellman in order to be able to encrypt the contract correctly, and then used their own private key S_H to encrypt the signature so that when Bob decrypted the signature, he thought he was using P_A when he was really using P_H. This explanation is not all that plausible given the fact that Alice's browser should already have access to the correct certificate authority's public key. Also, if we continue to roll with the original assumption that everyone has everyone else's correct public keys, then looking up Alice's public key wouldn't have been necessary in the first place, so I would rule against Alice.

5. To compute Sig_CA, CA would first start by compiling the data included in the rest of the certificate. This includes things like Bob's name, his public key, the current and expiration dates of the certificate, and more. This data is then hashed, and the resulting digest is encrypted using S_CA. So all in all, Sig_CA = E(S_CA, H(data)).

6. No, it's not enough, because anyone could have Cert_B based on a past communication with Bob. Bob would probably want to concatenate a signature to the communication containing Cert_B. This signature Sig would be derived by hashing the rest of the message (including Cert_B) into a digest D, and then encrypting D using Bob's private key with E(S_B, D). This way, when Alice receives the message, she can decrypt Sig using E(P_B, Sig), hash the message she received from Bob, and compare the two, and verify that Bob sent the message if they are identical. If they are, only Bob could have encrypted that signature with his private key. In this way, the certificate proves that Bob is associated with the public key inside, and that public key is confirmed to be the same as that of the person Alice is now communicating with through the security of the signature.

7. One way is that Mal could fool the certificate authority into associating Bob's website and name with her own public key. Mal would likely have to provide fake documents to the certificate authority to do this, and it could take a while (or more likely fail outright, if the CA does its due diligence). Another way is that Mal could try to hack into the CA server that contains its secret key, steal the secret key, and use it to create a fake certificate associating her public key to Bob's website and identity. This certificate would appear perfectly legitimate to anyone it was presented to (because it would indeed be identical to a real one, had it been created in good faith by CA employees).