1 - AITM is impossible, which means that we don't need to worry about Alice sending a message to Mal (who is pretending to be Bob), and Mal reading/modifying the message before sending it to Bob (who thinks Mal is Alice).
To make the communication secure, they can do the Diffie-Hellman key exchange in plain text, using appropriate numbers for their key.
In this case, Eve will have g, p, B, and A. However, as she does not have b (which is randomly generated by Bob), and as she does not have a (which is randomly generated by Alice), and as the numbers of the Diffie-Hellman Key exchange are appropriate, it will take longer than the heat death of the universe for Eve to be able to find any of those numbers. So, Alice and Bob can be sure they are not being eavesdropped.
There are also some assumptions in this case. The first is that the numbers used for this exchange are good. Second, the randomly generated numbers a, and b are actually randomly generated for this specific communication. Numbers a, and b not only have to be of appropriate size but also should not be used in every single communication Alice and Bob have. In addition to that, there should be no obvious (or semi-obvious) pattern that can be inferred based on previous communications (like b increasing by one after every single communication).
After they have an appropriate AES key, Alice can do AES(K, M) where the algorithm is AES and they are using an appropriate block-cipher-mode (so Eve cannot decrypt the message based on the frequency at which each letter happens). Bob will decrypt the message using AES_D(K, C) and safely read it.
As the AES algorithm using CBC is secure and as the keys are hard to crack, then the communication is secure.

2 - There is a BIG assumption in this assignment. Alice has BOB's key, and she is sure it is the correct key. This is a big assumption, as having the correct key pretty much eliminates the AITM threat. Another thing that is worth noting is that Alice does not care about eavesdroppers in this case, only about someone changing the message and Bob not realizing it. Therefore, Mal can read the message, but as long as he can't modify it, it's fine.
Alice should first write the message and then Hash it, therefore getting M and H(M). An additional security measure that she should take is to encrypt the Hash using her private key. Finally, she should concatenate her message with that. Let M' = M || E(S_A, H(M)).

If there is an adversary in the middle, they will be able to read the message, but if they modify any of its bits it will not match the hash. What is more, they can try to hash the new message, but they don't have Alice's public key to encrypt it. So even if they do change the message, they cannot change the hash as it is encrypted with Alice's private key. This ensures that Bob will be sure whether any changes were made to the message.

Bob will receive M'. The last bytes of the message are the encrypted hash E(P_B, H(M)) and the first bytes are M. He should separate the message from the encrypted hash. Hash the message M, getting H(M). Then he should decrypt the hash with E(P_A, E(S_A, H(M))) and compare it to H(M). If they are equal then the message has not been modified. If they are different then the message has been modified.

3 - In this case, an adversary in the middle is impossible. So Alice and Bob can do the DH key exchange (as described in 1) to get a shared key and exchange messages encrypted with AES. Alice then should write the message she wants to send. This way, getting M. Then, she should hash it using SHA-256, getting H(M). After that, she should encrypt the hash using her private key, getting E(S_A, H(M)). Alice then should concatenate both. The result will be M||E(S_A, H(M)). Last but not least, she should encrypt it all using AES before sending it to Bob. Therefore, the entire message will be C = AES(K, M || E(S_A, H(M))) .

The message is AES encrypted so Eve cannot read it. If the key of Diffie-Hellman is appropriate (as described in 1), Eve will not be able to discover the key. All the packages will be gibberish to her, and she will not read the contract.

Once the message arrives at Bob's side, he has to follow some steps to read the contract. First of all, he should do AES_D(K, C). Then, he will get M || E(S_A, H(M)). He can get Alice's signature, and decrypt it with her public key like: E(P_A, E(S_A, H(M))), getting H(M). Then all he has to do is Hash the M he received with SHA-256. Let's call the result of hashing the M he received D. If E(P_A, E(S_A, H(M))) == D, then the message was sent by Alice. He can be confident as the hash of the message was signed by Alice's private key, and can only be decrypted by Alice's public key. Only Alice has Alice's private key, and decrypting with her public key ensures that the message was written by Alice. If the hash does not match the message either got corrupted or someone else sent it.

4 - Alice can claim that there was an AITM attack. Alice can say she was indeed talking to Bob. But that the contract she sent Bob was a different one. She can say that there was an adversary in the middle, who changed the document and signed it with their own private key. Alice would also have to say that the key Bob used to decrypt the signature is not her Public Key.
If Alice and Bob did a DH key exchange, there are no certificates involved and the assumption that everybody has everybody else's correct public keys is no longer true, then this is a very plausible scenario. An adversary in the middle is a common form of attack. It is easy to change the document and forge Alice's signature. It would also be easy to show that her public key does not match the one in the contract.
If Bob has Alice's correct public key then this is not plausible, as while decrypting the signature Bob would have realized that it was not Alice's.
Alice can claim that she has never sent a contract to Bob in the first place and Bob has been scammed (maybe she has never been in touch with Bob). Or she can simply say that she has not sent the last version of the contract and Bob has been scammed. She can claim that Bob was talking to Mal the whole time. What is more, she can say that Mal impersonated her and it is plausible that Bob believed Mal was Alice. This way Mal sent the contract, with a fake handwritten signature, and signed it with his private key.
In other words, Alice can claim that she wasn't the one who sent the contract. The problem in this claim, which makes it **not plausible**, is that Bob has Alice's public key and he is sure he has the correct key. Therefore, he could not decrypt the signature and get the correct H(M) if he

had received a message encrypted by Mal's private key. Which makes this claim implausible. However, if the assumption everyone has everyone else's correct public key is no longer true then this is plausible. Scamming is a common form of attack.
However, if Mal (who was impersonating Alice) only did DH (**nothing involving RSA**), and somehow convinced Bob to update Alice's public key, then this is plausible. Maybe Mal could say that Alice updated her public key, and as the new value give Mal's public key. In this case, this would be a plausible scenario.
Alice can claim there was a spy in her own company, who modified the contract, signed it with the correct private key, and sent it to Bob. The spy (who we'll call Chad), has access to Alice's computer (they work in the same place). Alice can claim she left her computer open and he had the opportunity to upload a fake contract (via USB) and sign it with the correct private key and send it to Bob. However, this is highly implausible.

5 - The certificate must hash the following message: "bob.com" || P_B. Doing the following H("bob.com" || P_B). What is more H("bob.com" || P_B) should be encrypted using the certificate authority's private key, like E(S_CA, H("bob.com" || P_B)). If they just hash the public key and not the domain name, an adversary in the middle can intercept Bob's certificate, and change the signature of the certificate for the signature of the certificate of Mal.Com (and also change the public key in the certificate for Mal's public key). This way replacing the public key to Mal's public key, and replacing the certificate authorities signature in Bob's certificate for it to be the same as in Mal's certificate. If they just encrypt "Bob.com" and not the key, the adversary in the middle can intercept the certificate (sent as plain text), remove Bob's key, and add their own key. If the signature contains both the name and the public key, then the correct domain is associated with the correct public key, so any change would demand substantially more work. In other words, if they hash both, the domain name, and the public key, any changes in the public key and/or domain name would cause a substantial change in the hash. So both the domain and the key should be correct for the hash to be correct.
What is more, it is of paramount importance for the signature to be encrypted with S_CA, otherwise it is easy to forge it. If the signature is not encrypted anyone can hash the real domain name and fake public key in an AITM attack.

6 - She could send Bob a challenge. She could send a random number R for Bob to encrypt with his private key. When R gets to Bob's side, he should concatenate R with $g^b mod\ p$, this way getting R|| $g^b mod\ p$. He should encrypt that with his private key, E(S_B, R|| $g^b mod\ p$), and send it to Alice.

Alice decrypts the message with Bob's public key, like E(P_B, E(S_B, R|| $g^b mod\ p$)). If the random number matches, and if the DH number is what Alice is expecting, then she is talking to Bob.
This happens because only Bob has Bob's private key. Any messages encrypted with Bob's private key can only be decrypted by his public key, which Alice has. This guarantees that Bob is in the conversation as he correctly encrypted the random number. And Alice has decrypted it using his public key. What is more, sending $g^b mod\ p$ guarantees that no one else is listening to

the conversation. This happens because if Mal was between Alice and Bob, the DH numbers of the Alice-Mal communication would likely be very different than the DH numbers in the Mal-Bob communication. So Alice would expect a very different $g^b mod\ p$. Alice would get a $g^b mod\ p$ and would be expecting a $g^m mod\ p$, where m is Mal's randomly generated number. But if only Alice and Bob are in the conversation, then Bob will send $g^b mod\ p$, which is exactly what she is expecting.
7 - One way is scamming. Imagine if you have the site Bob.com (that's an uppercase beta), or yootube.com, gogle.com (likely caused by typos). In this case, these sites can get 100% legitimate certificates, impersonate real sites, but be used for scamming. Scammers can get a certificate for a site that looks legit but it's not, the domain might be a bit different than the correct one. This way the site will be trusted by the browser but will have malicious intents.

Another way is forcefully downgrading. Imagine Mal is in the middle of Alice and Bob. Mal connects to Bob.com, checks the validity of Bob's certificate, and does a DH key exchange. This way initiating a Mal-Bob connection. Alice thinks she is trying to connect to Bob, while she is, in fact, trying to connect to Mal. In this case, Mal says that bob.com does not have a certificate, and initiates a connection with Alice. They do the DH key exchange but Alice does not check Bob's certificate. Alice will be using http in her connection to Mal instead of HTTPS. This way, by downgrading from HTTPS to http, Mal can steal information from Alice, modify the messages she is sending to Bob.com, just by downgrading.
Last but not least is that you are trusting a third-party individual, who might not be trustworthy. Imagine that Mal works for the certificate authority, has the private key, uses it to create his own fake certificates, and takes part in AITM attacks. In this case, Mal has access to the S_CA and is able to steal it. This gives him power to create fake certificates, using the site's real domain, a fake public key (Mal's public key), and encrypt it with the S_CA. This way it is not possible for Alice to know she is not actually talking to Bob.