

Assumptions

- M: plaintext
- C: ciphertext
- K: shared secret key, obtained by using Diffie-Hellman
- $AES(K, M)$: encrypting M with K
- $AES_D(K, C)$: decrypting C with K
- $H(M)$: hash of M, obtained by using SHA-256
- P_A, P_B : public keys
 - every entity has copy of every public key
- S_A, S_B : secret/private keys
 - no entity has copy of any other secret/private key
- E: encryption/decryption function for operations involving personal keys
 - if M is small enough to be in domain of $E \rightarrow B$ sends $C = E(P_A, M)$, A computes $E(S_A, C) = E(S_A, E(P_A, M)) = M$

1]

- Alice and Bob agree on shared key K by using Diffie-Hellman.
- Alice encrypts message M with K by computing $AES(K, M) = C$.
- Alice sends C to Bob.
- Bob decrypts C by computing $AES_D(K, C) = AES_D(K, AES(K, M)) = M$.
- *Explanation:*
 - We have a long message, so we cannot use public key encryption. This is why we use AES.
 - Since AES encrypts and AES_D decrypts, these equalities hold: $AES_D(K, C) = AES_D(K, AES(K, M)) = M$

2]

- Alice hashes message M by computing $H(M) = H_1$.
- Alice sends $M \parallel H_1$ to Bob.
- Bob computes $H(M) = H_2$.
- Bob checks if $H_1 = H_2$.
 - If this is true, then Mal did not modify M.
 - If this is false, then Mal did modify M.
- *Explanation:*
 - We use " \parallel " to denote concatenation.
 - Hash functions are deterministic, pre-image resistant, collision resistant, and input-sensitive. This means that if $H_1 \neq H_2$, then the messages being hashed are different, which implies that Mal has modified the original message.

3]

- Alice and Bob agree on shared key K by using Diffie-Hellman.
- Alice encrypts message M with K by computing $AES(K, M) = C$.
- Alice hashes message M by computing $H(M) = H_1$.
- Alice encrypts H_1 by computing $E(S_A, H_1)$.

- Alice sends $C \parallel E(S_A, H_1)$ to Bob.
- Bob decrypts C by computing $AES_D(K, C) = AES_D(K, AES(K, M)) = M$.
- Bob computes $H(M) = H_2$.
- Bob decrypts $E(S_A, H_1)$ by computing $E(P_A, E(S_A, H_1)) = H_1$.
- Bob checks if $H_1 = H_2$.
 - If this is true, then Alice sent the message.
- *Explanation:*
 - Only Alice has S_A , so if $E(P_A, E(S_A, H_1))$ is not the hashed version of the message, then $E(S_A, H_1)$ was not an encryption by Alice.
 - Since Eve does not have K , Eve cannot read the message encrypted to C .
 - Since M is a long message, we cannot use public-key encryption, so we use AES encryption.
 - Since $H(M)$ is a fixed size, we can use public-key encryption.

4]

- Claim #1: $H_1 = H_2$ but the messages hashed are not the same. In other words, different messages are hashed to the same digest.
 - Not plausible. Hash functions are collision resistant and input sensitive.
- Claim #2: Bob decrypts the message C incorrectly because Alice and Bob are not using the same shared key K .
 - Not plausible. If Alice and Bob are using different keys for K , then it is indeed unlikely that $AES_D(K_A, C) = AES_D(K_B, AES(K_A, M)) = M$. Thus, at this point Bob would believe a different message. However, Bob would realize that the message was not sent by Alice when Bob checks if $H_1 = H_2$.
- Claim #3: Mal modified C .
 - Not plausible. Although Mal would have access to K and AES/AES_D to encrypt/decrypt, Bob would realize that the message was not sent by Alice when Bob checks if $H_1 = H_2$.
- Claim #4: The key pair (P_A, S_A) is invalid. Since we assume that Bob has the correct P_A , this means that S_A is incorrect.
 - Not plausible. If this was the case, then $E(P_A, E(S_A, H_1)) \neq H_1$, and Bob would find that $H_1 \neq H_2$. (Even if the message was modified so Bob obtained a different H_2 anyway, it is unlikely that these two errors would balance out to make $H_1 = H_2$.)

5]

- $sig_ \{CA\} = E(S_ \{CA\}, H(\text{"bob.com"} \parallel P_B))$

6]

- Alice validates certificate.
 - Alice computes $X_1 = H(\text{"bob.com"} \parallel P_B)$.
 - Alice computes $X_2 = E(P_ \{CA\}, sig_ \{CA\})$.
 - Alice checks if $X_1 = X_2$.
- Alice validates P_B .
 - Alice sends message M to Bob.
 - Bob encrypts message M by computing $E(S_B, M) = C$.

- Bob sends C to Alice.
- Alice decrypts C by computing $E(P_B, C)$.
 - If $E(P_B, C) = M$, then (P_B, S_B) is a valid key pair.

7]

- Mal requests a certificate for “bob.com” with a different public key (denote P_M), and CA approves. Mal would also have a secret key S_M for which (P_M, S_M) is a valid key pair. When Alice requests communication with “bob.com”, the validation steps occur with Mal, although Alice believes she is communicating with Bob.
- As Alice and Bob are deciding on a shared key K , Mal intercepts so that there are two different keys (one for communication between Mal and Alice, and one for communication between Mal and Bob). Now Mal can read/send messages from/to Alice/Bob, while making Alice and Bob believe that they are communicating directly with each other.