Cryptography III

Question 1 ***Public-key encryption and digital signature*** **(10 min)**

Alice and Bob want to communicate over an insecure network using public-key cryptography. They know each other's public key.

(a) Alice receives a message: Hey Alice, it's Bob. You owe me $100. Plz send ASAP.
The message is encrypted with Alice's public key.
◇ *Question:* Can Alice be sure that this message is from Bob?

(b) Bob receives a message: Hey Bob, it's Alice. I don't think I owe you $100. You owe me.
The message is digitally signed using Alice's private key.
◇ *Question:* Can Bob be sure that this message is from Alice?

◇ *Question:* How does Bob verify this message?

(c) Alice receives a message: Hey Alice, it's Bob. Find that $100 in my online wallet, my password is xxxxxx.
The message is encrypted with Alice's public key.
Alice decrypted this and tested the password, and it was in fact Bob's.

◇ *Question:* Can an eavesdropper also figure out the password?

Question 2 ***Encryption provides no integrity, signature provides no confidentiality*** **(25 min)**

Alice and Bob want to communicate with confidentiality and integrity. They have:

- Symmetric encryption.
  - Encryption: $\mathsf{Enc}(\mathsf{k}, \mathsf{m})$.
  - Decryption: $\mathsf{Dec}(\mathsf{k}, \mathsf{c})$.
- Cryptographic hash function: $\mathsf{Hash}(\mathsf{m})$.
- MAC: $\mathsf{MAC}(\mathsf{k}, \mathsf{m})$.
- Signature: $\mathsf{Sign}_{\mathsf{sk}}(\mathsf{m})$.

They share a symmetric key K and know each other's public key.

We assume these cryptographic tools do not interfere with each other when used in combination; *i.e.*, we can safely use the same key for encryption and MAC.

Alice sends to Bob
1. $\mathsf{c} = \mathsf{Hash}(\mathsf{Enc}(\mathsf{k}, \mathsf{m}))$
2. $\mathsf{c} = \mathsf{c}_1, \mathsf{c}_2$: where $\mathsf{c}_1 = \mathsf{Enc}(\mathsf{k}, \mathsf{m})$ and $\mathsf{c}_2 = \mathsf{Hash}(\mathsf{Enc}(\mathsf{k}, \mathsf{m}))$
3. $\mathsf{c} = \mathsf{c}_1, \mathsf{c}_2$: where $\mathsf{c}_1 = \mathsf{Enc}(\mathsf{k}, \mathsf{m})$ and $\mathsf{c}_2 = \mathsf{MAC}(\mathsf{k}, \mathsf{m})$
4. $\mathsf{c} = \mathsf{c}_1, \mathsf{c}_2$: where $\mathsf{c}_1 = \mathsf{Enc}(\mathsf{k}, \mathsf{m})$ and $\mathsf{c}_2 = \mathsf{MAC}(\mathsf{k}, \mathsf{Enc}(\mathsf{k}, \mathsf{m}))$
5. $\mathsf{c} = \mathsf{Sign}_{\mathsf{sk}}(\mathsf{Enc}(\mathsf{k}, \mathsf{m}))$
6. $\mathsf{c} = \mathsf{c}_1, \mathsf{c}_2$: where $\mathsf{c}_1 = \mathsf{Enc}(\mathsf{k}, \mathsf{m})$ and $\mathsf{c}_2 = \mathsf{Enc}(\mathsf{k}, \mathsf{Sign}_{\mathsf{sk}}(\mathsf{m}))$

(a) Which ones of them can Bob decrypt?

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6

(b) Consider an eavesdropper Eve, who can see the communication between Alice and Bob.

Which schemes, of those decryptable in (a), also provide *confidentiality* against Eve?

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6

(c) Consider a man-in-the-middle Mallory, who can eavesdrop and modify the communication between Alice and Bob.

Which schemes, of those decryptable in (a), provide *integrity* against Mallory? *i.e.*, Bob can detect any tampering with the message?

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6

(d) Many of the schemes above are insecure against a *replay attack*.

If Alice and Bob use these schemes to send many messages, and Mallory remembers an encrypted message that Alice sent to Bob, some time later, Mallory can send the exact same encrypted message to Bob, and Bob will believe that Alice sent the message *again*.

How to modify those schemes with confidentiality & integrity to prevent replay attack?

◇ The first scheme providing confidentiality & integrity is Scheme ☐.

The modification is:

◇ The second scheme providing confidentiality & integrity is Scheme ☐.

The modification is:

Question 3 ***Why do RSA signatures need a hash?*** **(20 min)**

This question explores the design of standard RSA signatures in more depth. To generate RSA signatures, Alice first creates a standard RSA key pair: (n, e) is the RSA public key and d is the RSA private key, where n is the RSA modulus. For standard RSA signatures, we typically set e to a small prime value such as 3; for this problem, let $e = 3$.

To generate a **standard** RSA signature S on a message M, Alice computes $S = H(M)^d \bmod n$. If Bob wants to verify whether S is a valid signature on message M, he simply checks whether $S^3 = H(M) \bmod n$ holds. d is a private key known only to Alice and $(n, 3)$ is a publicly known verification key that anyone can use to check if a message was signed using Alice's private signing key.

Suppose we instead used a **simplified** scheme for RSA signatures which skips using a hash function and instead uses M directly, so the signature S on a message M is $S = M^d \bmod n$. In other words, if Alice wants to send a signed message to Bob, she will send (M, S) to Bob where $S = M^d \bmod n$ is computed using her private signing key d.

(a) With this **simplified** RSA scheme, how can Bob verify whether S is a valid signature on message M? In other words, what equation should he check, to confirm whether M was validly signed by Alice?

(b) Mallory learns that Alice and Bob are using the **simplified** signature scheme described above and decides to trick Bob into beliving that one of Mallory's messages is from Alice. Explain how Mallory can find an (M, S) pair such that S will be a valid signature on M.

You should assume that Mallory knows Alice's public key n, but not Alice's private key d. The message M does not have to be chosen in advance and can be gibberish.

(c) Is the attack in part (b) possible against the **standard** RSA signature scheme (the one that includes the cryptographic hash function)? Why or why not?