Question 1 ***Key Exchange Protocols*** ()

Recall that in a Diffie-Hellman key exchange, there are values a, b, g and p. Alice computes $g^a \bmod p$ and Bob computes $g^b \bmod p$.

Q1.1 Which of these values (a, b, g, and p) are publicly known and which must be kept private?

Q1.2 Mallory can eavesdrop, intercept, and modify everything sent between Alice and Bob. Alice and Bob perform Diffie-Hellman to agree on a shared symmetric key K. After the exchange, Bob gets the feeling something went wrong and calls Alice. He compares his value of K to Alice's and realizes that they are different. Explain what Mallory has done.

Now consider the following key exchange protocols which can be used by Alice (A) and Bob (B) to agree upon a shared key, K.

ElGamal-Based Key Exchange

Message 1	$A \rightarrow B$:	$\{K\}_{PK_B}$
	Key exchanged	
Message 2	$A \leftarrow B$:	$\{secret1\}_K$
Message 3	$A \rightarrow B$:	$\{secret2\}_K$

Diffie-Hellman Key Exchange

Message 1	$A \rightarrow B$:	$g^a \mod p$
Message 2	$A \leftarrow B$:	$g^b \mod p$
	Key exchanged	
	$K = g^{ab} \mod p$	
Message 3	$A \leftarrow B$:	$\{secret1\}_K$
Message 4	$A \rightarrow B$:	$\{secret2\}_K$

Some additional details:

- PK_B is Bob's long-lived public key.
- K, the Diffie-Hellman exponents a and b, and the messages themselves are destroyed once all messages are sent. That is, these values are not stored on Alice and Bob's devices after they are done communicating.

Eavesdropper Eve records all communications between Alice and Bob, but is unable to decrypt them. At some point in the future, Eve is lucky and manages to compromise Bob's computer.

Q1.1 Is the confidentiality of Alice and Bob's prior ElGamal-based communication in jeopardy?

Q1.2 What about Alice and Bob's Diffie-Hellman-based communication?

Question 2 ***Public Key Encryption*** ()

The El Gamal encryption scheme is reproduced below:

- **Key Generation**: public key $= (g, h, p)$, where $h = g^k \pmod{p}$, private key $= k$
- **Encryption**: $c = (c_1, c_2) = (g^r \bmod p, m \times h^r \bmod p)$, where r is randomly sampled from $\{1, \ldots, p-1\}$.
- **Decryption**: $m = c_1^{-k} \times c_2 \pmod{p}$

Look at each scenario below and select the appropriate options.

Q2.1 With El Gamal, it is not a problem if the adversary can learn the value of g somehow.

◯ (A) True

◯ (B) False

◯ (C) ——

◯ (D) ——

◯ (E) ——

◯ (F) ——

Q2.2 With El Gamal, it is not a problem if the value r used during encryption is accidentally revealed after the encryption is complete.

◯ (G) True

◯ (H) False

◯ (I) ——

◯ (J) ——

◯ (K) ——

◯ (L) ——

Now imagine that Alice (A) and Bob (B) want to communicate over an insecure network and they know each other's public key. Consider the following message exchange:

A: Hey Bob, it's Alice. How many dollars do I owe you?
B: 10000

The message is encrypted with Alice's public key using ElGamal encryption.
Alice decrypted this successfully, but suddenly remembered that she only owed Bob $100.

Q2.1 Assume Bob would not lie. How did an attacker tamper with the message?

Q2.2 What could Bob have additionally sent that would've stopped this attack?

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

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 $e = 3$.

Suppose we used a **simplified** scheme for RSA signatures that skips using a hash function and instead uses message 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.

Q3.1 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?

Q3.2 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.

Q3.3 Is the attack in Q3.2 possible against the **standard** RSA signature scheme (the one that includes the cryptographic hash function)? Why or why not?