Cryptography III

Question 1 ***Diffie–Hellman key exchange*** **(15 min)**

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 $g^b \bmod p$.

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

(b) Eve can eavesdrop on everything sent between Alice and Bob, but can't change anything. Alice and Bob run Diffie-Hellman and have agreed on a shared symmetric key K. However, Bob accidentally sent his b to Alice in plain text. If Eve viewed all traffic since the beginning of the exchange, can she figure out what K is?

(c) Mallory can not only view all Alice—Bob communications but also intercept and modify it. 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.

Question 2 ***Perfect Forward Secrecy*** **(15 min)**

Alice (A) and Bob (B) want to communicate using some shared symmetric key encryption scheme. Consider the following key exchange protocols which can be used by A and B to agree upon a shared key, K_{ab}.

El Gamal-Based Key Exchange

Message 1	$A \rightarrow B$:	$\{K_{ab}\}_{K_B^{pub}}$
	Key exchanged	
Message 2	$A \leftarrow B$:	$\{secret1\}_{K_{ab}}$
Message 3	$A \rightarrow B$:	$\{secret2\}_{K_{ab}}$

Diffie-Hellman Key Exchange

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

Some additional details:

- K_B^{pub} is Bob's long-lived public key.
- K_{ab}, 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's 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.

(a) Is the confidentiality of Alice and Bob's prior El Gamal-based communication in jeopardy?

(b) What about Alice and Bob's Diffie-Hellman-based communication?

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?