Question 1 ***Ra's Al Gamal***

Recall the ElGamal scheme from lecture:

- KeyGen() $= (b, B = g^b \bmod p)$
- Enc$(B, M) = (C_1 = g^r \bmod p, C_2 = B^r \times M \bmod p)$

Q1.1 Is the ciphertext (C_1, C_2) decryptable by someone who knows the private key b? If you answer yes, provide a decryption formula. You may use C_1, C_2, b, and any public values.

○ Yes ○ No

Q1.2 Consider an adversary that can efficiently break the discrete log problem. Can the adversary decrypt the ciphertext (C_1, C_2) without knowledge of the private key? If you answer yes, briefly state how the adversary can decrypt the ciphertext.

○ Yes ○ No

Q1.3 Consider an adversary that can efficiently break the Diffie-Hellman problem. Can the adversary decrypt the ciphertext (C_1, C_2) without knowledge of the private key? If you answer yes, briefly state how the adversary can decrypt the ciphertext.

○ Yes ○ No

Question 2 ***Dual Asymmetry***

Alice wants to send two messages M_1 and M_2 to Bob, but they do not share a symmetric key.

Assume that p is a large prime and that g is a generator $\bmod p$, like in ElGamal. Assume that all computations are done modulo p in Scheme A.

Q2.1 Scheme A: Bob publishes his public key $B = g^b$. Alice randomly selects r from 0 to p - 2. Alice then sends the ciphertext $(R, S_1, S_2) = (g^r, M_1 \times B^r, M_2 \times B^{r+1})$.

Select the correct decryption scheme for M_1:

- ○ $R^{-b} \times S_1$
- ○ $B^{-b} \times S_1$
- ○ $R^b \times S_1$
- ○ $B^b \times S_1$

Q2.2 Select the correct decryption scheme for M_2:

- ○ $B^{-1} \times R^{-b} \times S_2$
- ○ $B^{-1} \times R^b \times S_2$
- ○ $B \times R^{-b} \times S_2$
- ○ $B^{-1} \times R \times S_2$

Q2.3 Is Scheme A IND-CPA secure? If it is secure, briefly explain why (1 sentence). If it is not secure, briefly describe how you can learn something about the messages.

Clarification during exam: For Scheme A, in the IND-CPA game, assume that a single plaintext is composed of two parts, M_1 and M_2.

- ○ Secure
- ○ Not secure

Q2.4 Scheme B: Alice randomly chooses two 128-bit keys K_1 and K_2. Alice encrypts K_1 and K_2 with Bob's public key using RSA (with OAEP padding) then encrypts both messages with AES-CTR using K_1 and K_2. The ciphertext is $\text{RSA}(\mathsf{PK}_{\mathsf{Bob}}, K_1 \| K_2), \text{Enc}(K_1, M_1), \text{Enc}(K_2, M_2)$.

Which of the following is required for Scheme B to be IND-CPA secure? Select all that apply.

- ☐ K_1 and K_2 must be different
- ☐ A different IV is used each time in AES-CTR
- ☐ M_1 and M_2 must be different messages
- ☐ M_1 and M_2 must be a multiple of the AES block size
- ☐ M_1 and M_2 must be less than 128 bits long
- ☐ None of the above

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?