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

Solution: The decryption formula is $M = C_1^{-b} \times C_2$.

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

Solution: An adversary that can break the discrete log problem can recover r from $C_1 = g^r$ or b from $B = g^b$, so they can compute g^{br} and recover the original message.

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

Solution: An adversary that can break the Diffie-Hellman problem can recover g^{br} from $C_1 = g^r$ and $B = g^b$, so they can recover the original message.

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:

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

Solution:

$S_1 = M_1 \times B^r$	Given in the question
$S_1 = M_1 \times g^{br}$	Substitute $B = g^b$
$M_1 = g^{-br} \times S_1$	Multiply both sides by g^{-br}
$M_1 = R^{-b} \times S_1$	Substitute $R = g^r$

Q2.2 Select the correct decryption scheme for M_2:

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

Solution:

$S_2 = M_2 \times B^{r+1}$	Given in the question
$S_2 = M_2 \times g^{b(r+1)}$	Substitute $B = g^b$
$S_2 = M_2 \times g^{br+b}$	Exponentiation properties
$M_2 = g^{-br-b} \times S_2$	Multiply both sides by g^{-br-b}
$M_2 = g^{-br} \times g^{-b} \times S_2$	Exponentiation properties
$M_2 = R^{-b} \times B^{-1} \times S_2$	Substitute $B = g^b$ and $R = g^r$
$M_2 = B^{-1} \times R^{-b} \times S_2$	Rearrange terms

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

Solution: This scheme is not IND-CPA secure. Eve can determine if $M_1 = M_2$ by checking if $S_2 = S_1 \times B$.

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

Solution:

A: False. Because Enc is an IND-CPA secure encryption algorithm, the key does not need to be changed between two encryptions.

B: True. AES-CTR requires that a unique nonce is used for each encryption, or it loses its confidentiality guarantees.

C: False. A secure encryption algorithm would not leak the fact that two messages are the same.

D: AES-CTR can encrypt any length of plaintext. Padding is not needed in AES-CTR.

E: AES-CTR can encrypt any length of plaintext.

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?

> **Solution:** $S^3 = M \bmod n$.

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.

> **Solution:** Mallory should choose some random value to be S and then compute $S^3 \bmod n$ to find the corresponding M value. This (M, S) pair will satisfy the equation in part (a).
>
> **Alternative solution:** Choose $M = 1$ and $S = 1$. This will satisfy the equation.

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?

> **Solution:** This attack is not possible. A hash function is one way, so the attack in part (b) won't work: we can pick a random S and cube it, but then we'd need to find some message M such that $H(M)$ is equal to this value, and that's not possible since H is one-way.
>
> Comment: This is why the real RSA signature scheme includes a hash function: exactly to prevent the attack you've seen in this question.