Question 1

1. (5 points) You are given an El Gamal ciphertext $c = (x, y)$ for which you know the public parameters: the group $\mathcal{G}$, the generator g and the public key h. Explain exactly how you can compute a new ciphertext $c' = (x', y')$ such that $c \neq c'$ that will decrypt to the same plaintext. Note: you not know the secret key k ($h = g^k$).

   Hint: this is know as re-randomisation.

We are given an El Gamal ciphertext c = (x, y). Based on the El Gamal Encryption Algorithm:

- In order to encrypt a message $m \in G$ and obtain ciphertext c = (x, y), we compute $x=g^r$ and $y=h^r m$, where g is the generator, $r \in (0,q-1)$ is the randomness, and $h=g^x$ is the public key.
- In order to decrypt ciphertext c = (x, y) and obtain the message $m \in G$, we compute $(h^r m/g^{rx}) = (h^r m/h^r) = m$.

We want to compute a new ciphertext c' = (x', y'), such that c /= c', that will decrypt to the same plaintext. We are given that we know the public parameters: the group G, the generator g, and the public key h. Therefore, we have to perform re-randomisation as follows:

- We choose a random number s.
- We have to encrypt the new ciphertext c' = (x', y') using that s. Thus, we have: $x'=g^{s+r}$, and $y'=h^{s+r}m$.
- Now we have to perform decryption: $(h^{s+r}m/g^{(s+r)x}) = (h^{s+r}m/h^s h^r)$, where $g^{(s+r)x} = g^{xs}g^{xr} = h^s h^r$. So we have, $(h^s h^r m/h^s h^r) = m$.

Thus, we have computed a new ciphertext c' = (x', y'), such that c /= c', that decrypts to the same plaintext m.

Question 2

2. (3 points) Explain how this can be used to allow a server to shuffle and anonymise a set of ciphertexts.

When re-randomising, we choose a new random number s for each ciphertext. A server can then shuffle the new re-randomised ciphertexts by using secure shuffling algorithms, such as random permutation of the order of ciphertexts. The server then sends these shuffled re-randomised ciphertexts to the corresponding recipients. Each shuffled re-randomised ciphertext can only be decrypted by the recipient's corresponding secret key, as shown in Question 1. Re-randomisation and shuffling make it is infeasible to map the shuffled ciphertexts to their original positions (i.e., before shuffling) in the set. This in turn ensures anonymity of the messages.

Question 3

3. (5 points) Recall the multiplicative homomorphic property of El Gamal:

$$\mathcal{E}(m) \cdot \mathcal{E}(n) = \mathcal{E}(m \cdot n)$$

   Note: multiplication of El Gamal ciphertexts is defined as pairwise multiplication:

$$c_1 \cdot c_2 = (x_1, y_1) \cdot (x_2, y_2) = (x_1 \cdot x_2, y_1 \cdot y_2)$$

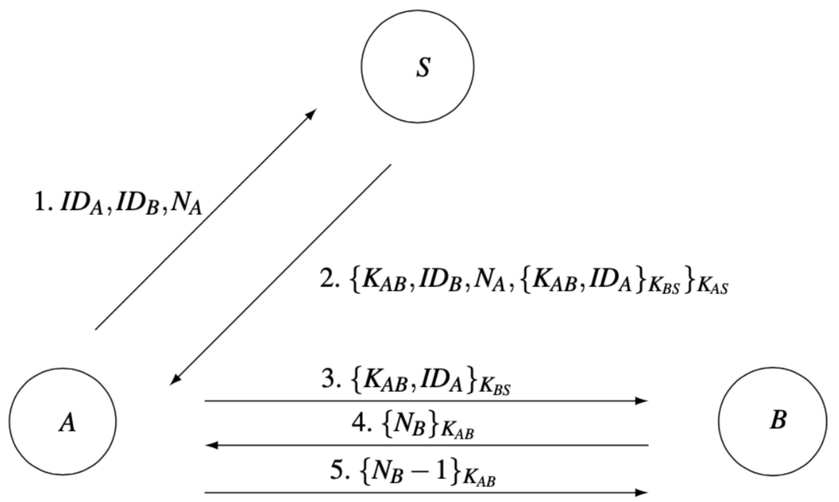
   Explain how re-encryption can be thought of as a consequence of this homomorphism.

   Hint: express re-encryption as multiplication of the given ciphertext c by the encryption of a special value in $\mathcal{G}$.

We know that re-encryption entails the encryption of a pre-existing ciphertext (originally encrypted under one public key) using a different public key, without the need to decrypt the original ciphertext. Re-encryption can be expressed as the multiplication of the given ciphertext c (i.e., the originally encrypted plaintext) by the encryption of a special value in group G. Let's denote the latter with $\mathcal{E}(b)$, and introduce randomness $r \in (0,q-1)$ to it. Thus we have, $\mathcal{E}(b)=(g^r, h^r)$ for generator $g \in G$ and public key h.
We have that the original ciphertext is $\mathcal{E}(c)=(x,y)$. Then, re-encrypting c becomes $\mathcal{E}_{re\text{-}encrypt}(c)= \mathcal{E}(c) \cdot \mathcal{E}(b)=(x \cdot g^r, y \cdot h^r)$.
The new ciphertext $(x \cdot g^r, y \cdot h^r)$ is an encryption of the original plaintext under the new public key h^r. Furthermore, re-encryption is done without decrypting or altering the original plaintext, thanks to the homomorphic property of El Gamal. Therefore, re-encryption can be thought of as a consequence of the El Gamal's multiplicative homomorphic property.

Question 4

4. (7 points) How can the adversary launch an attack on the protocol defined in the Figure below? Note that the adversary may have knowledge of an old session key K'_{AB} (due to leak) and the whole transcript of protocol execution in which K'_{AB} has been established.



Hint: a similar attack has been shown in lecture on AKEs.

The protocol defined in the figure is the Needham-Schroeder's shared key protocol.
If the adversary has knowledge of an old session key K'_{AB} (due to leak) and the whole transcript of protocol execution in which K'_{AB} has been established, then the adversary can do an attack on P4 (i.e., $\{N_B\}_{K_{AB}}$).
Knowing the whole transcript of protocol execution in which K'_{AB} has been established means that the adversary knows all messages exchanged between the parties and the trusted third party (TTP). Thus, the adversary can launch an impersonation attack as follows:

- Let's say C is the ID of the adversary's communication entity.
- C performs an attack on P4 (i.e., $\{N_B\}_{K_{AB}}$) as follows:
  - C uses the old session key K'_{AB} to masquerade as A. P3 now changes to $\{K'_{AB}, A\}_{K_{BS}}$. Thus, C is able to persuade B to also use the old session key K'_{AB}.
  - C replays the whole transcript of the previous protocol execution (in which K'_{AB} has been established).
  - Due to the valid session key K'_{AB} and the valid replay of previous transcript, B believes it's communicating with A, and thus, B ends up establishing a new session key with C. P4 now changes to $\{N_B\}_{K'_{AB}}$. P5 now changes to $\{N_B\text{-}1\}_{K'_{AB}}$.