**Q1: Homomorphic Encryption (10 points)**

(a) (5 points) Show that the above Elgamal encryption scheme is homomorphic with respect to multiplication.

(b) (5 points) Show that the above Elgamal encryption scheme is not homomorphic with respect to addition.

Assume:

Show that:

Recall due to (a): , or specifically show:

To equate them, Let

🡪 True for

Hence, Elgamal Encryption scheme is not homomorphic with respect to addition.

**Q2: Homomorphic-Based Yao Millionaire Problem (15 points)**

(a) (5 points) Explain why does the Homomorphic based protocol for Yao’s millionaire problem (in Lecture 11 slides 22-23) fail when using unpadded RSA?

Because it is insecure, as unpadded RSA produces the same plaintext for the same ciphertext. Furthermore, the matrix T will produce the same encryption for 1 since it is calculated using , which will reveal Sender information to the Receiver.

(b) (10 points) Design a protocol that uses unpadded RSA. Verify that your protocol works by implementing your proposed protocol using the notebook file (”Yao RSA.ipnyb”).

Done.

**Q3: Oblivious Transfer (OT) (10 pts)**

1. (10 points) Design a simple protocol for 1-out-of-n OT starting from 1-out-of-2 OT. Assume that both Alice and Bob are honest-but-curious. i.e., they follow the protocol but from time to time they collect extra information looking for exposing private data about each other. In your protocol, Alice and Bob can access the 1-out-of-2 functionality n times. Explain your protocol n details (Hint: Think of how to extend 1-out-of-2 to 1-out-of-3 and then generalize it to 1-out-of-n)

The sender will have *n* messages, and the receiver has an index *i*, and the receiver wishes to receive the *i*-th message among the sender's messages, without the sender learning *i*. Furthermore, the sender wants to ensure that the receiver receive only one of the *n* messages.

|  |  |
| --- | --- |
|  |  |
| Step 1 Alice | 1. Generates an RSA key paid PK = (N,e) and SK = (d) 2. Generate n random values, r0, r1, r2 … , and she sends them to Bob along with PK |
| Step 2 Bob | Bob picks a value (v) between 0 and n, and select |
| Step 3 Bob | Bob generates a random value k and blinds it with by computing:  and sends it to Alice |
| Step 4 Alice | Alice does not know which of  Bob did choose. Alice computes |
| Step 5 Alice | Alice combines the n secret messages with each of the possible keys, i.e.  , and she sends them to Bob |
| Step 6 Bob | Bob knows which of the n messages can be unblinded with , so he is able to compute exactly one of the messages |