# KFUPM College of Computer Science and Engineering Computer Engineering Department COE 449: Privacy Enhancing Technologies

**Faris Hijazi s201578750**

Fall 2019 (191)

Assignment 3: Due date Thursday 21/11/2019

## Tasks

**Question1: Oblivious Transfer (OT) (50 pts)**

(a) (20 points) Implement 1-out-of-2 with Socket programming using a programming language of your choice. The program should be run on two machines (or two terminals) to emulate a sender, i.e., Alice, and a receiver, i.e., Bob. Alice will generate two messages, e.g., two numbers, and will send exactly one number to Bob. On other hand, Bob needs to choose which number he wants to receive. When received, Bob should know nothing about the other number with Alice, and Alice should not know which number Bob chose.

(b) (20 points) Give a simple, deterministic protocol for 1out-of-n OT. Assume that both Alice and Bob are honestbut-curious. In your protocol, Alice and Bob can access the 1-out-of-2 functionality n times. (Hint: Think of how to extend 1-out-of-2 to 1-out-of-3 and then generalize it to 1-out-of-n)

We have 2 choices: either to extend 1-out-of-2 OT and do it iteratively, or just parameterize the 1-out-of-2 OT to 1-out-of-n OT, such that we send *n* messages instead of 2, and *b* is a *log2(n)*-bit integer, and can take any value in the range [0, n-1]

(c) (10 points) Implement the 1-out-of-n OT protocol in the previous task.

**Question2: Homomorphic Encryption (20 points)** Consider Elga-

mal encryption ([http://homepages.math.uic.edu/~leon/ mcs425-s08/handouts/el-gamal.pdf)](http://homepages.math.uic.edu/~leon/mcs425-s08/handouts/el-gamal.pdf)

1. (10 points) Is Elgamal encryption additive homomorphic?

no

1. (10 points) Is Elgamal encryption mulitplicative homomorphic?

Yes, according to this artictle: <https://nvotes.com/multiplicative-vs-additive-homomorphic-elgamal/>

**Question3: Dining Cryptographer (30 points)** In the lecture, we discussed the Dining Cryptographers protocol. In this problem, we will explore how to use that protocol as a building block to construct a general protocol for anonymous communication. Consider a group of n agents. [[1]](#footnote-1)

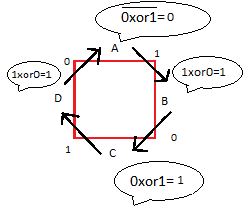
1. Describe a protocol using which one of the n agents can send an m-bit message. Explain informally why the protocol is correct (i.e., all agents receive exactly the message that was sent) and anonymous (i.e., none of the other agents have any clue who the real sender is).

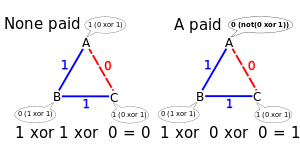
We can use the same protocol of sending a single bit, but we perform the same protocol *m* times, once for each bit.

The reason this protocol works is because the setup for each bit is completely independent of the previous one, this if the protocol is valid for a single bit, then it is valid for *m* bits.

1. Sketch a prove that the anonymity is preserved by the protocol for the case where n=4 and m=1.  
   (You need to show that from the point of 2 view of any nonsender, the probability of any of the other agents being the sender is 1/3).

The protocol: each one flips a coin and tells the result to his left neighbor, if there is an **odd** number of “same”, then everyone was honest

Bellow is a sketch for the 3-person DC Protocol.



Here we have an **odd** # of “same”s (1)

So one of them is paying (A).

POV of C and D:

C can’t know what the result for A is since he didn’t see the toss between D and A

Same thing for D for not knowing A, B or C are paying since it didn’t see the ones between A and B or B and C

1. [↑](#footnote-ref-1)