Exercise 10

1 Reversing Oblivious Transfer

Suppose Alice and Bob have a primitive that performs a 1-out-of-2 Oblivious Transfer (OT) for *one bit* from Alice (as sender) to Bob (as receiver). How can they reverse the direction of oblivious transfer?

Describe a protocol that uses their OT primitive any number of times and implements 1-out-of-2 OT of bits from Bob to Alice. You may use additional error-free communication in either direction, secret-key cryptosystems like block ciphers, and possibly other methods.

2 Secure Two-Party Computation

Consider secure two-party computation between Alice with input x and Bob with input y. A protocol for secure computation generates $(a, b) = f(x, y)$ for a function f known to both parties, such that Alice privately obtains a and Bob privately obtains b. Neither one of them learns more information about the input of the other one than what follows from the private output.

Which ones of the following functionalities can be computed simply by exchanging messages? Describe a protocol that does not using a secure computation primitive for those cases.

a) For $x, y \in \{0, 1\}$, compute $a = b = x \cdot y$ in $GF(2)$ (i.e., multiplication of bits).

b) For $x, y \in \mathbb{Z} \setminus \{0\}$, compute $a = b = x \cdot y$ in $\mathbb{Z}$.

c) For $x, y \in \{0, 1\}^k$, compute $a = b = x \oplus y$ in $GF(2^k)$ (i.e., interpret the values as bit strings).

d) For $x, y \in \{0, 1\}$, compute $a = x \oplus y$ and $b = x \cdot y$ in $GF(2)$.