CS670: Assignment 2

Instructions

- Using LaTeX to typeset your solutions will fetch +5 bonus marks in the assignment.
- You can ask the instructor for help in the assignment during office hours or via appointment.
- All help will stop 48 hours before the assignment deadline.
- **Deadline: March 27th 2024, EOD**

1 Secure Multiparty Computation Using Cards

In this question we will playing cards to do MPC. Alice and Bob want to decide if they should go on a date. However they want to reveal their choices only if Alice and Bob like each other. They come up with the following protocol which uses playing cards. Alice and Bob both have a "King" and "Ace". Here is an incomplete protocol to decide if they want to go on a date. Your task is to complete the protocol.

1. If Alice likes Bob she will place the cards face down in order: "King", "Ace".
2. If Alice does not like Bob she will place the cards face down in order: "Ace", "King".
3. If Bob does not like Alice he will place the cards face down in order: "King", "Ace".
4. If Bob likes Alice he will place the cards face down in order: "Ace", "King".
5. In between their cards, they place an additional: "Ace" facedown.

Write the remaining steps of the protocol and explain why it works. **10 marks**

2 Secure Multiparty Computation on Secret Shares

Secure Multiparty Computation for Dot Products

First recall the Du-Atallah protocol for multiplication. P_0 holds (x_0, y_0) and P_1 holds (x_1, y_1). They compute z_0 and z_1 respectively such that $z_0 + z_1 = (x_0 + x_1) \cdot (y_0 + y_1)$. The protocol works as follows:

1. The protocol begins with a (trusted) dealer P_2 sampling five random values $(X_0, X_1, Y_0, Y_1, \alpha)$.
2. P_2 sends $(X_0, Y_0, X_0 \cdot Y_1 + \alpha)$ to P_0 and P_2 sends $(X_1, Y_1, X_1 \cdot Y_0 - \alpha)$ to P_1.
3. P_0 sends $(x_0 + X_0)$ and $(y_0 + Y_0)$ to P_1. Similarly, P_1 sends $(x_1 + X_1)$ and $(y_1 + Y_1)$ to P_0.
4. Next, P_0 computes, $z_0 \leftarrow x_0 \cdot (y_0 + (y_1 + Y_1)) - Y_0 \cdot (x_1 + X_1) + (X_0 \cdot Y_1 + \alpha)$
5. Next, P_1 computes, $z_1 \leftarrow x_1 \cdot (y_1 + (y_0 + Y_0)) - Y_1 \cdot (x_0 + X_0) + (X_1 \cdot Y_0 - \alpha)$
6. Now observer, $z_0 + z_1 = (x_0 + x_1) \cdot (y_0 + y_1)$

Now answer the following questions:

1. Suppose the time to send a message from P_i to P_j is $x_{i,j}$. In the most optimized implementation, how much time will the implementation spend on communication. Show your calculations. **5 marks**
2. Suppose that P_0 and P_1 hold shares of vectors $\vec{x}$ and $\vec{y}$. Write an MPC protocol to compute the shares of $z \leftarrow \langle \vec{x}, \vec{y} \rangle$. Assume the presence of trusted dealer P_2. **10 marks**

Implementing an `xorif` functionality

We would like to implement the following function in MPC. **15 marks**

```
// Define a function to perform XOR operation conditionally
uint64_t xor_if(uint64_t x, uint64_t y, bool b) {
    if (b == 1)
        x = x ^ y;    // XOR operation
    return x;
}
```

To describe it in more detail,

1. P_0 and P_1 hold (x_0, y_0, b_0) and (x_1, y_1, b_1) respectively. They have the property that, $x = x_0 \oplus x_1$, $y = y_0 \oplus y_1$, and $b = b_0 \oplus b_1$.
2. Their goal is compute z_0 and z_1 respectively such that, both the following conditions hold true:
   - $z_0 \oplus z_1 = x$ if $(b_0 \oplus b_1 = 0)$
   - $z_0 \oplus z_1 = x \oplus y$ if $(b_0 \oplus b_1 = 1)$

3 Optimizations in Garbled Circuits

Point-and-Permute Optimization

While doing the point-and-permute optimization, for each gate we need to send four cipher texts. However, there are some (albeit very few) gates for which you can get away with sending just two cipher texts. What gates are those and why is it the case? **5 marks**

Choice of the encryption algorithm

In the plain non-optimized Garbled Circuits, we cannot use One Time Pads as the cipher. Explain why? Why does this change, when we use Point-and-Permute Optimization. **10 marks**