CS 601.442/642 – Modern Cryptography Instructor: Abhishek Jain

Homework 8

Deadline: December 1, 2020; 11:59 AM

1. **(10 points)** Let Alice and Bob be two parties with inputs $a \in \mathbb{Z}_q$ and $b \in \mathbb{Z}_q$, respectively. They wish to check if their inputs are equal, i.e., whether $a = b$. They want to do this while making sure that they do not learn any other information about the other party's input. In other words, if $a \neq b$, then Alice should not learn b and Bob should not learn a.

   Let $\mathbb{G}$ be a cyclic group of prime order q with generator g. They run the following protocol:

   - Alice samples a random value $r \leftarrow\$ \mathbb{Z}_q$. It then computes $X = g^r$ and $Y = g^{ar}$. It sends (X, Y) to Bob.
   - Bob computes X^b. It outputs 1 if $X^b = Y$, and 0 otherwise.

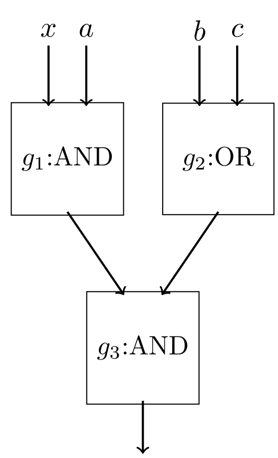
   Explain why this protocol is not secure against semi-honest Bob.

2. **(15 points)** Let Alice and Bob have inputs a and b, respectively. They want to securely send $(a + b)$ to a third-party Carol. Devise a protocol where Alice and Bob are only allowed to send **at most one message to each other** and **at most one message each to Carol**. Your protocol should satisfy all of the following security properties:

   - *Security against Semi-honest Alice:* Alice should not learn b.
   - *Security against Semi-honest Bob:* Bob should not learn a.
   - *Security against Semi-honest Carol:* Carol should not learn a and b.

   Argue that your protocol indeed satisfies all three security conditions, and gives the correct output to Carol. (You don't need to give a formal proof).

3. **(15 points)** Let C be a Boolean circuit as shown in the following figure.



Let $(\mathsf{Garble}, \mathsf{Eval})$ be the garbling scheme discussed in class. Recall that the $\mathsf{Garble}()$ function, when given this Boolean circuit C as input, outputs the following:

$$(\hat{G} = \{\hat{g}_1, \hat{g}_2, \hat{g}_3\}, \hat{\mathsf{In}} = \{K_0^1, K_1^1, K_0^2, K_1^2, K_0^3, K_1^3, K_0^4, K_1^4\}) \leftarrow \mathsf{Garble}(C),$$

where $\hat{G}$ is the set of 3 garbled gates and $\hat{\mathsf{In}}$ is the set of wire keys for the 4 input wires in this circuit. In this question, we will see that the privacy of inputs in a garbled circuit does not hold if the adversary has both the keys for a wire.

Consider an adversary who knows the description of C, garbled gates $\hat{G}$ and input wire keys $\{K_0^1, K_1^1, K_a^2, K_b^3, K_c^4\}$. Note that the adversary gets both the input wire keys for the first input wire, and only one key for each of the remaining 3 input wires. Also note that the values a, b, c are not known to the adversary.

Show how this adversary can use this information to learn at least one out of a, b or c.

(**Hint: Use the truth table of the gates to derive information.**)

4. **(10 points)** Recall the garbled circuit construction discussed in class. Let k_b^w be the key for the w^{th} wire corresponding to input b. For every gate g in C with input wires (i, j), output wire ℓ, the garbled gate is computed as follows:

First Input	Second Input	Output
k_0^i	k_0^j	$z_1 = \mathsf{Enc}_{k_0^i}(\mathsf{Enc}_{k_0^j}(k^\ell_{g(0,0)}))$
k_0^i	k_1^j	$z_2 = \mathsf{Enc}_{k_0^i}(\mathsf{Enc}_{k_1^j}(k^\ell_{g(0,1)}))$
k_1^i	k_0^j	$z_3 = \mathsf{Enc}_{k_1^i}(\mathsf{Enc}_{k_0^j}(k^\ell_{g(1,0)}))$
k_1^i	k_1^j	$z_4 = \mathsf{Enc}_{k_1^i}(\mathsf{Enc}_{k_1^j}(k^\ell_{g(1,1)}))$

The garbled gate is set as $\hat{g} = \mathsf{RandomShuffle}(z_1, z_2, z_3, z_4)$.

The encryption scheme used in this construction is a multi-message secure "special" secret-key encryption scheme as discussed in the class. Note, however, that in the garbling process, each key of the encryption scheme is used to encrypt only *two* messages. For example key k_0^i is used when computing both z_1 and z_2.

We now consider a modified garbling scheme where instead of using a multi-message secure encryption scheme, we use "special" one-time pads with the extra decryption property as defined for the special encryption scheme discussed in the class. This special one-time pad encryption can be obtained in a similar way, by first appending zeroes to the message and then encrypting the modified message using regular one-time pads. Recall that since one-time pads do not require any cryptographic assumptions, the advantage of designing a garbling scheme that only uses these "special" one-time pads is that it will be secure against *unbounded adversaries.*

Since one-time pads are only one-message secure, they cannot be used to encrypt two-messages. Therefore, we make the following modifications to the garbling scheme. First we assume that each key in the modified scheme is of the form $k_b^w = k_b^{w,0} \| k_b^{w,1}$. Then, for every gate g in C with input wires (i, j), output wire ℓ, the garbled gate is computed as follows:

First Input	Second Input	Output
$k_0^{i,0}$	$k_0^{j,0}$	$z_1 = k_0^{i,0} \oplus k_0^{j,0} \oplus \left(0^n \| k_{g(0,0)}^{\ell}\right)$
$k_0^{i,1}$	$k_1^{j,0}$	$z_2 = k_0^{i,1} \oplus k_1^{j,0} \oplus \left(0^n \| k_{g(0,1)}^{\ell}\right)$
$k_1^{i,0}$	$k_0^{j,1}$	$z_3 = k_1^{i,0} \oplus k_0^{j,1} \oplus \left(0^n \| k_{g(1,0)}^{\ell}\right)$
$k_1^{i,1}$	$k_1^{j,1}$	$z_4 = k_1^{i,1} \oplus k_1^{j,1} \oplus \left(0^n \| k_{g(1,1)}^{\ell}\right)$

The garbled gate is set as $\hat{g} = \mathsf{RandomShuffle}(z_1, z_2, z_3, z_4)$. Note that this modification, ensures that each half of the key is only used once.

Calculate the ratio between the length of an input wire key and the length of a wire key on the last layer, when garbling a circuit of depth d, using this modified construction.