Problems are assigned sporadically and are due 2 weeks after being assigned. This file will be updated as new problems are assigned.

- I encourage you to “batch” your solutions, and submit several problems together.
- Working in groups is fine with me. However, please submit individual solutions. Your write up should be based on your own understanding.
- Solutions must be typed. I strongly suggest \texttt{LaTeX}, or, if you haven’t yet been assimilated to \texttt{LaTeX}, \texttt{LyX} offers a gentler learning curve.

1. [assigned 8 Jan] Which of the following functions are negligible? ($\log$ means $\log_2$)

   (a) $f(n) = n^2/2^n/2^n$
   (b) $f(n) = 1/n\log n$
   (c) $f(n) = 1/n\log\log n$
   (d) $f(n) = 1/(\log n)^4$
   (e) $f(n) = 1/2\sqrt{n}$
   (f) $f(n) = n\log n/2^n$
   (g) $f(n) = 1/2^{4\log n}$
   (h) $f(n) = 1/2(\log n)^4$
   (i) $f(n) = (n \mod 2)/n^2$

2. [assigned 8 Jan] Let $X = \{X_n\}_{n \in \mathbb{N}}$, $Y = \{Y_n\}_{n \in \mathbb{N}}$, $Z = \{Z_n\}_{n \in \mathbb{N}}$. We will consider the definition of computational indistinguishability from [G02].

   Show that if $X$ and $Y$ are computationally indistinguishable, and $Y$ and $Z$ are computationally indistinguishable, then $X$ and $Z$ are computationally indistinguishable.

3. [assigned 15 Jan] When considering secure computation, is it a violation of security for there to exist an adversary that can achieve the following attack against a protocol?

   If it’s an attack, show how the attack cannot be done in the ideal world. If it’s not an attack, show how the “attack” can be done in the ideal world.

   (a) Alice & Bob each hold an input in $\{0, \ldots, N-1\}$ and wish to compute their sum modulo $N$. A malicious adversary corrupts Alice and learns Bob’s input in its entirety.

   (b) Alice & Bob each hold an input in $\{0, \ldots, N-1\}$ and wish to compute their sum modulo $N$. A malicious adversary corrupts Alice and forces Bob to always output zero.

   (c) Alice holds $x$ and Bob holds $y$, where $x, y \in \{0, \ldots, 7\}$, and wish to compute whether $x < y$ (i.e., they get output 1 if $x < y$ and output 0 otherwise). A malicious adversary corrupts Alice and learns the most significant bit of $y$.

   (d) Alice holds $x$ and Bob holds $y$, where $x, y \in \{0, \ldots, 7\}$, and wish to compute whether $x < y$ (i.e., they get output 1 if $x < y$ and output 0 otherwise). A malicious adversary corrupts Alice and causes Bob to output the least significant bit of his input $y$.

   (e) Alice holds $x$ and Bob holds $y$, where $x, y \in \{0, 1\}^n$. They wish to compute the inner product of those strings modulo 2: $\sum_{i=1}^n x_i y_i \pmod{2}$. A malicious adversary corrupts Alice and learns whether the $y$ string has a majority of 0s or a majority of 1s (assume that $n$ is odd).

   (f) Alice and Bob each hold a single bit and wish to compute the boolean-AND of their inputs. A malicious adversary corrupts Alice and learns Bob’s input in its entirety.
4. [assigned 15 Jan] Suppose Alice has an input $x \in \{0, 2, 4, \ldots, 8\}$ and Bob has an input $y \in \{1, 3, 5, \ldots, 9\}$. Here is a protocol that computes the function $f(x, y) = \max\{x, y\}$:

- If Bob has input $y = 9$, he announces “yes” and both parties output 9 and halt. Otherwise he announces “no” and the protocol continues.
- If Alice has input $x = 8$, she announces “yes” and both parties output 8 and halt. Otherwise, she announces “no” and the protocol continues.
- ...

The protocol continues until some party says “yes”, at which point the output is determined and the protocol is finished.

Show that this protocol is secure against semi-honest adversaries by describing appropriate simulators.

5. [assigned 15 Jan] Consider a variant of the above protocol, where $x, y \in \{0, \ldots, 9\}$. It still computes $f(x, y) = \max\{x, y\}$:

- If Bob has input $y = 9$, he announces “yes” and both parties output 9 and halt. Otherwise he announces “no” and the protocol continues.
- If Alice has input $x = 9$, she announces “yes” and both parties output 9 and halt. Otherwise, she announces “no” and the protocol continues.
- ... repeat for $y = 8$ then $x = 8$, and so on...

Show that this protocol is not secure against semi-honest adversaries.

6. [assigned 22 Jan] Consider the basic 2PC protocol from lecture for semi-honest adversaries, using garbled circuits:

Alice garbles the circuit and sends $(F, d)$ to Bob
Alice encodes her input $x_A$ to obtain $X_A$, and sends it to Bob for each of Bob’s input bits $(i)$:

- let $K_0$ and $K_1$ be the wire labels for false/true for this input wire
- the parties run a 1-out-of-2 OT protocol,
- with Alice sending $K_0, K_1$ and Bob giving choice bit $(x_B)_i$

let $X_B$ denote the collection of wire labels Bob learns from all OTs
Bob evaluates $F$ on garbled input $(X_A, X_B)$ to obtain garbled output $Y$

**Bob sends $Y$ to Alice** (fixed)
Both parties decode $Y$ (using $d$) to obtain $y$, then both output $y$
(a) Can you describe an attack by a *malicious* Alice in which she learns Bob’s input in its entirety, while Bob always outputs all zeroes? This protocol doesn’t specify what Bob does when the evaluation algorithm returns an error (since this is not a possibility with semi-honest behavior). So try to avoid an attack that causes such an error and relies on unspecified protocol behavior. Also, assume that the OT subprotocol is ideally secure (you can imagine that OT is carried out by a trusted device / magical entity).

*Note:* The attack I have in mind doesn’t work for all possible circuits \( f \). In particular, the only message from Bob in this protocol is \( y \). If \( y \) is shorter than Bob’s input, then there is no hope of Alice learning all of Bob’s input. Try to characterize exactly what kinds of \( f \) you can attack (in terms of properties of the circuit for \( f \)).

(b) Suppose that Alice & Bob want to do 2PC in which only Bob learns the output (for example, Alice has a restaurant database, Bob has a location, and Bob wants to learn a list of nearby restaurants. Alice should only learn that a query happened). Argue that, if we modify the protocol so that Bob does not send \( y \) to Alice, then the resulting protocol is secure against *malicious* Bob.

*Note:* we haven’t discussed how to prove malicious security in great detail. Think about (1) what capabilities a malicious Bob has in this protocol, and (2) what capabilities a malicious Bob has in the ideal world. Argue that an appropriate #2 can achieve the same effect of any #1.

(c) Suppose that Alice & Bob should *both* learn the output. Describe how to modify the protocol to still be secure against a *malicious* Bob. Give some argument to justify your protocol.

*Hint:* [BHR12] define an *authenticity* security property of garbling schemes, which is relevant here. See a summary on the wiki.