Quiz 5 - COMS E6261: Advanced Cryptography

Question 1

Select all the worlds where the following is true: There is an efficient way to sample instances of a hard search problem such that those instances are guaranteed to have a solution (and solutions can be efficiently verified).

□ Algorithmica

□ Heuristica

☑ Pessiland

☑ Minicrypt

☑ Cryptomania

Explanation: First, the fact that we can efficiently sample such hard instances gives us a TFNP problem that is hard on average (under some samplable distribution). Hence, this cannot be true in Algorithmica or Heuristica. Next, note that we *can* sample such hard "promise-true" problems if we have TFNP hard-on-average problems (where every instance is guaranteed to have a solution) and also if we have OWF (where we can sample from the set of outputs $\{f(x)\}_{x\in\{0,1\}^n}$ and a solution would be a preimage). In Pessiland, there are TFNP hard-on-average problems, as we saw last class. In Minicrypt and Cryptomania there are OWF.

Question 2

Suppose that there's an efficient way to sample instances of a hard search problem such that those instances are guaranteed to have a solution, *along with solutions to those instances.* This is equivalent to the existence of OWF.

● True

○ False

Explanation: Given a OWF, it is easy to sample hard instances of a search problem along with solutions to those instances: just sample a uniformly random x and output $(f(x), x)$. The other way is also true: suppose $D(r)$ takes random string r and outputs an instance and solution pair (z, w) of the hard search problem. Then the function D' mapping $r \mapsto z$ must be one way, since otherwise, the solver could invert D' on an input z to obtain r, and then feed r to D to obtain w.

Note that this argument directly works if the hardness of the search problem means that no adversary succeeds except with negligible probability (and this was our intention). If we use a weaker hardness-on-average definition, say no adversary succeeds with more than some constant probability, the statement is still true: D' is a weak one-way function, and can be boosted to a standard (strong) OWF by repetition. In particular, $D''(r_1, \ldots, r_n) = (z_1, \ldots, z_n)$ is a OWF.

Question 3

Select all the worlds where the following is true: There is an efficient way to sample instances of a hard search problem such that those instances are guaranteed to have a solution, but there is no efficient algorithm that *also* generates the solutions along with the samples.

□ Algorithmica

□ Heuristica

☑ Pessiland

□ Minicrypt

□ Cryptomania

Explanation: From Q1, the first part of the premise is only true in Pessiland, Minicrypt and Crpytomania. But the fact that there is no algorithm that samples hard promise-true problems *and* solutions means that OWF do not exist, by Q2.

Question 4

Suppose OWF exist, and let f be a OWF. Consider the following three-round protocol.

Round 1: Attacker sends an n-bit string y

Round 2: Challenger sends a uniformly random n-bit string r

Round 3: Attacker sends a string x

Challenger accepts iff the x sent by attacker satisfies $f(x) = y \oplus r$.

Question 4.1

For every OWF f, the above is a valid three-round public-coin puzzle with perfect completeness.

○ True

● False

Explanation: The only property that is guaranteed to hold is public verifiability (and the protocol is also public coin). Completeness does not necessarily hold, as if the OWF is not onto, whenever $y \oplus r$ is not in the range of the function, there is no solution x to send in round 3. In fact, it's possible that there's only negligible probability for any attacker, even computationally unbounded, to convince the challenger to accept. Eg, consider $f(x) = 0^k || f'(x)$ for some OWF $f' : \{0, 1, \}^k \to \{0, 1\}^k$. The probability that a random string has a preimage under this f is negligible. Soundness does not necessarily hold either. It's possible that for some OWF f it's easy to efficiently find inverses of a random string, as we saw in the first quiz, question 3.

Question 4.2

Now suppose that in the above protocol, f is a one-way permutation. For every OWP f, the above is a valid three-round public-coin puzzle with perfect completeness.

● True

○ False

Explanation: Since f is a permutation, every string is guaranteed to have a preimage. Therefore, an unbounded attacker can always find a satisfactory x.

Question 4.3

We still assume that f is a one-way permutation, but change the above protocol into a two round protocol, as follows.

Round 1: Challenger sends a uniformly random n-bit string r

Round 2: Attacker sends two n-bit strings y, x

Challenger accepts iff the x sent by attacker satisfies $f(x) = y \oplus r$.

The resulting protocol is a valid two-round public-coin puzzle.

○ True

◉ False

Explanation: The point of this problem is to understand why the first attempt in Yizhi and Jiaqian's slide 14 doesn't work. If the attacker gets to choose the string y *after* seeing the randomness send by the Challenger, they could always choose a string y so that $y \oplus r$ is any target that the Attacker wants. For instance, the Attacker can compute $f(0^n)$ and choose y so that $y \oplus r = f(0^n)$, and then by sending $x = 0^n$ the Attacker can always solve the puzzle. In other words, this puzzle has no soundness.

Question 5

If OWFs exist, there exists a 2-round puzzle with perfect completeness.

◉ True

○ False

Explanation:

- Round 1: Challenger chooses uniformly random n-bit string x, and sends $y = f(x)$
- Round 2: Attacker sends a string x'
- Challenger accepts iff $f(x') = f(x)$

Assuming f is a OWF, this is a puzzle satisfying perfect completeness, as any message sent by C has an inverse, based on how the protocol for C was defined. Note that this puzzle is *not* public-coin (if C would just sent its random coins, soundness would not hold).

Question 6

If there exists a 2-round puzzle with perfect completeness, then there exists a hard-on-average problem in TFNP.

○ True

◉ False

Explanation: Only true if the puzzle is *public*-coin; otherwise, with the above problem, you would have that OWF implies TFNP is hard-on-average.