Quiz 6 - COMS E6261: Advanced Cryptography

Question 1

(Throughout the quiz, we use "black-box" to mean "fully black-box" – the only thing we've mentioned in class; there are other weaker notions of black-box reductions in the literature, and even in the paper that was presented, but we did not address them).

Recall that in a black-box construction of a primitive Q from primitive P, the "proof of security" of primitive Q must look like this: a description of a randomized PPT algorithm that breaks the security of primitive P, which interacts only with an *oracle* $\mathcal{A}_Q$ for breaking Q, and an oracle from the primitive P. Importantly, the queries submitted to $\mathcal{A}_Q$ can themselves contain queries to the P oracle built in to them; that is, the queries submitted to $\mathcal{A}_Q$ can contain "oracle gates" that query the P oracle (review the lecture for a more detailed exposition).

We call a black-box reduction *oblivious* if, in the addition to being black-box it makes all its queries to $\mathcal{A}_Q$, the Q-solver, before making *any* queries to the P oracle (the primitive it is trying to break). Oblivious is a good term for this because, intuitively, the P-solver learns nothing about the specific P instance it is trying to solve before querying the $\mathcal{A}_Q$, the solver of the new primitive.

The paper [FGHMY] we saw last lecture, showed that there is no black-box construction of a hard-on-average TFNP problem from OWF, where the reduction is oblivious and calls $\mathcal{A}_Q$ one time.

For the following constructions of one primitive from another, choose whether the construction is black-box and/or oblivious.

Question 1.1

The construction of OWF from CRHF.

Reminder of the security proof: the CRHF-solver works as follows. Query the CRHF on a random input x, get y. Submit y to $\mathcal{A}_{\text{OWF}}$, get back a preimage x'. With non-negligible probability, $x' \neq x$ and (x, x') is a solution.

- ○ Not black-box
- ◉ Black-box but not oblivious
- ○ Black-box and oblivious

Explanation: This was mentioned in class. The reduction queries the CRHF first, before calling the adversary (as it needs to know what y to submit to the adversary), so it is not oblivious.

Question 1.2

The construction of hard PPP instances from OWP.

Reminder of the security proof: the OWP-solver works as follows. Given y to invert, create the circuit C_y which on input x runs the OWP on x (i.e. using an oracle gate) and outputs the same, unless the output was the 0 string, in which case the circuit outputs y. Submit C_y to Adv_{PPP}, and get a solution (x, x') where $x \neq x'$ that both map to y and one of which is the OWP preimage of y.

- ○ Not black-box
- ○ Black-box but not oblivious
- ◉ Black-box and oblivious

Explanation: This construction is oblivious: given y, you can create the circuit C_y with oracle gates to the OWP. You do not need to query the OWP first in order to prepare the input C_y.

Question 1.3

The construction of hard PWPP instances from CRHF.

Reminder of the security proof: the CRHF solver works as follows. Pass the CRHF to the PWPP solver (i.e., give $\mathcal{A}_{\text{PWPP}}$ a circuit composed of one oracle gate for the CRHF); the solution returned by $\mathcal{A}_{\text{PWPP}}$ is a collision of the CRHF.

- ○ Not black-box
- ○ Black-box but not oblivious
- ◉ Black-box and oblivious

Explanation: Again, this construction is oblivious.

Question 1.4

The construction of PRG from PRF.

Since we did not cover this in class, here's a recap of the construction (for specific parameters, just for ease of exposition) and the security proof.

Construction: Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be a length preserving PRF (so for each $k \in \{0,1\}^n$ we have $F_k : \{0,1\}^n \to \{0,1\}^n$). Define G to be the following length-doubling function: For $x \in \{0,1\}^n$, define $G(x) = F_x(0^n)||F_x(1^n)$. Then G is a PRG.

Security proof: If D_G is a ppt adversary breaking G as a PRG, define the ppt D_F (which has access to an oracle function and needs to distinguish whether it is F_k for a random k vs a truly random function) as follows. $D_F(1^n)$: Query the function at the point 0^n (call its answer y), and at the point 1^n (call its answer z). Run $D_G(y||z)$ and output the same.

○ Not black-box

◉ Black-box but not oblivious

○ Black-box and oblivious

Explanation: The adversary D_F in this proof needs to call the function oracle in order to prepare the input that it calls the adversary D_G on. As an informal aside, this reduction seems "a little more oblivious" than the construction of OWF from CRHF we saw above – can you see why?

Note that all the constructions above are black-box – the construction uses the primitive in a black-box way, and the adversary in the proof of security uses the given adversary in a black-box way – in both cases, only the input-output behavior is invoked, we do not care about any other properties of the function implemented by the oracle (e.g, what the implementation looks like, efficiency, etc). We will see in upcoming classes other reductions that are not black-box and need to use the actual implementation/circuit.

Question 2

Question 2.1

You have an idea – you think you can prove that if OWF exist, there exists a worst-case hard problem in TFNP: no PPT algorithm solves this problem on all inputs. Your reduction is oblivious black-box and calls the TFNP-solver oracle once.

○ We have seen in class that this is already known – nothing new.

○ We have seen in class that this is not possible – you must be wrong.

◉ None of the above. Time to work out the details and start getting excited – could be interesting!

Explanation: We showed it's not possible to construct an average-case hard problem in TFNP with such a reduction.

Question 2.2

You have an idea – you think you can prove that if OWF exist, and moreover if for every input length n the OWF is a bijection on $\{0,1\}^n$, then there exists a average-case hard problem in TFNP.

- ◉ We have seen in class that this is already known – nothing new.
- ○ We have seen in class that this is not possible – you must be wrong.
- ○ None of the above. Time to work out the details and start getting excited – could be interesting!

Explanation: This is a OWP – we saw in class it implies average case hardness of PIGEON (in PPP – a subset of TFNP)