

Assignment 6

November 7, 2024

Instructor: Chethan Kamath

Exercise 1 (One-way function (OWF) and $\mathbf{NP} \cap \mathbf{coNP}$). Recall the separation of OWF and one-way permutation from Lecture 21. Adapt this argument to show that OWF does not imply hardness of $\mathbf{NP} \cap \mathbf{coNP}$. (*Hint*: Consider a language $\mathcal{L} \in \mathbf{NP} \cap \mathbf{coNP}$. Let $\mathcal{R}$ and $\bar{\mathcal{R}}$ denote the $\mathbf{NP}$ relations corresponding to $\mathcal{L}$ and $\bar{\mathcal{L}}$, respectively. Observe that for every x there exists either $w : (x, w) \in \mathcal{R}$ or $\bar{w} : (x, \bar{w}) \in \bar{\mathcal{R}}$. Use this fact to learn a fresh query about x 's "witness" w or "non-witness" $\bar{w}$.)

Exercise 2 (How to use virtual black-box obfuscator (VBBO)). Recall the definition of VBBO from Lecture 22 (Definition 1). Given VBBO, construct:

1. one-way function (*Hint*: try obfuscating a point function)
2. trapdoor permutation
3. non-interactive commitment
4. fully-homomorphic encryption (FHE) from any SKE

If you need more properties of VBBO (e.g., auxiliary-input VBBO), first formally define them.

Exercise 3 (Obfuscation of obfuscation). Suppose Obf is an indistinguishability obfuscator (IO). Consider Obf' defined as $\text{Obf}'(\mathbf{P}; r_1 \| r_2) := \text{Obf}(\text{Obf}(\mathbf{P}; r_1); r_2)$. Is Obf' also an IO? If you believe it is, provide a formal proof; otherwise, come up with a counterexample. What happens when Obf is a VBB obfuscator?

Exercise 4 (IO is "best-possible obfuscation"). Show that if it is possible to VBB obfuscate a program $\mathbf{P}$, then obfuscating $\mathbf{P}$ with an IO results in VBB obfuscating $\mathbf{P}$. (*Hint*: prove and use the observation that IO of VBBO is still VBBO.)

Exercise 5 (Constructing puncturable pseudo-random function (PPRF)). Recall our informal definition of PPRF from Lecture 23 (Definition 2).

1. Formalise this definition. (*Hint*: define pseudorandomness at punctured point for fixed $x^* \in \{0, 1\}^n$)
2. Tweak the tree-based construction of PRF from Lecture 5 to construction a puncturable PRF. (*Hint*: the punctured key depends on the path from x^* to the root in the tree.)