CS789: Introduction to Probabilistic Proof Systems Spring 2025

Assignment 2

February 14, 2025

Instructor: Chethan Kamath

Assignment Policy:

1. The deadline for submitting solutions is 06/March, midnight. (But it is advisable to attempt them before the mid-sem exam, since the questions there may be based on the problems here.)
2. Please use LaTeX to typeset up your solutions.
3. You are free to collaborate with others to solve the problems. But in the end you must *write up* the solutions on your own. Please list the persons you collaborated with on each problem.

In Lecture 07, we encountered *pair-wise independent* hash functions and saw a construction based on matrices. In the following problem, we will see an alternative construction based on polynomials over finite fields.

Problem 1 ((3+3=6 points))**.** Recall that a family of functions $\mathcal{H} := \{h_K\}_{K\in\mathcal{K}}$, where $h_K : \mathbb{F}_p \to \mathbb{F}_p$ for a prime p and $\mathcal{K}$ is some key-space, is said to be pair-wise independent if for every *distinct* $x_1, x_2 \in \mathbb{F}_p$ and any $y_1, y_2 \in \mathbb{F}_p$,

$$\Pr_{K\leftarrow\mathcal{K}}[h_K(x_1) = y_1, h_K(x_2) = y_2] = 1/p^2.$$

1. Show that $\mathcal{H} := \{h_{a_1,a_0}\}_{a_1,a_0\in\mathbb{F}_p}$ where $h_{a_1,a_0}(x) := a_1x + a_0$ (over $\mathbb{F}_p$) constitutes a family of pair-wise independent hash functions.

Next, let's extend the above definition: a family $\mathcal{H} := \{h_K\}_{K\in\mathcal{K}}$ as above is said to be *k-wise independent* if for every *pair-wise distinct* $x_1, \ldots, x_k \in \mathbb{F}_p$ and any $y_1, \ldots, y_k \in \mathbb{F}_p$,

$$\Pr_{K\leftarrow\mathcal{K}}[h_K(x_1) = y_1, \ldots, h_K(x_k) = y_k] = 1/p^k.$$

2. Show that $\mathcal{H} := \{h_{a_k,\ldots,a_1}\}_{a_k,\ldots,a_1\in\mathbb{F}_p}$ where $h_{a_k,\ldots,a_1} := \sum_{i=1}^{k} a_i x^{i-1}$ (over $\mathbb{F}_p$) constitutes a family of k-wise independent hash functions.

Notice the similarity between the above construction and Reed-Solomon encoding.

In Lecture 07 we also saw the *set lower-bound protocol*, which allows a prover to convince a verifier that a certifiable set $\mathcal{S}$ is of size at least 2^k. Recall that $\mathcal{S}$ is certifiable if membership in $\mathcal{S}$ can be checked efficiently (either through an oracle or a witness). In the problem below, we will solve the converse task.

Problem 2 ((4+2=6 points)).

1. Using pairwise-independent hash function, design a *set upper-bound protocol*. That is, given an instance $(\mathcal{S}, k)$, where $\mathcal{S}$ is a certifiable set and $k \in \mathbb{N}$, your protocol needs to satisfy the following properties:

   (a) Completeness: if $|\mathcal{S}| \leq 2^k$ then the verifier accepts with overwhelming probability (say 1).

   (b) Soundness: if $|\mathcal{S}| > 2^{k+1}$ then the verifier rejects with noticeable probability (say 1/2).

   (Hint: You may assume it is possible to efficiently sample randomly from $\mathcal{S}$.)

2. Does your protocol satisfy some form of ZK? If not, can you tweak it to get ZK? Describe your simulator and argue why it works.

Set upper bound protocol (in conjunction with set lower bound protocol) was used in [For87] to show that **NP** is unlikely to be contained in **PZK** (i.e., SAT, e.g., is unlikely to have perfect zero-knowledge proof (ZKP)).

Recall that a language $\mathcal{L}$ is in the class **BPP** (bounded-error probabilistic polynomial-time) if there exists a probabilistic polynomial-time decider D such that:

- $\forall x \in \mathcal{L}$: $\Pr[\mathsf{D}(x) = 1] \geq 2/3$
- $\forall x \notin \mathcal{L}$: $\Pr[\mathsf{D}(x) = 1] \leq 1/3$,

where the probabilities are over random coins of D. Note that **BPP** has a trivial (perfect) ZKP: the prover sends nothing and the verifier simply decides the membership of an instance x in $\mathcal{L}$ *on her own*. In the following problem, we will show that for some restricted cases, ZKP can only hold trivially as above.

Problem 3 (3+2=5 points).

1. If a language $\mathcal{L}$ has a non-interactive (NI) *perfect* ZKP (i.e., the prover sends a single message a_1 to the verifier) then $\mathcal{L} \in$ **BPP**. Show that the same holds when relaxing to *statistical* ZKP. (Hint: Analyse what happens when your run the verifier on the simulated transcript.)

2. If a language $\mathcal{L}$ has a ZKP with *deterministic* verifier then $\mathcal{L} \in$ **BPP**. (Hint: Reduce to the above case.)

In Lecture 08, we saw a *honest-verifier* zero-knowledge protocol (HV-ZKP) $\Pi_{\text{GNI}} = (\mathsf{P}, \mathsf{V})$ for graph non-isomorphism (GNI). In the following problem, we will figure out how a *malicious verifier* (MV) can *break* zero-knowledge of Π_{GNI} and then try to fix the protocol.

Problem 4 ((2+2+2+2=8 points)).

1. Describe a MV strategy V^* that gains non-trivial knowledge when interacting with the honest prover P. Point out exactly what the non-trivial knowledge is.

2. Next, design an MV-ZKP $\tilde{\Pi}_{\text{GNI}} = (\tilde{\mathsf{P}}, \tilde{\mathsf{V}})$ for GNI. Formally describe your interactive protocol. (Hint: you will need to use the ZKP for *graph isomorphism* somehow.)

3. First show that $\tilde{\Pi}_{\text{GNI}}$ is HV-ZKP: you only need to argue on a high level why it is complete and sound, but ZK needs to be argued formally (in particular, describe the simulator formally).

4. Then show that $\tilde{\Pi}_{\text{GNI}}$ is MV-ZKP. Describe your simulator formally, and then argue why it works.

In Lecture 12 we saw an explicit error-correcting code (ECC) in the form of Walsh-Hadamard code. In the problem below, we will learn two more explicit ECCs: Reed-Solomon code and Reed-Muller code. In the following, $\mathbb{F}_p$ is a finite field of prime order p.

Problem 5 ((3+2=5 points))**.**

1. For parameters $n < m < p \in \mathbb{N}$ and parsing $\bar{a}$ as $(a_1, \ldots, a_n)$, the Reed-Solomon code $E_{RS} : \mathbb{F}_p^n \to \mathbb{F}_p^m$ is defined as

$$E_{RS}(\bar{a}) := f_{\bar{a}}(0), \ldots, f_{\bar{a}}(m-1),$$

   where $f_{\bar{a}}(x) := \sum_{i=1}^{n} a_i x^{i-1}$ is the $(n-1)$-degree univariate polynomial that encodes $\bar{a}$. Thus it is a *restricted* Reed-Solomon encoding, which we saw in Lecture 02. Show that E_{RS} is an ECC with distance $1 - n/m$. (Hint: Exploit linearity again.)

2. For parameters $d < p \in \mathbb{N}$ and $v \in \mathbb{N}$, and parsing $\bar{a}$ as $(a_{i_1,\ldots,i_v})_{i_1+\ldots+i_v \le d}$, the Reed-Muller code

$$E_{RM}(\bar{a}) : \mathbb{F}_p^{\binom{v+d}{d}} \to \mathbb{F}_p^{p^v}$$

   is defined as

$$E_{RM}(\bar{a}) := (f_{\bar{a}}(i_1, \ldots, i_v))_{i_1+\ldots+i_v \le d}$$

   where

$$f_{\bar{a}}(x_1, \ldots, x_v) := \sum_{i_1+\ldots+i_v \le d} a_{i_1,\ldots,i_v} x_1^{i_1} \cdots x_v^{i_v}$$

   is the d-degree v-variate polynomial that encodes $\bar{a}$. *Show* that E_{RS} is an ECC with distance $1 - d/p$. (Hint: You need to invoke Schwartz-Zippel Lemma.)

Finally, a *bonus* problem about ZKP for you to ponder on.

Problem 6 (3 points)**.** In the puzzle *Where's Wally*, you are given a large (physical) poster with lots going on in which *Wally* (highlighted in picture below) is craftily hidden. Think of how you (the prover) can convince your friend (the verifier) that you have found out where Wally is *without revealing* the location. You are free to use whatever real-world tool you wish to. Describe the simulator for your physical protocol.

Image credit: `http://waldo.wikia.com`

References

[For87] Lance Fortnow. The complexity of perfect zero-knowledge (extended abstract). In Alfred Aho, editor, *19th ACM STOC*, pages 204–209. ACM Press, May 1987.