Proofs, Arguments, and Zero-Knowledge
Solutions by L. Russo

Preface

This document contains some of my solutions to the excellent manuscript *Proofs, Arguments, and Zero-Knowledge* by Justin Thaler. That book is regularly updated (even on a daily basis) and it is not, as far as I know, versioned: so it becomes a bit hard to track whether or not some exercises are added, modified or removed. For this reason, together with the solution, I have decided to attach the full text of the exercise as it is at the time of my reading.

I am always happy to receive feedback. Also, no document is error-free: if you do find any, I would greatly appreciate it if you let me know; you can open an issue on GitHub for this.

Solutions

Exercise 3.3

Let $p = 11$. Consider the function $f : \{0,1\}^2 \to \mathbb{F}_p$ given by $f(0,0) = 3$, $f(0,1) = 4$, $f(1,0) = 1$ and $f(1,1) = 2$. Write out an explicit expression for the multilinear extension $\tilde{f}$ of f.

We first write the Lagrange interpolation for f as $\tilde{f}(x_1, x_2) = \sum_{w \in \{0,1\}^2} f(w)\mathrm{X}_w(x_1, x_2)$, where $\mathrm{X}_w(x_1, x_2) = \prod_{i=1}^{2}(x_i w_i + (1 - x_i)(1 - w_i))$. We easily determine:

- $\mathrm{X}_{00}(x_1, x_2) = (1 - x_1)(1 - x_2)$
- $\mathrm{X}_{01}(x_1, x_2) = (1 - x_1)x_2$
- $\mathrm{X}_{10}(x_1, x_2) = x_1(1 - x_2)$
- $\mathrm{X}_{11}(x_1, x_2) = x_1 x_2$

What is $\tilde{f}(2, 4)$?

We only need to compute $\tilde{f}(x_1, x_2) = 3(1-x_1)(1-x_2)+4(1-x_1)x_2+x_1(1-x_2)+2x_1x_2$ in $(x_1, x_2) = (2, 4)$. $\tilde{f}(2,4) = 3(1-2)(1-4)+4(1-2)4+2\cdot 4+2\cdot 2\cdot 4 = 3$

Now consider the function $f : \{0,1\}^3 \to \mathbb{F}_p$ given by $f(0,0,0) = 1$, $f(0,1,0) = 2$, $f(1,0,0) = 3$, $f(1,1,0) = 4$, $f(0,0,1) = 5$, $f(0,1,1) = 6$, $f(1,0,1) = 7$, $f(1,1,1) = 8$. What is $\tilde{f}(2,4,6)$?

$\tilde{f}(2,4,6) = 0$[1]

Exercise 3.4

Fix some prime p of your choosing. Write a Python program that takes as input an array of length 2^l specifying all evaluations of a function $f : \{0,1\}^l \to \mathbb{F}_p$ and a vector $r \in \mathbb{F}_p$, and outputs $\tilde{f}(r)$.

Check **multilinear_extension.py** on my GitHub repository.

[1] Simply run multilinear_extension.py with the input values of the exercises. See also the next exercise.