PicoCTF Cryptography Challenge Solutions

Stephen Kelman

9 September 2024

flag_printer

I made a program to solve a problem, but it seems too slow :(
Download the program here.
Download the message here.

Solution: The code creates the following matrix:

$$\begin{pmatrix} 0^0 \pmod m & 0^1 \pmod m & \cdots & 0^n \pmod m \\ 1^0 \pmod m & 1^1 \pmod m & \cdots & 1^n \pmod m \\ \vdots & \vdots & \ddots & \vdots \\ n^0 \pmod m & n^1 \pmod m & \cdots & n^n \pmod m \end{pmatrix}$$

where m is the prime number 7514777789, and n is 1769610. We're also given a vector of $n+1$ y-values which are meant to correspond to each row of this matrix. The flag, encoded into bytes, is the solution to the linear system of equations given by the above matrix and the y-values, taken mod m.

Here's the thing: Evaluating this as-is, with just row-reduction or something similar, is $O(n^3)$, so it makes sense to instead use special properties of this specific matrix to help us find the solution. Of course, since the rows in the matrix are $(x^0, x^1, x^2, \ldots, x^n)$ over the ring $\mathbb{Z}/m\mathbb{Z}$ (field, in fact, since m is prime), this is screaming, "POLYNOMIAL!!!" Unfortunately, I don't really know anything yet about figuring out polynomials over finite fields based on a given list of points (or even how to do this over $\mathbb{R}$ in general), so I'll come back and finish this solution once I do.

Credits to Peter-Simon Dieterich for posting a solution to give me a nudge in the right direction. From his solution, I'll be going through the following before I come up with my own solution and return to finish this:

1. Lagrange Interpolation
2. Newton Interpolation
   (a) As mentioned by Dieterich's solution, neither of these are sufficiently fast to give the solution in a reasonable amount of time. However, I feel like it would be unreasonable to start learning about polynomial interpolation just through highly optimized computer algebra techniques, as opposed to the actual mathematical fundamentals

3. This MathOverflow thread about optimizing polynomial interpolation
4. Chapter 10: Fast polynomial evaluation and interpolation, of Modern Computer Algebra