Homework 6 (v2)

Problem 1

Create a graph with 3 nodes and 3 edges and write constraints for a 3-coloring. Conver the 3-coloring to a rank 1 constraint system.

Problem 2

Write python code that takes an R1CS matrix A, B, and C and a witness vector w and verifies.

$Aw \odot Bw - Cw = 0$

Where ⊙ is the hadamard (element-wise) product.

Use this to code to check your answer above is correct.

Problem 3

Given an R1CS of the form

$$L[\vec{\mathbf{s}}]_{\mathbf{1}} \odot R[\vec{\mathbf{s}}]_{\mathbf{2}} = O[\vec{\mathbf{s}}]_{\mathbf{1}} \odot [\vec{G_2}]_2$$

Where L, R, and O are n x m matrices of field elements and **s** is a vector of G1, G2, or G1 points

Write python code that verifies the formula.

You can check the equality of G12 points in Python this way:

```
a = pairing(multiply(G2, 5), multiply(G1, 8)) b = pairing(multiply(G2,
10), multiply(G1, 4)) eq(a, b)
```

Hint: Each row of the matrices is a separate pairing.

Hint: When you get **s** encrypted with both G1 and G2 generators, you don’t know whether or not they have the same discrete logarithm. However, it is straightforward to check using another equation. Figure out how to discover if sG1 == sG2 if you are given the elliptic curve points but not s.

Solidity cannot multiply G2 points, do this assignment in Python.

Problem 4

Why does an R1CS require exactly one multiplication per row?

How does this relate to bilinear pairings?