I could not get part 5 to work. I'm not sure why h and h' do not match because the public and private keys were working fine for part 4 and 5. The values between h and h1 seem very close in decimal value and seem very similar, but do not match identically. I followed the pseudo-code but I think the issue lies around the area of U ← z*G + h*V. I feel like the calculation around that area is getting is messed up by something. Another possible position of error is the z ← (k-hs) mod r.

I also mod the h ← KMACXOF256(Ux, m, 512, "T"); because it returns a negative BigInteger value and I feel that it messed up the equation even more, but it could be the reason to my failure. Again, I used the same passphrase "test" because that is the passphrase I have created the public key output file for in part 3.

File.txt is the message file that is un-altered.

Pk.txt is the public key output text file generated from part 3.

Signed.txt is the text file that is signed using the sign feature.

I am unsure if the sign function is the issue or the verify function is the error. I have no vectors to test them against and have run out of time to make any more attempts in trying to solve the issue. The issue seems very simple viewing from how simple the pseudo-code is but I am unable to see it.

The Z cryptogram is displayed in the console and not written to any file.