Homework 3

Use this resource as a reference: https://cryptobook.nakov.com/digital-signatures/ecdsa-sign-verify-messages

The following may also be helpful:

https://www.rareskills.io/post/finite-fields

https://www.rareskills.io/post/elliptic-curve-addition

https://www.rareskills.io/post/elliptic-curves-finite-fields

Implement ECDSA from scratch. You want to use the secp256k1 curve. When starting off, use the Elliptic curve multiplication library used in the blog post linked here: https://www.rareskills.io/post/generate-ethereum-address-from-private-key-python

1) pick a private key

2) generate the public key using that private key (not the eth address, the public key)

3) pick message m and hash it to produce h (h can be though of as a 256 bit number)

4) sign m using your private key and a randomly chosen nonce k. produce (r, s, h, PubKey)

5) verify (r, s, h, PubKey) is valid

You may use a library for point multiplication, but everything else you must do from scratch.

Pay close attention to the distinction between the curve order and the prime number p we compute the modulus of $y^2 = x^3 + b \pmod{p}$.