Homework #4

Q 1-) (30pnts) Let the parameters for a Paillier PKC are as follows:
p = 293, q = 433 , g= 6497955158, μ= inverse of L($g^{\lambda(n)}$ mod n^2) = 53022
Consider 5 random numbers chosen randomly uniformly from a PRN

r_1= 35145
r_2= 74384
r_3= 10966
r_4=17953
r_5= 7292

Write a program to build a Paillier crypto counter such that r_i is the random value generated at round (step/iteration) i. Set m=1 for the initial value.

1.1. [15 pnts] Show step by step how to increment the value of the counter at 1,..,4'th iterations.

1.2. [15 pnts] Show step by step how to decrement the final counter value from a. to obtain m=1

Q2-) (30pnts) Suppose p=23 is a large prime, α=5 is a primitive root, a=7 is the secret exponent such that $\beta \equiv \alpha^a (\text{mod } p)$. The numbers p, α, β are public. Peggy wants to prove Victor that it knows the discrete logarithm without revealing it. Show the steps of a zero-knowledge proof.

Q3-) (40pnts) In Shamir's secret sharing scheme, a secret is split among *n* members using apolynomial of degree *k*. A collusion of *k* members has *k* shares.

2.1) [10pnts] Describe under what condition the collusion can reveal the secret.

2.2) [30pnts] Consider Shamir *(t, w)-Threshold Scheme* in Z_P. That is, given *t* public *x*-coordinates x1, x2, x3, …., xt, and *t y*- coordinates y1, y2, y3, …., yt, the key is computed by using the *Lagrange Interpolation* formula. Write a program to

a-) [15pnts] Find the key for *p* = 31847, *t* = 5 and *w* = 10 with following shares.

x1 = 413, y1 = 25439
x2 = 432, y2 = 14847
x3 = 451, y3 = 24780
x4 = 470, y4 = 5910
x5 = 489, y5 = 12734
x6 = 508, y6 = 12492
x7 = 527, y7 = 12555
x8 = 546, y8 = 28578
x9 = 565, y9 = 20806
x10 = 584, y10 = 21462

b-) [15pnts] Compute the share that would be given to a participant with *x*-coordinate equal to 10000? Can this be done without computing the whole secret polynomial? How?

Bonus question: [30pnts] Encrypt & decrypt the message M = {NETSEC} with BG (Blum & Goldwasser) where p = 499, q = 547; and let random quadratic residue for encryption is x_0 = 159201. Show your work and include a readme file for your code.