**Q1 -** Let P be an EC point. What is the **minimum** number of EC operations necessary to compute [63]P? And more specifically which are these operations?

|  |
| --- |
|  |

**Q2 - Consider both commitments introduced in our classes (Feldman and Pedersen)**, and assume they “commit” a value x. Under which (eventually different) assumptions they can be considered secure?

Feldman Pedersen

⭘ ⭘ **a)** no specific assumptions

⭘ ⭘ **b)** must use a large prime p in the modular exponentiations

⭘ ⭘ **c)** require that the committed value x is drawn from a large space

⭘ ⭘ **d)** both large prime p and x drawn from large space

**Q3 -** A strong prime p is defined as:

⭘ **a)** a prime number p much larger than usual

⭘ **b)** a prime p such as 2p+1 = q is also prime

⭘ **c)** a prime p such as p = 2q+1 and q is also prime

⭘ **d)** a prime p such as the Euler (p) is also prime

**Q4 -** Describe the Boneh-Franklin Identity Based Encryption scheme, specifying in particular, i) how a message is encrypted, ii) how a message is decrypted, and iii) what is the private key used by the receiver.

|  |
| --- |
|  |

**Q5 -** Consider an RSA digital signature based on a (2,2) secret sharing, and assume all following operations are based on modulo n, with n being the RSA parameter. The tag H(m)d is reconstructed by:

⭘ **a)** Summing the tags constructed using the two shares

⭘ **b)** Multiplying the tags constructed using the two shares

⭘ **c)** Interpolating the tags constructed using the two shares using Lagrange coefficients

⭘ **d)** Using a special approach proposed by Shoup.

**Q6 - Assume arithmetic modulus 100.** A Linear secret sharing scheme involving 3 parties is described by the following access control matrix:

A: 1 1 0

B: 0 1 1

C: 0 0 -1

Assume that the following shares are revealed:

A 🡪 51

B 🡪 63

D 🡪 11

What is the secret?

**a)** 1 **b)** 3 **c)** 23 **d)** 25 e) 75 **f)** 77 **g)** 97 **h)** 99 **i)** another result = \_\_\_\_\_\_\_\_\_\_

**Q7 -** A same message M is RSA-encrypted using two different public keys e1 = 5 and e2 = 7, but same RSA modulus n=143. The two resulting ciphertexts are: c1=23 and c2=4. Decrypt the message applying the Common Modulus Attack (show the detailed computations required).

*Just in case you need to rapidly compute inverses modulus 143, here a few ones:*

*𝑥={4,5,7,17,20,23,29,92} 🡪 x-1 mod 143 ={36,86,41,101,93,56,74,14}*

|  |
| --- |
|  |

**Q8 -** A Shamir Secret Sharing scheme uses a non-prime modulus p=55 (if you need modular inverses see table on the right). Of the 5 participating parties P1,…,P5, with respective x coordinates xi = {1,2,3,4,5}, parties P1, P3 and P5 aim at reconstructing the secret.

a) compute the Lagrange Interpolation coefficients for parties 1,3,5;

b) Reconstruct the secret, assuming that the shares are:

P1 🡪 46

P3 🡪 51

P5 🡪 2

c) Prove that the system is NOT unconditionally secure, by showing that the knowledge of the two shares P3 and P5 leak information about the secret – specifically, after knowing shares P3 and P5 which would be the only possible remaining secret values?

**Q9 -** Prove that **any** linear secret sharing scheme is homomorphic with respect to the sum operation.

|  |
| --- |
|  |

**Q10 –** 1) Determine the access control matrix that implements the policy: , and then 2) turn it into a linear secret sharing scheme, by computing the shares to assigned to the 5 parties (use modulus 100, share secret S=10, inventiyour own random values if/when necessary)

|  |
| --- |
|  |