CS 380/525: Intro to Crypto
Fall 2022
Assignment 4, due 11/8, before class

Please show your work. For modular exponentiations, etc. use any or all of the tricks/shortcuts we covered.

1. (30 points) Consider $\mathbb{Z}_{17}^*$:

   (a) For every element, check if it is a generator of $\mathbb{Z}_{17}^*$? Does it generate a cyclic subgroup? If so, show the subgroup. What is the order of the subgroups – prime or composite?

   (b) Verify that the prime-order cyclic subgroups tally with the *residues modulo p* formula.

   (c) Verify that every element of a prime-order sub-group is a generator.

2. (20 points) Consider $\mathbb{Z}_n^*$ for $n = 15$. Using Fermat's little theorem, how many witnesses can you find? Which ones? Are there any strong liars? Which ones? (See class notes for a worked-out example, $\mathbb{Z}_9^*$).

3. (10 points) Let's say, instead of using a composite $N = pq$ in the RSA cryptosystem, we just use a prime modulus p. As in RSA, we will have an encryption exponent e, and the encryption of a message $m \mod p$ would be $m^e \mod p$. Is this modified RSA secure? Either argue why it is, or give a counter-example that breaks it (i.e., an adversary given only public parameters $p, e, C = m^e \mod p$, can easily decrypt C to get plaintext m).

4. (10 points) Consider an RSA system with the following parameters: $p = 17, q = 23, N = 391, e = 3$. Find d. Encrypt $M = 55$, show the resulting ciphertext C. Now, decrypt C (using the d you computed), and verify we get back M.

5. (30 points) Consider an encryption scheme defined thus:

   Definition 0.1 *Some encryption scheme*

(a) $(PK, SK) \leftarrow \mathsf{KeyGen}(1^\lambda, \mathbb{G})$: *This is a randomized algorithm that takes in a security parameter, λ, group $\mathbb{G}$, and outputs a public/secret keypair. It first picks random $a, b \leftarrow \mathbb{Z}_p$. Then it picks $g_1, g_2, g_3 \in \mathbb{G}$, such that $g_1^a = g_2^b = g_3$. Return $PK = (g_1, g_2, g_3)$ and $SK = (a, b)$.*

(b) $C \leftarrow Encrypt(PK, m)$: *This is a randomized algorithm that takes in a public key, a message $m \in \mathbb{G}$, and returns a ciphertext C. It picks $x, y \leftarrow \mathbb{Z}_p$, and computes $C = (g_1^x, g_2^y, m \cdot g_3^{x+y})$.*

(c) $m \leftarrow Decrypt(SK, C)$: *This is a deterministic algorithm that takes in a secret key and ciphertext C, and returns the message m.*

Fill in the rest...(you may assume decryption algorithm knows PK. It might help to use El Gamal encryption/decryption as a template/example).