Week 3 Lab: Break RSA

Background:

RSA is the most used Public Key Cryptography as it is relatively simple to understand and implement and, ignoring quantum supremacy, near impossible to break. As a review, RSA public and private keys are derived as:

Public key = (e, n) where $n = p * q$ where p and q are two large primes

Private key = d where the relationship between e and d is

$$ed \bmod \varphi(n) \equiv 1$$

and

$$\varphi(n) = (p-1)*(q-1)$$

If you can find p and q, it is very possible to find the private key d. So, we are going to simulate a situation where, given n and e, you find one of the primes in a given file, and decrypt the message given to you. In this case, p and q are big enough that it will be difficult to solve by hand, but small enough that it can be solved.

Keep in mind that to encrypt a message using the public key use:

$$m^e \equiv c \bmod(n)$$

And to decrypt:

$$c^d \equiv m \bmod(n)$$

Note: This lab is adapted from The Hong Kong Polytechnic University, used in course COMP 3334.

Files:

Under the Week 3 RSA Folder in Files, you each have a different folder with 2 files. One with your name, n, e, and the ciphertext and another with a list of numbers, which includes the prime you need to calculate n.

Lab Steps:

Step 1:

Write a script to find the prime values for your *n* from the given file with various numbers. When you find *p* and *q*, you should be able to find *φ(n)*.

Step 2:

Now that you have *φ(n)* and *e*, write a script that uses the Extended Euclidean Algorithm to find *d*. The pseudocode is below:

```
/*Pseudocode*/
Specification:
Input: public exponent (e), modulus(phi_n)
Output: modular multiplicative inverse of e
\BEGIN
1. (A1, A2, A3) = (1, 0, phi_n);
   (B1, B2, B3) = (0, 1, e);
2. if B3 = 0
     return A3 which is GCD(phi_n, e) and there is no inverse;
3. if B3 = 1
     return B3 which is GCD(phi_n, e) and B2 which is the
     inverse of e;
4. Q = floor(A3 div B3);
5. (T1, T2, T3) = (A1 - Q * B1, A2 - Q * B2, A3 - Q * B3);
6. (A1, A2, A3) = (B1, B2, B3);
7. (B1, B2, B3) = (T1, T2, T3);
8. goto 2;

\END
```

Step 4:

When you have *d*, decrypt the message. The message is a list of ascii characters encrypted by the public key *(n, e)*. The resulting flag will form words and be obviously correct.

To Turn In:

Your p, q, and d.

The flag you get from decrypting the message.

Due:

Mon: 10/12 at 8pm

Tues: 10/13 at 5pm