MA398 Test 2 part 2

For the second part of the test you will implement the Elgamal public-key cryptosystem using elliptic curves as discussed over the past few lectures.

Requirements

The functionality of your final product should be similar to that of the RSA code on the course web page. In particular:

- You should have a `generateKeys()` function for creating public and private keys. Running this function should produce two text files containing the keys.
- You should have an `encrypt()` function whose arguments are two text files, one containing a public key and one containing the plaintext. Similarly, you should have a `decrypt()` function whose arguments are two text files, one containing a private key and one containing the ciphertext.
- Your code should be compatible with Python 3.6.

Implementation

Your code should be an implementation of the method discussed in class. In particular:

- Use 2048-bit primes p that are congruent to 3 modulo 4.
- Use elliptic curves of the form $E : y^2 = x^3 + ax$.
- Use Euler's Criterion to decide whether $x \in \mathbb{F}_p^*$ is a square modulo p, and if it is, use $x^{(p+1)/4}$ as a square root of x.
- Divide the plaintext into blocks of 128 characters. Convert each block to an integer n by interpreting the ASCII encodings of the characters in the block as the base-256 digits of n. Then encode the block as a point on E with coordinates either (n, y) or $(p - n, y)$.

Rules

- You may not collaborate with anybody on any part of this test.
- As references you may use your notes from class, the HPS textbook, Sweigart's book, and the Python documentation at `https://docs.python.org/3/`. No other references are allowed.
- You may use any code you have written previously for this course, as well as any code from the course web page. Beyond that, all new code must be designed and implemented by you.

Instructions

The deadline for this part of the test is 11 PM on Monday, April 24. By that time you must email me your code as well as instructions on how to create keys, encrypt, and decrypt. In addition, send me a text file with your public key. I will then test your code by exchanging messages with you.