COMP 443 / 543 – Fall 2022 - Project #2
Due: 29.12.2022 11:59 PM

- *By submitting this assignment, you agree to fully comply with the course syllabus and the Koç University Student Code of Conduct.*
- *Projects that are submitted after the due time will NOT be graded.*
- *Check the provided helper.py file for the frequency analysis dictionary.*
- *You can use any programming language.*
- *You have to submit only one script, named as kusisID.zip (i.e. ptap22.zip)*
- *Please only submit one zip file*
- *There will be a grade deduction if you do not follow the submission criteria!*

In this project, you are expected to implement a simple encryption–decryption messaging system using ElGamal.

A brief reminder of ElGamal:

1. Bob **generates** public and private keys:
    a. Bob chooses a large number q and a cyclic group F_q
    b. Bob chooses a random generator g for F_q and an element a such that gcd(b,q) = 1
    c. Bob computes $h = g^b$
    d. Bob publishes F, h, q and g, retains b as private key
2. Alice **encrypts the** message using Bob's public key:
    a. Alice chooses k from cyclic group F such that gcd(a, q) = 1
    b. Alice computes $p = g^k$ and $s = h^k = g^{ab}$
    c. Alice encrypts M with s
    d. Alice publishes $(p, M \times s) = (g^k, M \times s)$
3. Bob **decrypts the** message:
    a. Bob calculates $s' = p^b = g^{ab}$
    b. Bob decrypts M x s with s' and obtains M.

Encryption – Decryption with ElGamal

You will have to implement one script that contains the necessary code for the ElGamal key generation, encryption, and decryption. It will work as the following:

1. You will manually create an empty server.txt file in your directory. For this project, we did not want you to deal with network connections, so the purpose of the server.txt is to mimic a real server. Everything you will write in server.txt should be in ASCII format.
2. When you first run your script, it should check the contents of the server.txt file, if it contains any public keys and encrypted messages, your code should first decrypt that message and output the plaintext.
3. Then, your code should prompt the user for input. You will take this input message, encrypt it and write the encrypted message to the server.txt together with your public key.
4. The communication continues as one message on each side basis, i.e., after sending a message the other side will send one. This way only after sending a message the program needs to screen the file for the other side's reply with 5 seconds intervals.

Important Remarks

- Encode and decode messages as byte64 or your homework will be graded as 0.

- Make the size of the cyclic group maximum 1024 bits.(i.e it will be a number between [2^{1023} , 2^{1024} -1])
- You can choose the generator of the cyclic group using primality methods you have seen in the class and find a code snippet online.
- While writing decrypted messages to the server.txt, output it in the ASCII format.
- Public key content and output format to the server.txt should be the following:

  ****

  F: ####

  H: ####

  Q: ####

  G:####

  ****

  (i.e, publish each component of the public key in a newline and show which component it is in the beginning of the line as F: H: Q: G:)

  Please follow this format.

- **Your code will be graded in the following manner:**
  - We will run your script in 2 different terminals. One will be the sender and the other one will be the receiver
  - Both parties will initially read the empty server.txt, and we will send a message from one of them. The sender should output its public key in the first line and the encrypted message in the second line.
  - The receiver should check the server.txt 5 seconds later, read the encrypted message, decrypt it, **PRINT** it, and delete it from server.txt. AND it should prompt the user for input a message to encrypt and send to the sender (other party).
  - During this time, the sender is continuously checking server.txt for another message in 5-second intervals. When the message is written to the server.txt, it will follow the same steps as above.
- **YOUR CODE MUST WORK AS DESCRIBED ABOVE!**
- **DO NOT SUBMIT CODE WITH ERRORS, CODE THAT DOES NOT COMPILE OR YOUR WHOLE WORKSPACE.**