**Lab 3 Frequency Analysis and Paillier Homomorphic Encryption**

**First Half: Frequency Analysis**

Go to: <https://www.101computing.net/frequency-analysis/>

Q1. [40 pts] Select one of the Cipher other than Cipher #1 we did in class and recover the plaintext using frequency analysis. Please attach the substitution mapping as well as the original plaintext. [Take a final screenshot of the results.]

Text

Description automatically generated

Bar chart

Description automatically generated with medium confidenceText

Description automatically generated

**Second Half Paillier Homomorphic Encryption:**

The second half of this Lab is to implement Paillier Homomorphic Encryption in Python. We will continue to use the Anaconda environment. First, let us create a new environment called paillier-test.

conda create –-name paillier-test

conda activate paillier-test

conda install -c conda-forge phe

conda install numpy

from phe import paillier

#generate public and private key pairs.

public\_key, private\_key = paillier.generate\_paillier\_keypair()

Q1[35 pts]. Now suppose we have two numbers *a = 1* and *b = 2*, perform the following operations:

1). Encrypt *a* and *b* using the public key, and perform their summation “*enc\_c = enc\_a + enc\_b”, where “enc\_x” is the mathematical representation for the ciphertext of a – see below for the syntax*. Use what we have discussed in class, run the following and screen shot the ciphertext using:

enc\_c.\_EncryptedNumber\_\_ciphertext

Text

Description automatically generated

Text

Description automatically generated

2). Then perform enc\_d = enc\_c \* 2. Screen shot the ciphertext using

Enc\_d.\_EncryptedNumber\_\_ciphertext

Text

Description automatically generated

3) Decrypt d with the private key and screenshot the results.

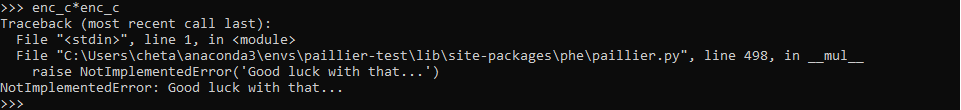


[Hints: Some useful commands:

public\_key.encrypt(a) 🡪 Encrypt the value of a.

private\_key.decrypt(enc\_d) 🡪 Decrypt the value of d.]

Q2[25 pts]. Now suppose we want to do enc\_c\*enc\_c (the product of two ciphertext), do we encounter an error? According to the lecture slides, please explain why we have this error. Please attach a screenshot of the error if there is one.



Yes. There is an error because paillier encryption only allows scalar multiplication. Cipher texts can only be multiplied by constants and no other cipher texts.

**[Bonus + 20 pts - Secured 2D Convolution]** Q3. Remember we had computed the 2D Convolution in the previous homework. Now assume Alice has a CPS device and she does not have the computational power to conduct 2D Conv. Thus, Alice uses Paillier Encryption to encrypt her Private Image P with the public key and sends it to Bob. Bob has the kernel K in the plaintext format. Bob performs secure computation Enc(R) = Enc(P)\*K, where \* denotes the convolution operation. The entire procedure is shown below:

Alice

Bob

Private Image P

Enc(P)

Decrypt R

Enc (R)

Enc(R) = Enc(P)\* K

Kernel K

**Write a new function that takes in the encrypted list of array elements and conducts *Secured 2D Convolution*. Then decrypt the results by Alice using the private key. The decrypted results should 100% match the one with Plaintext Results. You need to take a screenshot of the final decrypted results from Alice as well as take a screenshot of your python code implementation.**

[Hints: Note that with the Paillier Python package, all encrypted numbers are put into a Python List rather than Numpy array. That means we have to implement the convolution by hand in an element-wise manner – you can ignore the 180-degree flip of the image here. See the comments in the paillier-conv.py provided and the example in Lecture slides, p.45. If you do not know how to do it, first write a similar program using list rather than numpy array in plaintext and confirm its correctness first and then write the Paillier encrypted version.]

**[Bonus + 5 pts]** Q4. What is the execution time for the secure convolution? Please take a screenshot.

**[Bonus + 5 pts]** Q5. Now change the input image to 7x7 with any number you like, then take a screen shot of the execution time and compared with the image with 5x5. Make sure the sizes are correct.

**[Bonus + 10 pts]** Q6. Now change the input image to 9x9, 11x11. Record their execution time as well. Draw a figure in Excel to see the relations between execution time and homomorphic scalar multiplications: 1) The x axis of the first figure should be the total number of multiplications for 5x5, 7x7, 9x9, 11x11 inputs. The y axis of is the execution time you just recorded on your computer. There should be 4 points in the figure (e.g., as a scatter plot). The figure should have an increasing trend.