COL759
ASSIGNMENT 1

1. **encrypt.py**
   This is file containing the “encryption” function in which the arguments are the plain text and the key.
   All the characters of the english alphabet are stored in a map along with their corresponding numbers.
   Let the size of the key be “n”. If the number of characters in the plain text are not a multiple of n then we add extra ‘X’ to the plain text.
   Then the cipher text is calculated by multiplying the Key matrix with the matrix of chunks of size n of the Plain text : C = K x P.

2. **decrypt.py**
   This is file containing the “decryption” function in which the arguments are the cipher text and the key.
   All the characters of the english alphabet are stored in a map along with their corresponding numbers.
   Let the size of the key be “n”. Since we have added extra characters in the plain text before encrypting it, so we can have the assumption that the total number of characters in the cipher string which this decryption function is getting will be a multiple of n.
   Then the inverse of the key is calculated by multiplying its multiplicative inverse with its determinant and multiplicative inverse of determinant modulo 26.
   Then the plain text is calculated by multiplying the Key inverse matrix with the matrix chunks of size n of the cipher text [mod] 26 (which is the character size) : P = C x K_inverse.

3. **analysis.py**
   In this part the key is being calculated with the help of the index of coincidence as well as the known cipher text and its corresponding plain text.
   The loop of the key size will run from 1 to 10. Let the assumed size of the key in a particular iteration be i. So, now to calculate the key by the formula K = C x P_inverse, we need to make i x i matrix of cipher text and and i x i matrix of corresponding plain text. But we

do not know from where the grouping started in the actual encryption, so we will have a sliding window to select starting of the grouping from place 0 to i-1. And then we will first calculate the inverse of the plain matrix in the same way we calculated the inverse of the key. If it exists we will go to the next step and it it doesn't we will go to the next iteration of the loop. The next step will be to find the key by multiplying C and P_inverse, now we will decrypt the complete cipher text given using the key if its inverse exista and then check the Index of Coincidence of the plain text obtained after decrypting the complete cpher text. If IC >= 0.065, then this is the correct key and the program will return the key, else it will go to the next iteration. In this while we pass the complete cipher text to the decryption function, we skip the l%i characters of it (l is the length of the completed cipher text and i is the assumed key size), our assumption is that in a long text these characters don't effect IC significantly, so instead of adding i-l%i characters we remove l%i characters which also produces the same effect as adding extra characters.

Submitted By:-
Shruti Kumari : 2018CS50420
Nikita Bhamu : 2018CS50413