190031930_M.Varun

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SUBJECT CODE: 19CS3041S
CRYPT ANALYSIS AND CYBER DEFENSE WORKBOOK
3. Implementation of Hill Cipher Substitution technique

Date of the Session: **09/08/21** **Time of the Session: 9:00-10:40**

Learning Outcomes:

- To understand the concept multilettered encryption and decryption.
- To understand the applications of substitution techniques.

Pre-Lab Task:

1. Hill Cipher is a block cipher. Justify.

Hill ciphers (invented in 1929) are **a type of block cipher**: the ciphertext character that replaces a particular plaintext character in the encryption will depend on the neighboring plaintext characters. The encryption is accomplished using matrix arithmetic.

2. If A= $\begin{matrix} 4 & 5 \\ 2 & 7 \end{matrix}$, find |A|.

|A| = ad – bc

|A|=(7*4-2*5)

=28-10 =18

3. Write the mathematical formula for encryption and decryption in Hill Cipher.

E(K, P) = (K*P) mod 26 Where K is our key matrix P

is the plaintext in vector form.

D(P,K)=(C*K^-1) mod 26

1 3 1

3. If A = 3 2 5 , find A^{-1}.

Pre-Lab:-

3) If $A = \begin{bmatrix} 1 & 3 & 1 \\ 3 & 2 & 5 \\ 2 & 2 & 2 \end{bmatrix}$ find A^{-1}

we know that

$$A^{-1} = \frac{1}{|A|} adj(A)$$

Calculating $|A|$:-

$$|A| = \begin{bmatrix} 1 & 3 & 1 \\ 3 & 2 & 5 \\ 2 & 2 & 2 \end{bmatrix}$$

$$= 1(4-10) - 3(6-10) + 1(6-4)$$

$$= -6 - 3(-4) + 2 \Rightarrow -6 + 12 + 2$$

$$= -6 + 14 = 8$$

Since $|A| \neq 0$

finding $Adj(A)$:-

$$adj\ A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}' = \begin{bmatrix} A_{11} & A_{22} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 3 & 2 \\ 3 & 2 & 2 \\ 1 & 5 & 2 \end{bmatrix}$$

2 2 2

$$A = \begin{bmatrix} 1 & 3 & 2 \\ 3 & 2 & 2 \\ 1 & 5 & 2 \end{bmatrix}$$

$A_{11} = (4-10) = -6$

$A_{12} = -[6-2] = -4$

$A_{13} = (15-2) = 13$

$A_{21} = -[6-10) = 4$

$A_{22} = (2-2) = 0$

$A_{23} = -[5-3] = -2$

$A_{31} = [6-4] = 2$

$A_{32} = -[2-6] = 4$

$A_{33} = [2-9] = -7$

$$\text{adj } A = \begin{bmatrix} -6 & 4 & 2 \\ -4 & 0 & 4 \\ 13 & -2 & -7 \end{bmatrix}$$

Now,

$$A^{-1} = \frac{1}{|A|} \text{adj } A$$

$$A^{-1} = \frac{1}{8} \begin{bmatrix} -6 & 4 & 2 \\ -4 & 0 & 4 \\ 13 & -2 & -7 \end{bmatrix}$$

5. Can we consider the matrix $\begin{matrix} 6 & 6 \\ 6 & 6 \end{matrix}$ as a key matrix in Hill Cipher. Justify.

The most important item that must be discussed regarding the use of the Hill Cipher is that **not every possible matrix is a possible key matrix**. This is because, in order to decrypt, we need to have an inverse key matrix, and not every matrix is invertible.

In-Lab Task:

Q.1) Write a program to implement Hill Cipher Substitution technique for the following input.

Sample Input:

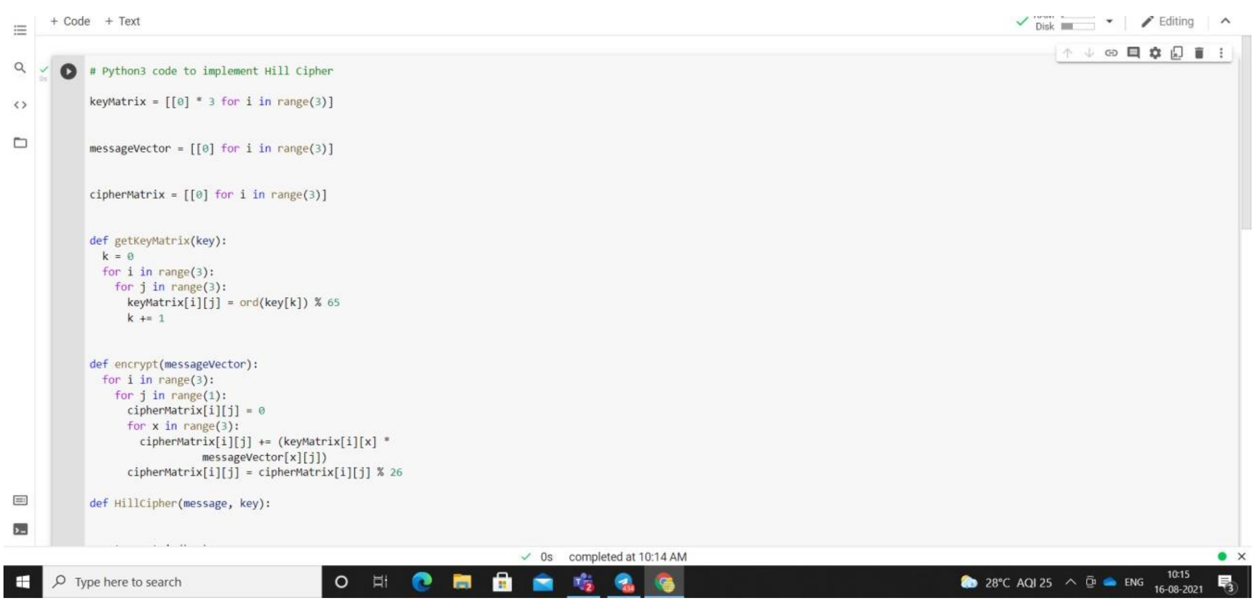
Plain Text: *Student to consider His/Her name*

Key: *ALBO*

Note: White space in the plaintext can be ignored and the key matrix must be 2×2 matrix.

Filer character to be taken as 'x

Sol)



```
# Python3 code to implement Hill Cipher

keyMatrix = [[0] * 3 for i in range(3)]


messageVector = [[0] for i in range(3)]


cipherMatrix = [[0] for i in range(3)]


def getKeyMatrix(key):
  k = 0
  for i in range(3):
    for j in range(3):
      keyMatrix[i][j] = ord(key[k]) % 65
      k += 1


def encrypt(messageVector):
  for i in range(3):
    for j in range(1):
      cipherMatrix[i][j] = 0
      for x in range(3):
        cipherMatrix[i][j] += (keyMatrix[i][x] *
                   messageVector[x][j])
      cipherMatrix[i][j] = cipherMatrix[i][j] % 26

def HillCipher(message, key):
```

```
    for i in range(3):
      messageVector[i][0] = ord(message[i]) % 65

    encrypt(messageVector)

    CipherText = []
    for i in range(3):
      CipherText.append(chr(cipherMatrix[i][0] + 65))

    print("Ciphertext: ", "".join(CipherText))

# Driver Code
def main():

    message = "LASYA"

    key = "ALBOALBOAL"

    HillCipher(message, key)

if __name__ == "__main__":
    main()
```

```
Ciphertext:  SOL
```

✓ 0s completed at 10:14 AM

Post-Lab Task:

1. Write a Pseudocode to find the inverse of a 3×3 matrix.

Importing NumPy Library

import numpy **as** np **import**

sys

Reading order of matrix n =

int(input('Enter order of matrix: '))

Making numpy array of n x 2n size and initializing

to zero for storing augmented matrix a =

np.zeros((n,2*n))

Reading matrix coefficients

print('Enter Matrix Coefficients:')

for i **in** range(n): **for** j **in**

range(n):

a[i][j] = float(input('a['+str(i)+']['+ str(j)+']='))

Augmenting Identity Matrix of Order n

for i **in** range(n): **for** j **in**

range(n): **if** i == j:

a[i][j+n] = 1

Applying Guass Jordan Elimination

for i **in** range(n): **if** a[i][i] == 0.0:

sys.exit('Divide by zero detected!')

for j **in** range(n):

if i != j:

```
        ratio = a[j][i]/a[i][i]

        for k in range(2*n):
            a[j][k] = a[j][k] - ratio * a[i][k]

# Row operation to make principal diagonal element to 1
for i in range(n):     divisor = a[i][i]     for j in
range(2*n):         a[i][j] = a[i][j]/divisor

# Displaying Inverse Matrix
print('\nINVERSE MATRIX
IS:') for i in range(n):     for j in
range(n, 2*n):         print(a[i][j],
end='\t')     print()
```

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation
	Marks Secured:_________out of __________ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation