Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Experiment 5-A

Date of Performance : 7 March 2023 **Date of Submission:** 7 March 2023

SAP Id: 60004200132 **Name :** Ayush Jain

Div: B **Batch :** B3

Aim of Experiment

Implement Hill Cipher. Create two functions Encrypt () and Decrypt(). Demonstrate these ciphers using Color Images/Gray Scale Images.

Theory / Algorithm / Conceptual Description:

HILL CIPHER

Hill cipher is a polygraphic substitution cipher based on linear algebra. Each letter is represented by a number modulo 26. Often the simple scheme A = 0, B = 1, ..., Z = 25 is used, but this is not an essential feature of the cipher. To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, against modulus 26. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption. The matrix used for encryption is the cipher key, and it should be chosen randomly from the set of invertible n × n matrices.

Encryption

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} = \begin{bmatrix} 67 \\ 222 \\ 319 \end{bmatrix} \equiv \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} \pmod{26}$$

Decryption

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}^{-1} \equiv \begin{bmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{bmatrix} \pmod{26}$$

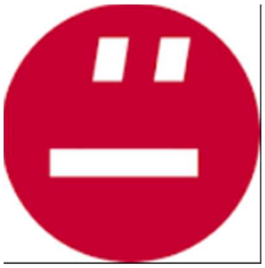
ALGORITHM FOR HILL CIPHER:

- A pixel matrix of the original image of dimensions n×n is constructed.
- The plain image is divided into n×n symmetric blocks.
- Generate a random key matrix of n×n
- For grayscale images, the number of levels is equal the number of alphabets.
- For color images-
- decompose the color image into (R-G-B) components.
- encrypt each component (R-G-B) separately.
- obtain the cipher image.

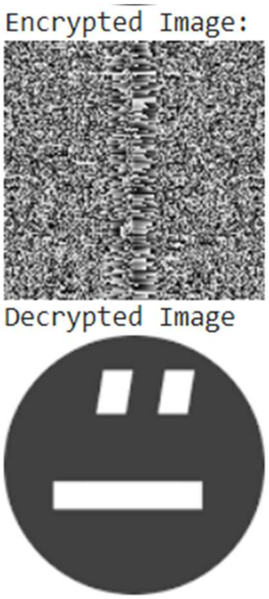
Program

```python
# Hill Cipher
from PIL import Image
import numpy as np
import sys
import numpy
import random
import requests
from io import BytesIO

numpy.set_printoptions(threshold=sys.maxsize)
url = "https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcRx8qFpKYKIfC-
Y166zoONiYOvYgC6vAE5XSY8IBqIovw&s"

response = requests.get(url)
img = Image.open(BytesIO(response.content)).convert("L")
img1 = img.convert("L")
print("Original image")
img1.show()
img_matrix=np.array(img1)
dimensions=img_matrix.shape

key_matrix=[]
for _ in range(dimensions[0]):
  l=[]
  for i in range(dimensions[1]):
    l.append(random.randint(0,1))
  key_matrix.append(l)
key_matrix=np.array(key_matrix)

def matmul(text,key):
  return np.matmul(text,key)

def encrypt(key , img):
  return matmul(key, img)
encp_matrix=encrypt(key_matrix,img_matrix)

def decrypt(key_matrix , img_matrix):
  inv_matrix=np.linalg.inv(key_matrix)

  decp_matrix=matmul(inv_matrix,encp_matrix)
  err_matrix=decp_matrix-img_matrix
  zero_matrix=matmul(key_matrix,inv_matrix)
  round_list_decp=[]
  for i in range(130):
    l=[]
    for j in range(130):
      l.append(round(decp_matrix[i][j]))
    round_list_decp.append(l)

  # print(round_list_decp==img_matrix)
  round_list_decp_np=np.array(round_list_decp)
  return round_list_decp_np

dec = decrypt(key_matrix  , img_matrix)
print("Encrypted Image:")
img = Image.fromarray(encp_matrix.astype(np.uint8))
img.show()
print("Decrypted Image")
img = Image.fromarray(dec.astype(np.uint8))
img.show()
```

Input



Output



Encrypted Image:

Decrypted Image

CONCLUSION: Hence we have successfully encrypted and decrypted image using hill cipher.