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 1

Date of Performance : 22-03-2022 **Date of Submission:** 23-03-2022

SAP Id: 60004190053 **Name :** Jayesh Kavedia

Div: A **Batch :** A4

Aim of Experiment

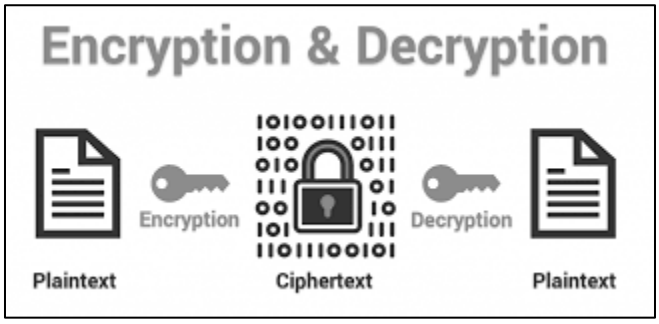
Design and Implement Encryption and Decryption Algorithm for

a. Caesar cipher cryptographic algorithm by considering letter [A..Z] and digits [0..9]. Create two functions Encrypt() and Decrypt(). Apply Brute Force Attack to reveal secret. Create Function BruteForce(). Demonstrate the use of these functions on any paragraph.
b. Hill Cipher. Your Program Must Input Image in Gray Scale. Choose keys according to Gray Scale Intensity level. Create two functions Encrypt() and Decrypt(). Make sure to have Multiplicative Inverse Exists for one of the Key in selected Key pair of Affine Cipher.

(CO1)

Theory / Algorithm / Conceptual Description

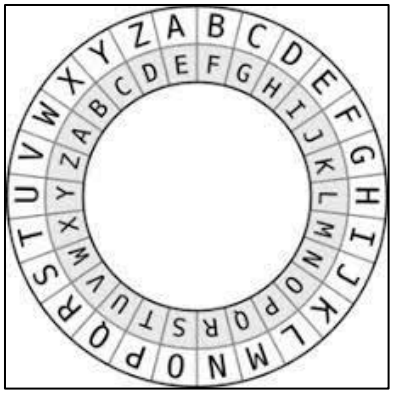
Encryption and Decryption



Encryption is the process of converting a readable message into an unreadable form so that it cannot be read by unauthorized parties. The process of converting an encrypted message back to its original (readable) format is known as decryption. The plaintext message is the original message. The ciphertext message is the encrypted message. Digital encryption techniques function by mathematically modifying the digital content of a plaintext message and producing a ciphertext version of the message using an encryption algorithm and a digital key. If the sender and recipient are the only ones who have access to the key, they can communicate safely.

Caesar Cipher Encryption Algorithm

The Caesar cipher, often known as the Shift cipher, is one of the oldest and most basic ciphers. It's a mono-alphabetic substitution cipher in which each letter in the plaintext is 'shifted' down the alphabet a fixed number of times. With a shift of one, for example, A would be replaced by B, B by C, and so on. If the attacker is aware that Caesar cipher is being used for encryption, decrypting it is simple. In this situation, the attacker can use the brute force method to determine the key by applying each key value in the range of characters and selecting the most meaningful answer.



Algorithm

CipherText = (PlainText + Key) % 26 (For A-Z)

PlainText = (CipherText - Key) % 26 (For A-Z)

Advantages

1. Only one short key is used in its entire process.
2. If a system does not use complex coding techniques, it is the best method for it.
3. It requires only a few computing resources.

Disadvantages

1. The message encrypted by this method can be easily decrypted.
2. It provides very little security.

Hill Cipher Encryption Algorithm

The Hill cipher is a polygraphic substitution cipher based on Linear Algebra principles. The Hill cipher is a more mathematical cipher than others since it uses modulo arithmetic, matrix multiplication, and matrix inverses. Because the Hill cipher is also a block cipher, it can theoretically function with blocks of any size. Encryption is based on modular arithmetic and matrix multiplication of the Key matrix and the Plain Text matrix or vector. The Cipher Text matrix or vector is multiplied by the inverse of the key matrix in modular arithmetic for decryption.

Algorithm

CipherText = (Key * PlainText) mod 26 (For A-Z)

PlainText = (Key^{-1} * CipherText) mod 26 (For A-Z)

where, Key^{-1} = ((Multiplicative Inverse of |D|) % 26 * (Adjoint (Key)) % 26) % 26 (For A-Z)

Limitation of Hill Cipher

Hill's cipher is susceptible to a known plaintext attack. If we had a set of n matching plaintext/ciphertext pairs, it is a straightforward process to recover the encryption matrix.

Program

A)

```
characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
# Encryption
PT = 'UkraineIsACountryInEasternEurope'
PT = PT.upper()
CT = ''
key = 5
for char in PT:
  temp = (characters.find(char) + key)%36
  CT += characters[temp]
print('Cipher Text for given Plain Text : ',CT)
# Brute Force for Decryption
for k in range(36):
  PT = ''
  for char in CT:
    diff = characters.find(char) - k
    if diff >= 0:
      PT += characters[diff]
    else:
      PT += characters[diff + 36]
  print(f'Plain Text for Cipher Text with Key {k} is {PT}')
```

Output

Encryption
Cipher Text for given Plain Text : ZPWFNSJNXFHTZSYW3NSJFXYJWSJZWTUJ

Decryption Brute Force
Plain Text for Cipher Text with Key 0 is ZPWFNSJNXFHTZSYW3NSJFXYJWSJZWTUJ
Plain Text for Cipher Text with Key 1 is YOVEMRIMWEGSYRXV2MRIEWXIVRIYVSTI
Plain Text for Cipher Text with Key 2 is XNUDLQHLVDFRXQWU1LQHDVWHUQHXURSH
Plain Text for Cipher Text with Key 3 is WMTCKPGKUCEQWPVT0KPGCUVGTPGWTQRG
Plain Text for Cipher Text with Key 4 is VLSBJOFJTBDPVOUSZJOFBTUFSOFVSPQF
Plain Text for Cipher Text with Key 5 is UKRAINEISACOUNTRYINEASTERNEUROPE
Plain Text for Cipher Text with Key 6 is TJQ9HMDHR9BNTMSQXHMD9RSDQMDTQNOD
Plain Text for Cipher Text with Key 7 is SIP8GLCGQ8AMSLRPWGLC8QRCPLCSPMNC
Plain Text for Cipher Text with Key 8 is RHO7FKBFP79LRKQOVFKB7PQBOKBROLMB
Plain Text for Cipher Text with Key 9 is QGN6EJAEO68KQJPNUEJA6OPANJAQNKLA
Plain Text for Cipher Text with Key 10 is PFM5DI9DN57JPIOMTDI95NO9MI9PMJK9
Plain Text for Cipher Text with Key 11 is OEL4CH8CM46IOHNLSCH84MN8LH8OLIJ8
Plain Text for Cipher Text with Key 12 is NDK3BG7BL35HNGMKRBG73LM7KG7NKHI7
Plain Text for Cipher Text with Key 13 is MCJ2AF6AK24GMFLJQAF62KL6JF6MJGH6
Plain Text for Cipher Text with Key 14 is LBI19E59J13FLEKIP9E51JK5IE5LIFG5

Plain Text for Cipher Text with Key 15 is KAH08D48I02EKDJHO8D40IJ4HD4KHEF4
Plain Text for Cipher Text with Key 16 is J9GZ7C37HZ1DJCIGN7C3ZHI3GC3JGDE3
Plain Text for Cipher Text with Key 17 is I8FY6B26GY0CIBHFM6B2YGH2FB2IFCD2
Plain Text for Cipher Text with Key 18 is H7EX5A15FXZBHAGEL5A1XFG1EA1HEBC1
Plain Text for Cipher Text with Key 19 is G6DW4904EWYAG9FDK490WEF0D90GDAB0
Plain Text for Cipher Text with Key 20 is F5CV38Z3DVX9F8ECJ38ZVDEZC8ZFC9AZ
Plain Text for Cipher Text with Key 21 is E4BU27Y2CUW8E7DBI27YUCDYB7YEB89Y
Plain Text for Cipher Text with Key 22 is D3AT16X1BTV7D6CAH16XTBCXA6XDA78X
Plain Text for Cipher Text with Key 23 is C29S05W0ASU6C5B9G05WSABW95WC967W
Plain Text for Cipher Text with Key 24 is B18RZ4VZ9RT5B4A8FZ4VR9AV84VB856V
Plain Text for Cipher Text with Key 25 is A07QY3UY8QS4A397EY3UQ89U73UA745U
Plain Text for Cipher Text with Key 26 is 9Z6PX2TX7PR39286DX2TP78T62T9634T
Plain Text for Cipher Text with Key 27 is 8Y5OW1SW6OQ28175CW1SO67S51S8523S
Plain Text for Cipher Text with Key 28 is 7X4NV0RV5NP17064BV0RN56R40R7412R
Plain Text for Cipher Text with Key 29 is 6W3MUZQU4MO06Z53AUZQM45Q3ZQ6301Q
Plain Text for Cipher Text with Key 30 is 5V2LTYPT3LNZ5Y429TYPL34P2YP52Z0P
Plain Text for Cipher Text with Key 31 is 4U1KSXOS2KMY4X318SXOK23O1XO41YZO
Plain Text for Cipher Text with Key 32 is 3T0JRWNR1JLX3W207RWNJ12N0WN30XYN
Plain Text for Cipher Text with Key 33 is 2SZIQVMQ0IKW2V1Z6QVMI01MZVM2ZWXM
Plain Text for Cipher Text with Key 34 is 1RYHPULPZHJV1U0Y5PULHZ0LYUL1YVWL
Plain Text for Cipher Text with Key 35 is 0QXGOTKOYGIU0TZX4OTKGYZKXTK0XUVK

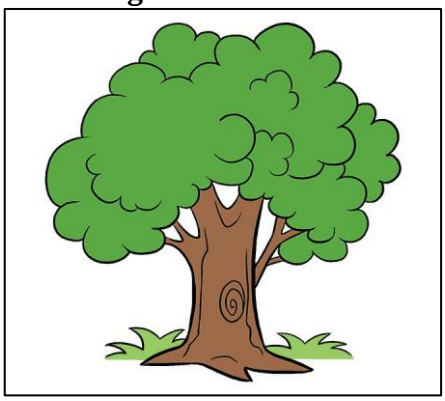
Program

B)

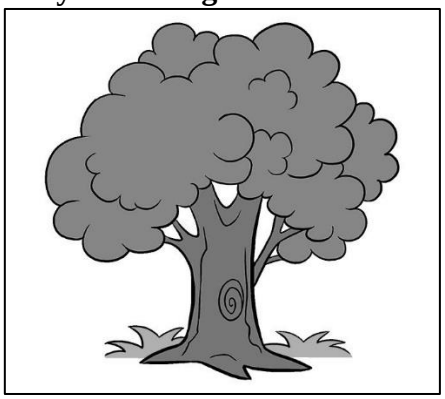
```
import cv2
import numpy as np
img = cv2.imread('tree.jpg',cv2.IMREAD_GRAYSCALE)
cv2.imwrite('gray.jpg',img)
def generate_key_matrix(n): #Creating a Involutory Matrix
  Mod = 256
  k = 23
  d = np.random.randint(256, size = (int(n/2),int(n/2)))
  I = np.identity(int(n/2))
  a = np.mod(-d,Mod)

  b = np.mod((k * np.mod(I - a,Mod)),Mod)
  k = np.mod(np.power(k,127),Mod)
  c = np.mod((I + a),Mod)
  c = np.mod(c * k, Mod)

  A1 = np.concatenate((a,b), axis = 1)
  A2 = np.concatenate((c,d), axis = 1)
  A = np.concatenate((A1,A2), axis = 0)
  return A
key = generate_key_matrix(img.shape[0])
encrypted_image = np.mod(np.matmul(key,img),256) #Encryption
cv2.imwrite('encrypted.jpg',encrypted_image)
key_inv = key # For a involutory matrix the matrix is its own inverse
decrypted_image = np.mod(np.matmul(key_inv,encrypted_image),256) #Decryption
cv2.imwrite('decrypted.jpg',decrypted_image)
```

Output

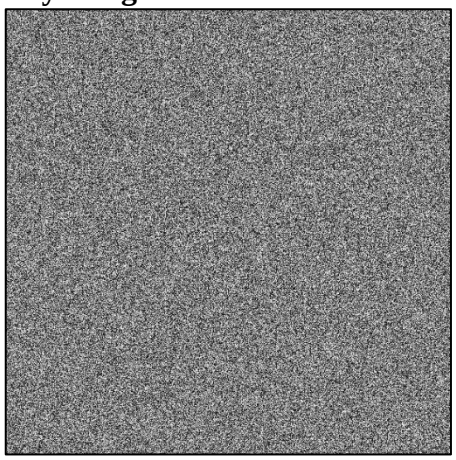
Real Image



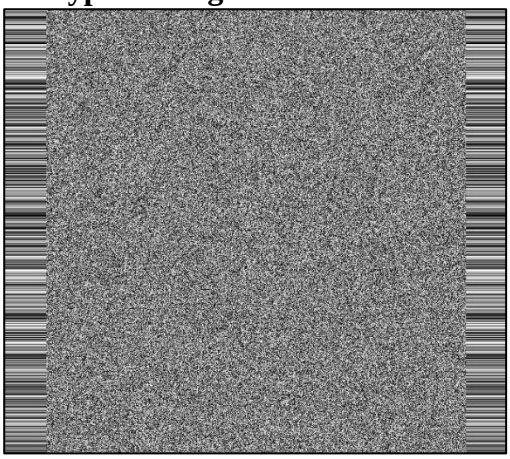
Grayscale Image



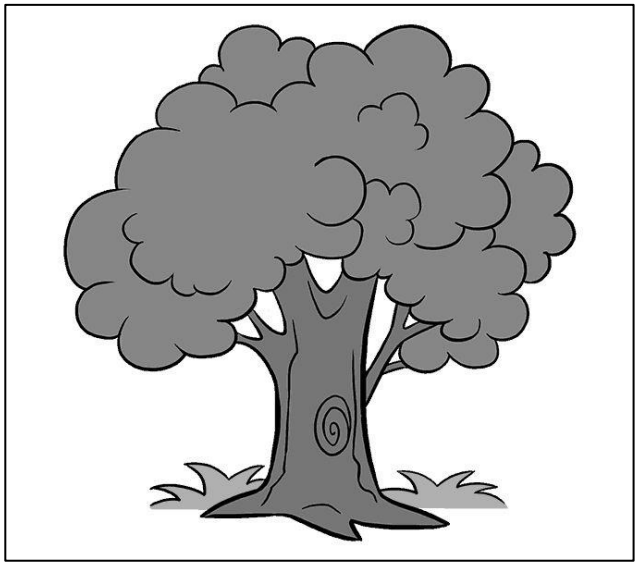
Key Image



Encrypted Image



Decrypted Image



Conclusion

Information Security of messages can be ensured by using Encryption and Decryption techniques and a common key between the sender and the receiver. Caesar Cipher Encryption Technique is the simplest substitution mono-alphabetic encryption technique. However, as it is easy to encrypt it provides very little security and can be decrypted easily. Hill Cipher is a Polygraphic substitution encryption algorithm which is more related to mathematical concepts to encrypt the data. It provides security to some extent but if set of PT's are mapped to set of CT's the key matrix can be easily determined and the data can be decrypted.