IMAGE CRYPTOGRAPHY BY RUBIKS CUBE PRINCIPLE

Fall Semester 2020-21

Team Members :

P.Sabiha – 17MIS0064

S.Swetha – 17MIS0084

Slot : D2

Guided By :

Dr.Navaneethan.C

Associate Proffessor Grade1

School Of Information Technology and Engineering

INFORMATION SECURITY ANALYSIS AND AUDIT – CSE3501

Abstract :

Cryptography is well known and widely used technique that manipulate information in order to cipher or hide their existence.

In the past years, a few encryption methods dependent on different algorithms have been proposed intended to secure digital pictures against cryptographic attacks. Chaos-based image encryption algorithms are generally used more often than others but require high computational cost. Moreover, this system is defined on real numbers whereas the cryptosystems are defined on finite sets of integers and also these systems are implemented using small key spaces which is insecure, especially in case of one-dimensional algorithms. The common image encryption algorithms such as DES, AES, RSA and the family of elliptic curve-based encryption (ECC) are also not helpful for fast and real-time communication applications for image encryption. Due to the high information redundancy some encryption schemes based on permutation have also been found insecure against various attacks.

In our project we are going to develop a code for more secure and fast encryption of images using a unique image encryption algorithm based on Rubik's cube principle. The pixels of the image are shuffled in a way similar to that of a Rubik's cube in a random manner using two randomly generated vectors. Then the same vectors are used for performing bitwise operation row-wise and column-wise. The XOR operator is applied to odd rows and columns of image using a key to decrease the association between original and encrypted images. The same key is flipped and applied to even rows and columns of image.

Finally, the algorithm which we used to develop the encryption system i.e.,Rubik's cube principle not only can achieve good encryption standards and perfect hiding ability but also can resist exhaustive attack, statistical attack and differential attack.

Various research, performance assessment tests and experimental tests done on similar type of image encryption algorithm shows it is suitable for real-time Internet encryption and transmission applications because of its fast encryption or decryption speeds it also demonstrates the robustness of the proposed algorithm against several types of attacks.

Keywords: Rubik's cube algorithm, XOR operator, Chaos-based image encryption algorithms.

Introduction :

In the recent age, technology has moved leaps but it has also come with its downsides. One of them is illegal copying of digital intellectual property. Several works have been done to curb this issue. Some of the main ones are done by using encryption. This project focuses on using encryption for protecting digital images. Encryption is a process of transforming data into an unreadable format using certain algorithms to make sure that the data is available to only legitimate users i.e. only authorized parties can access the data.

The objective of this project is encryption of mainly digital images. There are already well known encryption methods such as symmetric-key algorithms (DES,AES,IDEA), asymmetric-

key algorithms(RSA) and also algorithms based on Elliptic-curve-cryptography in place for data encryption but these are not the most suitable for image encryption. This is mostly applicable in real-time communication or in cases where fast encryption is needed.

The proposed encryption schemes in recent years can be mainly classified into categories as, value transformation, pixels position permutation, and chaotic systems.

Image encryption has been studied extensively. Some permutation-based encryption schemes have already been found to be insecure against certain attacks such as cipher-text only and chosen-plaintext attacks. It is due to the high data redundancy in such schemes and as secret permutations can be recovered by plaintext and ciphertext comparison,It is quite understandable. Although, chaos-based algorithms are used more often in image encryption in general, they have a high computational cost. Also, chaotic system are defined on real numbers whereas cryptosystems are defined on a finite set of integers.

In this project, we are going to consider the principle of rubik's cube for image encryption. We are going to implement this algorithm which has been specified and extensively studied in the paper mentioned below.

Literature Survey :

S.No	Authors	Challenges and Issues	Methodologies used	Pros	Cons	Applicatio-ns
1.	Sirisha,M.& Lakshmi, S. V. V. S. (2014)	In this paper, for analysing security issues can be done only by the key space and key sensibility.	The very concept of Rubik's cube depends on rows and columns. The principle is applied to rows and these rows rotated according to the requirement. The image is divided into rows and each row pixel transformation is applied.	Our proposed Rubik's cube image encryption is very fast and high secure.	In this paper the methods that used are implemented ethically or else the principle may not give the desired solutions	Real-time Internet encryption and transmissio n applications
2.	Amirtharaja n, Abhiram, Revathi.G, Reddy.J. B,	The encryption algoritm is created by	In this proposed methodology the pixels are first scrambled and then	This accounts for improved randomness that can resist	It creates complex in the security systems in	It is used in discovering more numerous

	Thanikaiselvan, V & Rayappan, J. B. B. (2013)	taking the Rubic cube fundamentals which create more complex along with the security as well.	data is embedded based on their intensity values. For scrambling, the famous Rubik's cube methodology is implemented The algorithm comprises of both image scrambling and embedding to pull off the aspiration of secret sharing. Scrambling procedure is defined which involves shifting of bits.	any steganalysis. And also it is benevolant idea for higing the secret messages	encryption algorithm.	theories and concepts in mathematics and in engineering for real time applications
3.	Nagarajan, B., & Manju, B.	It is complex because we use scrumbled image where the image is scrumbled using the cross over and mutation operations which were done by random number generator.	In the proposed method, first the bit level permutation is applied to the original image to shift the pixel positions. Then the genetic operators are used to diffuse the pixel values.	The entropy values of the encrypted images are very close to the ideal value of 8Sh, which means that the encryption algorithm is highly robust against entropy attacks.	It is a large process where there are many methods included for every step process	The encypted algorithm is high secure and used for all real time applications
4.	Abdullatif, A.A, Abdullatif F.A & Naji S. A. (2019)	using confusion techniques alone is not enough even though they are simple and fast techniques.	The proposed hybrid image encryption algorithm comprises two main phases: First a modified Rubik's cube encryption is	Bitwise processing enhances the performance in accordance to speed and it is very difficult to vision with	the use of different approaches in one integrated system, where one approach can compensate	The basic DNA encoding rules are based on the basic algebraic operations that offer

		To enhance performance, the system should be augmented with other techniques. In this study, the Rubik's cube algorithm is augmented with dynamic DNA algorithm.	employed to shuffle the pixels (i.e., confusion). Next a dynamic DNA algorithm is applied to provide an enhanced level of encryption (i.e., diffusion).	the human eye.	for the weaknesses of another approach.	high-speed encryption compared to the complex mathematic-cal operations that other techniques use. Thus, the proposed system is appropriate for many practical applications
5.	Majid Javid Moayed	Both of the existing and the proposed systems accompany security, during voting days from voter registration day to day of final results publication.	Method which In-corporates cryptography and vote security with voter verification. The bases of this method are Rubik's Cube game tools. Simple rules are used for selecting candidate and Rubik's cube is used for mixing candidate for encrypting ballot	RCV method is more secure and it is exoteric enough to be understood by everyone	In this paper that all the violanted actions in the proposed system are not rectified somehow security and cheating issues were decresed	Voting Systems and some game tool applications
6.	Venkata Krishna Pavan.K, Hemanth.V, Vishnu.R, Agilandeeswari.L	We cannot use the color images.	In this novel they propose a secure image encryption algorithm that uses both AES and Visual	In this process we can use different types of images. Here we use images and keys for image processing were the	In the image processing the image is not colored and we can't use big size cipher text and images.	It has many applications in many fields including Banking, Telecommu nication and Medical Image

			Cryptographic techniques to protect the image	hacker could not possibly get to know the algorithms used in the encryption		Processing etc.
7.	Savitha.B	The LSB Substitution Steganograp-hic method is only effective for bit images as they involve in loseless compression.	The current project aims to use steganography for an image with another image using spatial domain technique	In this paper the LSB method,the results obtained in the data hiding are pretty impressive.	In this the image and the text should be in the same size were it is not flexible.	This method is used for hobbyists, secretive data transmissio n, for privacy of users etc.
8.	Dr. Ekta Walia , Payal Jain, Navdeep	The text accepted in this steganograph y is only in LSB and DC coefficients	This paper presents analysis of Least Significant Bit (LSB) based Steganography and Discrete Cosine Transform (DCT) based Steganography.	DCT based steganograph y scheme is recommended because of the minimum distortion of image quality.	For the DCT, Less amount of the data is taken by this method when compared to LSB.	There are used to regain or identify the images with clarity that are not clear in camera
9.	Elena Acevedo, Antonio Acevedo, Fabiola Martínez, and Ángel Martínez	To use the system the developer should be experienced as it is a longer process.	we use the Alpha-Beta associative memory which has demonstrated to be a suitable tool for the Pattern Recognition area.	The main advantage of this method is that the encrypted image does not have the same size than the original image; therefore, since the beginning the adversary cannot know	There are less disadvantage s as it is hard to decrypt the message	To create the systems the patterns are used to slove the technical issues and system to be effective

				what the image means.		
10.	G.A. Papakostas, D.E. Koulouriotis and E.G. Karakasis	The efficiency of the system with this algorithm is less when compared to the systems with the other algorithm.	In this paper the method used is Discrete Cosine Transfer(DCT)	Lots of the information is stored in the images with minimum redandancy	The algorithm is somewhat complexity to study the computatio-nal efficiency	These are used in the multimedia systems.
11.	Image compression system using ANN Disha Parkhi ,S.S.Lokhande.(2018)	compression ratio provides a compromise between high compression ratio and minimal loss of quality	Discrete Cosine Transform (DCT) is one of the simplest commonly used compression methods.	DCTs are important to numerous applications in science and engineering, from lossy compression of audio and images	The disadvantage of using DCT image compression is the high loss of quality in compressed images, which is more notable at higher compression ratios.	desktop publishing, multimedia teleconferencing are the applications used
12	A METHOD TO COMPRESSION - THEN - ENCRYPTION FOR IMAGES S Sankar , S Nagarajan (2017)	Compressed data can vary considerably for small changes in the source data, therefore making it very Difficult to compress the data.	Differential Pulse Code Modulation (DPCM) method for dc and Run Length Encode (RLE) for ac coefficients respectively	The experimental results demonstrate the effectiveness of the proposed scheme in image compression	Compressed data can vary considerably for small changes in the source data, therefore making it very difficult to perform differential cryptanalysis on	Many data processing Applications are used

					compressed data	
13	Secure secret image carrier using Rubik's cube and modified LSB. Rehka,Goudar,Rohit (2018)	this image encryption is responsible to make confusion to unauthorized people	LSB method key is used to identify the initial position of embedding	by using the secret keys XOR administrator is connected into lines and sections of the scrambled picture.	it is difficult to unscramble (decrypt) secret picture because of properties of encryption system perplexity and dissemination, furthermore its substantial key space.	The process is done By utilizing matlab Tool
14	Binary Encryption Based On a Rubik's Cube. Vyom Chhabra, Tejeswini Sundaram	The challenge of not knowing the key length provides a stronger base for this algorithm, as the key specific attacks will involve more permutations	The chaotic encryption methods are used	The paper says that the offered scheme can be applied to many different data types, such as audio, image and video.	By this encryption technique we aim to make it computational very hard and difficult for interceptors to decrypt the message being passed	Signal processing
15	Effective and Key Sensitive Security Algorithm For An Image Processing Using Robust Rubik Encryption &	Ignorance of the verification process or difficulty in voter verification has caused voter confusion	Robust encryption method	Calculation of probability of cheating shows that RCV method is more secure and for using famous game tools is exoteric enough to be understood by everyone	Comparing long complexity text or tiny pixels for decryption or encryption [13] in order to verify the vote is another problem.	DRE systems, training poll

	Decryption Process Seetaiah Kilaru, Yojana Kanukuntla, Asma Firdouse, Mohammad Bushra & Sindhu chava(2013)					
16	KENKEN PUZZLE – BASED IMAGE ENCRYPTION ALGORITHM Adrian-Viorel (2015)	To use the system the developer should be experienced as it is a longer process	conventional methodology are used	At this point we can conclude that the newly proposed confusion strategy offers better performances in comparing with other Fridrich's structure based image encryption algorithm	Hard to decrypt the message	Military application. Visual encryption technique has many application areas like multimedia , biometrics, image processing, GIS, military communication etc. real-time multimedia application are used

17	An improved color algorithm with pixel permutation. and bit Lini Abraham , Neenu Daniel2 (2013)	Security is one of the main issues in transferring such sensitive information	substitution method based on DNA sequences are used to change the value of each pixel on the image	Chaos theory has proved to be an excellent alternative to provide a fast, simple, and reliable image encryption scheme that has a high enough degree of security	Security issues are faced	Complementary transformation
18	Enhancem-ent of Security in Visual Cryptography using DES Anisha Maria Coelho Prabhu , Pradyumna G. R. (2015)	To enhance performance,the System should be augmented with other techniques	visual information to be encrypted in such a manner that the decryption can be done by humans Cryptography refers to the study of mathematical techniques and related aspects of message security like data confidentiality, data integrity and of data authentication	The proposed system can be extended using lossless Image compression methodology. Compressed image has less redundancy than the original image, crypt analysis will be difficult.	The speed of embedding data into the Image is being increased.	These are used in multimedia System.
19	Secure Communication Based On Rubik's Cube Algorithm And Chaotic Baker Map Abitha K.A , Pradeep K Bharathan b.(2016)	Only by the key space analyzing Security issues are done	DRPE with baker map method is used.	There is no data loss during the process of encryption and decryption	We cannot use big size cipher text	Matlab is used to implement the proposed method.

20	Multimedia Encryption Using Visual Cryptography Krutika Solanki , Vidisha Vankani, Pooja Pukle, Sridhar Iyer(2016)	. It provides security to beat today's authentication challenges and is very easy to implement	symmetric encryption methods such as DES, AES are used.	This scheme can become a reliable solution suitable for today's authentication challenges and it can replace the existing traditional techniques.	The existing systems only encrypted and decrypted files in text format which was seen as a drawback	Encrypted algorithm is high Secure

Proposed System :

In the recent days image proceesing is the most used technique were the information should be hidden to everyone except to the authenticated access. Cryptography and Steganography are well known and widely used techniques that manipulate information (messages) in order to cipher or hide their existence. There are so many Existing systems for the image cryptography like pixel transformation using rubic cube principle, a way for random image steganography, secure and verificable cryptographic scheme using rubic principle these are some of the existing systems of the image cryptography were in every system there are many advantages and also disadvanatges that is the reason to propose the new systems were the proposed system may not be perfect but it may decrease the disadvantages when compared to the existing systems.

In the pixel transformation using the rubic cube principle is used to In this day and age of accelerated and convenient media for transmission and use of data resources the need for security has never been more profound and the methods that used in pixel transformation are implemented ethically or else the principle may not give the desired and the effective solutions which may lead to the wrong information.

In a way for random image steganography, were it is a best method for hiding the data bulit creates complex in the security systems in encryption algorithm.

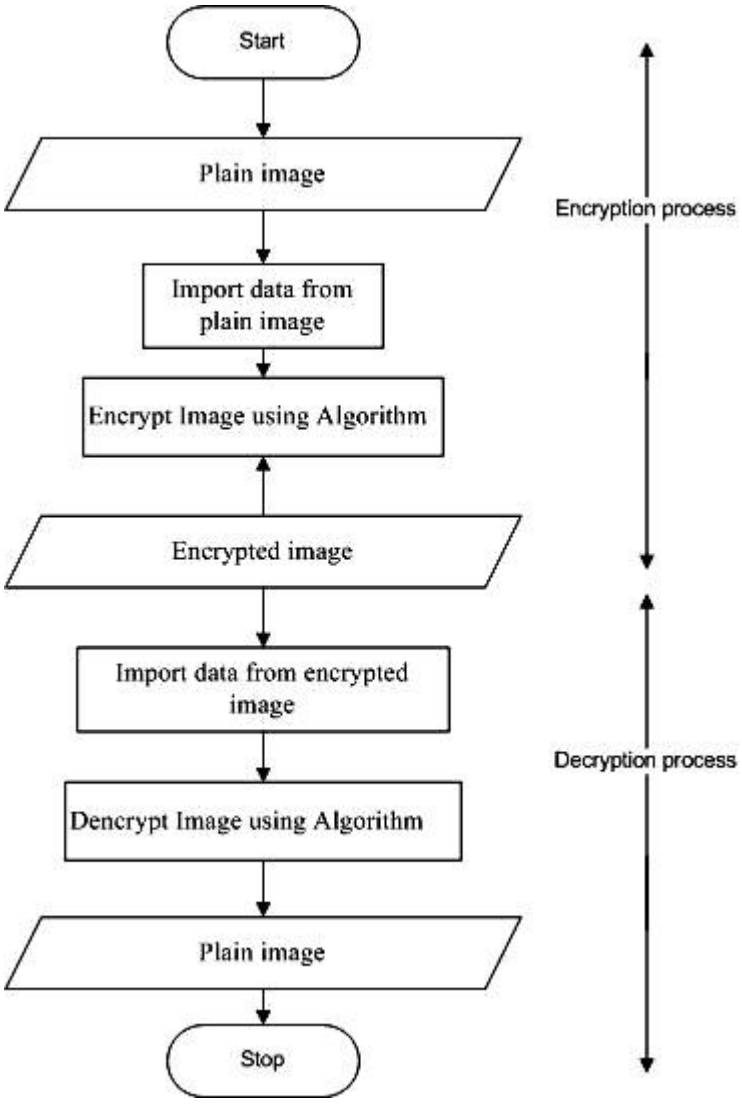
Where some image cryptography systems were implemented by using the AES, DES, ECC, DCT were they are most secure, Reasonable cost, Flexibility, Simplicity but these are common encryption techniques and also not helpful for fast and real-time communication applications for image encryption. Due to the high information redundancy some encryption schemes based on permutation have also been found insecure against various attacks.

So, the image cryptography using rubic cube principle the algorithm which we used to develop the encryption system i.e.,Rubik's cube principle not only can achieve good encryption

standards and perfect hiding ability but also can resist exhaustive attack, statistical attack and differential attack.

METHODOLOGY

Flow Chart:



Rubik's cube algorithm:

Encryption Algorithm

Let Io = pixels values matrix of a α-bit grayscale image ($M \times N$).
Steps involved in encryption are:

1. Create two vectors KR and KC with random values of length M and N, where the values in the vectors should be >0 and $<2^{\alpha}$ -1. (KR and KC should not have constant values)
2. Set the iteration value, ITERmax, and initialize the counter ITER at 0.
3. Increment the counter by one: ITER=ITER+1.
4. For each row i of image Io,
    a. compute the sum of all elements in the row i, this sum is denoted by
    $\alpha(i)$
    $(i)=Nj=1I(i,j),i=1,2,\ldots,M$
    b. compute modulo 2 of $\alpha(i)$, denoted by $M\alpha(i)$,
    c. row i is left, or right, circular-shifted by $KR(i)$ positions (image pixels are moved $KR(i)$ positions to the left or right direction, and the first pixel moves in last pixel.), according to the following: If $M\alpha(i)=0 \longrightarrow$right circular shift a. else$\longrightarrow$left circular shift

5. For each column j of image Io,
    a. compute the sum of all elements in the column j, this sum is denoted by
    $\beta(j)$,
    $(j)=Mi=1I(i,j),j=1,2,\ldots,N.$
    b. compute modulo 2 of $\beta(j)$, denoted by $M\beta(j)$.
    c. column j is down, or up, circular-shifted by $KC(i)$ positions, according to the following:
    if$(j)=0 \longrightarrow$up circular shift else$\longrightarrow$down circular shift.

Steps 4 and 5 above will create a scrambled image denoted by ISCR.

6. Using vector KC, the bitwise XOR operator is applied to each row of scrambled image using the following expressions:
    $I1(2i-1,j)=ISCR(2i-1,j) \oplus KC(j)$,
    $I1(2i,j)=ISCR(2i,j) \oplus \text{rot180}(KC(j))$
    where $\oplus$ and rot180(KC) represent the bitwise XOR operator and the flipping of vector KC from left to right, respectively.

7. Using vector KR, the bitwise XOR operator is applied to each column of image $I1$ using the following formulas:

$$\text{IENC}(i,2j-1)=\text{I1}(i,2j-1)\oplus KR(j)$$
$$\text{IENC}(i,2j)=\text{I1}(i,2j)\oplus \text{rot180}(KR(j))$$

with rot180(KR) indicating the left to right flip of vector KR.

8. If ITER=ITERmax, then encrypted image IENC is created and encryption process is done; otherwise, the algorithm branches to step 3.

KR, KC & ITER max are the secret keys.

Decryption Algorithm

The decrypted image, Io, is recovered from the encrypted image, IENC, and the secret keys, KR, KC, and ITERmax as follows in the following.

1. Initialize ITER=0.
2. Increment the counter by one: ITER=ITER+1.
3. The bitwise XOR operation is applied on vector KR and each column of the encrypted image IENC as follows:

$$\mathrm{I}_1(i,2j-1)=\mathrm{I}_{\mathrm{ENC}}(i,2j-1)\oplus K_R(j),$$
$$\mathrm{I}_1(i,2j)=\mathrm{I}_{\mathrm{ENC}}(i,2j)\oplus \mathrm{rot}\,180\left(K_R(j)\right),$$

4. Then, using the KC vector, the bitwise XOR operator is applied to each row of image
$I1$:

$$\mathrm{I}_{\mathrm{SCR}}(2i-1,j)=\mathrm{I}_1(2i-1,j)\oplus K_C(j),$$
$$\mathrm{I}_{\mathrm{SCR}}(2i,j)=\mathrm{I}_1(2i,j)\oplus \mathrm{rot}\,180\left(K_C(j)\right).$$

5. For each column j of the scrambled image ISCR,
    a. compute the sum of all elements in that column j, denoted as βSCR(j):

$$\beta_{\mathrm{SCR}}(j)=\sum_{i=1}^{M}\mathrm{I}_{\mathrm{SCR}}(i,j),\quad j=1,2,\ldots,N,$$

b. compute modulo 2 of βSCR(j), denoted by $M\beta$SCR(j),
c. column j is down, or up, circular-shifted by $KC(i)$ positions according to the following:

$$\text{if } M_{\beta_{\text{SCR}}(j)} = 0 \longrightarrow \text{up circular shift}$$
$$\text{else} \longrightarrow \text{down circular shift.}$$

6. For each row i of scrambled image ISCR,
   a. compute the sum of all elements in row i, this sum is denoted by αSCR(i):

$$\alpha_{\text{SCR}}(i) = \sum_{j=1}^{N} \text{I}_{\text{SCR}}(i,j), \quad i = 1, 2, \ldots, M,$$

   b. compute modulo 2 of αSCR(j), denoted by $M\alpha$SCR(j),
   c. row i is then left, or right, circular-shifted by $KR(i)$ according to the following:

$$\text{if } M_{\alpha_{\text{SCR}}(j)} = 0 \longrightarrow \text{right circular shift}$$
$$\text{else} \longrightarrow \text{left circular shift.}$$

7. If ITER=ITERmax, then image IENC is decrypted and the decryption process is done; otherwise, the algorithm branches back to step 2.

APPENDIX

CODE:

Encryption: To encrypt the image we need to save the images in a folder named “input”. Code takes image input using PIL library and converts it into RGB matrices and starts the encryption process using the Rubik’s cube algorithm. The encrypted images are then saved in another folder.

After the encryption the the values of Kr, Kc and Iterations(ITER_MAX) saved in a file named “keys”.

Encrypt.py

```
from PIL import Image

from random import randint
```

```
import numpy
import sys
from helper import *
im = Image.open(r"C:\Users\MashaAllah\Desktop\Image-cryptography\input\pic1.png");
pix = im.load()
#Obtaining the RGB matrices
r = []
g = []
b = []
for i in range(im.size[0]):
r.append([])
g.append([])
b.append([])
for j in range(im.size[1]):
        rgbPerPixel = pix[i,j]
        r[i].append(rgbPerPixel[0])
        g[i].append(rgbPerPixel[1])
        b[i].append(rgbPerPixel[2])

m = im.size[0]
n = im.size[1]
```

```
# Vectors Kr and Kc

alpha = 8

Kr = [randint(0,pow(2,alpha)-1) for i in range(m)]

Kc = [randint(0,pow(2,alpha)-1) for i in range(n)]

ITER_MAX = 1


print('Vector Kr : ', Kr)

print('Vector Kc : ', Kc)


f = open('keys.txt','w+')

f.write('Vector Kr : \n')

for a in Kr:

f.write(str(a) + '\n')

f.write('Vector Kc : \n')

for a in Kc:

f.write(str(a) + '\n')

f.write('ITER_MAX : \n')

f.write(str(ITER_MAX) + '\n')

for iterations in range(ITER_MAX):

# For each row

for i in range(m):

        rTotalSum = sum(r[i])
```

```
        gTotalSum = sum(g[i])
        bTotalSum = sum(b[i])
        rModulus = rTotalSum % 2
        gModulus = gTotalSum % 2
        bModulus = bTotalSum % 2
        if(rModulus==0):
            r[i] = numpy.roll(r[i],Kr[i])
        else:
            r[i] = numpy.roll(r[i],-Kr[i])
        if(gModulus==0):
            g[i] = numpy.roll(g[i],Kr[i])
        else:
            g[i] = numpy.roll(g[i],-Kr[i])
        if(bModulus==0):
            b[i] = numpy.roll(b[i],Kr[i])
        else:
            b[i] = numpy.roll(b[i],-Kr[i])
# For each column
for i in range(n):
        rTotalSum = 0
        gTotalSum = 0
        bTotalSum = 0
```

```
        for j in range(m):
                rTotalSum += r[j][i]
                gTotalSum += g[j][i]
                bTotalSum += b[j][i]
        rModulus = rTotalSum % 2
        gModulus = gTotalSum % 2
        bModulus = bTotalSum % 2
        if(rModulus==0):
                upshift(r,i,Kc[i])
        else:
                downshift(r,i,Kc[i])
        if(gModulus==0):
                upshift(g,i,Kc[i])
        else:
                downshift(g,i,Kc[i])
        if(bModulus==0):
                upshift(b,i,Kc[i])
        else:
                downshift(b,i,Kc[i])
# For each row
for i in range(m):
        for j in range(n):
```

```
            if(i%2==1):
                r[i][j] = r[i][j] ^ Kc[j]
                g[i][j] = g[i][j] ^ Kc[j]
                b[i][j] = b[i][j] ^ Kc[j]
            else:
                r[i][j] = r[i][j] ^ rotate180(Kc[j])
                g[i][j] = g[i][j] ^ rotate180(Kc[j])
                b[i][j] = b[i][j] ^ rotate180(Kc[j])
# For each column
for j in range(n):
    for i in range(m):
            if(j%2==0):
                r[i][j] = r[i][j] ^ Kr[i]
                g[i][j] = g[i][j] ^ Kr[i]
                b[i][j] = b[i][j] ^ Kr[i]
            else:
                r[i][j] = r[i][j] ^ rotate180(Kr[i])
                g[i][j] = g[i][j] ^ rotate180(Kr[i])
                b[i][j] = b[i][j] ^ rotate180(Kr[i])
for i in range(m):
for j in range(n):
    pix[i,j] = (r[i][j],g[i][j],b[i][j])
```

im.save(r"C:\Users\MashaAllah\Desktop\Image-cryptography\encrypted_images\pic1-encrypt.png")

Decryption: Decryption code takes Kr, Kc, Iterations(ITER_MAX) values along with the encrypted image and start the decryption process.

Decrypt.py

```
from PIL import Image

from random import randint

import numpy

import sys

from helper import *

im = Image.open(r"C:\Users\MashaAllah\Desktop\Image-cryptography\encrypted_images\pic1-encrypt.png")

pix = im.load()

#Obtaining the RGB matrices

r = []

g = []

b = []

for i in range(im.size[0]):

r.append([])

g.append([])

b.append([])

for j in range(im.size[1]):

        rgbPerPixel = pix[i,j]

        r[i].append(rgbPerPixel[0])
```

```
            g[i].append(rgbPerPixel[1])
            b[i].append(rgbPerPixel[2])
m = im.size[0]
n = im.size[1]
Kr = []
Kc = []

print('Enter value of Kr')

for i in range(m):
Kr.append(int(input()))

print('Enter value of Kc')
for i in range(n):
Kc.append(int(input()))
print('Enter value of ITER_MAX')
ITER_MAX = int(input())
for iterations in range(ITER_MAX):
# For each column
for j in range(n):
            for i in range(m):
                        if(j%2==0):
```

```
                r[i][j] = r[i][j] ^ Kr[i]
                g[i][j] = g[i][j] ^ Kr[i]
                b[i][j] = b[i][j] ^ Kr[i]
            else:
                r[i][j] = r[i][j] ^ rotate180(Kr[i])
                g[i][j] = g[i][j] ^ rotate180(Kr[i])
                b[i][j] = b[i][j] ^ rotate180(Kr[i])
    # For each row
    for i in range(m):
        for j in range(n):
            if(i%2==1):
                r[i][j] = r[i][j] ^ Kc[j]
                g[i][j] = g[i][j] ^ Kc[j]
                b[i][j] = b[i][j] ^ Kc[j]
            else:
                r[i][j] = r[i][j] ^ rotate180(Kc[j])
                g[i][j] = g[i][j] ^ rotate180(Kc[j])
                b[i][j] = b[i][j] ^ rotate180(Kc[j])
    # For each column
    for i in range(n):
        rTotalSum = 0
        gTotalSum = 0
```

```
        bTotalSum = 0
        for j in range(m):
                rTotalSum += r[j][i]
                gTotalSum += g[j][i]
                bTotalSum += b[j][i]
        rModulus = rTotalSum % 2
        gModulus = gTotalSum % 2
        bModulus = bTotalSum % 2
        if(rModulus==0):
                downshift(r,i,Kc[i])
        else:
                upshift(r,i,Kc[i])
        if(gModulus==0):
                downshift(g,i,Kc[i])
        else:
                upshift(g,i,Kc[i])
        if(bModulus==0):
                downshift(b,i,Kc[i])
        else:
                upshift(b,i,Kc[i])

# For each row
```

```
for i in range(m):
        rTotalSum = sum(r[i])
        gTotalSum = sum(g[i])
        bTotalSum = sum(b[i])
        rModulus = rTotalSum % 2
        gModulus = gTotalSum % 2
        bModulus = bTotalSum % 2
        if(rModulus==0):
                r[i] = numpy.roll(r[i],-Kr[i])
        else:
                r[i] = numpy.roll(r[i],Kr[i])
        if(gModulus==0):
                g[i] = numpy.roll(g[i],-Kr[i])
        else:
                g[i] = numpy.roll(g[i],Kr[i])
        if(bModulus==0):
                b[i] = numpy.roll(b[i],-Kr[i])
        else:
                b[i] = numpy.roll(b[i],Kr[i])

for i in range(m):
for j in range(n):
```

```
            pix[i,j] = (r[i][j],g[i][j],b[i][j])


im.save(r"C:\Users\MashaAllah\Desktop\Image-cryptography\decrypted_images\pic1.png")
```

Helper.py

```
import
numpy


def
upshift(a,index,n):
col = []     for j in
range(len(a)):
        col.append(a[j][index])
shiftCol = numpy.roll(col,-n)
for i in range(len(a)):
        for j in range(len(a[0])):
if(j==index):
                a[i][j] = shiftCol[i]


def
downshift(a,index,):
col = []     for j in
range(len(a)):
        col.append(a[j][index])
    shiftCol =
numpy.roll(col,n)     for i in
range(len(a)):
        for j in range(len(a[0])):
if(j==index):
                a[i][j] = shiftCol[i]


def rotate180(n):
    bits = "{0:b}".format(n)
return int(bits[::-1], 2)
```

IMPLEMENTATION & RESULT

Sample Execution:

Input Image 1:

('Vector Kr : ', [7, 114, 103, 17, 235, 153, 16, 86, 154, 178, 234, 144, 217, 215, 227, 32,
123, 71, 126, 165, 85, 70, 195, 250, 195, 161, 125, 137, 199, 1, 70, 15, 49, 207, 1, 33, 79,
180, 207, 229, 43, 193, 14, 105, 168, 14, 61, 203, 137, 23, 110, 226, 134, 0, 246, 240, 166,
71, 170, 225, 145, 76, 54, 187, 33, 95, 66, 137, 113, 204, 214, 164, 210, 142, 96, 167, 255,
207, 171, 120,
45, 25, 250, 232, 170, 13, 40, 94, 40, 213, 149, 182, 92, 35, 163, 121, 87, 72, 99, 229,
209, 236,
60, 204, 102, 103, 58, 227, 96, 94, 255, 26, 74, 196, 201, 196, 153, 20, 150, 153, 73, 68,
52, 229, 57, 223, 63, 180, 174, 15, 34, 77, 16, 28, 95, 167, 134, 173, 85, 21, 63, 111, 154,
170, 38, 10, 153, 40, 175, 165, 80, 92, 225, 174, 68, 85, 115, 246, 111, 10, 235, 201, 108,
61, 107, 82, 24, 134, 168, 215, 46, 106, 38, 182, 6, 85, 188, 142, 163, 182, 92, 124, 250,
148, 63, 38, 71, 70, 97, 223, 48, 96, 55, 5, 239, 246, 102, 192, 156, 54, 24, 69, 75, 75, 82,
250, 222, 25, 135, 162, 73, 20, 57, 176, 133, 67, 155, 80, 229, 203, 116, 29, 112, 20, 225,
200, 10, 178, 160, 54, 21, 247, 177, 56, 180, 58, 116, 19, 104, 139, 119, 86, 74, 24, 98,
201, 251, 6, 114, 145, 85, 118, 22, 93, 83, 116, 7, 222, 252, 193, 153, 160, 132, 5, 156,
165, 228, 114, 107, 129, 141, 255, 195, 102, 20, 59, 146, 128, 112, 207, 181, 210, 9, 57,
215, 57, 239, 39, 167, 102, 28, 39, 237, 113, 63, 15, 17, 146, 56])

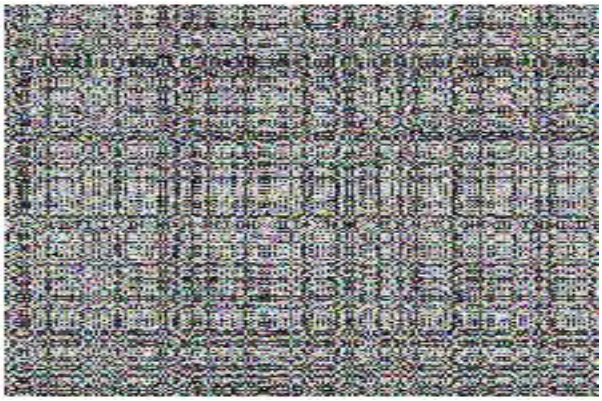
('Vector Kc) : ', [136, 205, 248, 88, 246, 149, 160, 166, 50, 137, 246, 2, 252, 58, 6, 100, 176,
209, 82, 162, 190, 19, 48, 106, 225, 141, 93, 206, 148, 185, 148, 196, 102, 159, 128, 184,
247, 138, 204, 24, 110, 35, 88, 119, 221, 72, 27, 53, 213, 123, 48, 88, 246, 111, 188, 202,
191, 43,

123, 210, 41, 176, 44, 24, 41, 88, 8, 32, 121, 192, 28, 176, 101, 102, 54, 222, 62, 92, 60,
2, 227,

135, 153, 150, 40, 200, 231, 198, 83, 234, 134, 160, 62, 151, 154, 250, 116, 123, 150, 3, 185, 60, 212, 232, 49, 68, 10, 16, 94, 65, 94, 31, 222, 142, 212, 173, 145, 250, 232, 162, 212, 36,
12, 72, 126, 246, 196, 48, 189, 192, 204, 90, 252, 254, 183, 137, 35, 203, 2, 252, 244, 241, 235, 226, 186, 113, 149, 6, 35, 47, 124, 13, 7, 207, 113, 60, 97, 198, 27, 192, 187, 187, 232, 242, 165, 108, 21, 31])

ITER_MAX: 1

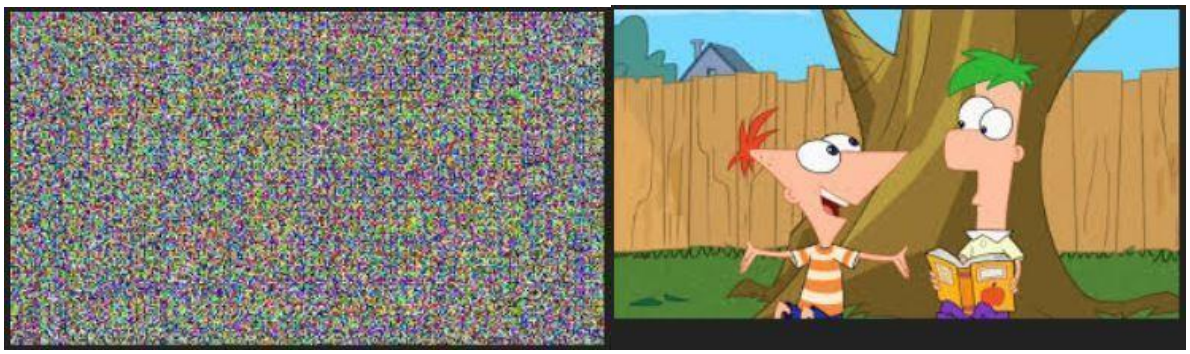
Encrypted image



Decrypted image:

Input Image 2:

Encrypted image: Decrypted image:



Conclusion:

Finally we have done the encryption and decryption using Rubik's cube algorithm, considering a image in the folder, which helps in security. Even this project can be developed in the future making the interface even better. This is all IMAGE CRYPTOGRAPHY BY RUBIKS CUBE project that we have done. So the Rubik's cube algorithm is one of the efficient algorithms that are present right one in the field of Image Cryptography.

References :

[1] Sirisha, M., & Lakshmi, S. V. V. S. (2014). Pixel transformation based on Rubik's cube principle. Int J Appl Innov Eng Manage, 3(5), 273-277.

[2] Amirtharajan, R., Abhiram, M. V., Revathi, G., Reddy, J. B., Thanikaiselvan, V., & Rayappan, J. B. B. (2013). Rubik's cube: A way for random image steganography. Res. J. Inform. Technol, 5, 329-340.

[3] Nagarajan, B., & Manju, B.(2016) Secure And Verifiable Cryptographic Scheme Using Rubik's Cube Principle. 4,198-205.

[4] Abdullatif, A. A., Abdullatif, F. A., & Naji, S. A. (2019). An enhanced hybrid image encryption algorithm using Rubik's cube and dynamic DNA encoding techniques. Periodicals of Engineering and Natural Sciences, 7(4), 1607-1617.

[5] Moayed, M. J. (2009). Voter Verification Using Rubik's Cube (Doctoral dissertation, Universiti Putra Malaysia).Universiti Putra Malaysia,1-25.

[6] Venkata krishna.K.P., Hemanth.v.,& Vishnu.R.,Agilandeswari.L.,Novel secure technique using visual cryptography and advance AES for images. International Journal of Knowledge Management & e-Learning,2016.1-7.

[7] Savitha.B., (2017). 'Image Steganography'.Wright State University,1-17.

[8] Dr Ekta Walia, Payal Jain and Navdeep. "An analysis of LSB & DCT based Steganography." Global Journal of Computer Science andTechnology, April2010.v-10.4-8.

[9] Elena.A.,Antonia.A.,Fabiola.M.,Angel.M., Associative Models for Encrypting Monochromatic Images.1-2.

[10] G.A. Papakostas, D.E. Koulouriotis and E.G. Karakasis (2009). Efficient 2-D DCT Computation from an Image Representation Point of View, Image Processing, Yung-Sheng Chen (Ed.), ISBN: 978-953-307-026-1, InTech,

[11] Parkhi, Disha And Lokhande, S. S. (2012) "Image Compression System Using ANN," International Journal Of Computer And Communication Technology: Vol. 3 : Iss. Article 4. https://Www.Interscience.In/Ijcct/Vol3/Iss1/4

[12] Shanmuganathan, Sankar. (2017). Cte: A Method To Compression-Then-Encryption For Images. International Journal Of Engineering Sciences & Research Technology. 6. 214 .

[13]Rekha,Goudar,Rohit(2018).Secure Secrete Image Carrier Using Rubiks Cube.Volume 120 No. 6 2018, 12111-12122 ISSN: 1314-3395 .

[14] Vyom Chhabra, Tejeswini Sundaram. Binary Encryption Based On A Rubik's Cube. http://Tejeswinisundaram.Github.Io/Assets/Rubik.Pdf

[15] Seetaiah Kilaru, Yojana Kanukuntla, Asma Firdouse, Mohammad Bushra & Sindhu Chava(2013). Effective And Key Sensitive Security Algorithm For An Image Processing Using Robust Rubik Encryption & Decryption Process.

[16] Adrian-Viorel (2015). .Kenken Puzzle – Based Image Encryption Algorithm.

[17] Lini Abraham1 , Neenu Daniel2(2013). An Improved Color Image Encryption Algorithm With Pixel Permutation And Bit Substitution.

[18] H. Sharma, N. Kumar And G. K. Jha, "Enhancement Of Security In Visual Cryptography System Using Cover Image Share Embedded Security Algorithm (CISEA)," *2011 2nd International Conference On Computer And Communication Technology (ICCCT-2011)*, Allahabad, 2011, Pp. 462-467, Doi: 10.1109/ICCCT.2011.6075137.

[19] K.A.Abitha Pradeep,K.Bharathi(2016) Secure Communication Based On Rubik's Cube Algorithm And Chaotic Baker Map. Volume 24, 2016, Pages 782-789.

[20] Krutika Solanki , Vidisha Vankani, Pooja Pukle, Sridhar Iyer(2016). Multimedia Encryption Using Visual Cryptography.