Image Encryption For Secure Transfer of Data

Rusabh K Shah [1], Dr. K Marriyappan [2]
[1]Dept of MTech Cyber Security
[2]Guide
[1, 2] Jain University, Bangalore

Abstract- *At the present time, the protection of multimedia data has become a very important process which can be achieved by encryption. Basically, so many different techniques have been used to protect private image data from those who illegally try to have access. An efficient cryptographic scheme is one that has a space of a large key that resists brute force search time, less execution time complexity/ High speed and should be able to provide high confusion and diffusion for good security. In this paper, we will provide an overview of an image encryption algorithm named Chaos-based encryption techniques. We propose a new and efficient method to develop secure image-encryption techniques with the goal of enhancing accuracy, security and privacy in a computer system. The Chaos based algorithm will be built using the following programming Language and its Libraries; Python V3, PIL and TKinter. It will be trained and tested by a specific group of network security indicators then the output will be assigned to one of the security levels. We will analyze the Chaos-based system which have many properties to achieve high security level, such as sensitivity to change initial conditions and parameters, ergodicity random behavior and unstable periodic orbits with long periods. The chaotic system is rich in significance and in implication because of sensitivity to change initial conditions, control parameters, ergodicity, random-like behaviour, repeated processing and very high diffusion and confusion properties that are desirable for cryptography.*

Keywords- Chaos based Image Encryption, Henon Map,

I. INTRODUCTION

The development of the Space Science and Technology has recently attracted a growing interest from researchers and industrial communities, mainly because of large number of possible applications capable to exploit remotely sensed data and satellite images. Advances in space science, data analysis, and communication technologies present new opportunities for users to increase productivity, reduce costs, facilitate innovation and create virtual collaborative environments for addressing the new challenges. GIS and Remote sensing technologies, along with related geospatial technologies, contribute powerful tools for preserving and protecting the nation’s critical infrastructure.

In such systems, a space borne platform collects scientific data and transmits them to a ground station and at the ground segment, a series of image products are created that can be made available to research or commercial organizations for exploitation. The data delivery and sharing process, usually based on CD/DVD-ROM or on shared network environment (Internet, LAN, WAN etc), provides the user with a digital version of the remote sensing data and images. In the same way as for multimedia contents, the digital format implies an inherent risk of unauthorized copy or use of the product.

Similarly, many digital services, such as Medical, Military, and Space imaging systems require reliable security in storage and transmission of digital images. The rapid progress of Internet in the digital world today, the security of digital images has become more and more important. The prevalence of multimedia technology in our society has promoted digital images to play a more significant role, which demands a serious protection of users‟ privacy. To fulfill such security and privacy needs in various applications, encryption of images is very important to minimize malicious attacks from unauthorized parties.

With the rapid growth of multimedia production systems, electronic publishing and widespread dissemination of digital multimedia data over the Inter-net, protection of digital information against illegal copying and distribution has become extremely important. To meet this challenge, a variety of traditional encryption algorithms have been proposed. Recently, along with the rapid development of theory and application of chaos, many researchers are now focusing on the chaotic cryptography. A lot of image encryption schemes based on chaos theory have been presented. These applications have been motivated by the chaotic properties such as ergodicity and sensitive dependence on initial conditions and system parameters, in addition to complex dynamics and deterministic behaviors.

A systematical method was suggested for adapting an invertible two-dimensional chaotic map on a torus or on a square to create a symmetric block encryption scheme. The main idea is to shuffle the positions of the pixels of the plain-image in the spatial-domain. A new chaotic key-based image encryption algorithm (CKBA) to change the pixel values of the plain-image. An algorithm for encoding binary images using one-dimensional chaotic maps is presented. In order to improve the security of the image encryption algorithm, many researchers prefer shuffling the positions and changing the grey values of image pixels simultaneously. For instance, in the two-dimensional chaotic map is generalized to 3D for designing a real-time secure symmetric encryption scheme. The new scheme employs the 3D map to shuffle the positions of image pixels and uses an-other chaotic map to confuse the relationship between the cipher-image and plain-image.

II. SYSTEM DESIGN

2.1 Proposed system

In this paper, we propose an efficient, selective chaos-based image-encryption and compression algorithm. The encryption and compression are the two main operations which take place. The new algorithm combines two techniques: encryption and compression. In this technique, a wavelet transform was used to decompose the image and decorrelate its pixels into approximation and detail components. The more important component (the approximation component) is encrypted using a chaos-based encryption algorithm. This algorithm produces a cipher of the test image that has good diffusion and confusion properties. The remaining components (the detail components) are compressed using a wavelet transform. A complete specification for the new algorithm is provided.

Encryption time- The time required to convert plaintext to cipher text is encryption time. Encryption time depends upon key size, plaintext block size and mode.

Decryption time- The time to recover plaintext from cipher text is called decryption time. The decryption time is desired to be less similar to encryption time to make system responsive and fast.

Throughput of Encryption: The throughput of the encryption scheme defines the speed of encryption.

Throughput of Decryption: The throughput of the decryption scheme defines the speed of decryption.

Power efficiency is measured in power usage during execution.

CPU time is measured in clock ticks or seconds.

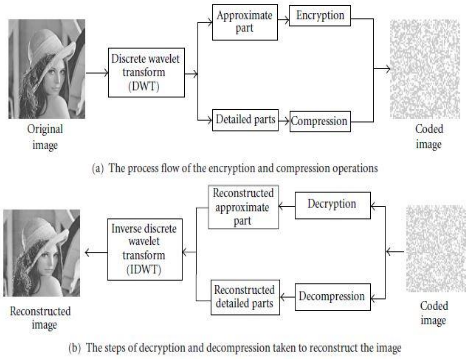
Memory usage is measured in memory used during execution.

2.2 Design Stage

The figure 2.1, explains how when an sample image which is taken to perform Discrete wavelet transform is split into two parts which include the approximate part and the detailed parts where the compression takes place. When both the images combined together the cipher image is obtained. To obtain back the reconstructed image or the original image we perform the same steps in the reverse order.

Wavelet Transform

The first step of a general image-compression technique is the wavelet transform. Distinguish between the visually important information and unimportant information. It also helps to reduce the statistical dependence between coefficients so that the source coding will be more efficient. It changes the scale-like geographical map. Every image will be transformed in each level of decomposition to a one low information image and three details image in horizontal, vertical, and diagonal axis image.


Original image
Discrete wavelet transform (DWT)
Approximate part
Encryption
Detailed parts
Compression
Coded image
Reconstructed approximate part
Decryption
Inverse discrete wavelet transform (IDWT)
Reconstructed detailed parts
Decompression
Reconstructed image
Coded image

(a) The process flow of the encryption and compression operations

(b) The steps of decryption and decompression taken to reconstruct the image

FIGURE 1: The processes of the new algorithm.

Figure 2.1

Types of wavelet transforms, includes Continuous Wavelet Transform (CWT) and the Discrete Wavelet Transform (DWT). The CWT is used for signals that are continuous in time, and the DWT is used when a signal is being sampled, such as during digital signal processing or digital image processing. In this project, discrete wavelet transform will be used.

Discrete Wavelet Transform (DWT) :

Image can be analyzed by passing it through an analysis filter bank, which consists of a low-pass and high-pass filter. When a signal passes, it is split into two bands. The low-pass filter extracts the coarse information of the signal. The high-pass filter extracts the detail information of the signal.

From the figure 2.2, a simple simulation of the algorithm is illustrated. We can see the that the plain image is taken as the input. We choose the encryption algorithm in this case which is Chaos based encryption algorithm. Then comes the part where the image is compressed which is a optional part for this algorithm. If we compress the image then the next step would be decompression. The decryption process takes place using the key to get back the original image. Thus by using this algorithm we are ensuring high level of security for confidential or secret images.

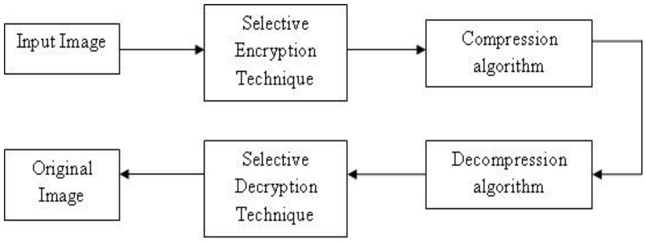


Figure 2.2

III. CHAOS BASED CRYPTOGRAPHY

3.1 Chaos based Cryptosystem

- Plaintext is broken into 8-byte blocks as:
  - P=P1 P2 P3 . .. .. . Pm
- The resulting ciphertext is similarly obtained in the form of 8-byte blocks:
  - C=C1C2C3 . . .. ..Cm
- The 128-bit key is broken into four 32-bit parts:
- K=A1 A2B1B2

3.2 KeyGeneration :

The use of chaos for image encoding yields to three types of keys; these keys may be used together or separately, in order to enhance the privacy.They are the external or (control) parameter α, the initial state $x0$, and the number of iterations.The number of iterations is fixed during all the cycles and equal to the size of the important part of the image.

3.3 Libraries Used :

Python3 - Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below.

PIL - Python Imaging Library is a free library for the Python programming language that adds support for opening, manipulating,.

Tkinter- Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

3.4 Cryptography Design:

The use of chaos for image encoding yields to three types of keys; these keys may be used together or separately, in order to enhance the privacy. They are the external or (control) parameter α, the initial state $x0$, and the number of iterations. The number of iterations is fixed during all the cycles and equal to the size of the important part of the image, which should be encrypted. In this project, one two or three external encryption keys are used, then three types of approaches are performed as following: first using one external encryption key: in this type, the chaotic map will generate a threshold vector of two values: 0 and 1. The pixels of the image vector that are corresponding to the zeros of the threshold vector will be encrypted by one external encryption key and the other pixels will be normalized in such a way to hide the details of the image vector. Second, using two external encryption keys: in order to increase the security, the second type uses two external encryption keys such that the chaotic map will also generate a threshold vector of two values: 0 and 1, and the pixels of the image vector that are corresponding to the zeros of the threshold vector will be encrypted by the first key, and other pixels that are corresponding to the ones will be encrypted by the second key. Third, using three external encryption keys: in order to increase the security further, the third type uses three external encryption keys. In this case the chaotic map will generate a threshold vector of three different values: 0, 0.5, and 1. The pixels of the image vector that are corresponding to the zeros will be encrypted using the first key; the pixels that are corresponding to 0.5 values will be encrypted by the second key, and finally the pixels that are corresponding to the ones will be encrypted by the third key. The encryption process is done for one-level decomposition and two-level decomposition; also it is repeated for two different sizes of external keys.

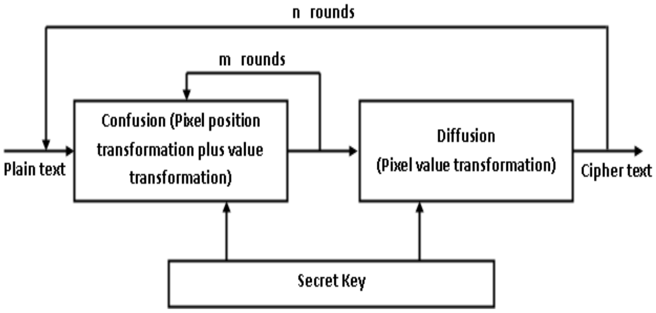

n rounds
m rounds
Plain text
Confusion (Pixel position transformation plus value transformation)
Diffusion (Pixel value transformation)
Cipher text
Secret Key

Figure 3.1

3.5 Combining with Henon Map :

The Henon map , sometimes called Henon-Pomeau map, is a discrete-time dynamical system. It is one of the most studied examples of dynamical systems that exhibit chaotic behavior. The Henon map takes a point (x_n, y_n) in the plane and maps it to a new point.

The map depends on two parameters, *a* and *b*, which for the classical Hénon map have values of *a* = 1.4 and *b* = 0.3. For the classical values the Hénon map is chaotic. For other values of *a* and *b* the map may be chaotic, intermittent, or converge to a periodic orbit. An overview of the type of behavior of the map at different parameter values may be obtained from its orbit diagram.

IV. SYSTEM CONFIGURATION

The basic system requirements for running the python code is given as follows:

Processor : Intel Core i5-6200 CPU @ 2.30 2.40GHz

Installed Memory: 8.00 GB (7.89 GB useable)

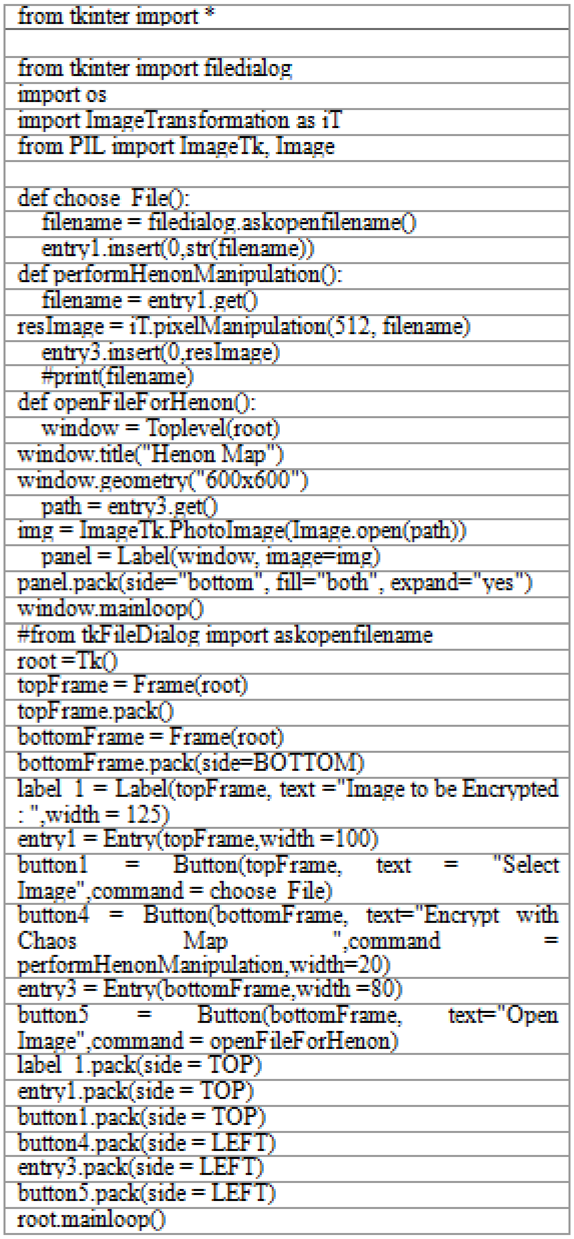
System type : 64-bit Operating System, x64 – based processor

Operating System : Windows 10 Pro

V. CODING AND EXECUTION

5.1 ENCRYPTION

For developing the algorithm we have used python programming language. The libraries which we used is majorly tkinter as it provides a very good graphical user interface. Other libraries used are Python3,PIL and some inbuilt. The code below shows the Encryption part of the algorithm.



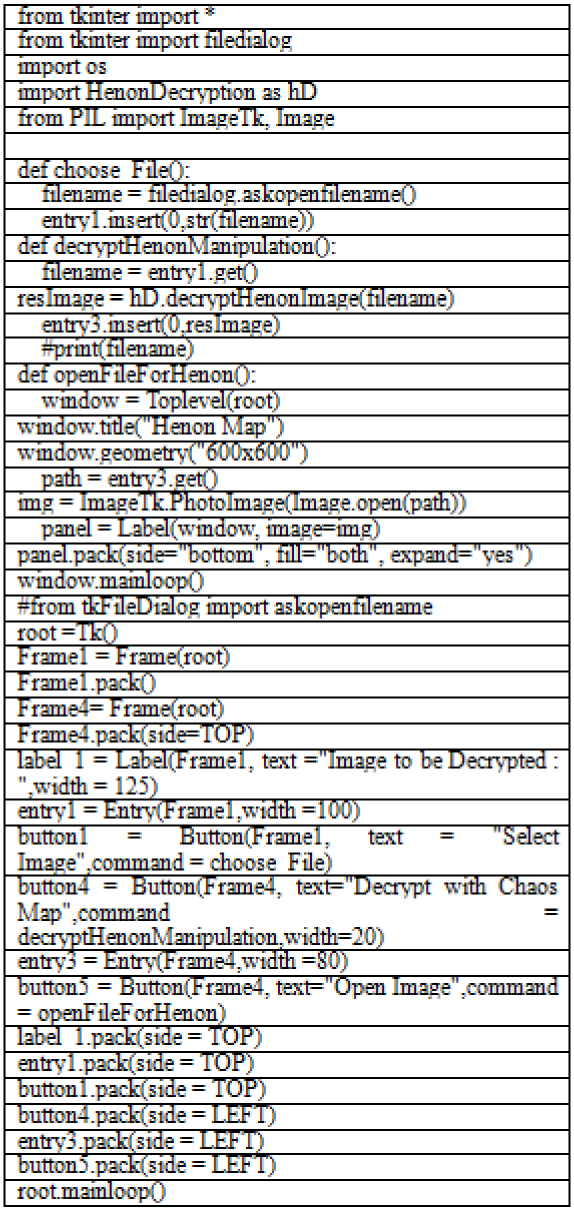
```
from tkinter import *

from tkinter import filedialog
import os
import ImageTransformation as iT
from PIL import ImageTk, Image

def choose  File():
    filename = filedialog.askopenfilename()
    entry1.insert(0,str(filename))
def performHenonManipulation():
    filename = entry1.get()
resImage = iT.pixelManipulation(512, filename)
    entry3.insert(0,resImage)
    #print(filename)
def openFileForHenon():
    window = Toplevel(root)
window.title("Henon Map")
window.geometry("600x600")
    path = entry3.get()
img = ImageTk.PhotoImage(Image.open(path))
    panel = Label(window, image=img)
panel.pack(side="bottom", fill="both", expand="yes")
window.mainloop()
#from tkFileDialog import askopenfilename
root =Tk()
topFrame = Frame(root)
topFrame.pack()
bottomFrame = Frame(root)
bottomFrame.pack(side=BOTTOM)
label  1 = Label(topFrame, text ="Image to be Encrypted : ",width = 125)
entry1 = Entry(topFrame,width =100)
button1 = Button(topFrame, text = "Select Image",command = choose  File)
button4 = Button(bottomFrame, text="Encrypt with Chaos Map ",command = performHenonManipulation,width=20)
entry3 = Entry(bottomFrame,width =80)
button5 = Button(bottomFrame, text="Open Image",command = openFileForHenon)
label  1.pack(side = TOP)
entry1.pack(side = TOP)
button1.pack(side = TOP)
button4.pack(side = LEFT)
entry3.pack(side = LEFT)
button5.pack(side = LEFT)
root.mainloop()
```

5.2 DECRYPTION

The decryption part of the algorithm also follows the same procedure as the encryption part but in a reverse order. In this decryption we use the encrypted image which we obtain after running the encryption part. Hence after running this code we get back the original image.



```
from tkinter import *
from tkinter import filedialog
import os
import HenonDecryption as hD
from PIL import ImageTk, Image

def choose  File():
    filename = filedialog.askopenfilename()
    entry1.insert(0,str(filename))
def decryptHenonManipulation():
    filename = entry1.get()
resImage = hD.decryptHenonImage(filename)
    entry3.insert(0,resImage)
    #print(filename)
def openFileForHenon():
    window = Toplevel(root)
window.title("Henon Map")
window.geometry("600x600")
    path = entry3.get()
img = ImageTk.PhotoImage(Image.open(path))
    panel = Label(window, image=img)
panel.pack(side="bottom", fill="both", expand="yes")
window.mainloop()
#from tkFileDialog import askopenfilename
root =Tk()
Frame1 = Frame(root)
Frame1.pack()
Frame4= Frame(root)
Frame4.pack(side=TOP)
label  1 = Label(Frame1, text ="Image to be Decrypted : ",width = 125)
entry1 = Entry(Frame1,width =100)
button1 = Button(Frame1, text = "Select Image",command = choose  File)
button4 = Button(Frame4, text="Decrypt with Chaos Map",command = decryptHenonManipulation,width=20)
entry3 = Entry(Frame4,width =80)
button5 = Button(Frame4, text="Open Image",command = openFileForHenon)
label  1.pack(side = TOP)
entry1.pack(side = TOP)
button1.pack(side = TOP)
button4.pack(side = LEFT)
entry3.pack(side = LEFT)
button5.pack(side = LEFT)
root.mainloop()
```

VI. FINAL OUTCOME

The first image(6.1) shows the Encryption step of the image. This user interface program helps us to select our own image. Once we click on the Encrypt with Chaos Map Button image gets encrypted and we obtain the cipher image. On clicking the open image button, we are able to view the image.After clicking on the encrypt button, it takes around 20-30 seconds to generate the cipher image.

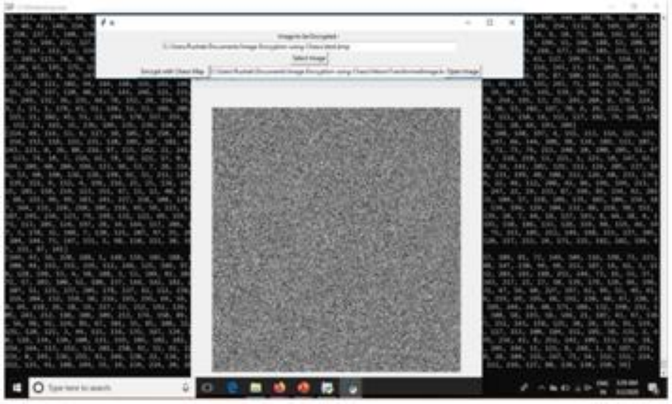


Image 6.1

The second image(6.2) which was performed to get back the original image using the Decryption python code. When we run the decryption part, which is also similar to the encryption. In this we select the cipher image which we had obtaioned in the previous step of encryption.

Click on the Decrypt with Chaos Map button to obtain back the original image. Click on the open image button to view the image. The decryption time taken will be around 20-30 seconds.

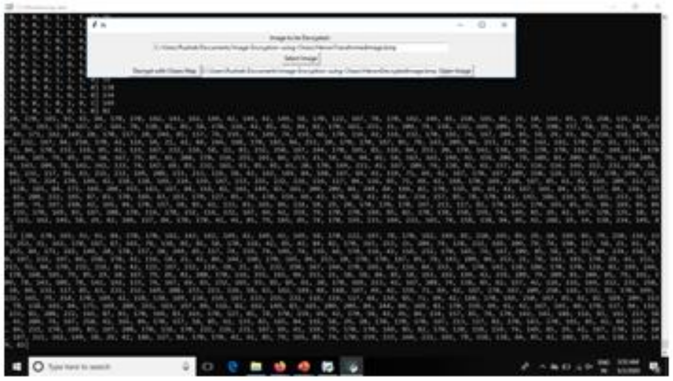


Image 6.2

This is the image(6.3) which we obtain after the decryption. We obtain the same original image that we used as the plain text. While looking at the image we can say the quality of the image is not compromised.

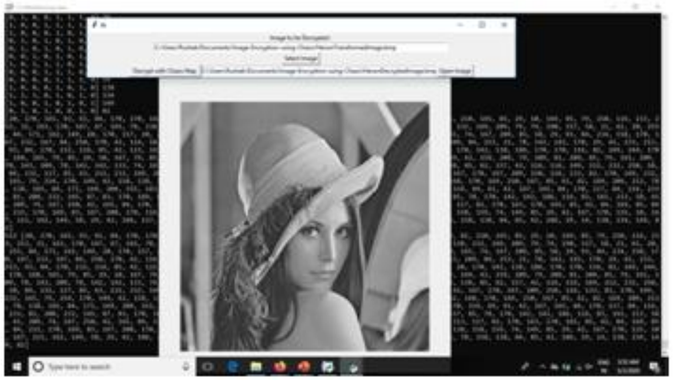


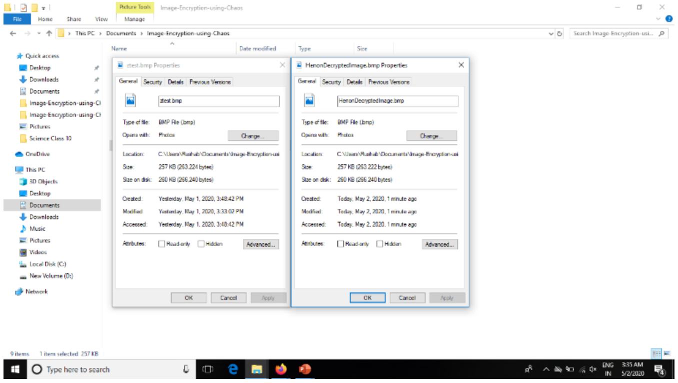
Image 6.3

The final image(6.4) that we analyze is the difference in quality between the original image and the image obtained after the encryption and decryption.

The table below shows a detail comparison between the two images

Size of image before encryption	263,224 bytes
Size of image after dencryption	263,222 bytes

Difference in qualiy of the image = **99.999%**



ztest.bmp Properties

ztest.bmp

Type of file: BMP File (.bmp)

Opens with: Photos

Location: C:\Users\Rushab\Documents\Image-Encryption-usi

Size: 257 KB (263,224 bytes)

Size on disk: 260 KB (266,240 bytes)

Created: Yesterday, May 1, 2020, 3:48:42 PM

Modified: Yesterday, May 1, 2020, 3:33:02 PM

Accessed: Yesterday, May 1, 2020, 3:48:42 PM

Attributes: □ Read-only □ Hidden

HenonDecryptedImage.bmp Properties

HenonDecryptedImage.bmp

Type of file: BMP File (.bmp)

Opens with: Photos

Location: C:\Users\Rushab\Documents\Image-Encryption-usi

Size: 257 KB (263,222 bytes)

Size on disk: 260 KB (266,240 bytes)

Created: Today, May 2, 2020, 1 minute ago

Modified: Today, May 2, 2020, 1 minute ago

Accessed: Today, May 2, 2020, 1 minute ago

Attributes: □ Read-only □ Hidden

Image 6.4

Chaos-based cryptographic algorithm provides the following :

- High security level
- Less computational time and power
- Efficient way to deal with difficult and intractable data
- No comprise in quality of the image.

VII. CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion

An effective image encryption algorithm with two independent chaotic functions allowing parallel computing is presented to enhance the diffusion and confusion functions. For low entropy plain images, which maintain their properties throughout many encryption rounds, a second chaotic function is incorporated to generate random numbers exploited together with exclusive-or operations for perturbing the integrity of such images even in first round. To increase the resistance of the crypto-system to differential attacks, the value of previously encrypted pixel is employed in the encryption of current pixel by exclusive-or and circular rotation operations. The resulting cipher-images indicates that these operations are effective for diffusing an infinitesimal change in single pixel intensity of plain image on many pixels in cipher-image. The complexity analyses shown prove that the proposed scheme requires less operations than the compared algorithms regarding key space, security and encryption speed. A variety of analyses and tests, such as statistical analysis, key-sensitivity and key-space analysis, plain image sensitivity analysis, and speed test have been carried out to present the security and the validity of the proposed algorithm. The proposed algorithm is suitable for parallel computing in two aspects. First, two independent chaotic functions are to be calculated separately without waiting for one another. Secondly, some equations in the algorithm can be converted into a form to be calculated by integer arithmetic.

I have successfully created a home page to encrypt the image, select the image from list and click on encrypt button. It will generate an encryption technique window, in which the output path location is specified. The user can change the output location. The method can be done for decryption process by selecting the cipher image to decrypt. Hence the successfully implementation of the Chaos Based Encryption algorithm is completed with the output results.

7.2 Recommendations / Future-Scope

The AES algorithm presents better security performance but slightly slower in terms of the encryption running speed, this allows us to recommend it for alternative image encryption algorithm. The Chaos-henon map shows exceptionally good confusion and diffusion properties. Due to the computational cost and the simplicity of implementation this map is a good alternative for image encryption in real time communication.

REFERENCES

[1] Zhang W, Wong K-w, Yu H, Zhu Z-l. An image encryption scheme using reverse 2-dimensional chaotic

map and dependent diffusion. Commun Nonlinear Sci Numer Simul2013;18(8):2066–80.

[2] Boriga R, Dascalescu AC, Priescu I. A new hyperchaotic map and its application in an image encryption scheme. Signal Process: Image Commun 2014;29(8):887–901.

[3] Wen C. An image encryption algorithm based on scrambling and chaos. J Inf Comput Sci2013;10(17):5725–33.

[4] Akhshani a, Behnia S, Akhavan a, Hassan HA, Hassan Z. A novel scheme for image encryption based on 2D piecewise chaotic maps. OptCommun2010;283(17):3259–66.

[5] Mohammad Zakir Hossain Sarker and Md. Shafiul Parvez, “A Cost Effective Symmetric Key Crypto-graphic Algorithm for Small Amount of Data”, Proceedings of the 9th IEEE International Multi topic Conference, pp. 1-6, December 2005

[6] Xun Yi Chik How Tan Chee Kheong Slew Rahman Syed, M., "Fast encryption for multimedia," IEEE Transactions on Consumer Electronics, vol. 47, no. 1, pp. 101-107, 2001.

[7] Yen J. C. and Guo J. I., “A new chaotic image encryption algorithm,” Proceeding of National Symposium on Telecommunications, pp. 358-362, December 1998.

[8] Jui-Cheng Yen and J. I. Guo, “A New Chaotic Mirror-Like Image Encryption Algorithm and its VLSI Architecture”, Pattern Recognition and Image Analysis, vol.10, no.2, pp.236-247, 2000.

[9] Jui-Cheng Yen and J. I. Guo, “Efficient Hierarchical Chaotic Image Encryption Algorithm and Its VLSI Realization”. IEEE Proceeding Vis. Image Signal Process, vol. 147, no. 2, pp. 167-175,2000

[10] Sam IS, Devaraj P, Bhuvaneswaran RS. Enhanced substitution-diffusion based image cipher. In: Information and Communication Technologies Communica-tions in Computer and Information Science. Berlin Heidelberg: Springer; 2010. p. 116–23.