NON ADAPTIVE PARTIAL IMAGE ENCRYPTION BASED ON CHAOS

A report submitted for the partial fulfillment in requirement for the
Degree of Bachelor of Science
in
Computer Science
at

Barrackpore Rastraguru Surendranath College

Suvasish Das
Roll NO: 6221103 02829
Registration NO:
1031911100364 of 2019

Samrat Mitra
Roll NO: 6221103 02812
Registration NO:
1031911400357 of 2019

Sayantan Das
Roll NO: 6221103 02810
Registration NO:
1031911600359 of 2019

Under the supervision of
Dr. Sukalyan Som
Department of Computer Science
Barrackpore Rastraguru Surendranath College
affiliated to West Bengal State University, Barasat

July. 2022

Barrackpore Rastraguru Surendranath College

85, Middle Road, 6, River Side Road, 24-Parganas (N), West Bengal 700120 *Affiliated Under West Bengal State University, Barasat & formerly under University of Calcutta *Registered under 2F & 12B of UGC Act with Autonomous Post Graduate Courses

Department of Computer Science

CERTIFICATE

The students **Suvasish Das** (Roll No:622110302829, Registration. No.: 1031911100364 of 2019), **Sayantan Das** (Roll No. 622110302810, Registration No.: 1031911600359 of 2019) **Samrat Mitra** (Roll No. 622110302812 Registration no.: 1031911400357 of 2019) have carried out their Project Work of paper **CMSADSE06P** for the partial fulfillment of requirements for the Degree of **Bachelor of Science in Computer Science** under my supervision. The project has been entitled as "**Non Adaptive Partial Image Encryption Based On Chaos**".

- -

(Sri Phulen Mahato)

HEAD
Department of Computer Science
Barrackpore Rastraguru Surendranath College

- -

(Dr. Sukalyan Som)

SUPERVISOR
Department of Computer Science
Barrackpore Rastraguru Surendranath College

- -

(EXTERNAL EXAMINER)

ACKNOWLEDGEMENT

Working for a dissertation work as a partial requirement towards the curriculum is no doubt a scientific step-ahead that has enormously helped us to gain immense experience to enrich the pathway towards the acquisition of knowledge.

In the preparation of this thesis work and its formation it is undoubtedly inevitable that there are many to whom we owe our sincere gratitude. We are greatly indebted to Dr. Monojot Ray (Principal, Barrackpore Rastraguru Surendranath College), Sri Phulen Mahato (HOD computer science, Barrackpore Rastraguru Surendranath College), Dr. Sukalyan Som (Department of Computer Science, Barrackpore Rastraguru Surendranath College), and other faculty members for their valuable suggestion and constant inspiration during the tenure of the work. We are also thankful to Sri Debasish Sarkar for their constant support.

We would like to convey our sincere thankfulness and cordial respect to Dr. Sukalyan Som, our supervisor for this work.

We are also indebted to all staff of our college for their extended arm of help to us all the time and grateful to our friends for their endless motivation, without which this dissertation work could not have taken its shape.

We are ever grateful to our parents and other family members for their moral support and encouragement without which this project would have never completed.

- - - - - - - - - - - - - - - - - -   - - - - - - - - - - - - - - - - - -   - - - - - - - - - - - - - - - - - -
(Suvasish Das)   (Samrat Mitra)   (Sayantan Das)

CONTENTS

Title **Page**

DEVELOPMENT DETAILS

- **Due Date:** March. 2022
- **Completion Date:** June, 2022
- **Project Summary:**

A chaos based symmetric key partial encryption of 8 bit/pixel grayscale image has been proposed. An original image is first decomposed into its 8 binary bit planes among which the significant ones are non-adaptively determined. The significant bitplanes are encrypted with the key stream generated by a logistic map based pseudo random binary number generator and then combined with the unencrypted ones to form the cipher image. Performance of the technique is verified by tests based on image statistics- Key sensitive and impermeability of cipher image. After encryption, the encrypted image is transmitted through a public local channel (a wireless network). During transmission the data gets distorted and we perform an image restoration process to restore the noisy image at the receiver end.

- **Developers Role:**

Developers Name	Role	Commit History
Samrat Mitra	Bitplane decomposition, Cipher-image generation	● Conversion functions ● Bitplane decomposition function ● Entropy and histogram calculation
Suvasish Das	Cipher image generation, Key sensitivity test, Noise removal	● Cipher image composition ● Noise profile adding and reduction model ● Analysis and plotting the model
Sayantan Das	Documentation, Library implementation	● Implement test images from image database ● Finding suitable OpenCV functions ● Documenting implemented functions

- **Development Environment:**

To develop this project requires Python 3.X, OpenCV 4.5, Jupyter Notebook 4.9, Numpy 1.22.3 and a python virtual environment.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MSB	Most Significant Bit
LSB	Least Significant Bit
RSA	Rivest Shamir Adleman
DES	Data Encryption Standard
CCCBG	Cross-Coupled Chaotic random Bit Generator

Chapter 1
Introduction

The field of encryption is becoming very important in the present era in which information security is of utmost concern. One way to achieve the goal of securing data to transmit is to convert the intelligible data into unintelligible form prior to transmission. Cryptography is the art and science of achieving security by converting sensitive information to an un-interpretable form such that it cannot be interpreted by anyone except the intended recipient. Security is an important issue in communication and storage of images, and encryption is one of the ways to ensure security. Image encryption has applications in internet communication, multimedia systems, medical imaging, telemedicine, military communication, etc. Information security is becoming more important in data storage and transmission. Images are widely used in several processes. Therefore, the protection of image data from unauthorized access is important. Chaos in popular language refers to any situation which is unpredictable and disorderly or "**A state of utter confusion**". Chaos theory describes the behavior of certain non-linear dynamic systems that under specific conditions exhibit dynamics that are sensitive to initial conditions. Encryption of images is different from that of texts due to some intrinsic features of images such as bulk data capacity, high correlation among pixels and high redundancy, which are generally difficult to handle by traditional methods. Due to the light relationship between chaos theory and cryptography, these properties make the chaotic systems a worthy choice. Thus, it is effective to use chaotic maps for cryptosystems. Chaotic systems may be used to generate pseudo-random numbers. They have a large key space, high sensitivity to key variation etc.

1.1 Motivation

Most of the chaos based encryption techniques are directly implemented by overlaying a chaotic sequence generated by a single chaotic map and the pixel value from the image. Only using a single chaotic map to encrypt images may result in lower security and smaller key space.

Most traditional or modern cryptosystems have been designed to protect textual data. An original important and confidential plaintext is converted into cipher text that is apparently random nonsense. Once the cipher text has been produced, it is saved in storage or transmitted over the network. Upon reception, the cipher text can be transformed back into the original plaintext by using a decryption algorithm. However, images are different from text. Although we may use the traditional cryptosystems (such as RSA and DES-like cryptosystems) to encrypt images directly, It is not a good idea for two reasons. One is that the image size is much greater than that of text, so the traditional cryptosystems need much time to directly encrypt the image data. The other problem is that the decrypted text must be equal to the original text.

1.2 Objective

A chaos based **non adaptive partial image encryption** scheme has been proposed [1]. The plain gray scale image is first decomposed into its corresponding biplanes followed by the significant biplanes are identified with the help of autocorrelation function. Then the significant biplanes are encrypted with the key generated by the **Cross-Coupled Chaotic Random Bit Generator** (CCCBG) [2]. The image is transmitted through a local public channel. During transmission the network has some asymptotic disturbance, due to which the transmitted data gets distorted with noise in random density. At the receiver end the encrypted image is obtained by combining both the significant encrypted biplanes and the insignificant unencrypted biplanes. And finally, the extra amount of unwanted noise is reduced using some degradation noise filters.

1.3 Organization of the report

The report is organized in six coherent chapters. In **Chapter 2** the fundamental theories of information security and encryption methodologies have been discussed. **Chapter 3** presents the overview of a non adaptive partial image encryption scheme. In **Chapter 4** the experimental results and their analysis has been given. And **Chapter 5** is dedicated towards the concluding remarks.

Chapter 2

Fundamentals of Information Security

The foundation for security is assets that need to be protected. An asset may be people, things created by people or parts of nature. In the area of information security, the assets are often labeled as information assets, and enclose not only the information itself but also resources that are in use to facilitate the management of information. IT artifacts in the shape of e.g. personal computers, networks, operative systems and applications constitute thus one of several types of supporting resources for managing information. It is not only IT artifacts to be counted as resources when managing information. Information may be managed manually, which makes humans an important resource. People are also indirectly an important resource because that is always people that handle tools that manage information.

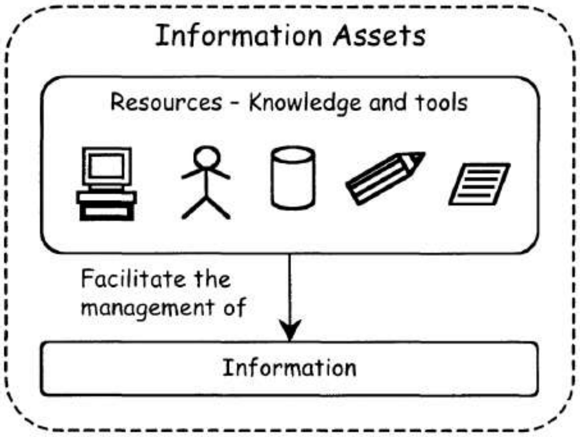


Figure 2.1: Information assets

2.1 Vulnerability

Vulnerability is absence of security mechanisms, or weaknesses in existing security mechanisms. Vulnerability may exist in all of the categories of security mechanisms, and may be known or unknown. The common mistake when using cryptography is the use of algorithms that are known to be weak or broken. Over the years, many algorithms have been declared broken, either due to vulnerability to brute-force attacks (like DES or MD5) or flaws in the protocol itself (like those failed AES candidates). This mistake is most common with hash algorithms, since many of the best-known and most commonly used encryption algorithms have been around for years and are still secure (like AES). Hash algorithms are often used in long-lived applications as well, which can make them difficult to change.

2.2 Threats against Information Assets

Most attacks can be categorized as one of six broad classes:

1. Malware: This is a generic term for software that has a malicious purpose. It includes virus attacks, worms, adware, Trojan horses, and spyware. This is the most prevalent danger to your system.
2. Security breaches: This group of attacks includes any attempt to gain unauthorized access to your system. This includes cracking passwords, elevating privileges, breaking into a server… all the things you probably associate with the term hacking.
3. Denial of service (DoS) attacks: These are designed to prevent legitimate access to your system. Web attacks: This is any attack that attempts to breach your website. Two of the most common such attacks are SQL injection and cross-site scripting.
4. Session hijacking: These attacks are rather advanced, and involve an attacker attempting to take over a session.
5. DNS poisoning: This type of attack seeks to compromise a DNS server so that users can be redirected to malicious websites, including phishing websites.

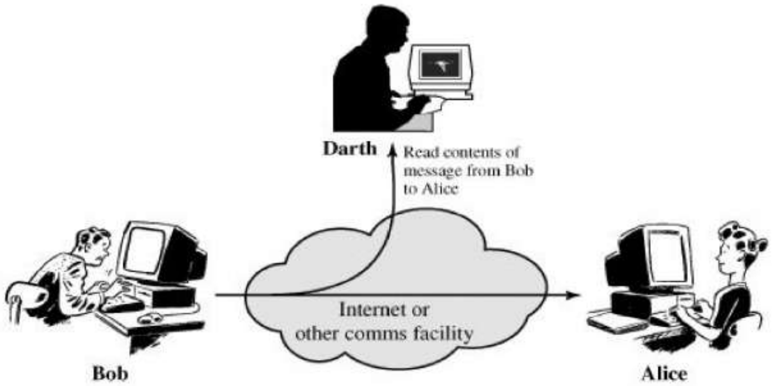

Darth
Read contents of message from Bob to Alice
Internet or other comms facility
Bob
Alice

Figure 2.2: Attacker read the content of the message

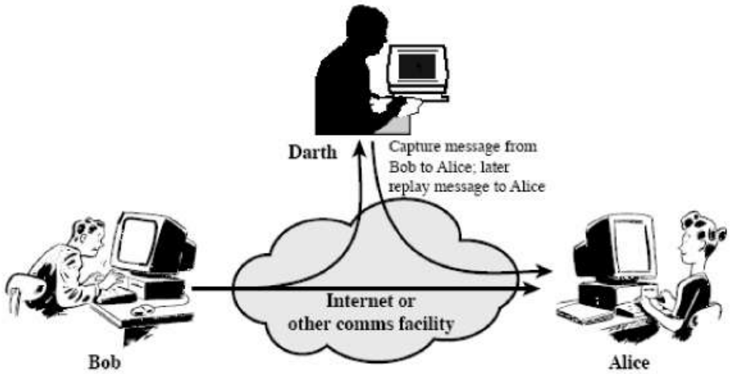

Darth
Capture message from Bob to Alice; later replay message to Alice
Internet or other comms facility
Bob
Alice

Figure 2.3: Attacker modify the content of the message

2.3 Types of cryptosystems

The aim of cryptography is not to hide the existence of a message, but rather to hide its meaning—the process known as encryption. To make a message unintelligible, it is scrambled according to a particular algorithm, which is agreed upon beforehand between the sender and the intended recipient. Thus, the recipient can reverse the scrambling protocol and make the message comprehensible (Singh, 2001a). This reversal of the scrambling is referred to as decryption. The advantage of using encryption/decryption is that, without knowing the scrambling protocol, the message is difficult to re-create. There are two basic types of cryptography: symmetric and asymmetric. Symmetric means the same key is used to encrypt the message and to decrypt the message. With asymmetric cryptography, a different key is used to encrypt the message than is used to decrypt the message. That may sound a bit odd, and some readers may be pondering how that is possible. Later in this chapter, we will explore exactly how that works. For now the important point is to understand the basic concept of symmetric and asymmetric cryptography. But first, let's take a brief look at the history of encryption.

2.3.1 Single-Key (Symmetric) Encryption:

Basically, single-key encryption means that the same key is used to both encrypt and decrypt a message. This is also referred to as symmetric key encryption. There are two types of symmetric algorithms: stream and block. A block cipher divides the data into blocks (often 64-bit blocks, but newer algorithms sometimes use 128-bit blocks) and encrypts the data one block at a time. Stream ciphers encrypt the data as a stream of bits, one bit at a time.

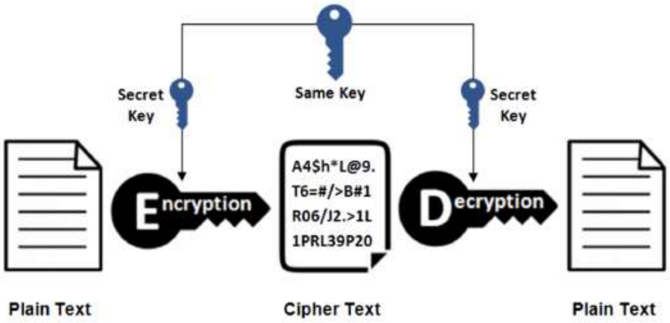


Figure 2.4: Single key encryption

There are two types of symmetric encryption algorithms:

1. Block algorithms. Set lengths of bits are encrypted in blocks of electronic data with the use of a specific secret key. As the data is being encrypted, the system holds the data in its memory as it waits for complete blocks.

2. Stream algorithms. Data is encrypted as it streams instead of being retained in the system's memory.

Some example of symmetric encryption are -
AES (Advanced Encryption Standard), DES (Data Encryption Standard), IDEA (International Data Encryption Algorithm), Blowfish (Drop-in replacement for DES or IDEA), RC4 (Rivest Cipher 4), RC5 (Rivest Cipher 5), RC6 (Rivest Cipher 6)

2.3.2 Public Key (Asymmetric) Encryption:

Public key encryption is essentially the opposite of single-key encryption. With any public key encryption algorithm, one key is used to encrypt a message (called the public key) and another is used to decrypt the message (called the private key). You can freely distribute your public key so that anyone can encrypt a message to send to you, but only you have the private key and only you can decrypt the message. The actual mathematics behind the creation and application of the keys will vary between different asymmetric algorithms. It should be pointed out, however, that many public key algorithms are dependent, to some extent, on large prime numbers, factoring, and number theory.

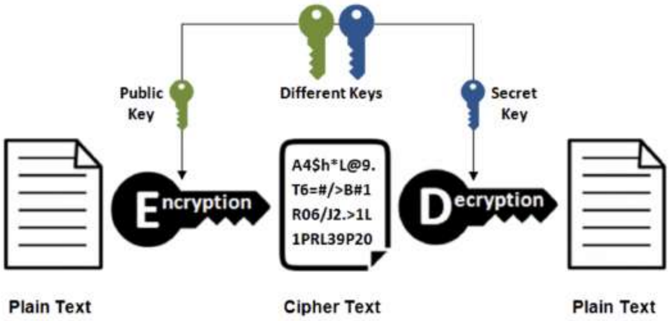


Figure 2.5: Public key encryption

The most well-known example of Asymmetric Encryption is the Digital Signature Algorithm (DSA). Developed by National Institute of Standards and Technology (NIST) in 1991, DSA is used for digital signature and its verification, and based on modular exponentiation and discrete logarithm.

2.4 Types of attacks on cryptosystems

Modern cryptography has become highly complex, and because encryption is used to keep data secure, cryptographic systems are an attractive target for attackers. What is considered strong encryption today will likely not be sufficient a few years from now due to advances in CPU technologies and new attack techniques.

Common types of cryptographic attacks include the following:

1. Brute force attacks attempt every possible combination for a key or password. Increasing key length boosts the time to perform a brute force attack because the number of potential keys rises. In a replay attack, the malicious individual intercepts an encrypted message between two parties (such as a request for authentication) and later "replays" the captured message to open a new session. Incorporating a time stamp and expiration period into each message can help eliminate this type of attack.

2. In a man-in-the-middle (MitM) attack, a malicious individual sits between two communicating parties and intercepts communications (including the setup of the cryptographic session). The attacker responds to the originator's initialization requests, sets up a secure session with the originator and then establishes a second secure session with the intended recipient using a different key and posing as the originator. The attacker has access to all traffic passing between the two parties.

3. An implementation attack takes advantage of vulnerabilities in the implementation of a cryptosystem to exploit the software code, not just errors and flaws but the logic implementation to work the encryption system.

4. A statistical attack exploits statistical weaknesses in a cryptosystem, such as floating-point errors. Another weakness that might lead to a statistical attack is the inability to produce truly random numbers. (Because software-based random number generators have a limited capacity, attackers could potentially predict encryption keys). Statistical attacks are aimed at finding vulnerabilities in the hardware or operating system hosting the cryptography application.

5. A ciphertext-only attack is one of the most difficult types of cyber-attack to perpetrate because the attacker has very little information to begin with. For example, the attacker might start with some unintelligible data that he or she suspects may be an important encrypted message but then gather several pieces of ciphertext that can help him or her find trends or statistical data that would aid in an attack.

6. In a known plaintext attack, an attacker who has a copy of both the encrypted message and the plaintext message used to generate the ciphertext may be able to break weaker codes. This type of attack is aimed at finding the link – the cryptographic key that was used to encrypt the message. Once the key is found, the attacker can then decrypt all messages that are encrypted using that key.

Chapter 3

Non-Adaptive Partial Image Encryption Based On Chaos

Cryptography deals with protecting the privacy of information during communication. Visual information through images plays an important role in all aspects of our lives. In recent days, the increasing use of computers leads to an increasing tendency to security and image fidelity verification. Transmitted images may have different applications, such as commercial, military and medical applications. So it is necessary to encrypt image data before transmission over the network to preserve its security and prevent unauthorized access. However, compared to text, much more processing power and bandwidth for processing is required. In recent years a number of different image encryption schemes has been proposed in order to overcome image encryption problems.

The chaos-based encryption has suggested a new and efficient way to deal with the intractable problem of fast and highly secure image encryption. Chaos is one of the possible behaviors associated with evolution of a nonlinear dynamic system and occurs for specific values of system parameters. The chaotic behavior of a nonlinear system apparently looks random. The chaotic state can be observed by the existence of a chaotic attractor in which all the system trajectories evolve following a certain pattern but are never the same. Existing techniques consume enough computational time for encryption due to its bulk size. Since an image can be considered as a combination of correlated and uncorrelated data wherein most of the information found to be present in the correlated part rather than the uncorrelated part, thus it would be sufficient to encrypt the correlated bit planes instead of encrypting the entire image in order to speed up the entire process and save time too.

In partial encryption techniques usually the significant information has to be encrypted leaving insignificant information unencrypted. In this project work we have tried to implement a simple, fast and secure algorithm for partial image encryption using the characteristics of chaotic functions. Finally, this algorithm is very sensitive to small changes in key so even with the knowledge of the key approximate values; there is the least possibility for the attacker to break the cipher.

3.1 Proposed Scheme

A chaos based symmetric key partial encryption of 8 bit/pixel grayscale image has been proposed. An original image is first decomposed into its 8 binary bit planes among which the significant ones are non-adaptively determined. The significant bitplanes are encrypted with the key stream generated by a '***Cross-Coupled Chaotic random Bit Generator***' and then combined with the unencrypted ones to form the cipher image. Performance of the technique is verified by tests based on image statistics- Key sensitive and impermeability of cipher image.

In the proposed scheme, we consider 8-bitplane images as two -dimensional matrices $I_{original} = [a]_{ij}$ with any $[a]_{ij}$ represents that the 8-bitplanes gray intensity value of the (i,j)th pixel and broadly classify the images under investigation into three categories:

1. Images where only a single bitplane contains the entire information.
2. Images where all the bitplanes contain significant information.
3. Images where some bitplanes are significant and some others are not.

Our interest mainly lies with third categories as encrypting the other categories is easier. Once we classify an image into the third category, a threshold is defined through which we evaluate the importance of the individual bitplanes before encrypting them. The important bitplanes, thus found, are then encrypted using '*chaos based PN Sequences*' started above, while leaving the other unencrypted.

The entire method of encryption is illustrated in *figure 3.1*.

In Figure 3.1 the entire method of illustration using flow charts. In section 3.1.1 the bit decomposition of the image is illustrated. The determination of significance and insignificance bitplane is done in section . In section 3.1.2 the chaotic system is described and in section 3.1.3 the method of encryption and decryption is illustrated.

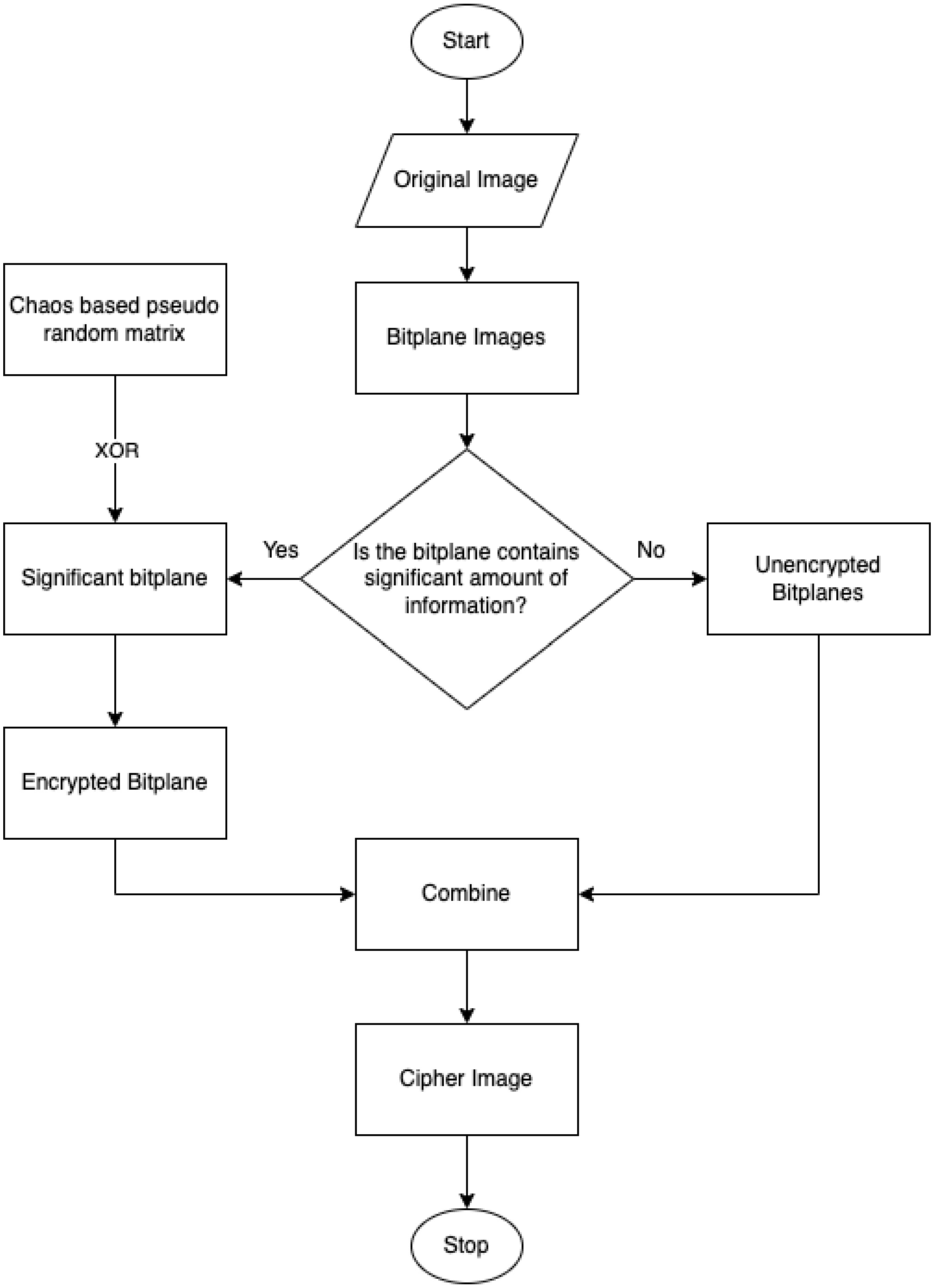


Figure 3.1: Flowchart representation of Method of Encryption

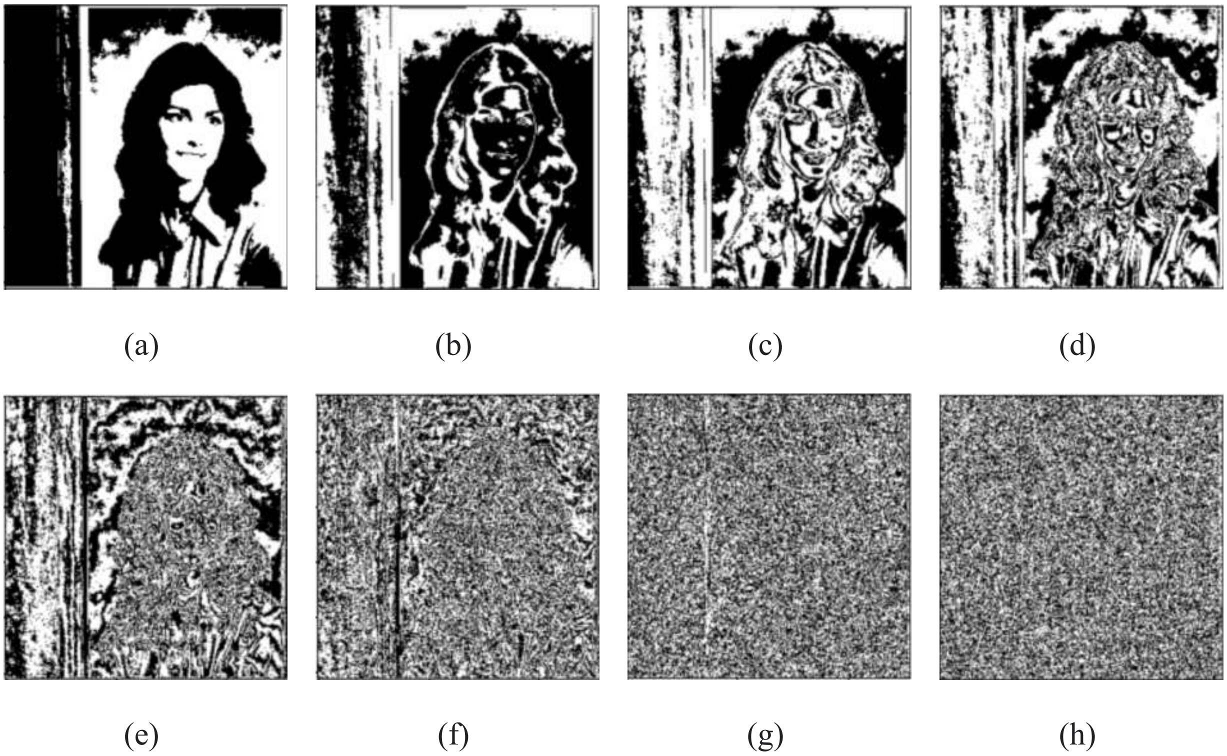
3.1.1 Bitplane Decomposition

In a gray-level image, the pixel intensity is quantized into an integer number of levels ranging from 0 to 255. The value of the pixel at coordinate (x, y) is denoted as f(x, y).Each pixel can be decomposed into an 8 bit binary value, given by :

$$\mathbf{f(x, y) = P(8)\ P(7)\ P(6)\ P(5)\ P(4)\ P(3)\ P(2)\ P(1)} \tag{3.2}$$

So, the input image can be divided into 8 binary images according to the bit locations within a pixel. In Fig 3.2 Original grayscale image "*Lena*" of size 512 × 512 and the binary images obtained by collecting the bits of all the plain-image pixels has been presented. Figure 3.3 shows the original image with the binary images obtained by collecting the ith bits of all the plain image pixels.

Figure 3.3: Original image of Lena of size 512 x 512



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.4: *Bitplane slicing of original image of lena. Corresponding bitplanes are- (a) Bitplane8- MSB, (b) Bitplane7, (c) Bitplane6, (d)Bitplane5, (e) Bitplane4, (f) Bitplane3, (g) Bitplane2, (d)Bitplane1- LSB*

3.1.2 Significant Bit-plane Determination

A bit can carry different amount of information depending on its position in an 8 bit binary number i.e. if there exists a 1 at the 8th position (MSB) of a 8 bit binary number then its contribution towards the formation of corresponding decimal number is 127 where as it contributes only 1 if it is present at the 1st position (LSB). Percentage of contribution of different bit positions in the formation of a pixel of intensity 255 is shown in Table I that can be derived by the formula as stated in figure 3.4

$$p(i) = \frac{2^{i}}{\sum_{i=0}^{7}(2^{i})} \tag{3.5}$$

Table 3.1: *Percentage of contribution in the formation of the pixel intensity*

Bitplanes	Percentage of contribution in the formation of a gray level intensity of 255	Significant/Insignificant
1	0.392156	Insignificant
2	0.784313	Insignificant
3	1.568627	Insignificant
4	3.137254	Significant
5	6.274509	Significant
6	12.549019	Significant
7	25.098039	Significant
8	50.196083	Significant

To determine whether the bitplane is significant or not we frame the null hypothesis as $\mathcal{H}_0$:ith bitplanes significant against the alternative hypothesis $\mathcal{H}_1$:ith bitplane is not significant. Considering the level $\alpha = 0.05$ (i.e 5% contribution of the level of significance from table 3.1 we consider the first 5 bitplanes significant in terms of their percentage contribution.

3.1.3 Chaotic Systems considered in this system

Chaotic systems have a number of interesting properties such as sensitivity on initial condition and system param- eter, ergodicity and mixing (stretching and folding) prop- erties, etc. These properties make the chaotic systems a worthy choice for constructing the cryptosystems (block ciphers as well as stream ciphers) as sensitivity to the ini- tial condition/system parameter and mixing properties respectively, are analogous to the confusion and diffusion properties of a good cryptosystem. A general way to design a chaotic stream cipher is to generate a random bit stream using a chaotic system.

In the proposed random bit generator, two cross coupled piecewise linear chaotic maps are employed (unlike to the pseudo random bit generator proposed, where also two piecewise linear chaotic maps are employed but they are not coupled with each other) to generate random sequences and the set up is abbreviated as CCCBG (Cross-Coupled Chaotic random Bit Generator). In the CCCBG, random bit streams are generated by comparing the two orbits generated by cross coupled piecewise linear chaotic maps; therefore it is difficult for an eavesdropper to extract information about both chaotic systems. The rest of the paper is organized as follows: In the Section, we discuss the dynamics of the skew tent map in brief and the construction of the proposed CCCBG is presented in this Section.

3.1.3.1 Dynamics of Skew Tent Map

The skew tent map is ergodic and has uniform invariant density function in its definition interval. It is the simplest kind of one-dimensional chaotic map which is defined as:

$$x_{i+1} = F(\alpha, x_i) = \begin{cases} \dfrac{x_i}{\alpha} & x_i = [0, \alpha) \\ \dfrac{1 - x_i}{1 - \alpha} & x_i = (\alpha, 1] \end{cases} \tag{3.6}$$

where α and x_i are system parameters and initial condition of the map respectively. It is a non-invertible transformation of unit interval onto itself and contains only one system parameter α, which determines position of the top of the tent in the interval [0,1]. A sequence computed by iterating F(α, x), is expansionary everywhere in the interval [0,1] and distributed uniformly in it.

Orbits for system parameter values 0.4 are shown in Figure 3.7 and in Figure 3.8, we have depicted the chaotic solutions of the equation in figure 3.5, which show sensitivity on initial condition as well as on system parameters.

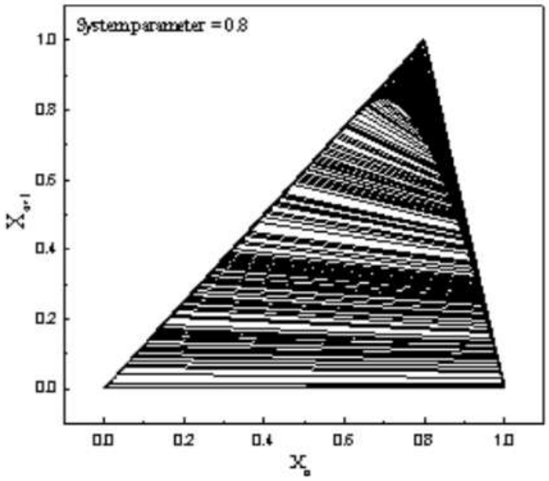


Figure 3.7: Shows the orbits of the skew tent map for system parameter value 0.4

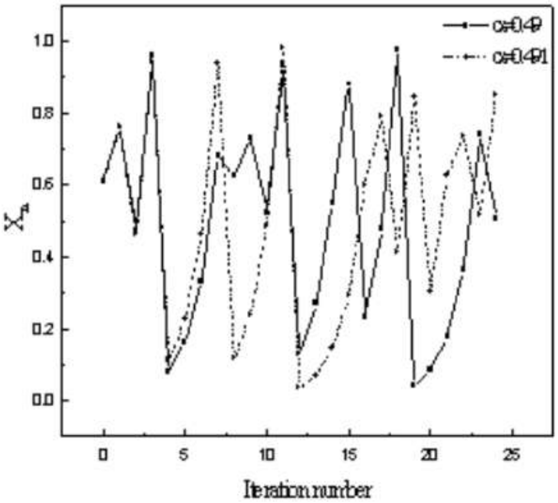


Figure 3.8: Shows the sensitivity of chaotic solution of skew tent map

3.1.3.2 Cross-coupled Chaotic Tent Map Based Bit Generator

In this Section, we discuss the arrangement of chaotic systems in CCCBG. In the proposed CCCBG, we choose two skew tent maps which are piecewise linear chaotic maps and cross-coupled. The output generated by the first tent map is fed to the second tent map as the input (initial condition) and vice versa. The system parameter for the both chaotic maps is kept the same and is in the chaotic regime. If $f_1(x_0,\alpha)$ and $f_2(y_0,\alpha)$ are two piecewise linear chaotic maps and are given as:

$$x_{i+1} = f_1(\alpha, x_i) \tag{3.9}$$

$$y_{i+1} = f_2(a, y_i) \tag{3.10}$$

where α is the system parameter and is the same for both chaotic tent maps, x_i and y_i are the initial conditions and x_i+1 and y_i+1 are their new corresponding states. The CCCBG produces the binary sequences by comparing the outputs of the cross coupled piecewise linear chaotic maps in the following way:

$$g(x_{i+1}, y_{i+1}) = \begin{cases} 0 & \text{if} \quad x_{i+1} < y_{i+1} \\ 1 & \text{otherwise} \end{cases} \tag{3.11}$$

If the binary sequences generated by the CCCBG are random and have no pattern in them, we can use them for the development of new chaotic stream ciphers. In the next section, we discuss basic statistical tests as well as NIST suite tests for testing the randomness and uniformity of the binary sequences generated by CCCBG.

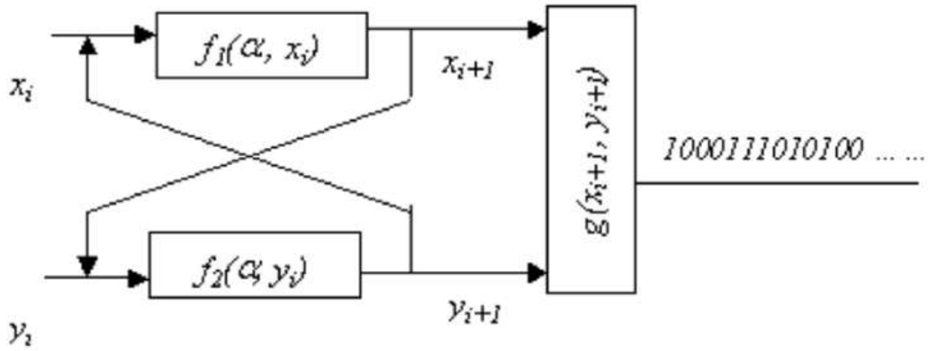


Figure 3.12: *The block diagram of Cross-Coupled Chaotic random bit Generator*

3.2 Method of Encryption and Decryption

3.2.1 Method of Encryption

- Step 1: Consider the plain image to be loriginal (x,y) of size M × N where x = 0,1,2,......M - 1, And y=0,1,2,....,N-1.
- Step 2: Each pixel value P(x,y) in $l_{original}$ (x,y) is decomposed into its corresponding 8 bit binary equivalent and thus 8- planes BP(x,y) ∀ i = 1,2,...,8 are formed.
- Step 3: Significant bit planes are determined by the level α critical region from the H_0:ith bit-planes significant against the alternative hypothesis H_1:ith bit-planes not significant where the test statistics is the percentage contribution of a bit plane in formation of a pixel.
- Step 4: Keys for diffusing the significant bit planes are generated using an ID logistic map based on CCCBG with chosen values of the triplet(x_0,$y_{0,}$µ). The final iterated values of (x_{n+1},y_{n+1}), for the highest significant bit plane, become the initial values (x_0,y_0) for generating the next key and so on.
- Step 5: The significant bit planes, determined by α% (0.05) level of significance, are ciphered as CBP_j= $BP_j \oplus K_j$ ∀ j= 1,...,4.

- Step 6: The cipher bit planes CBP_j and the encrypted bit plane BP_j are combined together to form cipher image as $C_i(x,y) = CBP_j + BP_k$ ∀ i = 1,2...,8, j = 1,..4 and k = 5,...8 where + is used for combining process.

3.2.2 Method of Decryption

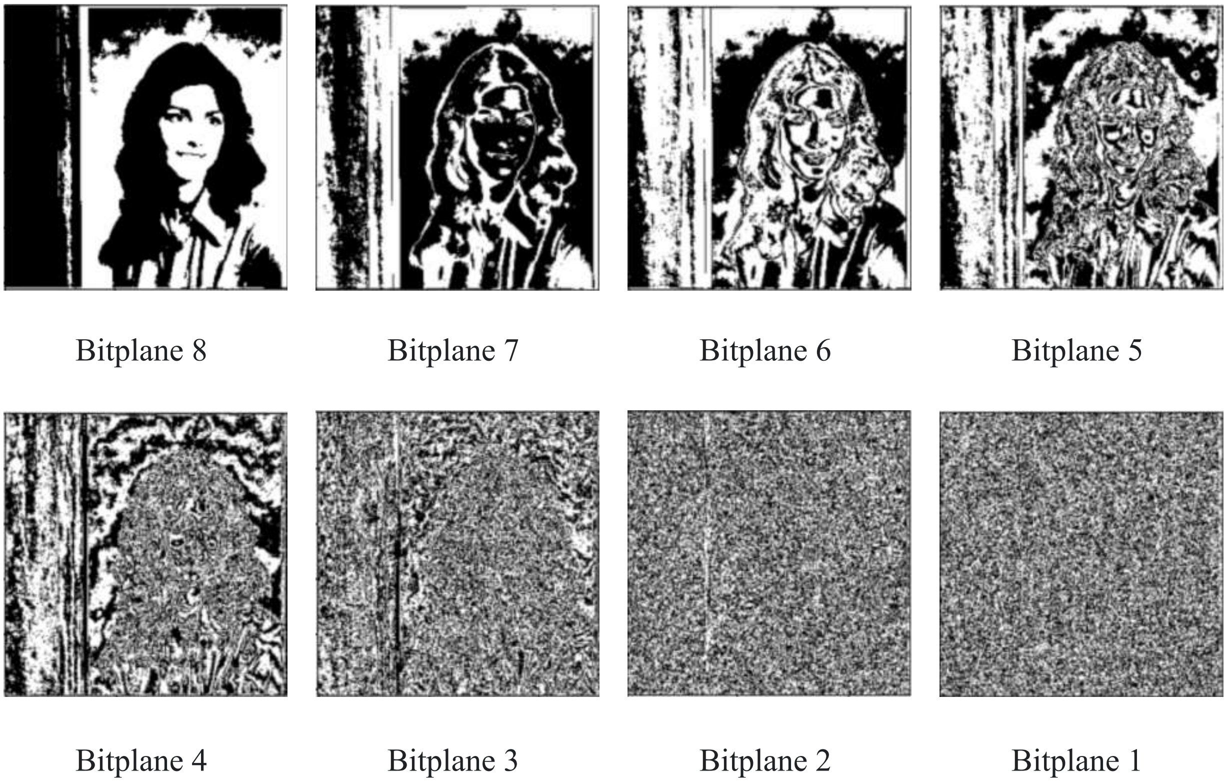
- Step 1: Consider the cipher image to be I_{cipher} (x, y) of size M × N where x = 0,1,2,.....M - 1 And y = 0,1,2,... N-1.
- Step 2: Each pixel value $P_i(x,y)$ in I_{cipher} (x, y) is decomposed into its corresponding 8 bit binary equivalent and thus 8 bit- plane $BP_i(x, y)$ ∀ i = 1,2,...,8 are formed.
- Step 3: Significant bit planes are determined by the level α critical region from theH_o:i^{th} bit-planes significant against the alternative hypothesisH_1: ith bit-planes not significant where the test statistics is the percentage contribution of a bit plane in formation of a pixel.
- Step 4: Upon receiving the triplet (xo, Yo, H) keys for diffusing the significant bitplanes are generated using ID logistic map basedPRNG. The final iterated values of (xn+1, Yn+1) for the highest significant bitplane become the initial values (xo, yo) for generating the next key and so on.
- Step 5: The significant bit planes, determined by a% (0.05) level of significance, are ciphered as CBP; = BPOK ,Vj = 1,...,4.
- Step 6: The cipher bit planes CBP,and the encrypted bit plane BPjare combined together to form cipher image as Ci(x,y) = CBP; + BPVi = 1,2...,8, j = 1,..4 and k = 5,...,8 where + is used for combining process.

3.3 An illustration

Step 1: Let us consider the plain image to be original (x,y) of size M ✕ N where x = 0,1,2,......M - 1, And y=0,1,2,....,N-1.

Figure 3.13: *Original image of lena of size 512 ✕ 512*

Step 2*:* Each pixel value P(x,y) in $l_{original}$ (x,y) is decomposed into its corresponding 8 bit binary equivalent and thus 8- planes BP(x,y) $\forall$ i = 1,2,...,8 are formed.



Bitplane 8

Bitplane 7

Bitplane 6

Bitplane 5

Bitplane 4

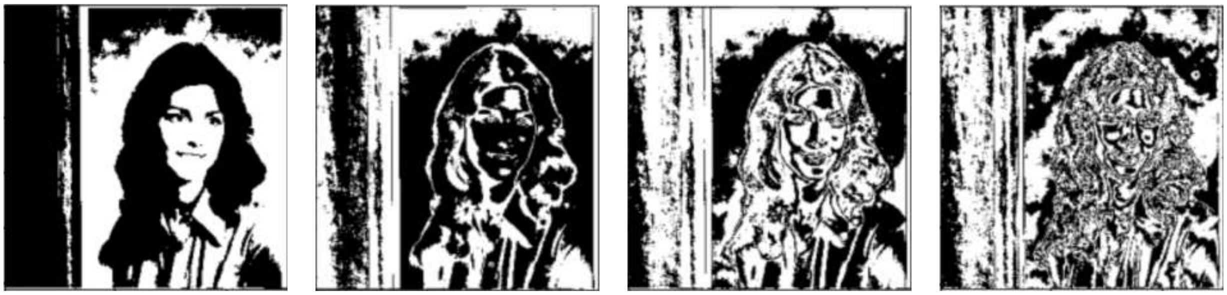
Bitplane 3

Bitplane 2

Bitplane 1

Figure 3.14: *8-bitplane binary image decomposition from grayscale image*

Step 3*:* Significant bit planes are determined by the level α critical region from the H_0:i^{th} bit-planes significant against the alternative hypothesis H_1:i^{th} bit-planes not significant where the test statistics is the percentage contribution of a bit plane in formation of a pixel.

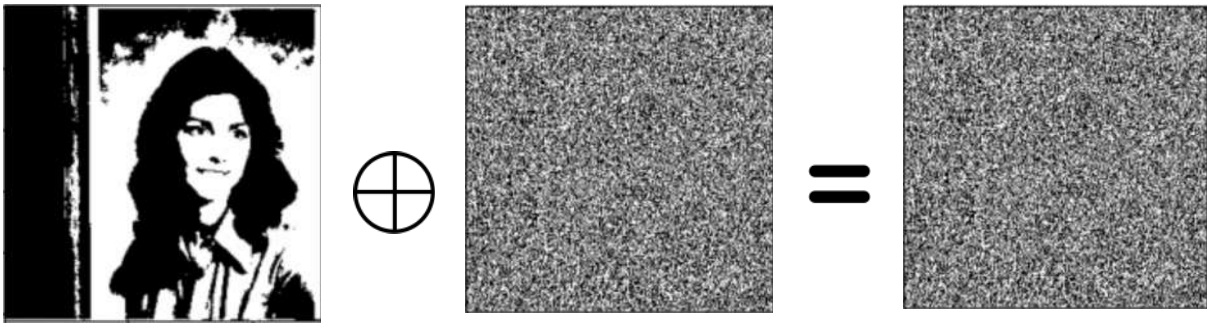


Bitplane 8 Bitplane 7 Bitplane 6 Bitplane 5

Figure 3.15: *Significant bitplane determination by the level α critical region*

Step 4*:* Keys for diffusing the significant bit planes are generated using an ID logistic map based on CCCBG with chosen values of the triplet(x_0,$y_{0,}$μ). The final iterated values of (x_{n+1},y_{n+1}), for the highest significant bit plane, become the initial values (x_0,y_0) for generating the next key and so on.

Step 5*:* The significant bit planes, determined by α% (0.05) level of significance, are ciphered as $CBP_j= BP_j \oplus K_j \ \forall \ j= 1,...,4$.



Cipher bitplane 8

Pseudo random binary matrix

Cipher bitplane8

Figure 3.16: *Cipher bitplane conversion from MSB bitplane*

Step 6*:* The cipher bit planes CBP_j and the encrypted bit plane BP_j are combined together to form cipher image as $C_i(x,y) = CBP_j + BP_k \forall i = 1,2...,8, j = 1,..4$ and $k = 5,...8$ where + is used for combining process.

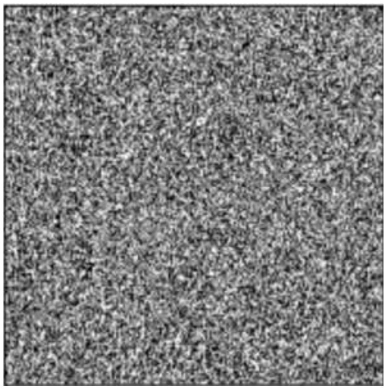
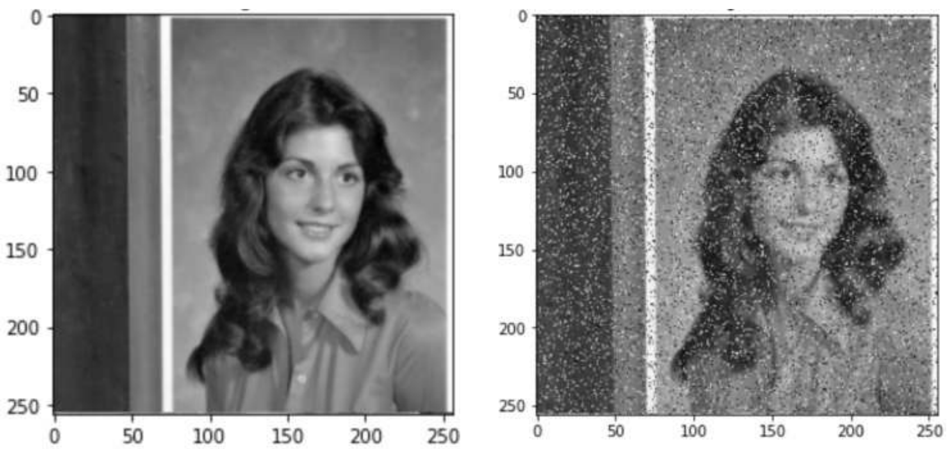


Figure 3.17: *Final cipher image of lena of size 512 × 512*

3.4 Transmission Through Public Channel

After generating the cipher image, we need to transmit the image through some computer networks. If we use some wireless network model or public channel data transmission, then there is a high chance of the data getting publicly exposed. This may cause serious damage for the organization, and put the organizations privacy in risk. Our proposed encryption method can efficiently handle this problem, and ensure a secure data transmission, even if the data gets exposed, it is impossible to decrypt the cipher image without proper seed key values.

Another major problem arises, when we transmit the data through some computer network, there is a high chance that the data gets distorted at some point due to some asymptotic disturbance. If the data gets distorted, then after decrypting the data, we get an image with noise added in it.



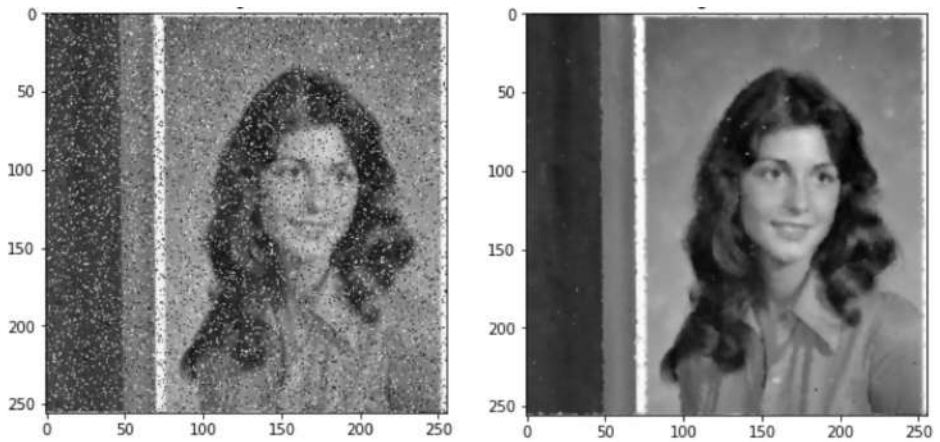
(a) (b)

Figure 3.18: *Encrypted and Decrypted image of lena of size 512x 512 (a) original image of lena (b) decrypted image of lena*

As we can see from the figure (b), noise is added to the decrypted image after decryption. Now the challenge is to denoise the image using a dome degradation algorithm. For our simulation purpose we use a salt and pepper noise model. The best-known order-statistic filter for salt and pepper noise in image processing is the median filter, which, as its name implies, replaces the value of a pixel by the median of the intensity levels in a predefined neighborhood of that pixel –

$$f'(x, y) \ = \ median\,\{\, g(r, c)\,\}\, where\ (r, c) \in S_{x,y} \tag{3.19}$$

where, as before, $S_{x,y}$ is a subimage (neighborhood) centered on point (x, y). The value of the pixel at (x, y) is included in the computation of the median. Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of both bipolar and unipolar impulse noise.



(a) (b)

Figure 3.20: *Encrypted and Decrypted image of lena of size 512x 512 (a) decrypted image of lena (b) final image after removing noise*

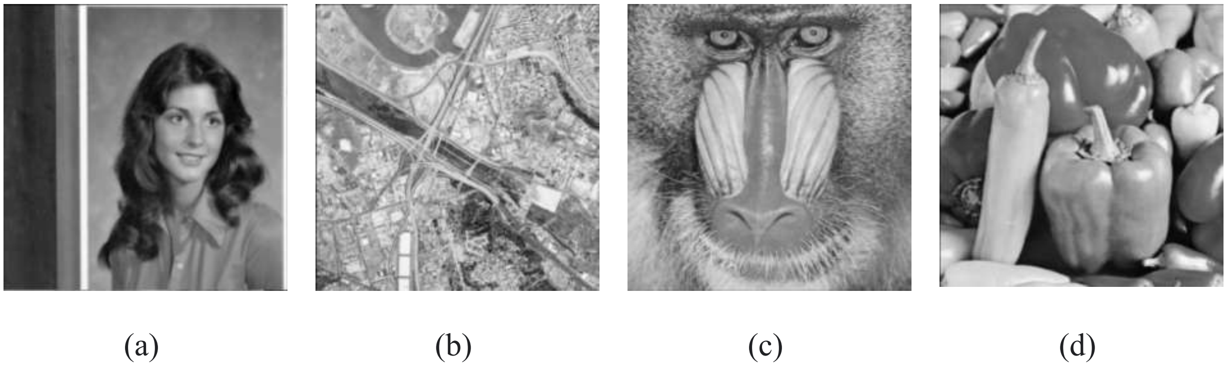
By doing this we can ensure the image’s quality is restored up to 80% – 90% depending on the image and the intensity of the noise model. In this application we use salt and pepper noise to simulate our model, but in real world application the noise may be added in different intensity. It is possible to automate the filter degradation models depending on the noise model in it. Those are described in our future scope.

Chapter 4

Experimental results and their analysis

In this chapter the efficiency of the schema is discussed by exhaustive simulation over a sample of four grayscale images. The grayscale test images were in "*.tiff*" format derived from USC-SIPI Image Database (USE-Viterbi school of engineering, University of South California), which provides a simplicity to perform various quantitative processing tasks.

We use four test images for our analysis purpose. The thumbnails of four test images are shown below–



(a) (b) (c) (d)

Figure 4.1: *thumbnail view of images (a) lena, (b) map, (c) baboon, (d) pappers*

4.1 Subjective Quality Assessment

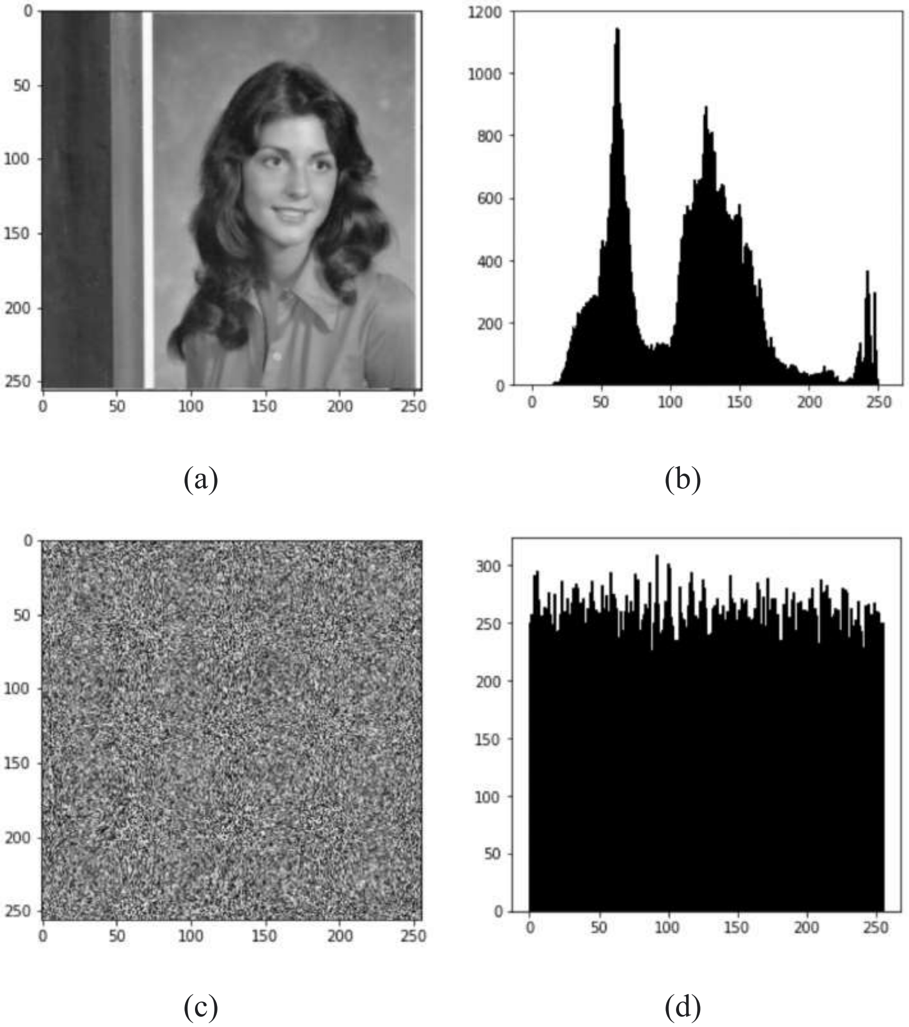
Human's visual quality assessment is "subjective". The human beings' ability to assess the visual quality of an image is influenced by many aspects such as, the level image is influenced by many aspects such as, the level of interaction with the scene, the comfortability of the viewing environment, and the viewer's state of mind. The guidelines for the subjective assessment test conditions such as the viewing distance, the test duration, and the observer's recruitment. As the proposed Algorithms are lossless (which means no loss in the image information), the subjective quality assessment will reveal an identical subjective match between the original and the decrypted images. Therefore, subjective assessment is inadequate. In order to evaluate the proposed Algorithms properly, the authors use objective quality assessment, which provides more reliable results, as shown in the next sub-section.

4.2 Objective Quality Assessment

4.2.1 Statistical test

4.2.1.1 Visual Test through Histogram Analysis

An image histogram demonstrates how pixels in an image are distributed by graphing the number of pixels at intensity level. In order to have a perfectly ciphered image the histogram of the image must exhibit uniformity of distribution of pixels against the intensity values. The histograms of original as well as encrypted images have been analyzed. In Figure (4.2) the histograms of the original image of Lena of size 512×512 and the histograms of corresponding Cipher Image has been plotted.



(a)

(b)

(c)

(d)

Figure 4.2: *Visual test through histogram analysis (a) plain image (b) histogram of the plain image (c) cipher image (d) histogram of cipher image*

From the histogram analysis we can say that the cipher image has a certain pattern whereas that of the Cipher image is uniformly distributed. The features of the original image are uniformly distributed in the cipher image. The cipher image is perfectly secure to transmit.

4.2.1.2 Measures of Central Tendency and Dispersion

Measures of Central Tendency and Dispersion have been used as a measure of homogeneity. The comparative analysis as presented inTable 4.1 depicts that the measures are different in original image and encrypted image. The measures in Cipher Images are uniform which shows that cipher images have uniform mean, median pixel intensities.

Table 4.1: *Measures of central tendency of original image and cypher image*

Image Name	**Size**	**Mean**		**Median**	
		Original	**Cipher**	**Original**	**Cipher**
Lena	512×512	111.17	127.26	118.0	127.0
Map	512×512	140.50	127.45	132.0	127.0
Baboon	512×512	129.61	127.13	130.0	127.0
Pappers	512×512	120.21	127.50	121.0	127.0

Table 4.2: *Measure of dispersion of original and cypher image*

Image Name	**Size**	**Standard Derivation**	
		Original	**Cipher**
Lena	512×512	49.64	73.85
Map	512×512	45.34	73.91
Baboon	512×512	42.31	73.96
Pappers	512×512	53.87	74.01

The measure of cypher images is uniform which shows that the cypher image has uniform mean, median pixel intensities in different images.

4.2.1.3 Correlation Coefficient Analysis

In most of the plain images, there exists high correlation among adjacent pixels whereas poor correlation between the neighboring pixels of corresponding cipher image is observed. Karl Pearson's Product Moment correlation coefficient, stated as follows, is used as a measure to find the correlation of horizontally, vertically and diagonally adjacent pixels of both the plain and cipher image and the correlation between the plain image and cipher image pixels. The formula used to get correlation coefficient is –

$$r_{xy} = \frac{cov(x,y)}{\sigma_x \sigma_y} \text{where } cov(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{4.3}$$

$$\sigma_x = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \; and \; \sigma_y = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 \, \text{with}\sigma_x \neq 0 \; and \; \sigma_y \neq 0 \tag{4.4}$$

Table 4.3: *measure of cross correlation between original images and cipher images*

Image Name	Size	Correlation Coefficient
Lena	512×512	0.0012
Map	512×512	0.0061
Baboon	512×512	0.0031
Pappers	512×512	0.0023

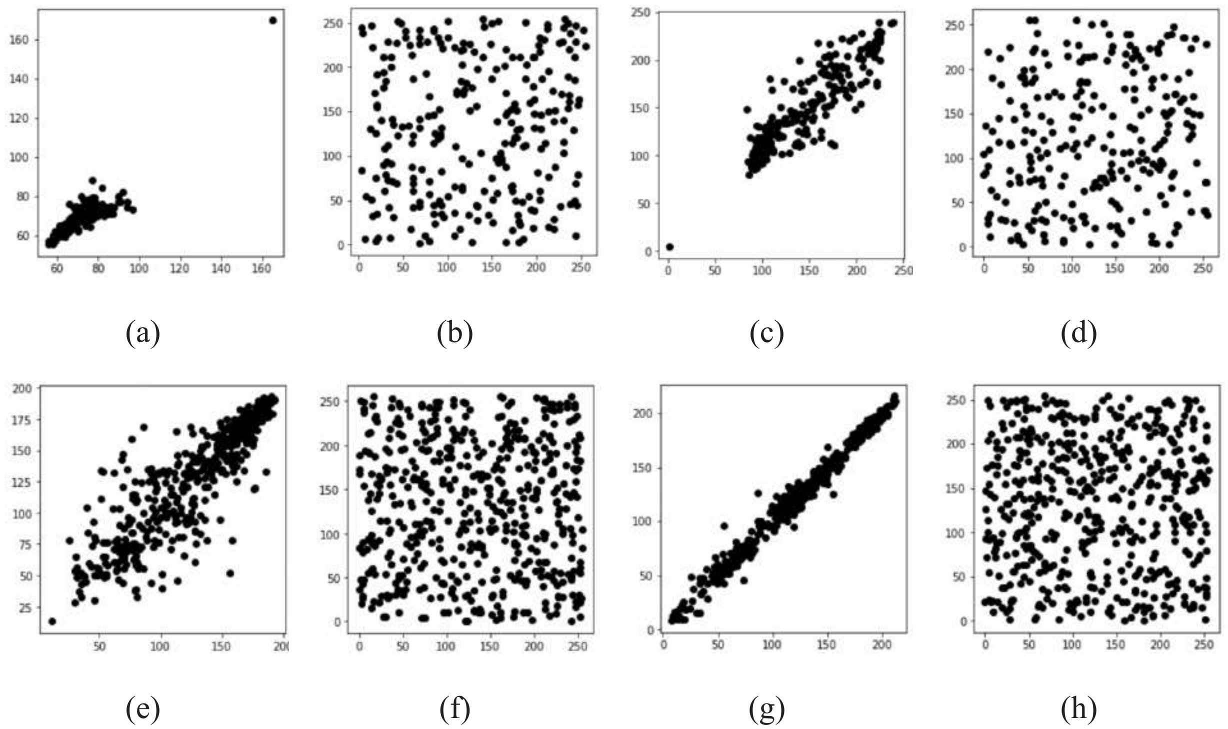
Table 4.4: *coefficient correlation between vertically and horizontally adjacent pixels of original and cypher images*

Image Name	Size	Horizontal Pixel		Vertical Pixel	
		Original	Cipher	Original	Cipher
Lena	512×512	0.9757	0.0026	0.9730	-0.0020
Map	512×512	0.9400	0.0045	0.9704	-0.0015
Baboon	512×512	0.9409	0.0016	0.9275	0.0013
Pappers	512×512	0.9404	-0.0020	0.9930	0.0036

In Table III correlation coefficient between two vertically, and horizontally adjacent pixels of four sample original images and corresponding encrypted images are presented from which it can be concluded that there is negligible correlation between the two vertically and horizontally adjacent pixels in encrypted image but high correlation in original image. Table III also presents the correlation coefficient between the original image and the cipher image.

4.2.1.4 Scatter Plot Analysis

A scatter plot (aka scatter chart, scatter graph) uses dots to represent values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables. We use scatter – plot to observe and show relationships between two numeric variables. Identification of correlational relationships is common with scatter plots. In these cases, we want to know, if we were given a particular horizontal value, what a good prediction would be for the vertical value.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.5: *Scatter Plot analysis (a) original image of lena (b) cipher image of lena (c) original image of map (d) cipher image of map (e) original image of baboon (f) cipher image of baboon (g) original image of papers (h) cipher image of papers*

4.2.2 Key Space Analysis

A good encryption scheme should be sensitive to the secret keys and the key space should be large enough to make brute force attack indefensible. In the proposed algorithm, the initial conditions and the system parameters xo, Yo, and the number of iterations for scrambling has been used as keys where xo,yo E [0,1], ue(10.5,2) if the calculation precision is 10-14, the secret key space is1014 x 1014 x 1014 1012, So the key space is large enough to resist exhaustive attack.

4.2.3 Information Entropy

It is well known that the entropy H(s) of a message sources can be calculated as:

$$H(s) = \sum_{i=0}^{2^N-1} p(S_i).\log_2 \frac{1}{p(S_i)} \tag{4.6}$$

Table 4.5: *Measurement of encryption entropy*

Image name	Size	Entropy of original image	Entropy of cypher image
Lena	512×512	7.5060	7.9971
Map	512×512	7.1914	7.9835
Baboon	512×512	7.8282	7.9967
Pappers	512×512	6.9940	7.9985

4.2.4 Measuring encryption quality: MSE and PSNR

Image encryption quality measure is a figure of merit used for the purpose of evaluation of image encryption techniques. With the application of encryption to an image a change takes place in pixel values as compared to those values before encryption. Such change may be irregular. This means that the higher the change in pixel values, the more effective will be the image encryption and hence the encryption quality. So the encryption quality may be expressed in terms of the total changes in pixel values between the original image and the encrypted one.

The performance of the decryption procedure is measured by the *Peak Signal-to-Noise Ratio* (PSNR). The PSNR of a given image is the ratio of the mean square difference of the component for the two images to the maximum mean square difference that can exist between any two images. It is expressed as a decibel value. Greater PSNR value (>30dB) reveals better image quality. For encrypted images, a smaller value of PSNR is expected. Let C(i,j)and P(i,j) be the gray level of the pixels at the throw and jth column of H x W cipher and plain-image, respectively.

The Mean Square Error (MSE) between these two images is defined as –

$$\mathrm{MSE} = \frac{1}{\mathrm{M} \times \mathrm{N}} \sum_{\mathrm{i}=0}^{\mathrm{M}-1} \sum_{\mathrm{j}=0}^{\mathrm{N}-1} |\, \mathrm{C}(\mathrm{i},\mathrm{j}) - \mathrm{P}(\mathrm{i},\mathrm{j})|^2 \qquad (4.7)$$

The *Peak Signal-to-Noise Ratio* is defined as –

$$\mathrm{PSNR} = 20 \times \log_{10}(\frac{255}{\sqrt{\mathrm{MSE}}}) \qquad (4.8)$$

Table 4.6: *Measurement of encryption quality: MSE, PSNR in DB*

Image name	Size	MSE	PSNR
Lena	512×512	20.809e +003	4.9481
Map	512×512	20.935e +003	4.9220
Baboon	512×512	20.794e +003	4.9513
Pappers	512×512	20.799e +003	4.9502

Chapter 5

Conclusion and future scope

This communication puts forward a non-adaptive partial encryption of grayscale images based on chaos. The proposed algorithm effectively determines significant bit planes on the basis of contribution made by them to form a pixel. The significant bit planes are encrypted by the key stream generated on the basis of a chaos based pseudorandom binary number generator where the insignificant bit planes are left over to reduce the computational time.

The simulation experiment and results show that the encryption algorithm is effective, simple to implement, its secret key space is reasonably large and can effectively resist exhaustive attack, statistical attack and so on.

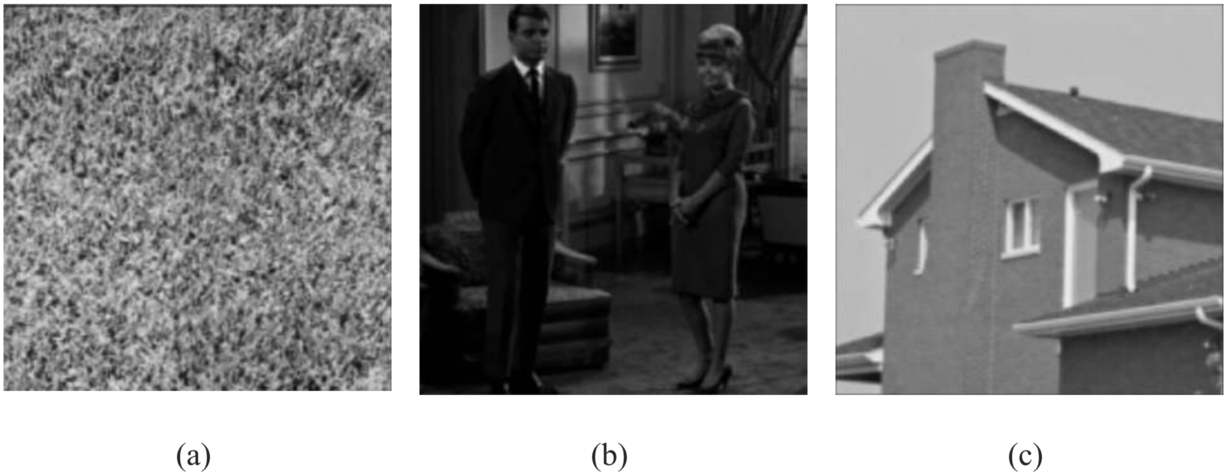
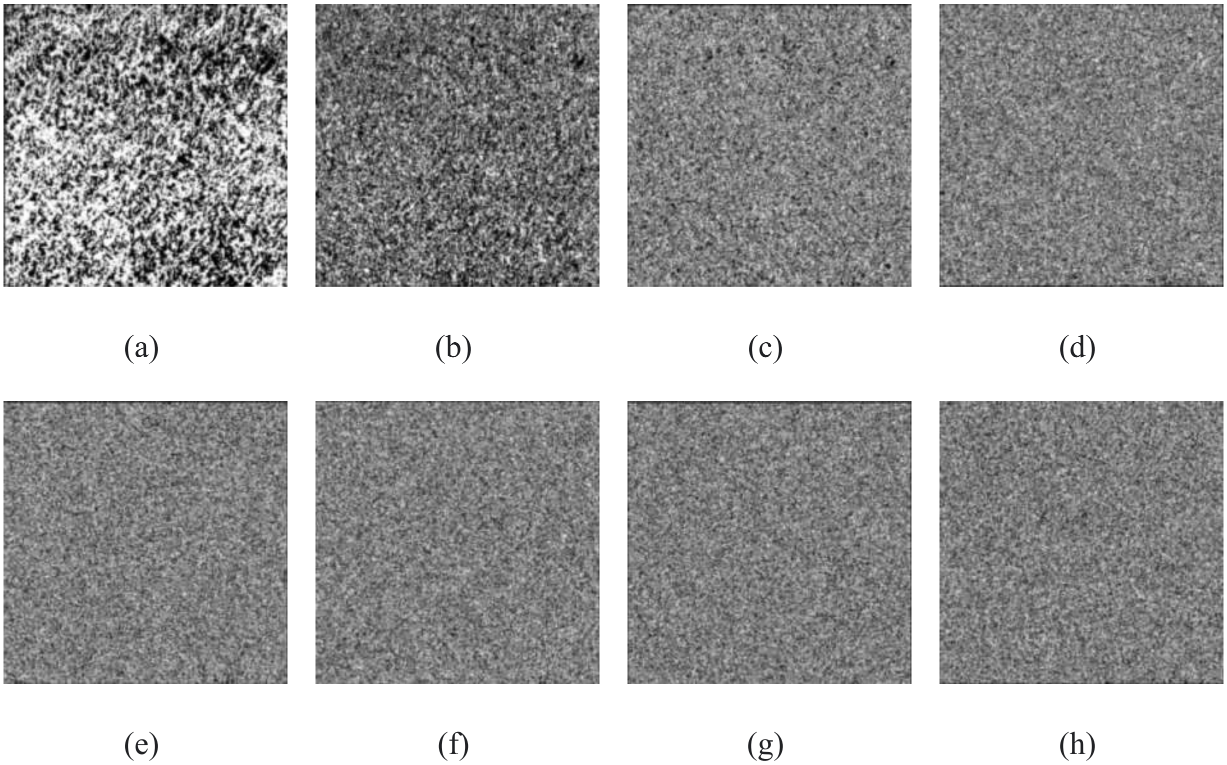


Figure: 5.1: Three types of sample images (a) Original grayscale image of grass, (b) original grayscale image of couple, (c) original grayscale image of house

An image can be classified in three categories of images :

(i) where only a single bit plane contains the entire information



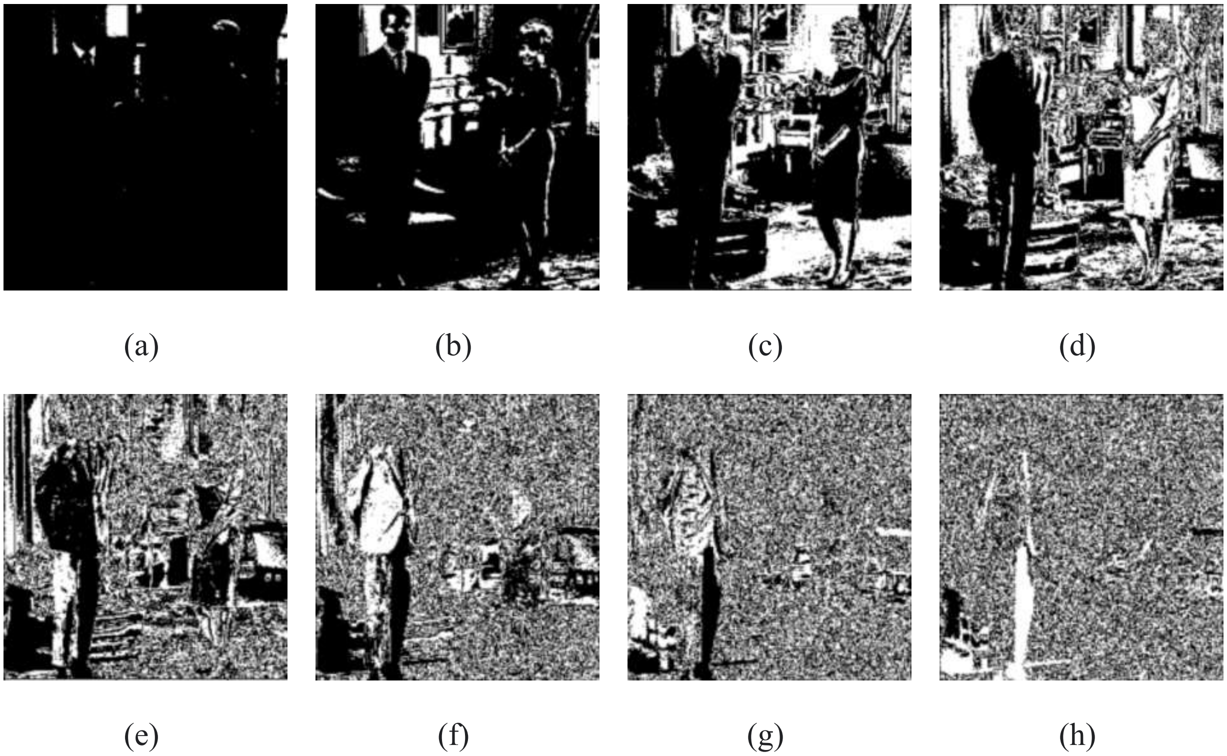
(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5.2: Bitplane Images of figure 5.1.a (a) bitplane8 - MSB, (b) bitplane7, (c) bitplane6, (d) bitplane5, (e) bitplane4, (f) bitplane3, (g) bitplane2, (h) bitplane1- LSB

As it is clearly visible that after dividing the original image of grass into 8 bitplane images, only bitplane 8 contains the most amount of information. We can say, this image only contains one significant bitplane (which is MSB bitplane). To process this type of image, we need to encrypt only a single bitplane which contains the entire data. This can reduce a significant amount of resources, in terms of computation.

(ii) images where all the bit planes contain significant information


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5.3: Bitplane Images of figure 5.1.b (a) bitplane8 - MSB, (b) bitplane7, (c) bitplane6, (d) bitplane5, (e) bitplane4, (f) bitplane3, (g) bitplane2, (h) bitplane1- LSB

As it is clearly visible that after dividing the original image of the couple into 8 bitplane images, all bitplane images contain the most amount of information. We can say, this image contains all significant bitplanes (bitplane 8 to 1). To process this type of image, we need to encrypt all significant bitplanes which contain the entire data. This can take a significant amount of resources, in terms of computation.

(iii) images where some bit planes are significant and some others are not

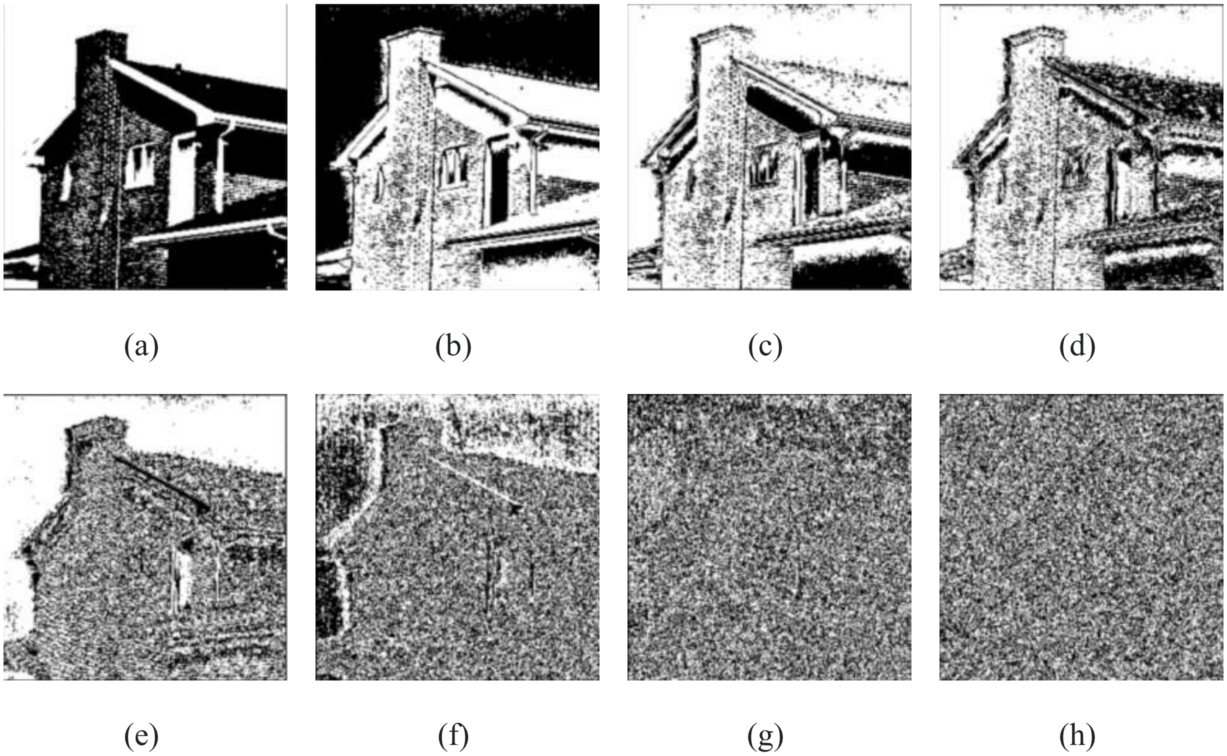


Figure 5.4 : Bitplane Images of figure 5.1.c (a) bitplane8 - MSB, (b) bitplane7, (c) bitplane6, (d) bitplane5, (e) bitplane4, (f) bitplane3, (g) bitplane2, (h) bitplane1- LSB

As it is clearly visible that after dividing the original image of home into 8 bitplane images, some bitplane images contain the most amount of information and some do not. We can say, this type of image contains some percentage of significant bitplanes which contribute the information. To process this types of image, we need to encrypt only most significant bitplanes which contains the most amount of data. The cost to process this type of image can depend on the image itself.

Our interest mainly lies with the third category as encrypting the other categories is easier. Once we classify an image into the third category, a threshold is defined through which we evaluate the importance of the individual bit planes before encrypting them. Designing an adaptive algorithm to detect the significant bit planes and thereafter encrypting them by chaos based PN sequence would be of future concern.

5.1 Future scope:

5.1.1 Image Encryption

At the end of the day we need to protect our data. Increasingly, encryption is being seen as the best way to ensure that data is protected, but the ever growing use of encryption creates a management challenge. The challenge, however, doesn’t need to be daunting. Implementing a flexible and extensible solution that automates many of the time-consuming and error-prone key management tasks in an automated enterprise-wide manner is rapidly becoming a priority for many organizations. A function ‘*number_of_significant _bitplane()*’ can be designed in such a way to automate the number of key significant bitplane images that can be found, to focus on further processing.

```
$(int) \leftarrow NumberOfSignificantBitplane\ (\ BitplaneImages)$:
    $sBit\ =\ int\ (\ number\ of\ significant\ bitplane\ )$
    $Return\ (sBit)$
```

By doing this we can ensure the efficiency of the process and can reduce significant amounts of processing time. In order for enterprise-wide encryption to be deployed correctly, organizations need to deploy the correct tool to manage the keys. In the same way that data protection has moved from an IT challenge to a C-level issue, key management has become a high-level business imperative.

5.1.2 Data Transmission

During data transmission through some public channel, the image gets distorted due to some asymptotic disturbance. In our proposed model, we ensure the image quality using some degradation model. For simulation purposes, we use *salt and pepper noise* and *median filter* degradation model for denoising purposes. But, in real life applications, the noise may be added in different types. The noise models may be independent of the transmitted data. Different noise models need different degradation filters for denoising. This can be automated using some function, which returns the most suitable noise degradation filters with respect to the noise added in it. By doing this, we ensure the image’s quality is restored upto *80% – 90%* (depends on the image and noise intensity).

5.2 Conclusion:

Image plays an important role in lives and they are used in many applications in our day to day lives. Therefore it is necessary to affirm the integrity and confidentiality of the digital image that is being transmitted. Some of the image encryption techniques are discussed that play an important role in image transmission. In this paper a survey of some important image cryptography is provided in the last decades.

These encryption methods are studied and analyzed well to promote the performance of encryption methods. Each technique is unique in its own way and this makes it suitable for its many applications. Everyday new techniques are evolving hence fast and secure conventional encryption techniques work with high security rate. This survey provides a way to realize the different aspects that are used from chaotic to Genetic algorithms approach and DNA sequence for image encryption.

Our model is based on grayscale image processing, but it can work on color images as well (where color images have 3 channels) with efficiency. The standard tricolor images produced by the SDSS are very good images. A picture that is processed to show faint asteroids may be useless to study the bright core of a galaxy in the same field.

REFERENCES

1. Sukalyan Som, Sayani Sen, "A Non-adaptive Partial Encryption of Grayscale Images Based on Chaos", International Conference on Computational Intelligence: Modeling, Techniques and Applications, CIMTA-2013.
2. Narendra K Pareek, Vinod Patidar, Krishan K Sud, "A Random Bit Generator Using Chaotic Maps", International Journal of Network Security, Vol.10, No.1, PP.32 (38, Jan. 2010).
3. USC-SIPI Image Database, USE-Viterbi school of engineering, University of South California, since 1981. (available on *sipi.usc.edu/database/database.php*)
4. "The Future of Encryption", by Richard Molds, published on Helpnetsecurity February 18, 2008. (available on helpnetsecurity.com/2008/02/18/the-future-of-encryption)
5. Research Group VITS, Orebro University, Sweden, "Information Security Fundamentals", 2003, Security Education and Critical Infrastructures, pp 95–107.
6. "Cryptosystem", by Corinne Bernstein, June 2019, article published on Techtarget (available on techtarget.com/searchsecurity/definition/cryptosystem).
7. "Computer Security Fundamentals", book by Chuck Easttom, 2E, 2012 by Pearson, ISBN-10: 0-7897-4890-8.
8. Digital Image Processing, 4'th global edition by Rafael C. Gonzalez –University of Tennessee, Richard E. Woods –Interoptics, person.
9. Prasenjit Kumar Das , Mr. Pradeep Kumar and Manubolu Sreenivasulu, "Image Cryptography: A Survey towards its Growth" Advance in Electronic and Electric Engineering. ISSN 2231-1297.