AN APPROACH TOWARDS DESIGNING ROBUST CRYPTOGRAPHY FOR DIGITAL CONTENT

**A THESIS SUBMITED TO THE
FACULTY OF ENGINEERING & TECHNOLOGY,
JADAVPUR UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE
DEGREE OF MASTER OF TECHNOLOGY
IN THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING**

**By
SOUMYA PAUL
Registration No. - 137128 of 2016-17
Exam Roll No. – M6TCT19025**

**Under The Esteemed Guidance Of
Prof. ATAL CHAUDHURI**

**Department of Computer Science & Engineering
JADAVPUR UNIVERSITY**

**Department of Computer Science & Engineering
Jadavpur University
Kolkata-700032
May 2019**

To whom it may concern

This is to certify that the work embodied in this thesis entitled ***"An approach towards designing Robust Cryptography for digital content"*** has been satisfactorily completed by Soumya Paul (Registration no. 137128 of 2016-2017, Exam Roll no. - M6TCT19025). It is a bona-fide piece of work carried out Jadavpur University, Kolkata, for partial fulfillment of requirements for the awarding of the Master of Computer Technology degree of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University during the academic year 2016-19.

(Prof. Atal Chaudhuri)
THESIS SUPERVISOR
Dept. of Computer Science & Engineering
Jadavpur University
Kolkata-700032

(Prof. Mahantapas Kundu)
HEAD
Dept. of Computer Science & Engg.
Jadavpur University
Kolkata-700032

(Prof. Chiranjib Bhattacharjee)
DEAN
Faculty of Engineering & Technology
Jadavpur University
Kolkata-700032

Certificate of Approval

This is to certify that the thesis entitled ***“An approach towards designing Robust Cryptography for digital content”*** is a bona-fide record of work carried out by Soumya Paul in partial fulfillment of the requirements for the award of the degree of Master of Technology in the department of Computer Science and Engineering, Jadavpur University during the period June 2018 to May 2019. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

Examiners

... **...**

(Signature of the Examiner) **(Signature of the Supervisor)**

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as part of my degree Master of Technology in Computer Technology studies.

All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original of this work.

Name : SOUMYA PAUL

Registration Number : 137128 of 2016-17

Exam Roll Number : M6TCT19025

Thesis Title : *AN APPROACH TOWARDS DESIGNING ROBUST CRYPTOGRAPHY FOR DIGITAL CONTENT*

Signature with Date :

Dedicated

to

My Parents

Whose affection, love, encouragement and prays of day and night makes me able to get such success and honor

and

My Guide

The touch bearer of my life

Acknowledgements

This dissertation is not only a signature of my research or the skills at keyboard but resembles a milestone in almost a year short of the three year spent at Jadavpur University; more specially in the Department of Computer Science and Engineering. Since my first days at Jadavpur University, starting in September of 2016, I have been encouraged to find the ‘me" in myself. With the knowledge filled lecture sessions of extremely qualified professors and sustained assistance, I not only tasted the beauty of Computer Science but also was motivated to pursue my research in the same domain. This thesis is also the summarization of all the experiences I have gained in the university and from substantial number of remarkable individuals in my life.

Firstly, I would like to thank GOD, almighty, my supreme guide, for bestowing his blessings upon me in my entire endeavor, then I would like to thank my research advisors, Prof. Atal Chaudhuri, Dept. of Computer Science and Engineering, Jadavpur University and Dr. Prabir Kumar Naskar, Dept. of Computer Science and Engineering, Govt. College of Engg. and Textile Technology, Serampore, Hooghly, INDIA. From the initial days, they have been highly supportive in my endeavor for a research topic. They never said me `no' in my amazing journey, right from the selection of a research topic to the completion of this thesis and in between.They has not only been my research advisors, but also my source of inspiration in troubled times. At times, when I did get low, their specific set of words – “Carry on, we are with you" gave me the much needed oxygen for me to get motivated. In short, they have not only guided my research but also me provided academic and emotional assistance through the rough path of this present research completion. Thank You, Sir.

I would like to thank Prof. Mahantapas Kundu, Head of Dept., Computer Science and Engineering, and Prof. Chiranjib Bhattacharjee, Dean, Faculty of Engineering and Technology, Jadavpur University, for providing me with all the assistance needed to pursue my research work. My growth of knowledge would not have been possible without the sessions and discussions with my respected professors.

I conclude this portion with my family, the basic source of sustenance of life. I thank my parents, Mr. Susanta Kumar Paul and Mrs. Maya Paul for believing in me, educating me and providing me with all the amenities to grow up into an educated human being.

In my wonderful journey of my M.Tech research, I have witnessed a plethora of notable individuals who supported and motivated me towards success. Some of them I have mentioned and the rest remain in my mind unforgotten for the rest of my life.

List of Publications & Awards

➢ List of Publications

1. Prabir Kr. Naskar, Soumya Paul, Dipta Nandy and Atal Chaudhuri, *“DNA Encoding and Channel Shuffling for Secured Encryption of Audio Data”*, Multimedia Tools and Applications, Springer Nature, May 2019. DOI :: https://doi.org/10.1007/s11042-019-7696-z

2. Soumya Paul, Pranjal Dasgupta, Prabir Kr. Naskar and Atal Chaudhuri, “*Secured Image Encryption Scheme Based on DNA Encoding &Chaotic Map"*, Review of Computer Engineering Studies, IIETA, June 2017, vol. 4, no. 2, pp. 70-75.

➢ List of Awards

1. The research paper titled *“Secured image encryption scheme based on DNA encoding and chaotic map"* has been awarded the best paper for the session *“Cyber Security and Encryption Technology"* in International Conference on Advances in Science Technology (ICAST 2017) organized by MCKV Institute of Engineering from 17th to 19th March, 2017.

2. The research paper titled *“Secured image encryption scheme based on DNA encoding and chaotic map”* has been awarded the best paper for the session *“Cryptograph"* in Regional Student Convention (RSC 2016) organized by Computer Society of India, Region II, Computer Society of India, Kolkata Chapter and MCKV Institute of Engineering, Liluah, Howrah on 20th August, 2016 at the Seminar Hall of MCKV Institute of Engineering, Liluah, Howrah.

Title of the Thesis

An approach towards designing Robust Cryptography for digital content

Contents

List of Figures

List of Tables

Abstract

Multimedia file like audio and image demands special encryption technique due to its large data capacity without compromising correlation between it's original and encrypted version (closer to zero). Most of the popular block cipher techniques work on multiple rounds whereas the proposed scheme guarantees the necessary low correlation between the original and the encrypted file without multiple rounds. The unique feature is that consecutive blocks use different keys derived from the original one using proposed key chaining algorithm and experimental results show that the correlation between the consecutive keys is also close to zero. The used encryption technique is based on DNA encoding with logistic chaotic map using the generated chain of keys. Furthermore, the concept of channel shuffling is introduced to make the encrypted data more secure. The experimental results confirm that the correlation between the original and ciphered block is close to zero and number of samples change rate value is close to 100. Again correlation between the two consecutive ciphered blocks is also close to zero, which conforms the acceptability of proposed scheme.

Chapter 1

Introduction

1.1 Concept of Cryptography

Cryptography is the science of information security. It is nothing but the art of secret writing. The word is derived from the Greek kryptos, meaning hidden. Cryptography is closely related to the disciplines of cryptology and cryptanalysis. Cryptography includes techniques such as microdots, merging words with images, and other ways to hide information in storage or transit. However, in today's computer-centric world, cryptography is most often associated with scrambling plaintext (ordinary text, sometimes referred to as clear text) into cipher text (a process called encryption), then back again (known as decryption). Individuals who practice this field are known as cryptographers. Modern cryptography concerns itself with the following four objectives:

- **Confidentiality:** Confidentiality is the assurance that information is not disclosed to unauthorized individuals, programs, or processes. So, loss of confidentiality is the unauthorized disclosure of information. We need to protect our confidential information. An organization needs to guard against those malicious actions that endanger the confidentiality of its information. In the military, concealment of sensitive information is the major concern.

In industry, hiding some information from competitors is crucial to the operation of the organization. In banking, customers account need to be kept secret.

- **Integrity:** Integrity means information must be accurate, complete, and protected from unauthorized modification or destruction. In a bank, when a customer deposits or withdraws money, the balance of his/her account needs to be changed. Integrity means that changes need to be done only by authorized entities and through authorized mechanisms. Integrity violation is not necessarily the result of a malicious act; an interruption in the system, such as a power surge, may also create unwanted changes in some information.

- **Non-repudiation:**The creator/sender of the information cannot deny at a later stage his or her intentions in the creation or transmission of the information.

- **Authentication:** The sender and receiver can confirm each other's identity and the origin/destination of the information.

Procedures and protocols that meet some or all of the above criteria are known as cryptosystems. The origin of cryptography is usually dated from about 2000 BC, with the Egyptian practice of hieroglyphics. These consisted of complex pictograms, the full meaning of which was only known to an elite few. The first known use of a modern cipher was by Julius Caesar (100 BC to 44 BC), who did not trust his messengers when communicating with his governors and officers. For this reason, he created a system in which each character in his messages was replaced by a character three positions ahead of it in the Roman alphabet.

In recent times, cryptography has turned into a battleground of some of the world's best mathematicians and computer scientists. The ability to securely store and transfer sensitive information has proved a critical factor in success in war and business. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

1.2 General Idea of Cryptography

Cryptography consists of two processes. Firstly, encryption which is the process of converting ordinary information (plaintext) into unintelligible gibberish (i.e., cipher text). Secondly, decryption which is the reverse, in other words, moving from the unintelligible cipher text back to plaintext, as shown in following figure. a cipher (or cipher) is a pair of algorithms for performing the process of encryption and decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a key. This is a secret parameter (ideally known only to the communicants) for a specific message exchange context. The key is the most important part in the process of cryptography. If it gets corrupted or lost after the process of encryption then the secret message cannot be decrypted.

Thus three characters participate in an information exchange scenario; the sender who needs to send secure data, the receiver who is the recipient of the data and the person who somehow disturbs the communication between “A” and “B” by intercepting messages to uncover the data or by sending her own disguised message.

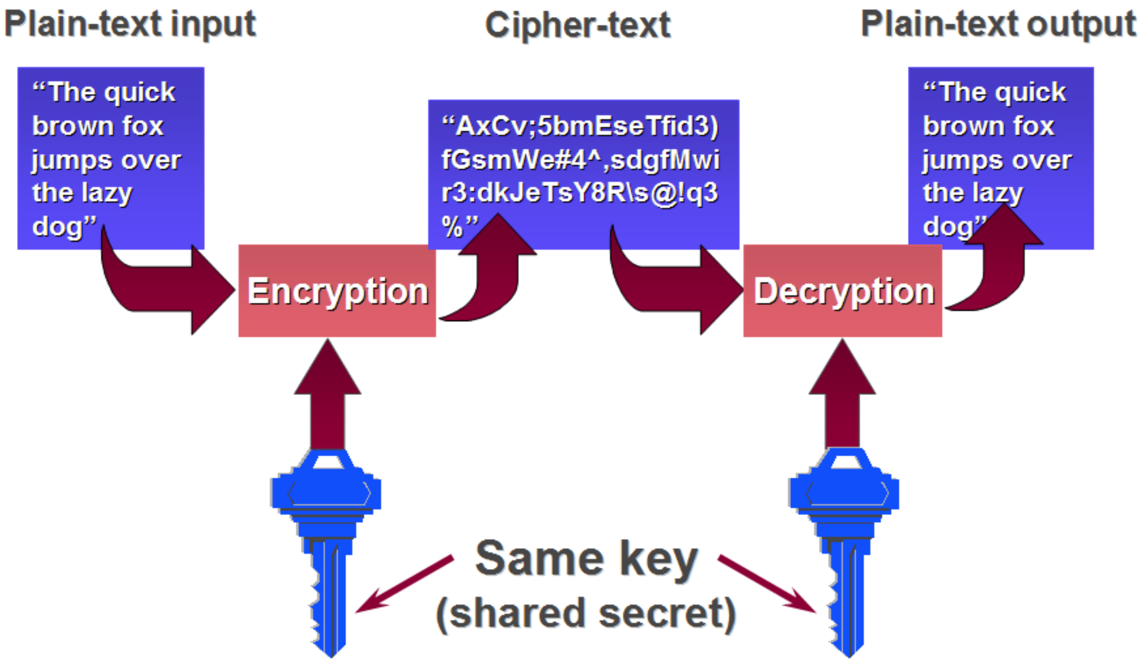

Plain-text input
Cipher-text
Plain-text output
“The quick brown fox jumps over the lazy dog”
“AxCv;5bmEseTfid3) fGsmWe#4^,sdgfMwi r3:dkJeTsY8R\s@!q3 %”
“The quick brown fox jumps over the lazy dog”
Encryption
Decryption
Same key
(shared secret)

Figure 1.1: The Process of Cryptography.

1.3 Various Types of Cryptography

Cryptography involves scrambling a message or creating a digest of the message. Based on the number of keys that are employed for encryption and decryption, their application and uses, we can categorize the overall cryptographic techniques, which are discussed briefly in following section.

1.3.1 Symmetric-Key Cryptography

Symmetric-key cryptographic started thousands of years ago when people needed to exchange secrets (for example in a war).We still mainly use symmetric-key cryptography in our network security. In symmetric-key cryptography, the same key is used by the sender as well as the receiver. The sender uses this key and an encryption algorithm to encrypt data; the receiver uses the same key and corresponding decryption algorithm to decrypt the data. The process is shown in following Figure 1.2.

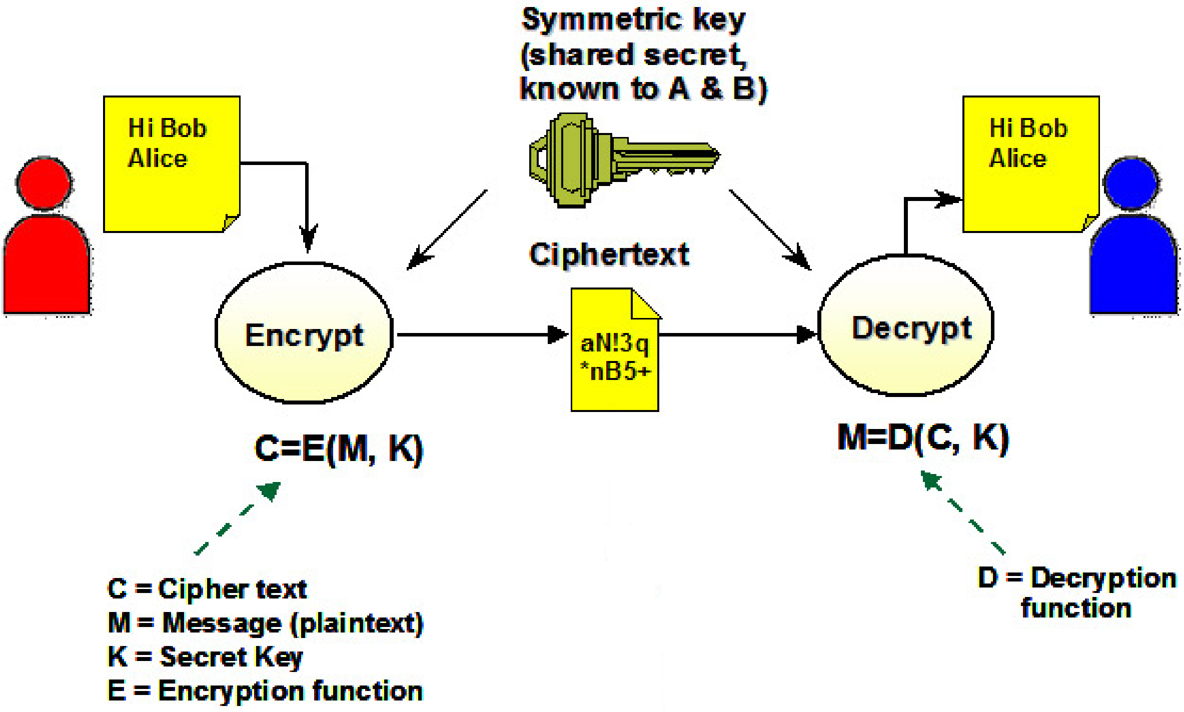


Figure 1.2: Symmetric-Key Cryptography

1.3.2 Asymmetric-Key Cryptography

In Asymmetric or public-key cryptography, there are two keys: a private key and a public key. The private key is kept by the receiver. The public key is announced to the public. The public key is used by the sender to encrypt the message while the private key kept with the receiver to decrypt the message. The process is shown in following Figure 1.3.

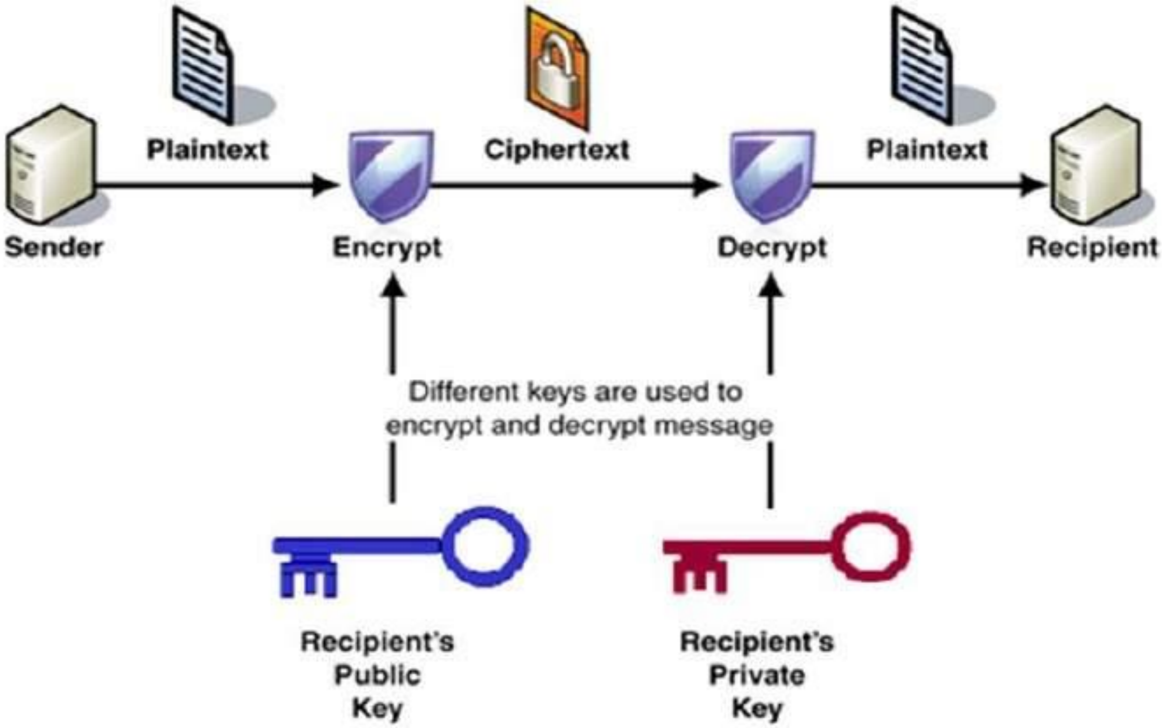


Figure 1.3: Asymmetric – Key Cryptography

1.4 Necessity of Cryptography [1]

Cryptography is a science that applies complex mathematics and logic to design strong encryption methods. It is also an art which allows people to keep confidence in the electronic world. Cryptography plays important role in various type of area, such as text, audio, image etc.

Main goal of cryptography process is to prevent the transmitting information from unwanted person of observing its meaning. When people started doing business online and needed to transfer funds electronically, the applications of cryptography for integrity began to surpass its use for confidentiality. In today's world thousands of people interact electronically every day by different means like e-mails, ATM machines, e-commerce or cellular phones. The rapid increase of information transmitted electronically resulted to an increased reliance on cryptography and authentication.

Key Management plays an important role in cryptography. We have to make sure no unauthorized access occurs to key. And in case we lose a key eventually we lost our data too protected by that key. So now we will discuss about some key management areas.

1.4.1 Principles of Key Management

Below are the required three conditions:-

- What **key strength** is adequate for the data protected?
- How will we ensure **keys are protected** but available when needed?
- Where will we **store the keys**?

- **Key Strength**

Use of weak keys may achieve compliance but it provides false sense of security to its customers and investors. For AES it can use 128, 192 or 256 bit keys. But 128 bit key is strong enough for business data as long as it is random. One can measure key strength by key size and attacker's ability to step through possible combination until right key is found. The best way to choose a key (all bit combinations) is that should be appear in the key space i.e. all possible keys.

- **Key Protection**

Encrypted keys can't be locked away and only brought out by trusted employees as needed. Instead, keep the keys available and safe. Access security for the key is at most basic level. It does not matter how well protected your keys are when not in use, valid users and applications must gain access. Addition to authentication also emphasis on identity verification should be strong and aggressively enforces separation of duties, need-to-know and least privileges.

- **Key Storage**

Many time we feel to store key on the same system and drive as the encrypted files are stored. This seems good idea when our key is encrypted but still its bad security practice. If in case our system fails and key is not recoverable. We might have backup for that system but backup restored do not work as intended.

1.4.2 Cryptography's Role in Society

Encryption does not guarantee that every piece of data is protected from unauthorized access. Now we will see where exactly encryption fits in overall security controls architecture. Encryption acts as an additional layer of security. The biggest mistake made by any organization is by considering encryption as solution for all security issues. For example Encryption is not in affect when data is not in transit like data at resides at server and when being processed at client end. So if attacker gets access to server he doesn't care whether organizations make use of encryption or not as he will get plaintext. In other case it may be possible key gets compromised. And Security team make assumption of 100% security through encryption and do not pay attention to (SEIM) or response policies. Before deploying encryption please implement following controls.

- Implement authentication & authorization controls between application and databases.
- Strong application access control.
- Separate data access management; employees should allow to access data based on the access control.
- Physical security should be implemented for storage, and system components.
- Implement log management and monitoring to find anomalous activity across network.
- Follow security best practices for database and system components.

1.4.3 When to Encrypt

- When data moves from one zone to another zone, moves between two external end points or moving between trusted wireless zones, then before transmission encryption is required.
- If we store all data of an organization in an encrypted database like spread sheet instead of a flat file, then we can prevent the unauthorized access. Hence for database protection encryption is required.
- In modern scenario encryption also required in various field, like research fields, defense, satellite communication, medical data transmission etc.

Chapter 2

Review of Different File Formats

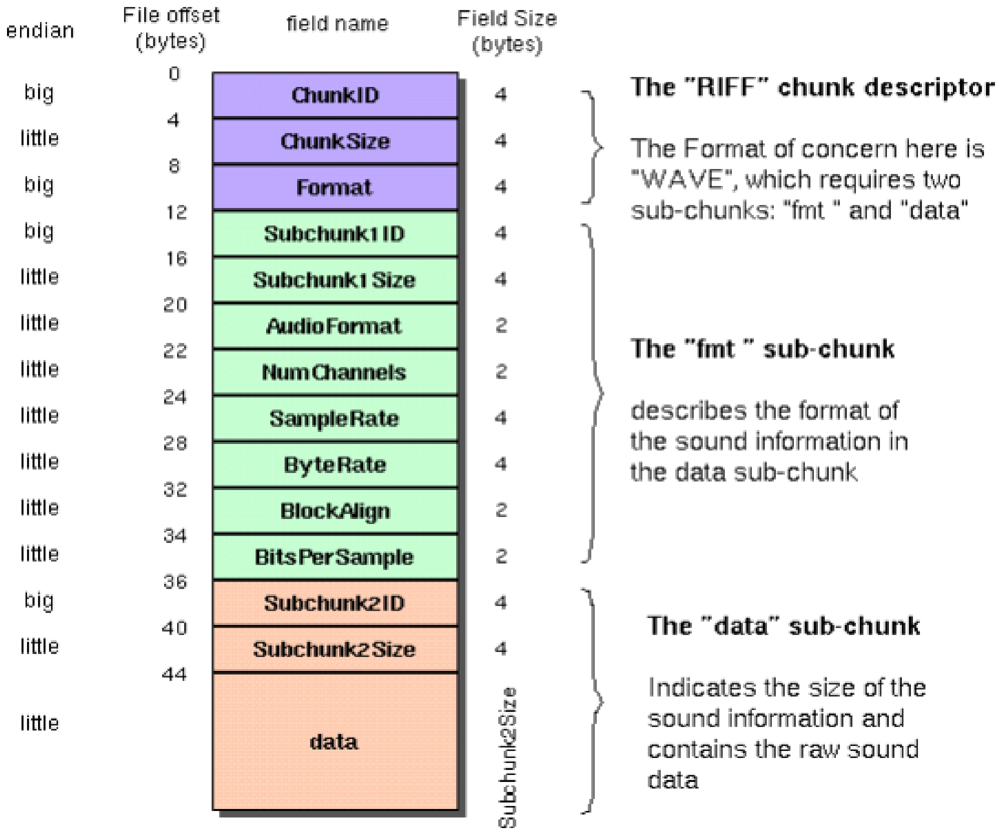
2.1 Audio File (.WAV File)

2.1.1 Format of the Digital Audio (. WAV) File

The WAV file is an instance of a Resource Interchange File Format (RIFF) defined by IBM and Microsoft. The RIFF format acts as a "wrapper" for various audio coding formats. Though a WAV file can contain compressed audio, the most common WAV audio format is uncompressed audio in the linear pulse code modulation (LPCM) format. LPCM is also the standard audio coding format for audio CDs, which store two-channel LPCM audio sampled at 44,100 Hz with 16 bits per sample. Since LPCM is uncompressed and retains all of the samples of an audio track, professional users or audio experts may use the WAV format with LPCM audio for maximum audio quality. WAV files can also be edited and manipulated with relative ease using software.

The WAV format supports compressed audio using, on Microsoft Windows, the Audio Compression Manager. Any ACM codec can be used to compress a WAV file. The user interface (UI) for Audio Compression Manager may be accessed through various programs that use it, including Sound Recorder in some versions of Windows. Beginning with Windows 2000, a WAVE_FORMAT_EXTENSIBLE header was defined which specifies multiple audio channel data along with speaker positions, eliminates ambiguity regarding sample types and container sizes in the standard WAV format and supports defining custom extensions to the format chunk. There are some inconsistencies in the WAV format: for example, 8-bit data is unsigned while 16-bit data is signed, and many chunks duplicate information found in other chunks.

2.1.2 The Canonical WAV File Format



endian	File offset (bytes)	field name	Field Size (bytes)
big	0	ChunkID	4
little	4	ChunkSize	4
big	8	Format	4
big	12	Subchunk1ID	4
little	16	Subchunk1Size	4
little	20	AudioFormat	2
little	22	NumChannels	2
little	24	SampleRate	4
little	28	ByteRate	4
little	32	BlockAlign	2
little	34	BitsPerSample	2
big	36	Subchunk2ID	4
little	40	Subchunk2Size	4
little	44	data	Subchunk2Size

The "RIFF" chunk descriptor

The Format of concern here is "WAVE", which requires two sub-chunks: "fmt " and "data"

The "fmt " sub-chunk

describes the format of the sound information in the data sub-chunk

The "data" sub-chunk

Indicates the size of the sound information and contains the raw sound data

Figure 2.1: Canonical WAV File Format

2.1.3 Header Structure

Table-2.1: WAV Header Structure

Offset	***Size***	***Name***	***Description***
The canonical WAVE format starts with the RIFF header:			
0	4	ChunkID	Contains the letters "RIFF" in ASCII form (0x52494646 big-endian form).
4	4	ChunkSize	36 + SubChunk2Size, or more precisely: 4 + (8 + SubChunk1Size) + (8+SubChunk2Size). This is the size of the rest of the chunk following this number. This is the size of the entire file in bytes minus 8 bytes for the two fields not included in this count: ChunkID and ChunkSize.
8	4	Format	Contains the letters "WAVE" (0x57415645 big-endian form).
The "WAVE" format consists of two subchunks: "fmt " and "data":			
The "fmt " subchunk describes the sound data's format:			
12	4	Subchunk1ID	Contains the letters "fmt " 0x666d7420 big-endian form).
16	4	Subchunk1Size	16 for PCM. This is the size of the rest of the Subchunk which follows this number.
20	2	AudioFormat	PCM = 1 (i.e. Linear quantization)Values other than 1 indicate some form of compression.
22	2	NumChannels	Mono = 1, Stereo = 2, etc.
24	4	SampleRate	8000, 44100, etc.
24	4	ByteRate	==SampleRate * NumChannels * BitsPerSample/8
32	2	BlockAlign	== NumChannels * BitsPerSample/8 The number of bytes for one sample including all channels. I wonder what happens when this number isn't an integer?
34	2	BitsPerSample	8 bits = 8, 16 bits = 16, etc.
	2	ExtraParamSize	if PCM, then doesn't exist
	X	ExtraParams	Space for extra parameters
The "data" subchunk contains the size of the data and the actual sound:			

36	4	Subchunk2ID	Contains the letters "data" (0x64617461 big-endian form).
40	4	Subchunk2Size	==NumSamples*NumChannels* BitsPerSample/8, This is the number of bytes in the data. You can also think of this as the size of the read of the subchunk following this numbe.
44		Data	The actual sound data.

2.1.4 Audio (.wav) File Structure

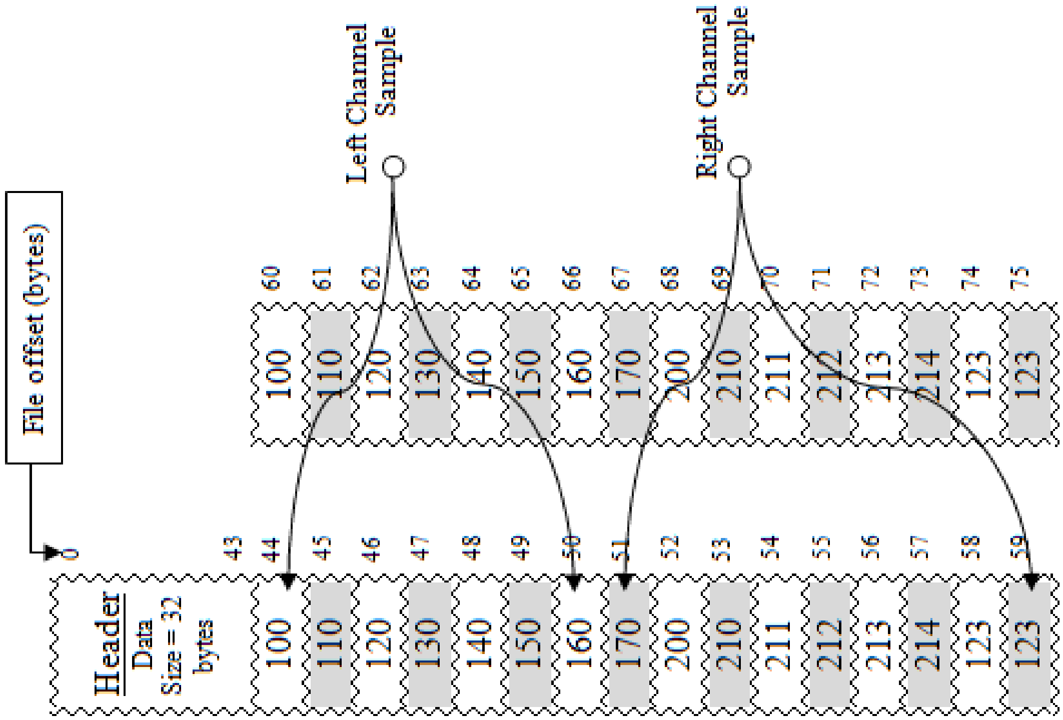


Figure 2.2: A Sample of Audio Structure of Size 32 Bytes

Audio (.wav) file header size is 44 bytes and after header structure actual data bytes are contained. An example, here are opening 32 data bytes (excluding the header information) of a WAV file with bytes shown in Figure 2.2. In Figure 2.2, shaded and non-shaded cell represents right channel sample and left channel sample respectively, where left channel sample is in even index position and right channel sample is in odd index position.

2.1.5 Discussion of WAV File with an Example

As an example, here are the opening 72 bytes of a WAVE file with bytes shown as hexadecimal numbers:

52 49 46 46 24 08 00 00 57 41 56 45 66 6d 74 20 10 00 00 00 01 00 02 00

22 56 00 00 88 58 01 00 04 00 10 00 64 61 74 61 00 08 00 00 00 00 00 00

24 17 1e f3 3c 13 3c 14 16 f9 18 f9 34 e7 23 a6 3c f2 24 f2 11 ce 1a 0d

Here is the interpretation of these bytes as a WAVE sound file:

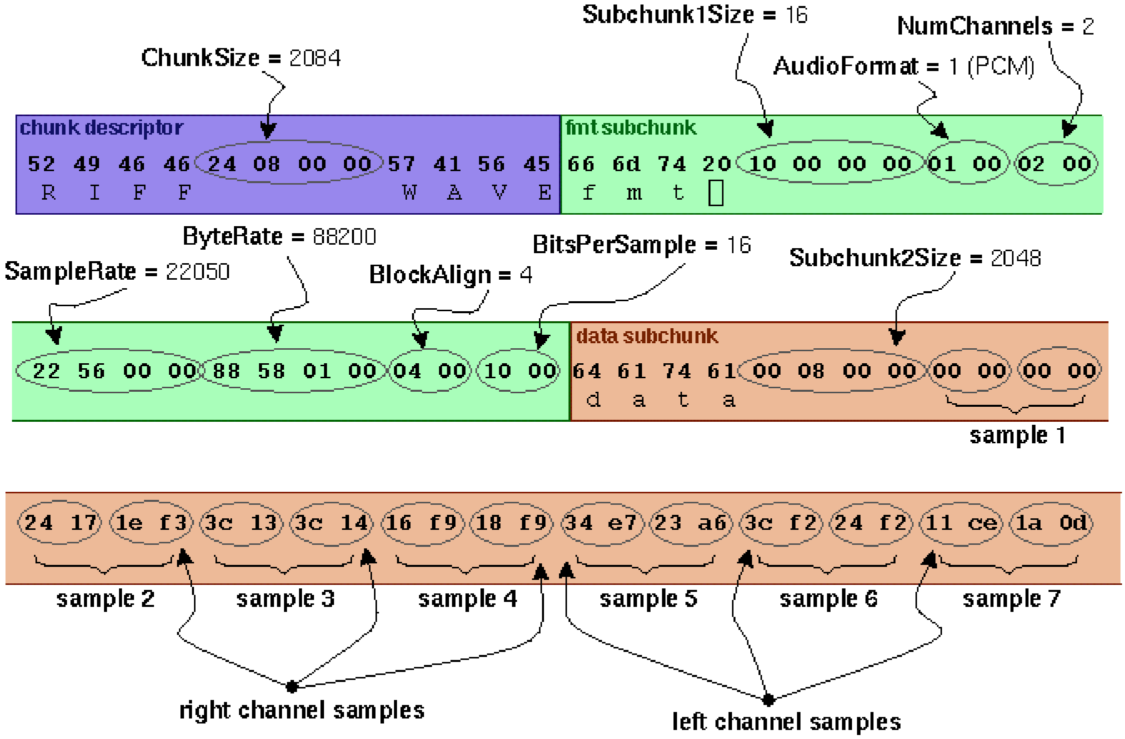


Figure 2.3: WAV Sound File Structure

- The default byte ordering assumed for WAVE data files is little-endian. Files written using the big-endian byte ordering scheme have the identifier RIFX instead of RIFF.
- The sample data must end on an even byte boundary. Whatever that means.
- 8-bit samples are stored as unsigned bytes, ranging from 0 to 255. 16-bit samples are stored as 2's-complement signed integers, ranging from -32768 to 32767.

- There may be additional subchunks in a Wave data stream. If so, each will have a char [2] SubChunkID, and unsigned long SubChunkSize, and SubChunkSize amount of data.
- RIFF stands for *Resource Interchange File Format*.

2.1.6 General Discussion of RIFF Files

Multimedia applications require the storage and management of a wide variety of data, including bitmaps, audio data, video data, and peripheral device control information. RIFF provides a way to store all these varied types of data. The type of data a RIFF file contains is indicated by the file extension. Examples of data that may be stored in RIFF files are:

- Audio/visual interleaved data (.AVI)
- Waveform data (.WAV)
- Bitmapped data (.RDI)
- MIDI information (.RMI)
- Color palette (.PAL)
- Multimedia movie (.RMN)
- Animated cursor (.ANI)
- A bundle of other RIFF files (.BND)

2.1.7 WAV Header Structure in Programming Concept

```
typedef unsigned char u8;
typedef unsigned short int u16;
typedef unsigned int u32;
typedef struct wav_hdr {
                    u32         ChunkID;
                    u32         ChunkSize;
                    u32         Format;
                    u32         Subchunk1ID;
```

```
        u32        Subchunk1Size;
        u16       AudioFormat;
        u16        NumChannels;
        u32        SampleRate;
        u32        ByteRate;
        u16        BlockAlign;
        u16        BitPerSample;
        u32        Subchunk2ID;
        u32        Suchunk2Size;
    }WAV_HEADER;
```

2.2 Image File (.BMP File)

2.2.1 Format of Digital Image (.BMP) File

A digital image is represented by a one dimensional array of numbers that represent the different light intensities of each pixel. The dimension of a 640 × 480 pixel image can literally be multiplied out to find the total number of pixel in the image, in this case 307,200 pixels. In digital photography this is known as the resolution of an image**,** a digital camera that takes pictures of 640 × 480 is known as a 0.3 Mega Pixel resolution camera for this reason.

Digital images usually use either 24-bits (standard bitmap) or 8 bits (standard GIF image, color or grey scale) for the storage of intensity information per pixel. This means that in a bitmap image (BMP) there are a potential 16.8 Million colors (224) per pixel and in a GIF, 256 different color combinations.

In a typical 24-bit image, each pixel has three color components, red, green and blue, each component using 8 bits to represent a value from 0 to 255. An 8 bit image on the other hand can either have a color palette of 256 different grey levels or color values. Naturally this results in the 8 bit color image having to perform a “best fit” in order to match a real world color to its limited palate. The number of pixels in an uncompressed BMP image contributes directly to file size, for instances, a 640 × 480 image has 3, 07,200 pixels in total, and each of this is represented by 24 bits which equals a total of 900 Kilobytes.

Due to this size overhead images are often compressed using either lossy or lossless compression. Lossy compression as the name suggests, reduces the file size but at the expense of the image's integrity, JPEG (Joint Photographic Experts Group) format is a prime example of this, trading quality of image for level of compression.

Lossless compression is sometimes used to save space while maintaining the images integrity as it always reconstructs the image exactly as it was before compression.

In bitmap images this is usually Run Length Encoding, succinctly describes it as method consisting of the process of searching for repeated runs of a single symbol in an input stream, and replacing them by a single instance of the symbol and a run count. In the case of a bitmap image the symbol it hopes to find runs of are the concurrent 1s or 0s that represent color values.

The BMP file format, sometimes called bitmap or DIB file format (for device-independent bitmap), is an image file format used to store bitmap digital images, especially on Microsoft Windows and OS/2 operating systems.

2.2.2 Pixel Storage

In uncompressed BMP files, and many other bitmap file formats, image pixels are stored with a color depth of 1, 4, 8, 16, 24, or 32 bits per pixel. Images of 8 bits and fewer can be either grayscale or indexed color. An alpha channel (for transparency) may be stored in a separate file, where it is similar to a grayscale image, or in a fourth channels that converts 24-bit images to 32 bits per pixel.

Uncompressed bitmap files (such as BMP) are typically much larger than compressed (with any of various methods) image file formats for the same image. The bits representing the bitmap pixels are packed within rows. Depending on the color depth, a pixel in the picture will occupy at least *n*/8 bytes (*n* is the bit depth, since 1 byte equals 8 bits). The approximate size for a n-bit (2^n colors) BMP file in bytes can be calculated, including the effect of starting each word on a 32-bit word boundary, as:

$$rowsize = 4.\left\lfloor\frac{(n.width)+31}{32}\right\rfloor \tag{1}$$

where the floor function gives the highest integer that is less than or equal to the argument; that is, the number of 32-bit words needed to hold a row of *n*-bit pixels; this value multiplied by 4 gives the byte count.

$$filesize \approx 54 + 4 \times 2^n + rowsize \times hight \qquad (2)$$

height and width are given in pixels. In the above formula, 54 is the size of the headers in the popular Windows V3 BMP version (14-byte BMP file header plus 40-byte DIB V3 header); some other header versions will be larger or smaller than that, as described in tables below. And $4 \cdot 2^n$ is the size of the color palette; this size is an approximation, as the color palette size will be $3 \cdot 2^n$ bytes in the OS/2 V1 version, and some other versions may optionally define only the number of colors needed by the image, potentially fewer than 2^n. Only files with 8 or fewer bits per pixel use a palette; for 16-bit (or higher) bitmaps, omit the palette part from the size calculation:

$$filesize = 54 + rowsize \times hight \qquad (3)$$

2.2.3 Header Structure

Here we describe 54-bytes header structure with an example of a 2x2 Pixel, 24-Bit Bitmap.

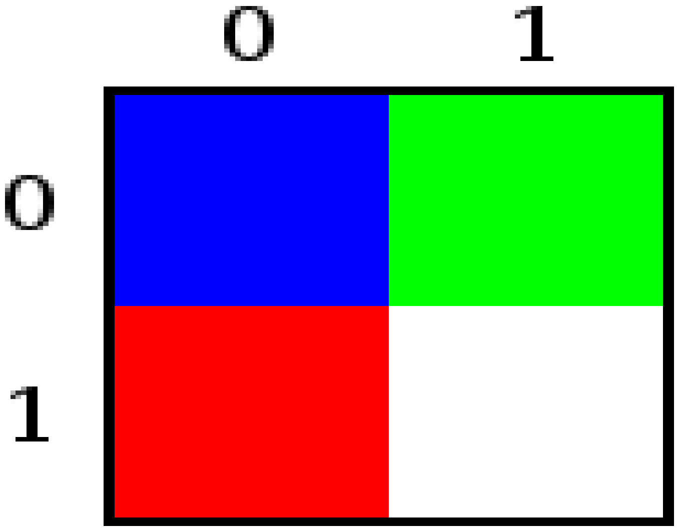


Figure 2.4: A 2 x 2 Pixel in a BMP Image

Table 2.2: BMP Header Structure

Offset	*Size*	*Hex Value*	*Value*	*Description*
0h	2	42 4D	"BM"	Magic Number (unsigned integer 66, 77)
2h	4	46 00 00 00	70 Bytes	Size of the BMP file
6h	2	00 00	Unused	Application Specific
8h	2	00 00	Unused	Application Specific
Ah	4	36 00 00 00	54 bytes	The offset where the bitmap data (pixels) can be found.
Eh	4	28 00 00 00	40 bytes	The number of bytes in the header (from this point).
12h	4	02 00 00 00	2 pixels	The width of the bitmap in pixels
16h	4	02 00 00 00	2 pixels	The height of the bitmap in pixels
1Ah	2	01 00	1 plane	Number of color planes being used.
1Ch	2	18 00	24 bits	The number of bits/pixel.
1Eh	4	00 00 00 00	0	BI_RGB, No compression used
22h	4	10 00 00 00	16 bytes	The size of the raw BMP data (after this header)
26h	4	13 0B 00 00	2,835 pixels/meter	The horizontal resolution of the image
2Ah	4	13 0B 00 00	2,835 pixels/meter	The vertical resolution of the image
2Eh	4	00 00 00 00	0 colors	Number of colors in the palette
32h	4	00 00 00 00	0 important colors	Means all colors are important
Start of Bitmap Data				
36h	3	00 00 FF	0 0 255	Red, Pixel (0,1)
39h	3	FF FF FF	255 255 255	White, Pixel (1,1)

3Ch	2	00 00	0 0	Padding for 4 byte alignment (Could be a value other than zero)
3Eh	3	FF 00 00	255 0 0	Blue, Pixel (0,0)
41h	3	00 FF 00	0 255 0	Green, Pixel (1,0)
44h	2	00 00	0 0	Padding for 4 byte alignment (Could be a value other than zero)

2.2.4 BMP Header Structure in Programming Concept

```
typedef unsigned char u8;
typedef unsigned short int u16;
typedef unsigned int u32;
typedef struct bmp_hdr {
        u16     signature;      // Magic Number (unsigned integer 66, 77)
        u32     file_size;      // Size of the BMP file
        u16     reserved1;      // Application Specific
        u16     reserved2;      // Application Specific
        u32     offset;         // The offset where the bitmap data (pixels) can be found.
        u32     info_hdr_size;  // The number of bytes in the header (from this point).
        u32     width;          // The width of the bitmap in pixels
        u32     height;         // The height of the bitmap in pixels
        u16     pannel;         // Number of color planes being used.
        u16     bpp;            // The number of bits/pixel.
        u32     compression;  // BI_RGB, No compression used
        u32     size_of_image_data;     // The size of the raw BMP data (after this header)
        u32     hor_reso_per_meter;  // The horizontal resolution of the image
        u32     ver_reso_per_meter;  // The vertical resolution of the image
        u32     colors_in_palette;      // Number of colors in the palette
        u32     imp_colors_used;        // Means all colors are important
}BMP_HEADER;
```

2.3 Text File (.txt File)

2.3.1 Format of the Digital Text (.txt) File

Text can be of various types: plaintext-consisting of fixed sized characters having essentially the same type of appearance, formatted text-where appearance can be changed using font parameters, and hypertext-which can serve to link different electronic documents and enable the user to jump from one to the other in a non-linear way. Internally text is represented via binary codes as per the ASCII table. The ASCII table is however quite limited in its scope and a new standard has been developed to eventually replace the ASCII standard. This standard is called the Unicode standard and is capable of representing international characters from various languages throughout the world. Text can be inserted into an application using various means. The simplest way is directly typing text into the application by using the keyboard; alternatively text can be copied from another pre-existing file or application and pasted into the application. Nowadays we also generate text automatically from a scanned version of a paper document or image using an Optical Character Recognition (OCR) software. When text is saved onto the hard disk, it can be compressed using various algorithms so as to reduce the file size. All algorithms however works in a lossless mode, i.e. all information in the original file is maintained intact without loss of data in any way. Depending on how various visual properties of text are stored and the compression scheme followed, text can be stored into a number of file formats each requiring its own specific application to open and modify the contents.

Now we will discuss briefly about different types of text. There are:-

2.3.1.1 Unformatted Text

Also known as plaintext, this comprise of fixed sized characters from a limited character set. The character set is called ASCII table which is short for American Standard Code for Information Interchange and is one of the most widely used character sets. It basically consists of a table where each character is represented by a unique 7-bit binary code. This means there are 2^7 or 128 code words which can be used to identify the characters. The characters include a to z, A to Z, 0 to 9, and other punctuation characters like parenthesis, ampersand, single and double quotes, mathematical operators, etc. All the characters are of the same height. In addition to normal alphabetic, numeric and punctuation characters, collectively called printable characters, the ASCII character set also includes a number of control characters. These include BS (backspace), LF (linefeed), CR (carriage return), SP (space), DEL (delete), ESC (escape), FF (form feed) and others. Whenever characters include in the table are required to be stored or processed by a computer, the corresponding numeric code is retrieved from the ASCII table in binary form and substituted for the actual text for internal processing. This includes both the printable and control characters, e.g each line of text in a document is terminated by al linefeed character.

2.3.1.2 Formatted Text

Formatted text are those where apart from the actual alphanumeric characters, other control characters are used to change the appearance of the characters, e.g. bold, underline, italics, varying shapes, size and colors etc. Most text processing software used such formatting option to change text appearance. It is also extensively used in the publishing sector for the preparation of papers, books, magazines, journals, and so on. In addition a variety of document formatting options are supported to enable an author to structure a document in chapters, sections and paragraphs, and with tables and graphics inserted at appropriate points. The control characters used to format the text is application dependent and may vary from one package to another, e.g. bold appearance in an MS-Word document may be produced by a different control code than that in an HTML document. To print such a document, the printer should also be capable of interpreting these control codes so that the appropriate appearance may be reproduced.

2.3.1.3 Hypertext

Documents provide a method of transmitting information. Reading a document is an act of reconstruction knowledge. A book or an article on paper has a given structure and is represented in a sequential form. Although it is possible to read individual paragraphs without reading previous paragraphs, authors mostly assume sequential reading. Novels as well as movies always assume a pure sequential reception. Technical documentation (e.g. manuals) consists often of a collection of relatively independent information units. There also exits many cross-references in such documentation which lead to multiple searches at different places for the reader. A hypertext document can be used to handle such situations. The term hyper is usually used to mean something excessive (beyond super), e.g. hyperactive, hypertension, etc. Here the term is used to mean certain extra capabilities imparted to normal or standard text. Like normal text, a hypertext document can be used to reconstruct knowledge through sequential reading but additionally it can be used to link multiple documents in such a way that the user can navigate non-sequentially from one document to the other for cross-references. These links are call hyperlinks. Typically hyperlinks take the form of an underlined text string and the user initiates the access and display of a particular document by pointing and clicking the mouse on the appropriate link. The underlined text string on which the user clicks the mouse is called an anchor and the document which opens as a result of clicking is called the target document.

Chapter 3

Literature Survey

Encryption is the conversion of data into a secret code usable over a general network. The intelligible form (original data) of information is called plain text/image and the unintelligible form (protected data) is called cipher text/cipher image. The process of converting the plain text/image into cipher text/ image is called encryption, while the invert process of transforming cipher text/ image into the corresponding plain text/image is called decryption. Because of the rapid development of the internet and innovation in technologies, the security of multimedia data especially during it's transmission over the internet is an important issue. Authorized people can send and receive information from a distance using computer networks. To be secured, information needs to be hidden from unauthorized access (confidentiality), protected from unauthorized change (integrity), and available to an authorized entity, when it is needed (availability). Media data types such as image, audio, video, text and animation sequences are included in the multimedia data. Text and image are treated as static media where audio and video are treated as dynamic media.

In general, most encryption algorithms use a secret value called a key. The security of encrypted data entirely depends on two things: the strength of the encryption algorithm and the secrecy of the key. The key is used for encryption and decryption and it must be kept secret, thereby requiring the sender and receiver to agree on the same key before making any data transmissions. The key is independent of the plain text/plain image. Therefore, the same plain text/image is encrypted to different cipher text/ image with different keys. Thus both processes are impossible to be fulfilled without the use of the correct key. Encryption can be strong or weak. Encryption strength is measured by the time and resources it would require to recover the plain text/plain image. The result of powerful encryption is cipher text/ image that is very difficult to decipher without possession of the appropriate decoding tool. The sender and the receiver must keep the key secret because anyone who knows the key can use it to encrypt the plain text/image. In addition, the strength of the algorithm is important. An unauthorized entity can take encrypted cipher text/image and endeavor to break the encryption by determining the key based on the cipher text/image. While cryptography is the science of securing data, crypt-analysis is the science of analyzing and breaking secure communication, and therefore it is the process of recovering the plain text/image or key, usually by using the cipher text/image and knowledge of the algorithm.

3.1 Different Audio Encryption Schemes

Moreover, the data content of dynamic media is larger than that of static media. Therefore, audio encryption is completely different from static media data encryption such as text encryption and image encryption. In audio data negative values are present which can be lost after some transformations and that is also an issue of complexity in audio data encryption compared to text and documented data encryption. Most of the data encryption algorithm when applied to audio data resulted in low conversion efficiency. Hence there is a need for designing faster and secured encryption algorithms for audio data and this is the booming area for researchers. The security of the audio data is achieved by various techniques like steganography, cryptography and other techniques. Steganography involves hiding of the secret information within the other information and Cryptography involves application of mathematical transformations on the secret information in order to protect the data over the insecure channel while transmission.

Now we will discuss about the comparative study among existing conventional encryption techniques based on their performances in terms of key size, block size, cipher type, keys, attacks, vulnerability and security. When most of the conventional text data encryption algorithms [3] such as DES, 3DES, RC2, RC4, RC6, Blowfish and AES are applied to audio data have not generated satisfactory results. The main drawback in the existing audio encryption algorithms is its tradeoff between real time processing and security [4]. We will also discuss about the recent techniques which are based on the encryption and stenographic techniques proposed to enhance the security of the audio data.

Comparative study among conventional encryption standards based upon different factors discuss in table-3.1. To enhance the security level, we use some extra technology along with traditional encryption algorithm which are specified in table-3.1. These securities enhancing technologies have specified in next section.

Table-3.1 (I-II): Comparative Study among Conventional Encryption Standards [5]

(I)

Factors	***DES***	***3DES***	***RC2***	***RC4***
Key Size	56 Bits	168 Bits	8-128 Bits	40-128 Bits
Block size	64 Bits	64 Bits	64 Bits	Byte Oriented
Cipher Type	Block Cipher	Block Cipher	Block Cipher	Stream Cipher
Keys	Private Key	Private Key	Single Key	Single Key
Attacks	Vulnerable to Differential and Linear Attacks	Vulnerable to Differential, Brute Force Attacks	Vulnerable to Differential, Brute Force Attacks	Vulnerable to Brute Force Attacks
Security	Proven Inadequate	Inadequate	Vulnerable	Weak Security

(II)

Factors	*RC6*	*Blowfish*	*AES*
Key Size	128,192 or 256 Bits	32-448 Bits	128,192 or 256 Bits
Block size	128 Bits	64 Bits	128,192 or 256 Bits
Cipher Type	Symmetric Algorithm	Symmetric Block Cipher	Symmetric Cipher Algorithm
Keys	Single Key	Private Key	Private Key
Attacks	Vulnerable to Differential, Brute Force Attacks	Vulnerable to Differential, Brute Force Attacks	Strong Against Differential, Linear, Brute Force Attacks
Security	Vulnerable	Less Secure	Considered Secure

3.1.1 Higher Dimensional Chaos Based Audio Encryption [6]

This is a good property like mixing and sensitive dependence of initial conditions which are chaotic in nature and control parameters are exploited .Variables are used as audio encryption keys. Chaos –based look up table are used for encrypting the audio files. A lookup table is constructed using a higher dimension cat map. Cipher block chaining architecture is used. Chaotic based cryptosystems are suitable for large scale data encryption i.e., audio, image etc. Characteristics of this algorithm are: the key space and security is increased by this algorithm. It is sensitive to initial parameters, pixel distribution uniformity, resilient to chosen/plain text attacks. Higher the dimension of the chaotic map, more is the confusion and hence more secured.

Though this cryptosystem provides security, security is function of dimension of the map. As the security requirement increases the memory requirement also increases in the same ratio hence increasing computational complexity of the systems. This methodology may not be suitable to implement in mobile network as memory and computation time are basic constraints.

3.1.2 Digital Audio Watermarking Based Audio Encryption [7]

Watermark is a secure link readable only by authorized persons with the knowledge about the secret. Watermarking techniques involves embedding information within another data in order to provide copyright protection. Only authorized person with the secret key can extract the secret information. Watermarking techniques can be categorized based on

1. Type of key used i.e. symmetric or asymmetric.
2. Amount of information used to extract-blind, semi blind and non-blind.
3. System robustness, semi-fragile and fragile.

Researchers have come up with various test statistics and embedded functions. In [8], they propose a combined approach of chaotic encryption and fragile image watermarking for secrecy and intrusion control and detection respectively. Here they hide the encrypted audio track in a 2D image to have more security and complexity in tattooing. This security algorithm is more secured and requires less computational time. It prevents plaintext attacks. It uses masking techniques and permutation dependent keys. The watermarking helps in intrusion detection and control but does not protect against eavesdropping, complex in tattooing.

3.1.3 Biometric Cryptography Based Audio Encryption [9]

Biometric cryptography is a method using biometric features to encrypt original data. Statistically analyzing these biological characteristics has become known as the science of biometrics. This method can improve the security of the encrypted data. The most accurate feature iris is used [15]. Iris image is converted to binary code to form the secret key. This key is used to encrypt the data which are audio signals. At the decryption phase same key is used to generate the original data. Conventional cryptosystems lack in authentication, this method provides the ultimate authentication through biometrics.

The advantages authentication may become difficult if pattern gets corrupted due to reflections to eyes, if isolation is not done properly because the upper and lower part of the iris is covered by eyelids and eyelashes. Limited combination of binary output i.e., iris binary key output can be exploited by brute force attack; if key is revealed in entire secret message.

3.1.4 Index Based Selective Audio Encryption [10]

This algorithm uses for modified discrete cosine transform (MDCT) based selective encryption scheme with resource allocation feature. MDCT audio index is used to determine the audio importance and energy efficient selective encryption takes place on the audio data. This particular scheme is energy efficient, has better encryption performance and good audio transmission quality.

3.1.5 Shuffling Based Audio Encryption [11]

This algorithm uses encryption scheme employing a shuffle of stream cipher. Audio file and keis taken as input and byte shuffling is done. Audio file is considered as a stream and encryption is dependent on both data and key. Brute force attack is impossible on encrypted audio file which are typically large. It is also resistive to statistical attacks. It is not suitable for low quality audio files as they will be prone to statistical attacks.

3.1.6 Perceptual Audio Coder (PAC) Based Audio Encryption [12]

This a popular algorithm, like MPEG's MP3 standard, used to compress digital audio by removing extraneous information. It provides efficient compression of high-quality audio over a variety of formats from 16 Kbit/s for a monophonic channel to 1024 Kbit/s for a 5.1 format with four or six auxiliary audio channels, and provisions for an ancillary (fixed rate) and auxiliary (variable rate) side data channel. Perceptual audio coders are used in many applications including Digital Radio and Television, Digital Sound on Film, Multimedia/Internet Audio, Portable Devices, and Electronic Music Distribution (EMD). Partial Encryption is one of most popular audio encryption technique.

3.1.7 Randomized Arithmetic Coding Based Audio Encryption [13]

In 2006, Grangetto et al. [13] proposed another novel security framework for multimedia data which based on randomized arithmetic coding. In this framework, multimedia data was encrypted by inserting randomized arithmetic coding procedure.

3.1.8 Scrambling Based Audio Encryption [14]

Progressive multimedia data such as audio scrambling in compressed domain was proposed by Yan et al. [14] in 2008, where a secret MP3 audio is scrambled with the help of a secret key before being transmitted. But this method is vulnerable to brute force attack and known plain text attack as discussed in [15].

3.1.9 Shared Cryptography Based Audio Encryption [16]

In the field of shared cryptography, some researchers have processed audio sharing separately so that it could be distributed amongst a qualified set of participants [16, 17].

3.1.10 Cosine Number Transform (CNT) Based Audio Encryption [18]

Lima and da Silva [18] had proposed an audio encryption scheme based on the cosine number transform (CNT). The scheme is recursively applied to a block of non-compressed audio data and the blocks are selected using a simple overlapping rule to provide diffusion of the ciphered data.

3.1.11 Chaotic Map Based Audio Encryption [19, 20, 21-26]

In literature, chaotic maps [19, 20] are widely used in multimedia encryption techniques because chaotic encryption algorithm must be secure in perception, have large key space, higher key sensitivity, and resistant to cryptanalytic attacks to the initial conditions. In the field of static multimedia data such as image, chaotic maps are extensively used to make the encryption schemes more secured [21-24]. Chaotic maps are also used for encrypting dynamic multimedia data such as audio. In 2011, Mosa et al. [25] proposed a speech encryption approach, which was based on the permutation of speech segments using chaotic Baker map and their substitution using masks in both time and transform domains. Sathiyamurthi et al. [26] have also proposed a speech encryption scheme using chaotic shift keying for secured speech communication, in which higher degree of security is achieved by multiple levels of permutation process on sampled speech.

3.1.12 DNA Encoding Based Audio Encryption [27-31]

In 1994, Adleman [27, 28] first introduced DNA computing into the field of encryption, which created a new era of information processing. DNA molecules have massive parallelism coupled with low energy consumption and high storage density [29]. This introduces some unique advantages for DNA-based encryption algorithms over traditional cryptographic algorithms. However, using only DNA encoding to encrypt images is not secure. Therefore, a combination of chaos encryption technology and secret data encryption based on DNA computing [30, 31] solves the existing hidden insecurity problems.

3.2 Different Image Encryption Schemes

Image encryption techniques try to convert original image to another image that is hard to understand and to keep the image confidential between users. In other words, it is important to clarify that without decryption key no one can access the content. Involvement of image in various fields, such as research field, defense, satellite communication, banking data and medical data transmission, photography etc. are the causes of necessity of encryption of image.

To get better security by encryption process, researchers have focused on methods that satisfy confusion and diffusion and use some extra technique since preliminary techniques, such as AES, DES, IDEA, and RSA, which are not efficient for proper encryption. Among these security enhancement technique, some are discuss in bellow:-

3.2.1 DNA Encoding Based Image Encryption [32 – 38]

One of the latest and most successful image encryption methods is DNA-based image encryption. DNA cryptography utilizes DNA sequences as information carrier and takes advantage of biological technology to achieve encryption. DNA-based image encryption methods can be generally divided into two stages. In the first stage, it transforms the plain-image into the DNA sequence matrices by the DNA encoding rules and generates the key streams. In the second stage, encryption is carried out using the key streams and DNA sequence operations and then it converts the encrypted DNA sequence matrices into the cipher-image according to the DNA decoding rules. Shyam et al. [32] developed a novel encryption scheme based on DNA computing and used the natural DNA sequences to encode the image. Zhang and Guo [33] created a new image encryption algorithm based on DNA sequence addition operation. Liu et al. [34] proposed RGB image encryption scheme adopting DNA encoding combined with logistic map. Wei et al. [35] suggested a colour image encryption method based on DNA and hyper chaotic system, where Chen's hyper-chaotic maps was employed to scramble the locations of DNA sequence matrices. Enayatifar et al. [36] suggested an image encryption algorithm based on a hybrid model of DNA masking, Genetic Algorithm (GA) and the logistic map. Guesmi et al. [37] have implemented hybrid image crypto system using DNA masking, secure hash algorithm and Lorenz system. The combination of DNA masking and Lorenz system has enhanced the strength of the security system and improved the information entropy. Liu et al. [38] introduced an image encryption method using DNA complementary rule and two chaotic maps with good stochastic property.

3.2.2. Chaotic Map Based Image Encryption [19, 39-51]

Chaotic maps are dynamical systems that are widely used to produce long range of unpredictable pseudo random sequences in any cryptography application. It has been used in image encryption to rearrange the image pixel in permutation phase and to modify the pixel positions in diffusion phase respectively. Chaos based encryption techniques have several advantages over the conventional encryption techniques in terms of speed, security and computational power. Typically chaos based image encryption can be classified into two categories: one dimension maps and multidimensional maps [39, 40].

Fridrich *et al*. [41] was the first to propose chaos based encryption scheme using two dimensional standard baker map. Pareek *et al*. [19] introduced an encryption system with one dimensional map and later fusion of one dimensional maps were carried out to obtain the sequence with enhanced chaotic behaviour. Patidar *et al*. [40] proposed substitution–diffusion based chaotic image cryptosystems. In this scheme, standard map was employed in both substitution–diffusion routine. Liu and Wang [41] presented an encryption technique with one time random key generation using chaotic map and Message Digest 5 (MD5) algorithm. Chen *et al*. [42] proposed a logistic map based image diffusion process along with 3D map for shuffling the pixel position. Liu *et al*. [45] developed a hyper chaotic system for colour image encryption using Piece Wise Linear Chaotic Map (PWLCM) and Choquet Fuzzy Integral (CFI). Ghebleh *et al*. [46] introduced a shuffling and masking structure based chaotic image scheme where 3D cat map was used to permute the equally divided plan image blocks and skew tent map was used to mask the shuffled image block. Zhang *et al*. [47] proposed an image encryption algorithm using chaotic skew tent map and permutation –diffusion architecture. This scheme generates P-box as the similar size of secret image using skew tent map, which permutes the positions of the image pixels. Zheng and Jin [48] proposed an image encryption method adopting henon map and compound spatiotemporal chaos to scramble and diffuse the image pixel.

Gao *et al*. [49] presented the chaotic image cipher using logistic map and combined Lorenz - Chen’s chaotic systems, where random sequence generated by logistic map was used to shuffle the image pixel. Combined Lorenz - Chen’s chaotic map key stream was used to perform the robust diffusion process. Wang *et al*. [50] proposed a chaotic image crypto system with two logistic maps. The first map was used to generate the control parameter for permutation and second was used to generate the keystream for diffusion stage. Seyedzadeh and Mirzakuchaki have implemented a scheme with interconnected constructs of two Combined Two-dimensional Piecewise Nonlinear Chaotic Map (CTPNCM) to encrypt images. In this scheme, a 256 bit large external secret key has been used to obtain the initial conditions and control parameter for CTPNCM [51].

3.2.3. Cellular Automata Based Image Encryption [52-56]

Cellular Automata (CA) is another kind of dynamical system that has excellent capacity to exhibit unpredictable and complex behaviour to build a robust image cryptosystem. Jian *et al*. [57] proposed permutation and substitution based image encryption, where 2-D Von Neumann CA was employed to perform the substitution process and permutation was accomplished by scan patterns. Rong *et al*. [58] presented an image encryption based on pixel value reallocation and the reallocated data was realized using recursive CA substitution and random key sequence. Jin *et al*. [34] proposed an encryption scheme based on elementary CA with 8 periodic boundary conditions. Periodic boundary reversible CA was adapted in diffusion stage to iterate the pixels. Tafti and Janosepah [54] proposed an image encryption algorithm in frequency domain using DCT and one dimensional CA. Here, higher frequencies of the image blocks are encrypted by one dimensional CA with the XOR local rule. Peng *et al*. [55] developed a hybrid image crypto system using coupled CA to encrypt an image in a parallel manner. Zamani *et al*. [56] introduced an image encryption using fuzzy CA and hyper chaotic systems where five one dimensional non – uniform fuzzy CA were employed in encryption stage.

3.2.4 Scan Based Image Encryption [57-60]

Scan is a formal language based two dimensional spatial methodology with massive varieties of Space Filling Curves (SFCs), which are used to permute the image pixels. There are different 9 scanning methods based on different application such as simple SCAN, extended SCAN and generalized SCAN, each of which has a specific set of scan patterns. Each different scannings has its own grammar and a set of basic scan patterns defined. However there are total fifteen scan patterns defined including all the scanning. Each basic scan patterns has a set of transformation and a set of laws to obtain complex scan pattern from basic ones if necessary. The laws for complex scan patterns from basics are defined by the production rules of the grammar of that specific scanning. The author presents lossless image encryption with compression methods based on scan patterns [57]. This scheme generates a wide range of scan patterns based on two-dimensional patterns and spatial retrieve method that can be used to encode the image signal. Bhatnagar *et al.* has proposed a SFC based selective encryption technique [58] in which SFC curve was used to scramble the image pixel positions. Further, non-linear chaotic maps were used to diffuse the image. The same author [59] has proposed an image encryption scheme in transform domain using Fractional Wavelet Transform (FrWT) and dual space-filling curves. FrWT is used to decompose the image into sub-band coefficients. Further, dual SFC is employed to shuffle the FrWT sub-band coefficients. Pareek *et al*. [60] has proposed a color image encryption scheme using zigzag scan pattern and 128 bit secret key is used. In this case, scan pattern is used to rearrange the pixels and the secret key is used to perform the substitution and mixing process respectively. Below in Figure 3.1 all the scan patterns have been shown.

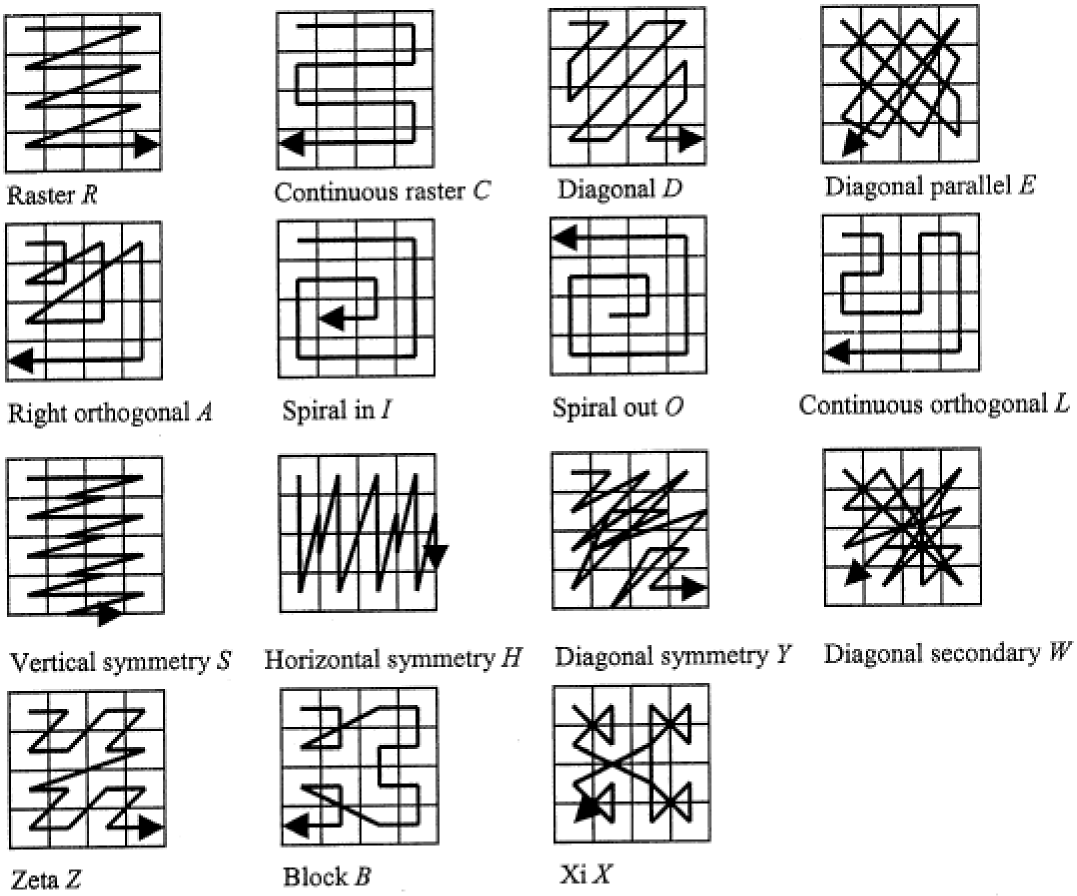


Figure 3.1: Basic Scan Pattern

3.2.5 Different Scrambling Algorithm for Image Encryption [61-68]

3.2.5.1 Sign Scrambling Algorithm Based Image Encryption [61-62]

Shi and Bhargava (1998 -1999) implemented a new method to encrypt the sign bit of every (Discrete Cosine Transform) DCT coefficient in JPEG [61] and every motion vector in MPEG [62]. This approach is really fast and easy to implement. It is robust to the compression because the sign bit is not changing with respect to different quantization tables. However it lacks security and affects compression efficiency as a result of encrypting DCT coefficients before run-length and Huffman coding. Since the sign bit could be viewed as the most significant bit of coefficients, the results of the sign encryption expected to be good. The experiments in [61] and [62] show that the image/video would be totally unrecognizable after sign bits are randomly altered. To see the results of the sign encryption algorithm, it is implemented on MATLAB. Sign encryption is applied to Lena image with *8x8* logical scrambling matrixes, which could only take *0 or 1* as a value. The results are not good as it is expected. Therefore, selectively encrypting sign bit of every DCT coefficient is not an effective way for image/video encryption. But, because of simplicity, it can be used with other encryption methods for higher security level.

3.2.5.2 Slice Scrambling Based Image Encryption [63]

It is another approach for image encryption. Wu and Kuo (2003) mentioned an encryption algorithm, which scrambles equal size of 8x8 DCT blocks with respect to a table No. 1 in [63]. This approach is also easy to implement and robust to the compression, since different quantization tables will only change coefficients of the block but not the arrangement of the blocks. However, it affects the compression efficiency because of placing uncorrelated DC coefficients next to each other and lacks security since one can change the block arrangement with respect to the correlation of the blocks. Moreover, the block arrangement permutation should be embedded to the key, which makes key sizes very high in small block cases. The contour is recognizable in the encrypted image when block size is large. It doesn’t seem very secure even visually. Moreover, the key size gets larger with small block sizes. On the other hand, the inner blocks are not encrypted. Therefore, the correlations of the originally near blocks are high. This information can be used to reconstruct original image. As a result, this method also can’t be used in practice.

3.2.5.3 DCT Coefficient Scrambling Based Image Encryption [64-65]

Tang (1996) suggested shuffling the DCT coefficients within an 8x8 block for JPEG/MPEG based transmission system [64]. This technique is fast and simple to implement. However, it changes the statistical property (run-length characteristic) of DCT coefficients. As a result, it may increase the bitrate drastically. But it lacks security. No contour is recognizable in the encrypted image. Visually, it is very secure, but since only inner blocks are encrypted, it has been shown in [65] that this chipper is vulnerable to frequency-based attack that exploits the property of none-zero coefficients that have the tendency to appear earlier in the zigzag order. In addition, it may increase the bitrate of the compressed video by as much as 50% as reported in [64] and this approach also may not be amenable to secure bitrate conversion. Therefore, this method also can't be used in practice.

3.2.5.4 Line Scrambling Based Image Encryption [66-68]

There are several video scrambling systems [66, 67] which rely on methods of directly distorting the image in the spatial domain. These scrambling techniques are not efficient for transmitting digital video signals because they, in general, will significantly chance the statistical property of the original video signal [68].

One of the spatial domain scrambling approaches is line scrambling which scrambles the lines of the image to experiment spatial domain scrambling effects, line scrambling algorithm is implemented on MATLAB. A permutation of the sequence from 1 to image height is used to rearrange the image lines No contour is recognizable in the encrypted image. Visually it is very secure. But the file size of the scrambled image is 30% more than original. Therefore, using of this method is not proper for the limited bandwidth systems.

3.2.6 Image Encryption Based On Different Transformations [69-71]

3.2.6.1 Fourier Transformation Based Image Encryption [69, 70]

Ran Tao et al (2010) introduced another technique for image encryption using multi-order fractional Fourier transform. In this technique, the encrypted image is obtained by the summation of different orders of Inverse Discrete Fractional Fourier Transform (IDFR FT) of the interpolated sub-images. The whole transform orders of the utilized FFT are used as the secret keys for the decryption of each sub-image. Compared with the traditional image encryption methods based on the FRFT, the method is with a larger key space and the amount of keys can be set as large as two times the amount of the pixels in the original image. In future work, one can also combine the proposed method with other image encryption methods to enhance the security of the system [69]. Zhengjun Liu et al (2011) proposed an image encryption algorithm based on fractional Fourier transform. A local random phase encoding is introduced into this algorithm. The data at the local area of complex function is converted by fractional Fourier transform [70].

3.2.6.2 Wavelet Transformation Based Image Encryption [71]

Zhu Yu et al (2010) proposed Chaos-Based image encryption algorithm using Wavelet Transform. Algorithm uses the wavelet decomposition concentrating image information in the high-frequency sub-band and then encryption is applied for the sub-band image. After that a wavelet reconstruction is introduced in order to spread the encrypted part throughout the whole image. A second encryption process is used to complete the encryption process. Theoretical analysis and experimental results show that this algorithm has an obvious increase in efficiency as well as satisfied security [71].

3.2.7 Different Image Encryption Based on Soft Computing

3.2.7.1 Genetic Algorithm Based Image Encryption [72-79]

Ankita Agarwal (2012) established a new approach of Genetic Algorithm in which the operations of GA (Crossover and Mutation) are used to produce this encryption method. This new method was applied to the candidate’s type of data i.e. images [72]. Sandeep Bhowmik (2011) tried to analyze the application of GA for image security using a combination of block-based image transformation and encryption techniques. The cases show that the correlation among pixels decreased when the proposed algorithm was applied to images [73]. Jalesh Kumar and S. Nirmala (2012) implemented a new technique to encrypt image,which utilizes selection crossover and mutation operations. This technique based on genetic algorithm, comprises three stages; the first stage deals with the selection of key sequence. In this stage, linear congruential pseudo random generator is used for generating key sequence. The second stage is deals with the crossover operation. The third stage deals with the mutation operation on the result obtained from the previous stage. This method combines transposition and substitution methods to encrypt the data [74]. Aarti Soni and Suyash Agrawal (2013) presented a new method based on genetic algorithm which is used to generate a key by the help of pseudo random number generator. Random number will be generated on the basis of current time of the system. Using Genetic Algorithm can keep the strength of the key good and make the whole algorithm good enough. Symmetric key algorithm AES has been proposed for encrypting the image as it is very secure method for symmetric key encryption. This algorithm increased the efficiency of the algorithm in terms of computation time required and complexity to attack the message. It uses the concept of pseudo random number generator and genetic algorithms to increase the complexity of key by increasing the irregularity of the key. Implementation of Genetic Algorithm with PRNG has a very complex key which is very difficult for cryptanalyst to attack. The AES symmetric key encryption algorithm will provide an efficient method for encrypting image and increasing the overall efficiency of the system [75]. Abdelsalam et al (2010) propose a new approach for e-security applications using the concept of genetic algorithms with pseudorandom sequence to encrypt and decrypt data stream. The feature of such an approach includes high data security and high feasibility for easy integration with commercial multimedia transmission applications. An experiment testing feasibility is reported in which several images are encrypted and decrypted. The experimental results show that the proposed technique achieved high throughput rate that is fast enough for real time data protection. In this technique used the concept of genetic algorithms in cryptography along with the randomness properties of Non-Linear Feedback Function Shift Register (NLFFSR).This total way of transferring secret information is highly safe and reliable. So, without the knowledge of the pseudorandom sequence no one will be able to extract the message. Since the NLFFSR pseudorandom binary sequence is unpredictable it is very difficult to decrypt correctly an encrypted signal by making an exhaustive search without knowing the initial value and the feedback function f and nonlinear output function [76]. Srikanth et al (2010) suggested a new algorithm to encrypt an image by using genetic algorithm. In this algorithm first the image is broken down into blocks Then the initial transformation steps are performed after that the functions similar to Vernon cipher are used to locate the pixels and further genetic algorithm is used to encrypt the images using one point cross-over. This algorithm can be classified as comparatively, more efficient and less complex when it is compared to the existing spatial domain techniques [77].Ahmed

Mahamood et al (2013) presented a novel efficient symmetric encryption technique which can be applied to medical images. It uses genetic algorithm to become highly adaptive. Standard Digital Imaging and Communications in Medicine (DICOM) images are segmented into number of blocks based on pixels intensity and entropy measurements. The novelty of this technique is variable key length which controls the relationship between processing time and resulting robustness. Using an evolutionary based technique in the form of genetic algorithms creates an adaptive optimized method that controls the processing time by applying five adjusting parameters. These parameters are: encryption algorithms, key-length, robustness parameter (CORR, NPCR), number of regions, and side information [78]. Dr. Dilbag et al (2013) designed a new algorithm using the concept of genetic algorithms. This algorithm enhances the quality, efficiency and effectiveness of the algorithm by using cryptography. Genetic algorithms are used to find an optimized solution within minimum possible time [79].

3.2.7.2 Fuzzy Sets Theory Based Image Encryption [80-83]

Said E. El- Khamy et al (2005) introduced a new color image encryption system. This system uses a Fuzzy Bit Generator (FBG) which was developed by the researcher. Moreover by using a fuzzy PN bit generator generated four binary sequences which are used to encrypt image pixels. According to these sequences, each pixel color byte value rotated as bits in right or left direction for some bits, after rotation process the pixel color value XOR- ed by one of binary sequences. According to image dimensions, the bits of S_0, S_1 are used to get the coordinate of the upper left corner of a 4×4 pixel blocks B_i of the image under encryption x_i and y_i. S_2, S_3 are used to get the coordinates x_{i+1} and y_{i+1} for B_{i+1}. Having theses randomly chosen blocks, the variance $\sigma_i^2 and\ \sigma_{i+1}^2$are calculated. The value of the variance was used in the permutation scheme [80].

Srinivasa et al (2012) proved a fuzzy logic method to fuse images from different sensors, in order to enhance the quality of proposed method. Along with quality evaluation parameters: image quality index (IQI), mutual information measure (MIM), root mean square error (RMSE), peak signal to noise ratio (PSNR), fusion factor (FF), fusion symmetry (FS) and fusion index (FI) and entropy. In this method, the potentials of pixel level image fusion using fuzzy logic approach have been explored along with quality evaluating measures. Fused images are primarily used to human observers for viewing or interpretation, and to be further processed by a computer using different image processing techniques [81].

Maneckshaw and Krishna Kumar (2013) proposed a novel Image encryption algorithm based on multiple fuzzy graph (FG) mapping technique. The Fuzzy graphs are obtained from a matrix of size *n* and then they are used to encrypt an image. They are discussed the fuzzy graphs with triangular and sigmoid membership functions. They are obtained a desired graph with the help of sigmoid function from a matrix by considering the vertices as the entries of the matrix and connecting edges between them whenever they are adjacent. To encrypt the image using the technique of multiple FG mapping , they are undergo the process of image encryption by dividing it into three stages – shuffling, mapping and encryption in order to have a greater security levels. This method has a larger key space, i.e., the randomly obtained pixel values which are combined with the pixels of the original image. These pixels are influenced by the factors such as the chosen matrix, the size of the matrix, the types of graphs obtained, the types of membership functions chosen, the patterns of fuzzy graphs obtained and the blocks they are plotted. This technique provides a multiple levels of securities for the image encryption. [82].

Shreyamsha Kumar and Chidamber Patil (2010) designed a new scheme for encrypt JPEG image. In the designed scheme the modified DCT blocks are confused by a fuzzy PN sequence. In addition to that, the DCT coefficients of each modified DCT block are converted to unique uncorrelated symbols, which are confused by another fuzzy PN sequence. Finally, the variable length encoded bits are encrypted by chaotic stream cipher. An amalgamation of all the three techniques with random combination of seeds will provide the required security against the casual listener/observer where the security needed is only in-terms of few hours. An image encryption algorithm that works co-operatively with JPEG compression has been proposed to meet three major requirements: (i) to provide temporal security against casual observer, (ii) to preserve the compression ratio, (iii) remain compliant with the JPEG file format. The experimental results of this scheme show that the encryption scheme provides the required security along with JPEG compression. Further, it preserves the compression ratio and the JPEG file format. All these advantages make it suitable for secure image coding and tactical communication [83].

3.2.7.3 Neural Networks Based Image Encryption [84, 85]

Hüsamettin UYSAL and Sinem KURT (2012) designed automatic system based on artificial neural networks for decrypting an image automatically without knowing what the decoder. In this study, decryption through MLP (Multilayer Perceptron) and RBF (Radial Basis Function) networks was tested using the interface which was designed in Matlab GUI and the image was shown and saved with minimum error by calculating the error rates of decrypted images. They are determined the differences between RBF and MLP. RBF was better than MLP with shorter processing time and less error rate. The main difference of this study from other related ones is using ANN for encoding and decoding. The procedure of this system consists of decoding the encrypted images by ANN to obtaining the original image. As a result, it is succeeded to obtain almost the same image of the original image easily by ANN [84]. Ismail I. A. et al (2012) proposed an algorithm to encrypt and decrypt satellite image by using neural networks back propagation. The goal of proposed algorithm to investigate the applicability of a back-propagation artificial neural network on the encryption of huge-sized satellite images. The used network is of *N* x *M* x *N* neurons representing the input, hidden, and output layers, respectively. The network is trained by adjusting the weights while the bias is given a constant value between 0 and 1 after normalizing the values. A bias is determined. The bias between the input layer and the hidden layer works as the first key *(K1)*, while the bias between the hidden layer and the output layer represents a second key *(K2)*. The training method uses *K1*, K2, or both and is done using images of small sizes to improve speed. Then, the network is used to encrypt and decrypt normal satellite images. Numerous trials were done for different satellite optical and SAR images and the goodness of fit (quality of decryption) between the original images and the decrypted ones was at least 98%, even for the images that the network was not previously trained to decrypt. It was also found that the network is not affected by geometrical image distortions like translation, size, and rotation [85].

3.2.7.4 Hybrid Algorithms Based Image Encryption [86-90]

Rasul Enayatifar and Abdul Hanan Abdullah proposed a new method based on a hybrid model composed of a genetic algorithm and a chaotic function for image encryption. In their method, firstly, a number of encrypted images are constructed using the original image with the help of the chaotic function. Secondly, these encrypted images are employed as the initial population for starting the operation of the genetic algorithm. Thirdly, the genetic algorithm is used to optimize the encrypted images as much as possible. Finally, the best cipher-image is chosen as the final image encryption [86].

Anil Kumar and Ghose and M. K. Ghose(2009) proposed a new approach of genetic algorithms (GA) with pseudorandom sequence to encrypt data stream. For transmitting the secured data over the channel there is the requirement of the high throughput. In these cases, the conventional encryption techniques are not a feasible solution for this reason a high throughput and secure encryption technique is proposed for real time data transmission like over the telephone link or video transmission. The concept of Genetic Algorithms used along with the randomness properties of chaos. This total way of transferring secret information is highly safe and reliable. The simulation results have indicated that the encryption results are (1) completely chaotic by the sense of sight, (2) very sensitive to the parameter fluctuation. The experimental results of the proposed approach confirm that high throughput rate needed for real time data protection is achieved [87].

Shubhangini et al (2013) suggested a novel method for image encryption by using chaotic function and genetic algorithm. This method, firstly images are encrypted using chaotic function and encryption key. Secondly genetic algorithm is used for optimization in which the best cipher image is selected. Finally the best cipher Images are selected based on correlation coefficient and entropy [88].

Nooshin B. et al (2012) presented a novel image encryption/decryption algorithm based on chaotic neural network (CNN). The employed CNN is comprised of two 3-neuron layers called chaotic neuron layer (CNL) and permutation neuron layer (PNL). The values of three RGB (Red, Green and Blue) color components of image constitute inputs of the CNN and three encoded streams are the network outputs. CNL is a chaotic layer where, three well-known chaotic systems i.e. Chua, Lorenz and Lü systems participate in generating weights and biases matrices of this layer corresponding to each pixel RGB features. Besides, a chaotic tent map is employed as the activation function of this layer, and makes the relationship between the plain image and cipher image nonlinear. The output of CNL, i.e. the diffused information, is the input of PNL, where three-dimensional permutation is applied to the diffused information. The overall process is repeated several times to make the encryption process more robust and complex. A 160-bit-long authentication code has been used to generate the initial conditions and the parameters of the CNL and PNL. Some security analysis are given to demonstrate that the key space of the new algorithm is large enough to make brute-force attacks infeasible and simulations have been carried out with detailed numerical analysis, demonstrating the high security of the new image encryption scheme [89].

Mouad et al (2011) presented a digital image encryption algorithm based on chaotic logistic maps and using fuzzy logic. The main idea of this algorithm is the usage of a fuzzy logic set of rules to control the next iteration of our proposed iterative mechanism using a set of logistic maps. The introduction of the fuzzy controller helped to use a set of logistic maps instead of one logistic map and therefore increased the randomness of the generated inputs [90].

3.3 Text File

Text Encryption have huge importance in our daily life, as similar as image and audio encryption, because it is related with various cases, such as: banking transactions, computer passwords, and e–commerce. Some related work on text encryption describe in bellow:-

3.3.1 Text Encryption Using Graph Theory

3.3.1.1 Complete Graph Based Text Encryption [91]

In the proposed algorithm by Wael Mahmoud Al Etaiwi [91], a text–graph is built from the plain text by adding a new vertex for each character in the plain text, and adding a new edge between each two sequential characters in the plain text. Each added edge is weighted by calculating the difference between the connected vertices' indexes in the predefined lookup table, which contains the character indexes in the language. For example, $\mathbf{E_{ac}}$ that connect vertex **a** with vertex **c** is weighted by calculating the deference between **c** and **a** indexes in the lookup table. When text–graph is built, it will converted to complete graph by adding missing edges with a sequential weights. The adjacency matrix of the resulted complete graph called **M1**. A minimum spanning tree algorithm is applied to form **M2** adjacency matrix. The final cipher text consists of **M1** and the results of multiplying **M1** with **M2** and a secret matrix key.

3.3.1.2 Spanning Tree Based text Encryption [92]

Ravinath et al [92] used a spanning tree to encrypt networks' packets in ad hoc networks, the proposed algorithm provides a privacy mechanism in ad hoc networks and reduces overhead in encryption by encrypt selected data packets. In the proposed algorithm, the minimum spanning tree were calculated using Prim algorithm, then each node need to know it's neighbor node key in order to communicated with, in order to exchange keys between each adjacent nodes, RSA key exchange mechanism were used.

3.3.2 Elliptic Curve Cryptography Based Text Encryption [93]

In 2015 Sing et al. described a faster method to encrypted and decrypted text data using Elliptic Curve Cryptography (ECC). The ECC was used in the process of transforming plaintext into coordinates to be encrypted with the help of ASCII code.

Chapter 4

Preliminaries

4.1 Chaos Theory and the Logistic Map

4.1.1 Chaos Theory

Chaos theory is a branch of mathematics that deals with nonlinear dynamical systems. *Nonlinear* means that due to feedback or multiplicative effects between the components, the whole becomes something greater than just adding up the individual parts. Lastly, *dynamical* means the system changes over time based on its current state. Chaos or chaotic system for short is an intervention between rigid regularity and unpredictability based on probability (Figure 4.1) [94]. Chaos can be defined by some special characteristics [95].

- *Nonlinearity*: Nonlinearity means that the change in an element at an initial time can escort to a change in the same or a different element at a later time, that is not depend to the change at the initial time.

- *Determinism*: It has not probabilistic (deterministic) which is governed by exact and correct rules with none of the element of chance.

- *Sensitivity to initial condition*: It means, negligible changes in its initial state can generate fully different final state.

- *Irregularity*: it means “order in disorder”.

- *Long term prediction*: chaos gives uncontrolled long term prediction due to sensitivity to initial conditions.

- *The logistic map*: the chaotic function uses logistic map. The map is one dimensional so it gives scalars for the encryption process.

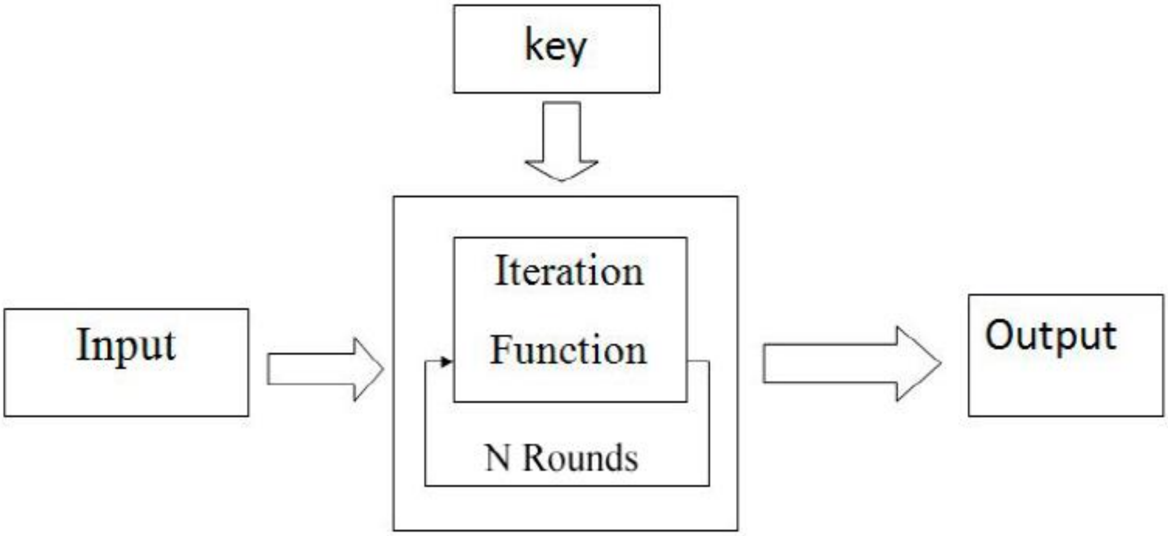


Figure 4.1: Chaos Iterative Function

4.1.2 Application Areas of Chaos [95]

Historically, the chaos is used in mathematics and physics in starting. It prolonged into engineering and more recently into information and social science. A few years ago there has been rising interest in commercial and industrial applications of chaotic systems. There are several types of latent commercial and industrial applications based on different aspects of chaos based system which are shown in Table 4.1 [95].

Table 4.1. Chaos Based Applications

Category	*Applications*
Control	Control of irregular behavior in devices and systems.
Synthesis	Potential control of epilepsy, improved hesitant of systems, such as ring laser gyroscopes. Switching of packets in computer networks.
Synchronization	Secure communications, chaotic broad band radio, and encryption.
Information Processing	Encoding, decoding, and storage of information in chaotic systems, such as memory elements and circuits. Better performance of neural networks. Pattern recognition.
Short Term Prediction	Contagious diseases, weather, economy.
Engineering	Vibration control, stabilization of circuits, chemical reactions, turbines, power grids, lasers, fluidized beds, combustion, and many more.
Computers	Switching of packets in computer networks. Encryption. Control of chaos in robotic systems.
Communications	Information compression and storage. Computer network design and management.
Medicine and Biology	Cardiology, heart rhythm (EEG) analysis, Prediction and control of irregular heart activity (chaos-aware defibrillator).
Management and Finance	Economic forecasting, restructuring, financial analysis, and market prediction and intervention.
Consumer Electronics	Washing machines, dishwashers, air conditioners, heaters, mixers.

4.1.3 Chaos and Cryptography [96]

Chaos and cryptography share some similar characteristics shown in Figure 4.2:

(1) Both chaotic map and encryption system are deterministic (not probable).

(2) Both are unpredictable and not simple. It any external observer which has not any knowledge of the algorithm and initial condition as key, cannot understand the random behavior of the system.

(3) A chaotic system is sensitive to initial condition means Small changes of any element can be fully changed the output. Cryptography is depending key based confusion and diffusion, *means* modification of one bit of plain text or key could change all bits of the cipher text with50% probability.

(4) The iterative chaotic system is topological transitive and cryptography is multi round transformation means Single chaotic map with iterative transformation.

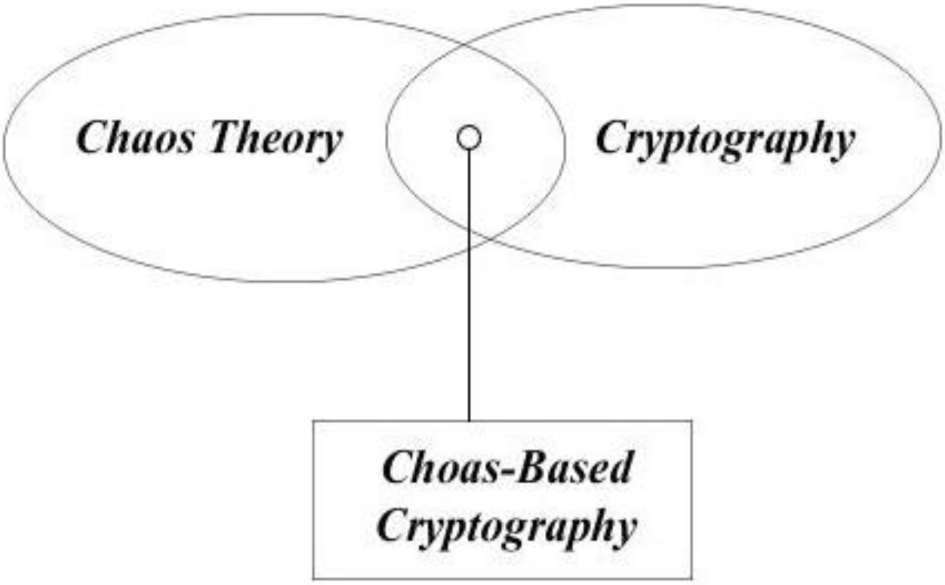


Figure 4.2. Relation between Chaos and Cryptography

4.1.4 Traditional Encryption and Chaos Based Encryption [96]

Chaos is also different from cryptography in some other features [96].

(1) Chaotic systems are based on real/complex number spaces (bounded continuous space) whereas cryptography defined binary sequences (finite discrete space).

(2) Chaos theory is providing the idea to understand the asymptotic behavior of iterative processes whereas cryptography defined the characteristics of first a few iterations.

Table 4.3. Comparison between Chaos Based and Traditional Cryptosystem

Chaos Based Cryptosystem	***Traditional Cryptosystem***
Floating point arithmetic	Integer arithmetic
Slow computation	Fast computation
Based on any nonlinear function	Usually based on the mod function
Does not necessitate prime numbers	Usually based on prime numbers
Low cycle length	High cycle length
Statistical bias	No statistical bias
Data superfluous	Data companionable
Chaos Theory	***Cryptography***
Chaos based system	Pseudo-chaos based system
Indiscriminate transformation	Indiscriminate transformation
Infinite number of stages	Finite number of stages
Infinite number of repetitions	Finite number of repetitions
Initial stage	Plain text
Final stage	Cipher text
Initial situation and/or parameters	Keys
Asymptotic autonomy of initial and final stages	Confusion

4.1.5 Logistic Map

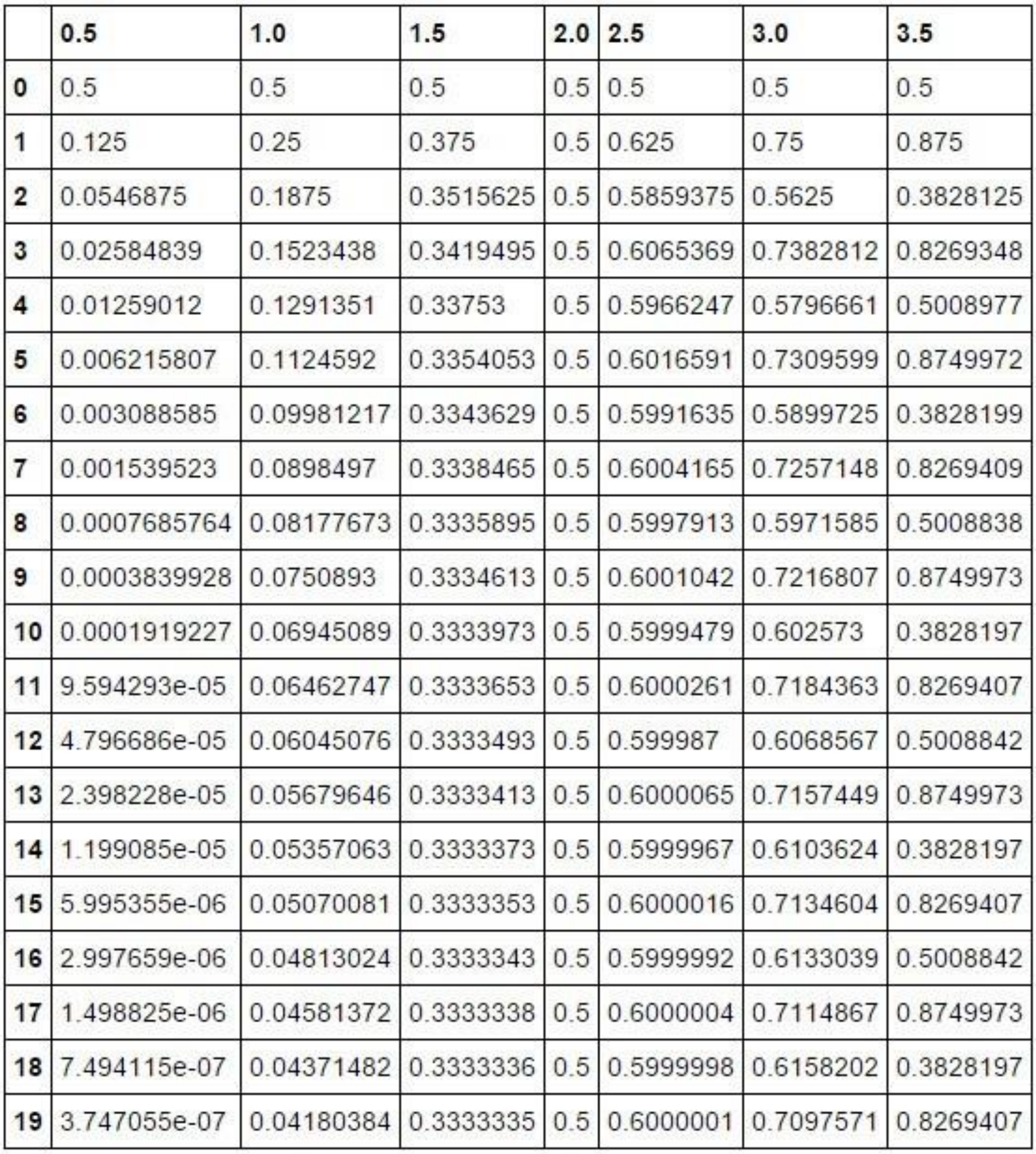
Logistic map [19, 20, 21, 31and 97] is one of the very popular simple chaotic map, which is used in cryptographic field. The logistic map uses a nonlinear difference equation to look at discrete time steps. It’s called the logistic *map* because it maps the population value at any time step to its value at the next time step. The mathematical form of logistic map is shown in equation-4.

$$\left.\begin{array}{c} f(x) = r\times x\times(1-x) \\ x_{n+1}=f(x_n) \end{array}\right\} \tag{4}$$

In the equation *x* represents the population at any given time n, and *r* represents the growth rate. In other words, the population level at any given time is a function of the growth rate parameter and the previous time step’s population level. If the growth rate is set too low, the population will die out and go extinct. Higher growth rates might settle toward a stable value or fluctuate across a series of population booms and busts.

Following table shows 20 time steps for growth rate parameters of 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, and 3.5. The values are:

Table 4.4: Logistic Model’s Value (Population Value) for Different Growth Rate Parameter



	0.5	1.0	1.5	2.0	2.5	3.0	3.5
0	0.5	0.5	0.5	0.5	0.5	0.5	0.5
1	0.125	0.25	0.375	0.5	0.625	0.75	0.875
2	0.0546875	0.1875	0.3515625	0.5	0.5859375	0.5625	0.3828125
3	0.02584839	0.1523438	0.3419495	0.5	0.6065369	0.7382812	0.8269348
4	0.01259012	0.1291351	0.33753	0.5	0.5966247	0.5796661	0.5008977
5	0.006215807	0.1124592	0.3354053	0.5	0.6016591	0.7309599	0.8749972
6	0.003088585	0.09981217	0.3343629	0.5	0.5991635	0.5899725	0.3828199
7	0.001539523	0.0898497	0.3338465	0.5	0.6004165	0.7257148	0.8269409
8	0.0007685764	0.08177673	0.3335895	0.5	0.5997913	0.5971585	0.5008838
9	0.0003839928	0.0750893	0.3334613	0.5	0.6001042	0.7216807	0.8749973
10	0.0001919227	0.06945089	0.3333973	0.5	0.5999479	0.602573	0.3828197
11	9.594293e-05	0.06462747	0.3333653	0.5	0.6000261	0.7184363	0.8269407
12	4.796686e-05	0.06045076	0.3333493	0.5	0.599987	0.6068567	0.5008842
13	2.398228e-05	0.05679646	0.3333413	0.5	0.6000065	0.7157449	0.8749973
14	1.199085e-05	0.05357063	0.3333373	0.5	0.5999967	0.6103624	0.3828197
15	5.995355e-06	0.05070081	0.3333353	0.5	0.6000016	0.7134604	0.8269407
16	2.997659e-06	0.04813024	0.3333343	0.5	0.5999992	0.6133039	0.5008842
17	1.498825e-06	0.04581372	0.3333338	0.5	0.6000004	0.7114867	0.8749973
18	7.494115e-07	0.04371482	0.3333336	0.5	0.5999998	0.6158202	0.3828197
19	3.747055e-07	0.04180384	0.3333335	0.5	0.6000001	0.7097571	0.8269407

The columns represent growth rates and the rows represent generations. The model always starts with a population level of 0.5and it's set up to represent population as a ratio between 0 and 1.

4.1.6 System Behavior and Attractors

If we trace down the column of Table 4.1 under growth rate 1.5, we'll see the population level settles toward a final value of 0.333… after 20 generations. In the column for growth rate 2.0, we'll see an unchanging population level across each generation. This makes sense in the real world – if two parents produce two children, the overall population won't grow or shrink. So the growth rate of 2.0 represents the replacement rate.

Visualization of result of Table 4.1 as a line chart shown in Figure 4.3:-

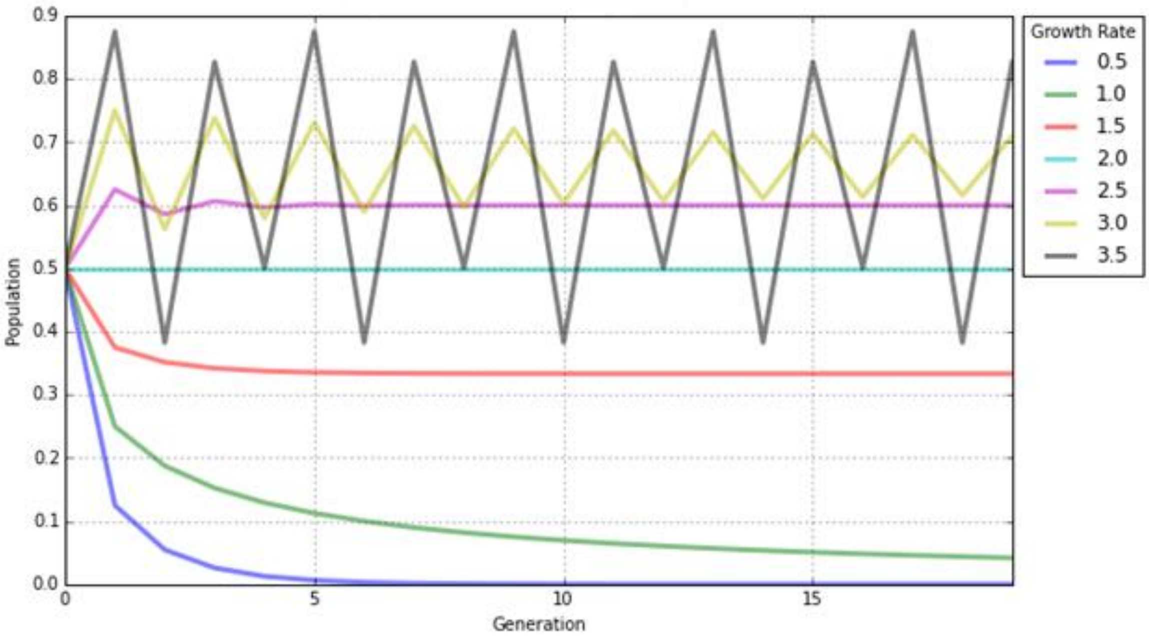


Figure 4.3: Logistic Model Result with Growth Rate

Here we can easily see how the population changes over time, given different growth rates. The blue line represents a growth rate of 0.5, and it quickly drops to zero. The population dies out. The cyan line represents a growth rate of 2.0 (remember, the replacement rate) and it stays steady at a population level of 0.5. The growth rates of 3.0 and 3.5 are more interesting. While the yellow line for 3.0 seems to be slowly converging toward a stable value, the gray line for 3.5 just seems to bounce around.

An *attractor* is the value, or set of values, that the system settles toward over time. When the growth rate parameter is set to 0.5, the system has a fixed-point attractor at population level 0 as depicted by the blue line. In other words, the population value is drawn toward 0 over time as the model iterates. When the growth rate parameter is set to 3.5, the system oscillates between four values, as depicted by the gray line. This attractor is called a limit cycle.

But when we adjust the growth rate parameter beyond 3.5, we see the onset of chaos. A chaotic system has a *strange attractor*, around which the system oscillates forever, never repeating itself or settling into a steady state of behavior. It never hits the same point twice and its structure has a fractal form, meaning the same patterns exist at every scale no matter how much you zoom into it.

4.1.7 Bifurcations and the Path to Chaos

To show this more clearly, let's run the logistic model again, this time for 200 generations across 1,000 growth rates between 0.0 to 4.0. When we produced the line chart above, we had only 7 growth rates. This time we'll have 1,000 so we'll need to visualize it in a different way, using something called a bifurcation diagram. Bifurcation analysis is the mathematical study of changes in the solutions when changing the parameters of the equations. These qualitative changes in the dynamics of the system are called bifurcations. The parameter values where they occur are called bifurcation points. It has two types :-

- ***Local bifurcations:*** This can be analyzed entirely through changes in the local stability properties of equilibria, periodic orbits or other invariant sets as parameters cross through critical thresholds.
- ***Global bifurcation:*** which often occur when larger invariant sets of the system ”collide” with each other, or with equilibria of the system. They cannot be detected purely by a stability analysis of the equilibria.

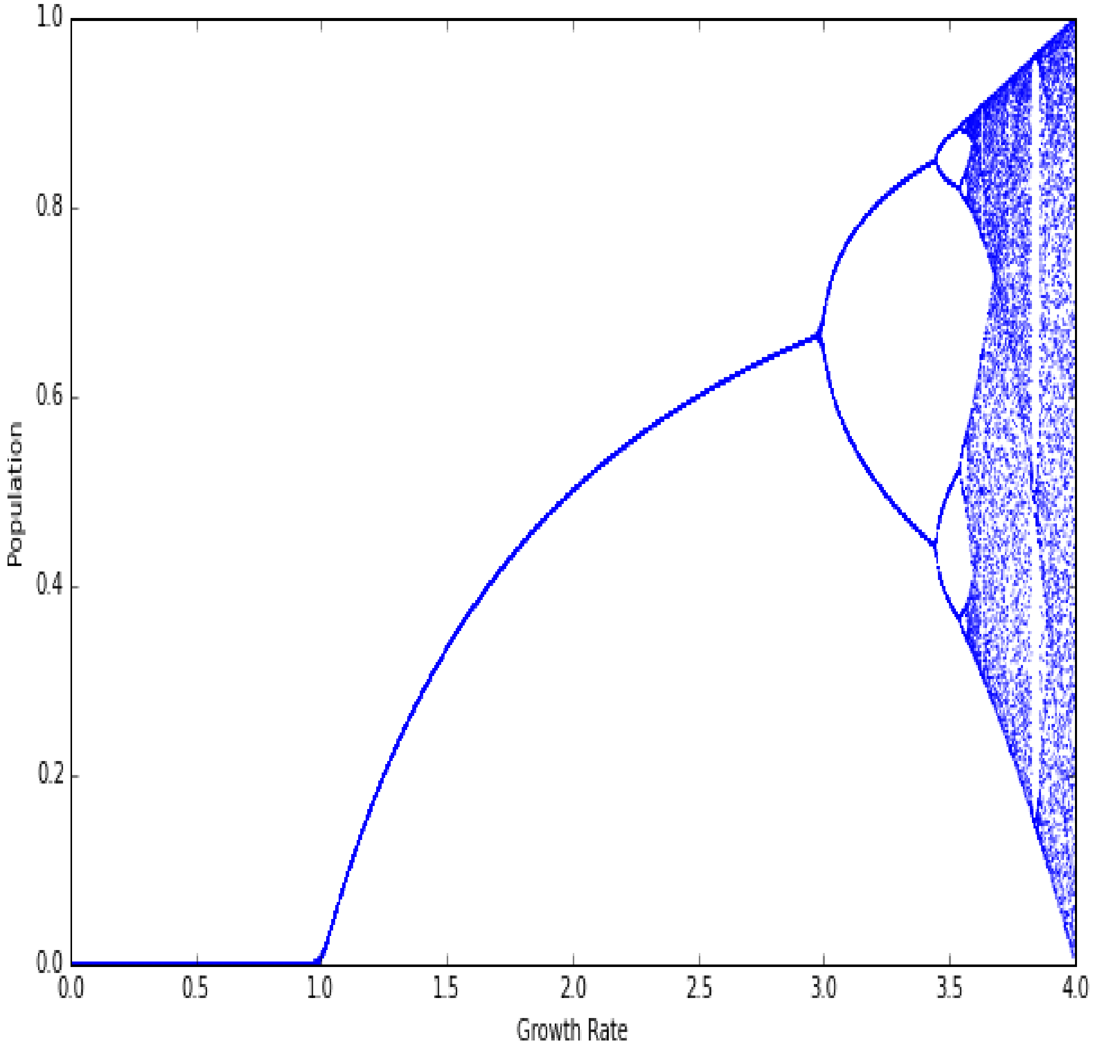


Figure 4.4: Bifurcation Diagram

In Figure. 4.4 we see that when the growth rates less than 1.0, the system always collapses to zero (extinction). For growth rates between 1.0 and 3.0, the system always settles into an exact, stable population level. Look at the vertical slice above growth rate 2.5. There's only one population value represented (0.6) and it corresponds to where the magenta line settles in the line chart shown earlier. But for some growth rates, such as 3.9, the diagram shows 100 different values – in other words, a different value for each of its 100 generations. It never settles into a fixed point or a limit cycle.

So, why is this called a bifurcation diagram? Let's zoom into the growth rates between 2.8 and 4.0 to see what's happening:

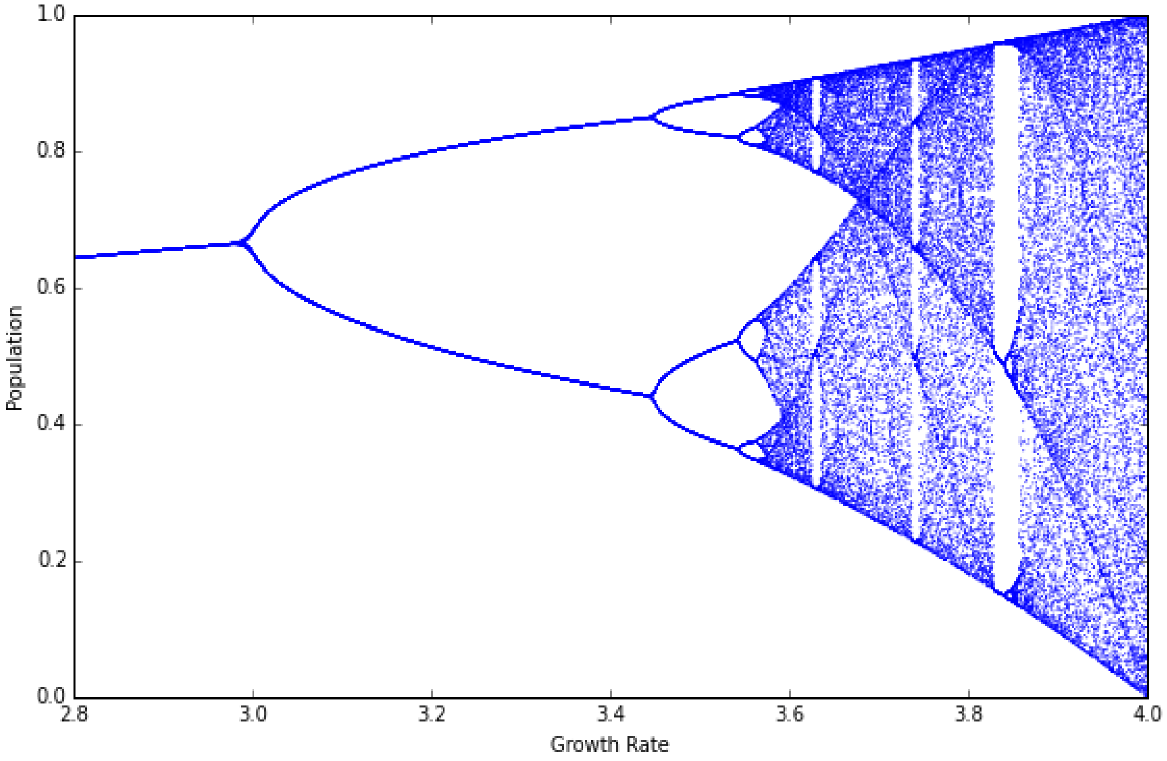


Figure 4.5: Zooming Portion of Bifurcation Diagram when Growth Rates between 2.8 and 4.0

At the vertical slice above growth rate 3.0, the possible population values fork into two discrete paths. At growth rate 3.2, the system essentially oscillates exclusively between two population values: one around 0.5 and the other around 0.8. In other words, at that growth rate, applying the logistic equation to one of these values yields the other.

Just after growth rate 3.4, the diagram bifurcates again into four paths. This corresponds to the gray line in the line chart we saw earlier: when the growth rate parameter is set to 3.5, the system oscillates over four population values. Just after growth rate 3.5, it bifurcates again into eight paths. Here, the system oscillates over eight population values.

4.1.8 The Onset of Chaos

Beyond a growth rate of 3.6, however, the bifurcations ramp up until the system is capable of eventually landing on *any* population value. This is known as the period-doubling path to chaos. As you adjust the growth rate parameter upwards, the logistic map will oscillate between two then four then eight then 16 then 32 (and on and on) population values. These are *periods*, just like the period of a pendulum.

By the time we reach growth rate 3.9, it has bifurcated so many times that the system now jumps, seemingly randomly, between all population values. We only say *seemingly* randomly because it is definitely *not* random. Rather, this model follows very simple deterministic rules yet produces apparent randomness. This is chaos: deterministic and aperiodic.

Let's zoom in again, to the narrow slice of growth rates between 3.7 and 3.9:

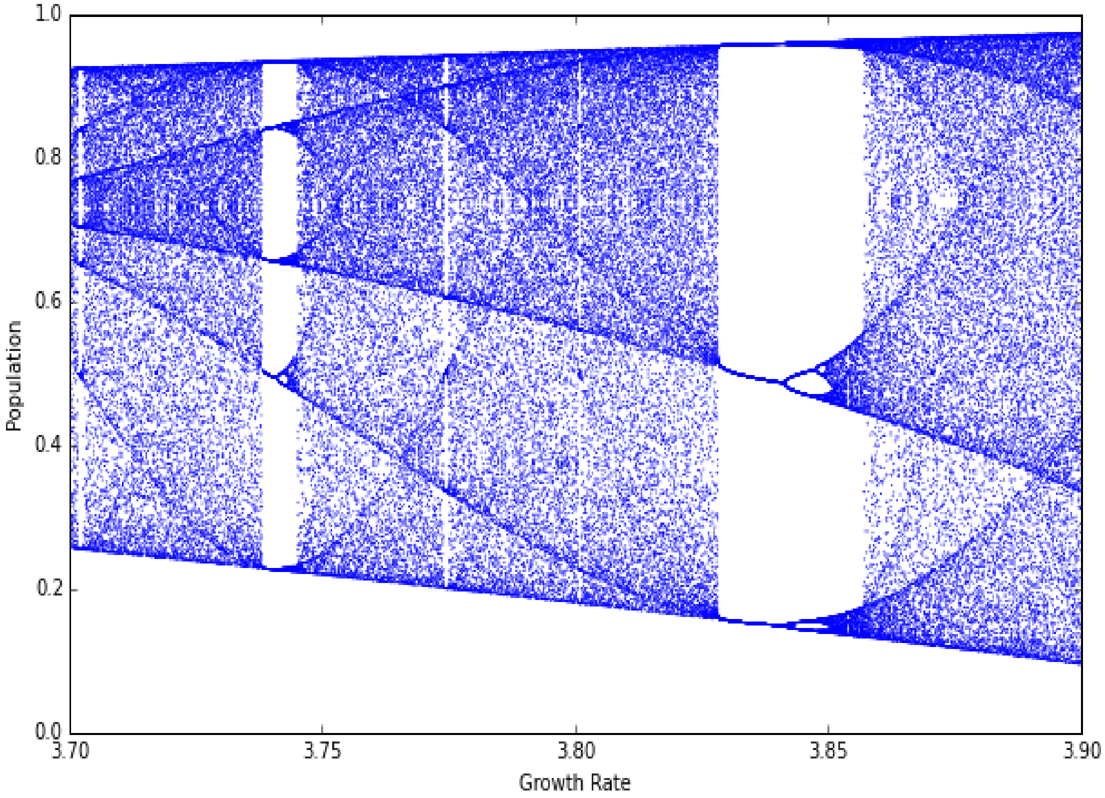


Figure 4.6: Zooming Portion of Bifurcation Diagram when Growth Rates between 3.7 and 3.9

As we zoom in, we begin to see the beauty of chaos. Out of the noise emerge strange swirling patterns and thresholds on either side of which the system behaves very differently. Between the growth rate parameters of 3.82 and 3.84, the system moves from chaos back into order, oscillating between just three population values (approximately 0.15, 0.55, and 0.95). But then it bifurcates again and returns to chaos at growth rates beyond 3.86.

4.1.9 Fractals and Strange Attractors

In the Figure 4.7, the bifurcations around growth rate 3.85 look a bit familiar. Let's zoom in to the center one:

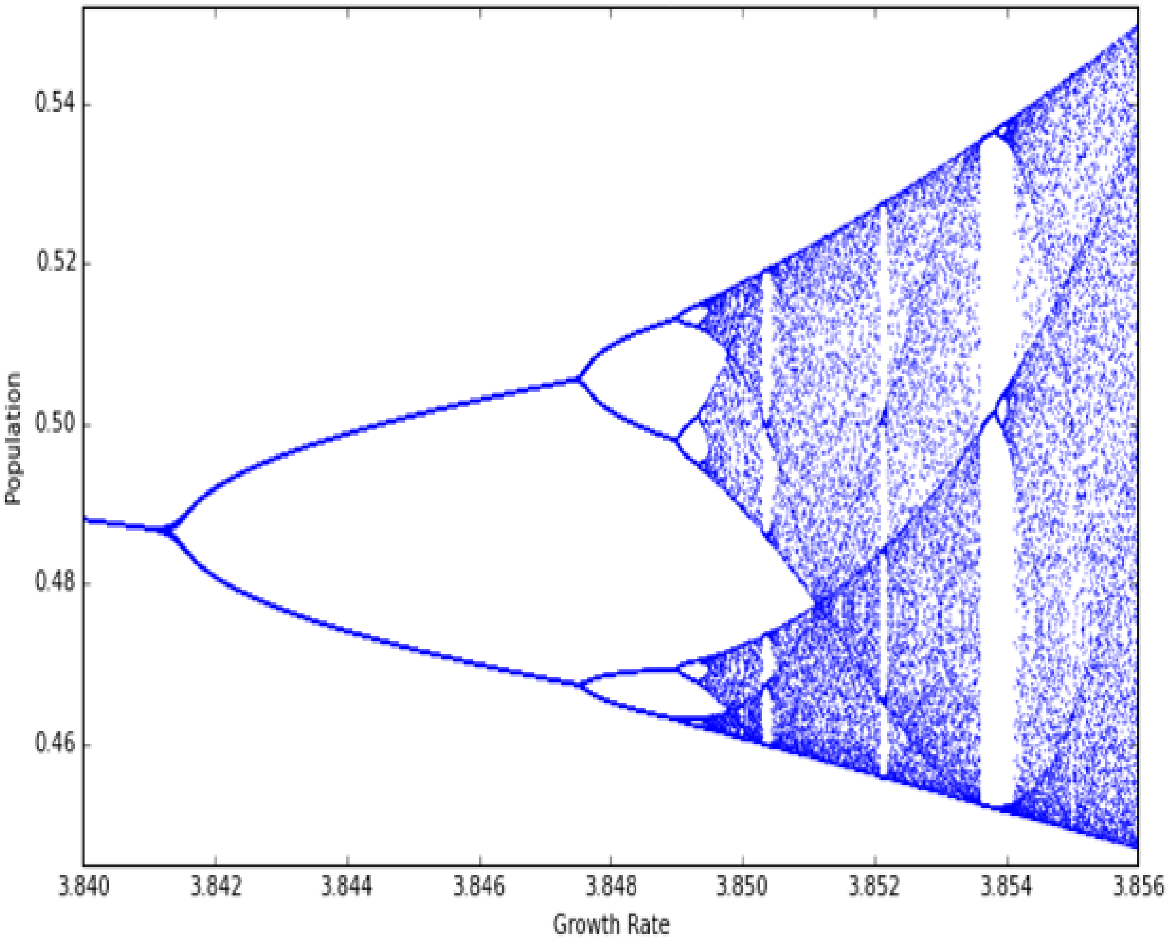


Figure 4.7: Zooming Portion of Bifurcation Diagram when Growth Rates around 3.85

Incredibly, we see the exact same structure that we saw earlier at the macro-level. In fact, if we keep zooming infinitely in to this plot, we'll keep seeing the same structure and patterns at finer and finer scales, forever.

Another way to visualize this is with a phase diagram, which plots the population value at generation *t + 1* on the y-axis versus the population value at *t* on the x-axis.

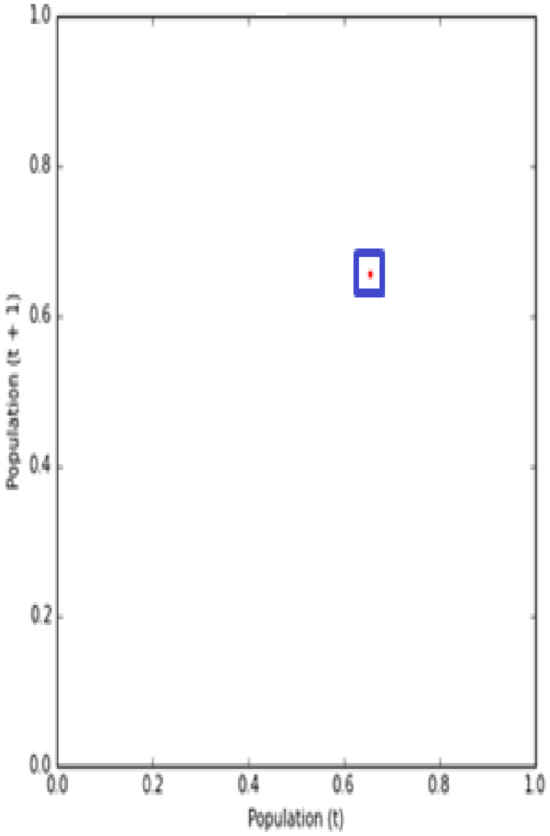


Figure 4.8: Phase Diagram when the Growth Rate Parameter (r) = 2.9

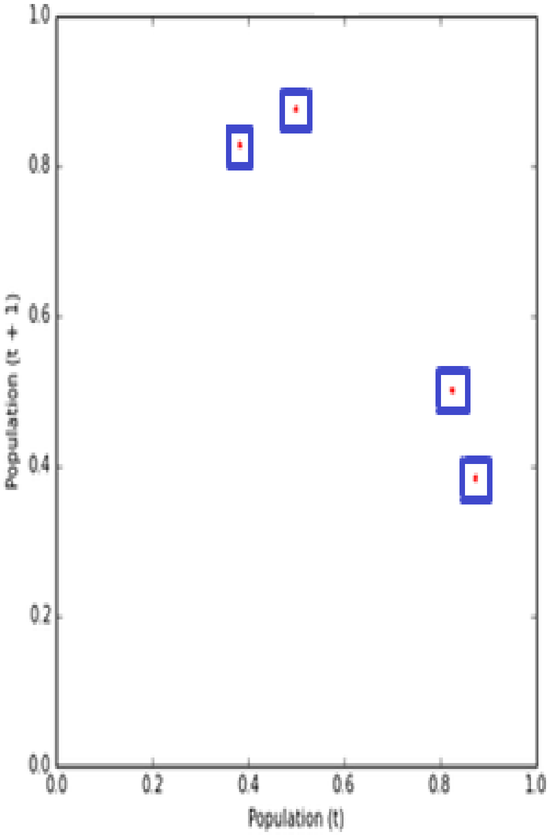


Figure 4.9: Phase Diagram when the Growth Rate Parameter (r) = 3.5

The phase diagram above on Figure 4.8 shows that the logistic map homes in on a fixed-point attractor at 0.655 (on both axes) when the growth rate parameter is set to 2.9. The plot on Figure 4.9 shows a limit cycle attractor. When the growth rate is set to 3.5, the logistic map oscillates across four points, as shown in this phase diagram.

Now we will see what happens when these period-doubling bifurcations lead to chaos. In Figure 4.10 depicts a parabola formed by a growth rate parameter of 3.9.The Figure 4.11 depicts 50 different growth rate parameters between 3.6 and 4.0. This range of parameters represents the *chaotic regime*: the range of parameter values in which the logistic map behaves chaotically. Each growth rate forms its own curve. These parabolas never overlap, due to their fractal geometry and the deterministic nature of the logistic equation.

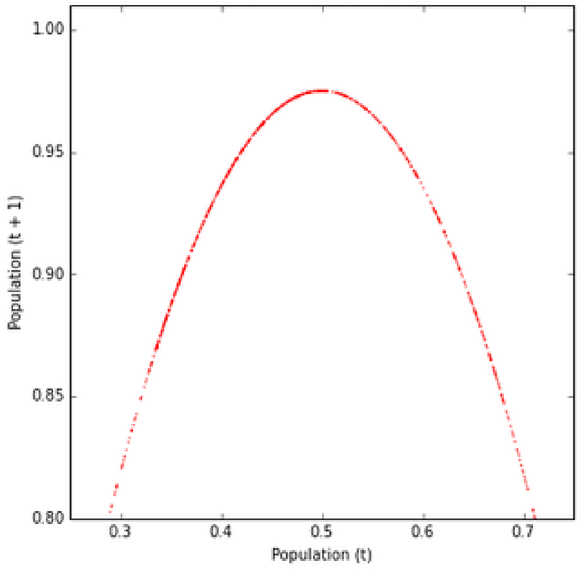


Figure 4.10: Representation of Chaos in Period-Doubling Bifurcations when Growth Rate Parameter(r) = 3.9

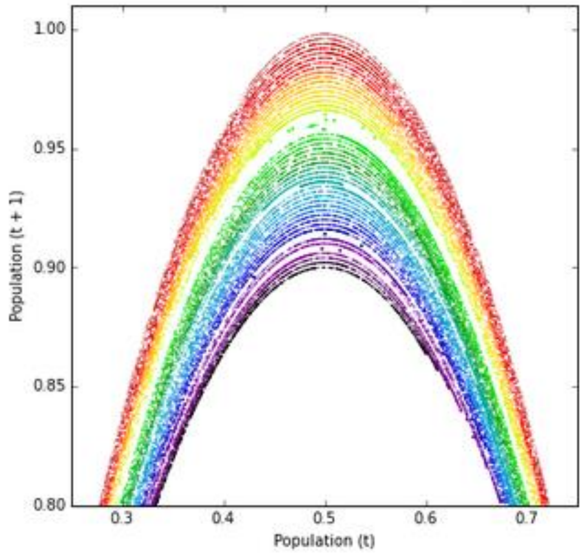


Figure 4.11: Representation of Chaos in Period-Doubling Bifurcations when Growth Rate Parameter(r) between 3.6 and 4.0

4.1.10 Maximum Randomness

By literature survey we know that, when growth rate parameter(r) = 3.99, then get maximum oscillation within interval [0, 1], i.e. is shown in Figure 4.12.

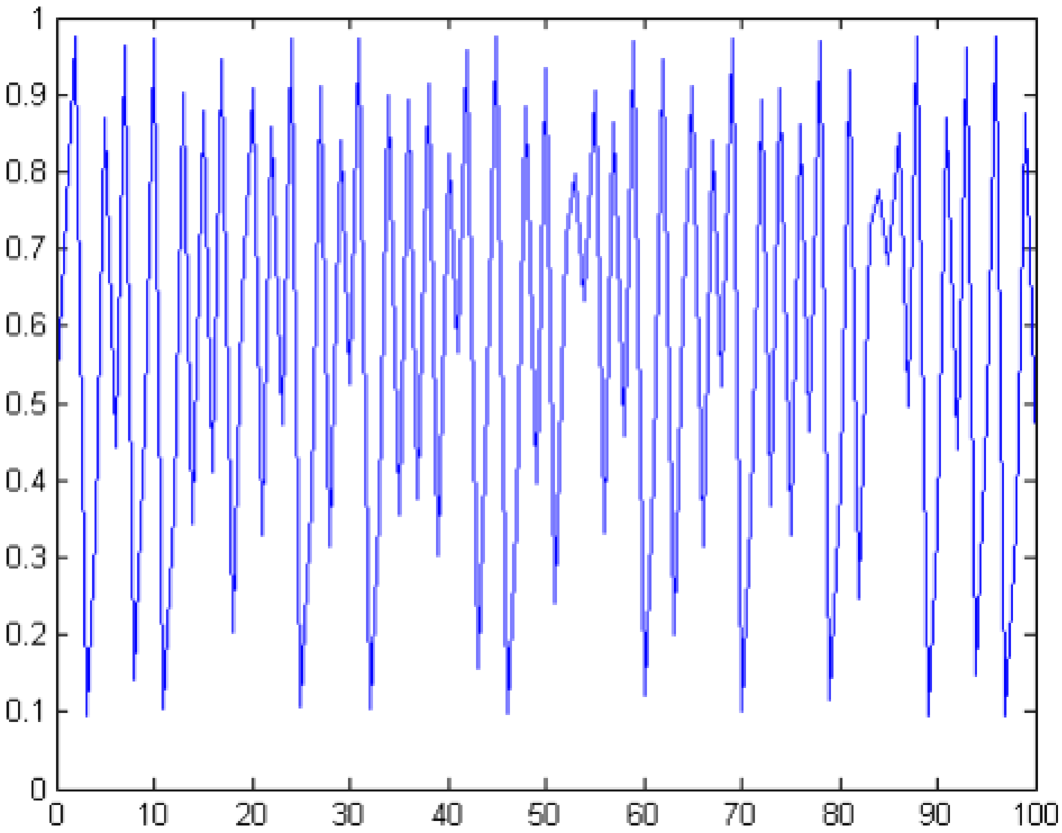


Figure 4.12: Logistic Map Function for $\boldsymbol{r}$=3.999 and $\boldsymbol{x}_t$ =0.66

For the inherent characteristics of chaotic functions, they are strongly sensitive to the initial parameter values and their related evolution function. This means a slight alteration in input parameter value causes huge changes in the value that is generated by the evolution function.

4.2 Deoxyribonucleic Acid Sequence Operation

4.2.1 DNA Encoding Rule:

Knowledge of deoxyribonucleic acid sequence (DNA) has become indispensable for basic biological research, and in numerous applied fields such as diagnostic, biotechnology, forensics, and biological systematics. There are four different nucleic acids in a DNA sequence: A (adenine), T (thymine), C (cytosine) and G (guanine). According to the rules of base pairing, the purine-adenine (A) always pairs with the pyrimidine-thymine (T), and the pyrimidine-cytosine (C) always pairs with the purine-guanine (G). It can be concluded that A and T are complementary, similarly G and Care.

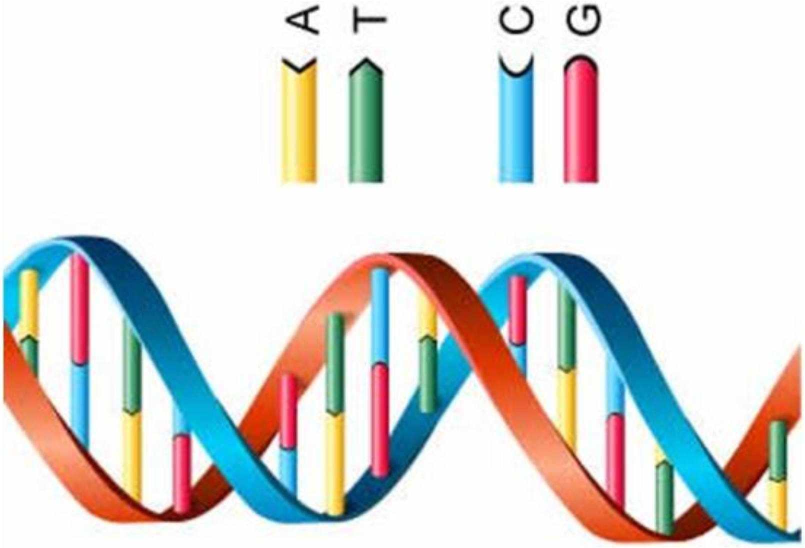


Figure 4.13: Simple DNA Structure

These relationships are often called Watson-Crick base pairing rules and are named after the two scientists who discovered their structural basis. By using the four bases A, C, G and T to encode 00, 01, 10 and 11, there are 24 kinds of coding schemes. But there are only 8 kinds of coding schemes satisfying the Watson-Crick complement rule, which are shown in Table 1.In the binary system, 0 and 1 are complementary. Therefore, it can be concluded that 00 and 11 are complementary and also 01 and 10 are complementary.

Moreover, with the develop of DNA computing, DNA cryptography has emerged as a new cryptographic field. There’re many new image encryption algorithms based on DNA computing [98-100].

In [98], a novel image fusion encryption algorithm DNA sequence operations to change the gray levels of the shuffled image pixels. Table 4.5 introduces the coding and decoding map rules for the DNA sequence.

Table 4.5: Encoding and Decoding Map Rules for DNA Sequences

Category	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***
A	00	00	01	01	10	10	11	11
T	11	11	10	10	01	01	00	00
G	01	10	00	11	00	11	01	10
C	10	01	11	00	11	00	10	01

4.2.2 XOR Algebraic Operation for DNA Sequence

In the encryption system, it also uses the XOR algebraic operation for DNA sequences. XOR operation for DNA sequences is performed according to traditional XOR in the binary. Corresponding to the 8 kinds of DNA encoding schemes, there also exists 8 kinds of DNA XOR rules. The XOR operation for DNA sequences used in the encryption system is shown in table 4.6.

Table 4.6: XOR Operation for DNA Sequences

XOR	***A***	***G***	***C***	***T***
A	G	A	T	C
G	A	G	C	T
C	T	C	G	A
T	C	T	A	G

4.2.3 Optical Realization of XOR Operation in DNA Theory

Optical logic gates have been investigated in various schemes utilizing nonlinear effects in nonlinear mediums, such as semiconductor optical amplifiers (SOAs). Figure 4.11 shows a method for implementing an optical XOR gate using SOAs. The optical XOR gate is very efficient for high-speed signal processing. Recently, studies have shown that SOA-based optical XOR gate can transmit data up to 160 Gb/s [101]. In this article, we use optical method to achieve XOR operation in DNA theory. It should be noted that we need to convert two DNA sequences into binary sequences according to decoding rules before they can be used as input to optical XOR gate. For example, there are two DNA sequences [TGCA] and [ATGC], which are first converted into binary sequences. The above two DNA sequences can be decoded as [11011000] and [00110110] using DNA decoding rule 1. Then, these two sequences are used as the input to the optical XOR gate in Figure 4.14 to obtain the output [11101110]. Finally, the same encoding rule (rule 1) is used to obtain the DNA sequence [TCTC]. If we use the incorrect decoding rules to decode the same DNA sequence, we will get incorrect binary sequence. In other words, the same rule is used for encoding and decoding to get the correct result.

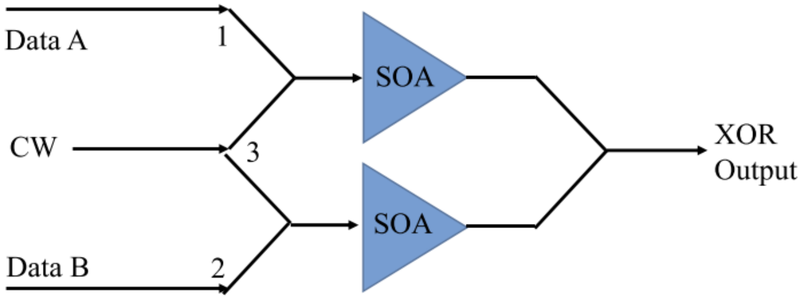


Figure. 4.14. Configuration of Optical XOR Gate Using SOAs (CW: Continuous-Wave)

4.2.4 Addition and Subtraction for DNA Sequence

Addition and subtraction phenomenon of DNA sequence is very similar to traditional algebraic computations. The addition and subtraction is done over modulo 4. The result of addition operation of DNA sequence are given below in Table 4.7 and subtraction operation of DNA sequence are given below in Table 4.8 when 00→A; 11→T; 01→C and 10→G.

Table 4.7: Addition Rule for DNA Sequence

+	***A***	***T***	***C***	***G***
G	G	C	T	A
C	C	A	G	T
T	T	G	A	C
A	A	T	C	G

Table 4.8: Subtraction Rule for DNA Sequence

-	***A***	***T***	***C***	***G***
G	G	T	C	A
C	C	G	A	T
T	T	A	G	C
A	A	C	T	G

Chapter 5

Our Audio Encryption Scheme

This scheme is applicable for the protection of multimedia data such as audio (.wav formatted). Audio file needs special encryption techniques due to its large data capacity without compromising the correlation between the original and the encrypted version (closer to zero). For this reason the proposed scheme is specially designed for the protection of audio data. The proposed scheme is established on key based block ciphering followed by key based channel shuffling, where both the size of the key and block is 32 bytes. The block ciphering is based on the concept of DNA encoding with logistic chaotic map. Next, channel (left, right) shuffling is applied after being encrypted using DNA encoding to make the cipher more confused against cryptanalysis. The unique feature is that consecutive blocks use different keys with key size 32 bytes derived from the original one using the proposed key chaining algorithm and experimental result shows that the correlation between consecutive keys is also close to zero. Hence, additional protection of the secret audio is ensured. Moreover experimental results from various statistical analysis are highly encouraging which contribute to establish the strength of this scheme.

The block diagram of the proposed scheme is shown in Figure 5.1 and the entire scheme is described using the following three sub-sections, where first section described on key chain generation, second section described on channel shuffling index generation and last or third section described on data encryption & decryption scheme.

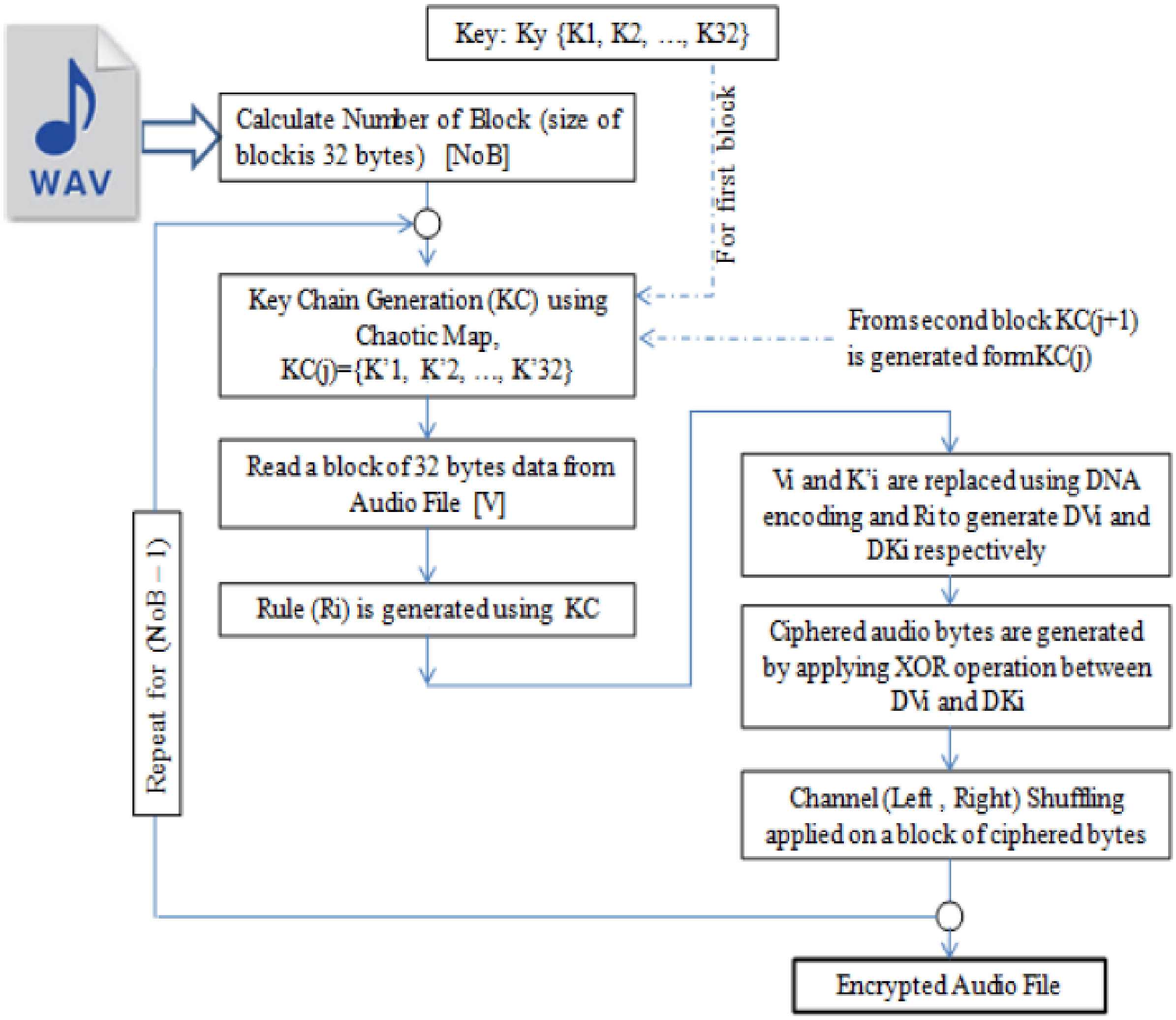

Key: Ky {K1, K2, ..., K32}
WAV
Calculate Number of Block (size of blockis 32 bytes) [NoB]
For first block
Key Chain Generation (KC) using Chaotic Map, KC(j)={K'1, K'2, ..., K'32}
From second block KC(j+1) is generated formKC(j)
Read a block of 32 bytes data from Audio File [V]
Rule (Ri) is generated using KC
Vi and K'i are replaced using DNA encoding and Ri to generate DVi and DKi respectively
Ciphered audio bytes are generated by applying XOR operation between DVi and DKi
Channel (Left , Right) Shuffling applied on a block of ciphered bytes
Repeat for (NoB – 1)
Encrypted Audio File

Figure 5.1: Block Diagram of Proposed Scheme

5.1 Key Chain Generation

The main focus of this algorithm is using different keys for consecutive blocks of audio files. The multiple keys so mentioned are derived from the original key using a key chaining is continued till the last 32 bytes block of audio file is encrypted.

Here *KC(j)* indicates 32 bytes key chain for the j^{th} block. First, 32 bytes *KC(1)* is generated from 32 bytes initial key (K_y) and next *KC(j)* is generated form *KC(j-1)*.

Following key chain generation method is used to generate first 32 bytes key chain *KC(1)* from K_y. First, the initial value of x_0 is calculated from the initial 32 bytes key (K_y) as explained by equation-5 and equation-6.

$$K_y = \{K_1, K_2, k_3, \dots, K_{32}\} \tag{5}$$

$$x_0 = \frac{\sum_{i=1}^{32} K_i \times (i+1)}{2^{21}} \tag{6}$$

Now, equation-7 is used to generate a new set of 32 bytes key chain *(KC)*, where x_0 and *xi* are the initial value and the i^{th} value of *x* in equation-4. In equation-7, the value of *C* is equivalent to K_{i-1}, where the initial value of *C* is K_1. Next, equation-8 is used to generate each K'_i. Therefore, a single byte difference in K_y causes a drastic change in *KC*.

$$K'_i = \lfloor x_i \times (K_i^2 + C) \rfloor \, \%256 \tag{7}$$

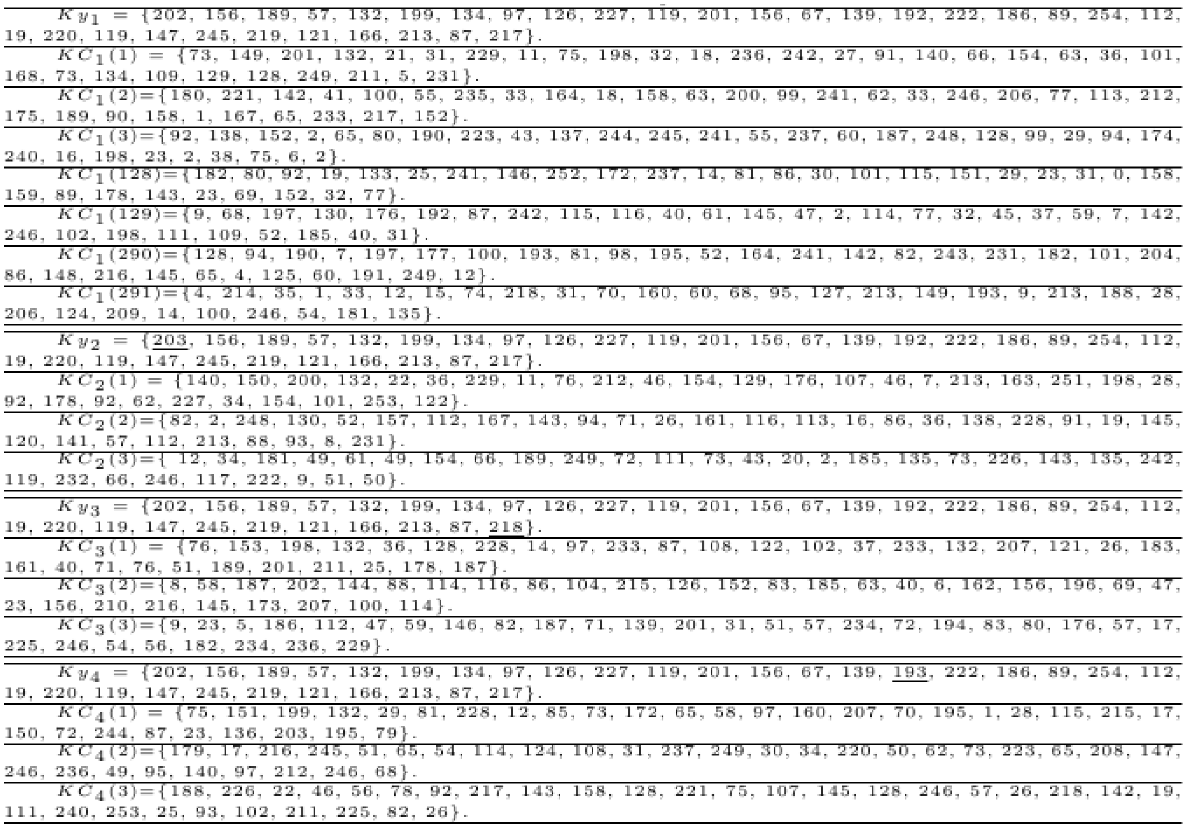
$$K'_i = \begin{cases} K'_i, & if\ K'_i \neq K_i; \\ \lfloor x_i \times 256 \rfloor, & if\ K'_i = K_i; \end{cases} \tag{8}$$

Now, the new set of 32-byte *KC(1)* is shown in equation-9 as

$$KC(1) = \{K'_1, K'_2, K'_3, \dots, K'_{32}\} \tag{9}$$

which is used to cipher 1st block of secret audio. Now another 32 bytes key chain *KC(j)* for j^{th} block is generated by above mentioned steps where K_y is replaced by *KC(j-1)*. Moreover, for each block of secret audio a new set of 32 bytes key chain *KC* is generated. Next, a single bit/byte change in K_y, generates a completely different key chain *KC(j)*, which adds avalanche effect in the generated key chain. Key sensitivity is also shown in the Figure 5.2 by considering four sets of K_y (K_{y1}, K_{y2}, K_{y3}& K_{y4}) keeping single byte differences and some sets of key chains are also shown for each K_y, where $KC_i(j)$ indicates the key chain of K_{yi} for block *j*.

The Figure 5.2 result shows the key sensitive in *KC* for a single byte difference K_y. The changed bytes in $K_{y2,3,4}$are indicated using underlines and they have a single byte difference from K_{y1}.



Ky_1 = {202, 156, 189, 57, 132, 199, 134, 97, 126, 227, 119, 201, 156, 67, 139, 192, 222, 186, 89, 254, 112, 19, 220, 119, 147, 245, 219, 121, 166, 213, 87, 217}.

$KC_1(1)$ = {73, 149, 201, 132, 21, 31, 229, 11, 75, 198, 32, 18, 236, 242, 27, 91, 140, 66, 154, 63, 36, 101, 168, 73, 134, 109, 129, 128, 249, 211, 5, 231}.

$KC_1(2)$={180, 221, 142, 41, 100, 55, 235, 33, 164, 18, 158, 63, 200, 99, 241, 62, 33, 246, 206, 77, 113, 212, 175, 189, 90, 158, 1, 167, 65, 233, 217, 152}.

$KC_1(3)$={92, 138, 152, 2, 65, 80, 190, 223, 43, 137, 244, 245, 241, 55, 237, 60, 187, 248, 128, 99, 29, 94, 174, 240, 16, 198, 23, 2, 38, 75, 6, 2}.

$KC_1(128)$={182, 80, 92, 19, 133, 25, 241, 146, 252, 172, 237, 14, 81, 86, 30, 101, 115, 151, 29, 23, 31, 0, 158, 159, 89, 178, 143, 23, 69, 152, 32, 77}.

$KC_1(129)$={9, 68, 197, 130, 176, 192, 87, 242, 115, 116, 40, 61, 145, 47, 2, 114, 77, 32, 45, 37, 59, 7, 142, 246, 102, 198, 111, 109, 52, 185, 40, 31}.

$KC_1(290)$={128, 94, 190, 7, 197, 177, 100, 193, 81, 98, 195, 52, 164, 241, 142, 82, 243, 231, 182, 101, 204, 86, 148, 216, 145, 65, 4, 125, 60, 191, 249, 12}.

$KC_1(291)$={4, 214, 35, 1, 33, 12, 15, 74, 218, 31, 70, 160, 60, 68, 95, 127, 213, 149, 193, 9, 213, 188, 28, 206, 124, 209, 14, 100, 246, 54, 181, 135}.

Ky_2 = {$\underline{203}$, 156, 189, 57, 132, 199, 134, 97, 126, 227, 119, 201, 156, 67, 139, 192, 222, 186, 89, 254, 112, 19, 220, 119, 147, 245, 219, 121, 166, 213, 87, 217}.

$KC_2(1)$ = {140, 150, 200, 132, 22, 36, 229, 11, 76, 212, 46, 154, 129, 176, 107, 46, 7, 213, 163, 251, 198, 28, 92, 178, 92, 62, 227, 34, 154, 101, 253, 122}.

$KC_2(2)$={82, 2, 248, 130, 52, 157, 112, 167, 143, 94, 71, 26, 161, 116, 113, 16, 86, 36, 138, 228, 91, 19, 145, 120, 141, 57, 112, 213, 88, 93, 8, 231}.

$KC_2(3)$={ 12, 34, 181, 49, 61, 49, 154, 66, 189, 249, 72, 111, 73, 43, 20, 2, 185, 135, 73, 226, 143, 135, 242, 119, 232, 66, 246, 117, 222, 9, 51, 50}.

Ky_3 = {202, 156, 189, 57, 132, 199, 134, 97, 126, 227, 119, 201, 156, 67, 139, 192, 222, 186, 89, 254, 112, 19, 220, 119, 147, 245, 219, 121, 166, 213, 87, $\underline{218}$}.

$KC_3(1)$ = {76, 153, 198, 132, 36, 128, 228, 14, 97, 233, 87, 108, 122, 102, 37, 233, 132, 207, 121, 26, 183, 161, 40, 71, 76, 51, 189, 201, 211, 25, 178, 187}.

$KC_3(2)$={8, 58, 187, 202, 144, 88, 114, 116, 86, 104, 215, 126, 152, 83, 185, 63, 40, 6, 162, 156, 196, 69, 47, 23, 156, 210, 216, 145, 173, 207, 100, 114}.

$KC_3(3)$={9, 23, 5, 186, 112, 47, 59, 146, 82, 187, 71, 139, 201, 31, 51, 57, 234, 72, 194, 83, 80, 176, 57, 17, 225, 246, 54, 56, 182, 234, 236, 229}.

Ky_4 = {202, 156, 189, 57, 132, 199, 134, 97, 126, 227, 119, 201, 156, 67, 139, $\underline{193}$, 222, 186, 89, 254, 112, 19, 220, 119, 147, 245, 219, 121, 166, 213, 87, 217}.

$KC_4(1)$ = {75, 151, 199, 132, 29, 81, 228, 12, 85, 73, 172, 65, 58, 97, 160, 207, 70, 195, 1, 28, 115, 215, 17, 150, 72, 244, 87, 23, 136, 203, 195, 79}.

$KC_4(2)$={179, 17, 216, 245, 51, 65, 54, 114, 124, 108, 31, 237, 249, 30, 34, 220, 50, 62, 73, 223, 65, 208, 147, 246, 236, 49, 95, 140, 97, 212, 246, 68}.

$KC_4(3)$={188, 226, 22, 46, 56, 78, 92, 217, 143, 158, 128, 221, 75, 107, 145, 128, 246, 57, 26, 218, 142, 19, 111, 240, 253, 25, 93, 102, 211, 225, 82, 26}.

Figure 5.2: Varied KC for Single Byte Difference K_y

5.2 Channel Shuffling Index Generation

To make the scheme more robust and secured, channel (left, right) shuffling operation is used. The positions of the left and the right channels are in odd and even sequences respectively. Therefore, two 16 bytes shuffling indexes (*Left_Index[16]* and *Right_Index[16])* are generated using *KC(j)* for j^{th} block of 32 bytes ciphered data, where each index byte is in the range of 1 to 32. Here, linear probing is used to make all indexes as collision free. The algorithm-1 is used to generate the channel shuffling indexes.

Algorithm 1. Channel Shuffling Index Generator

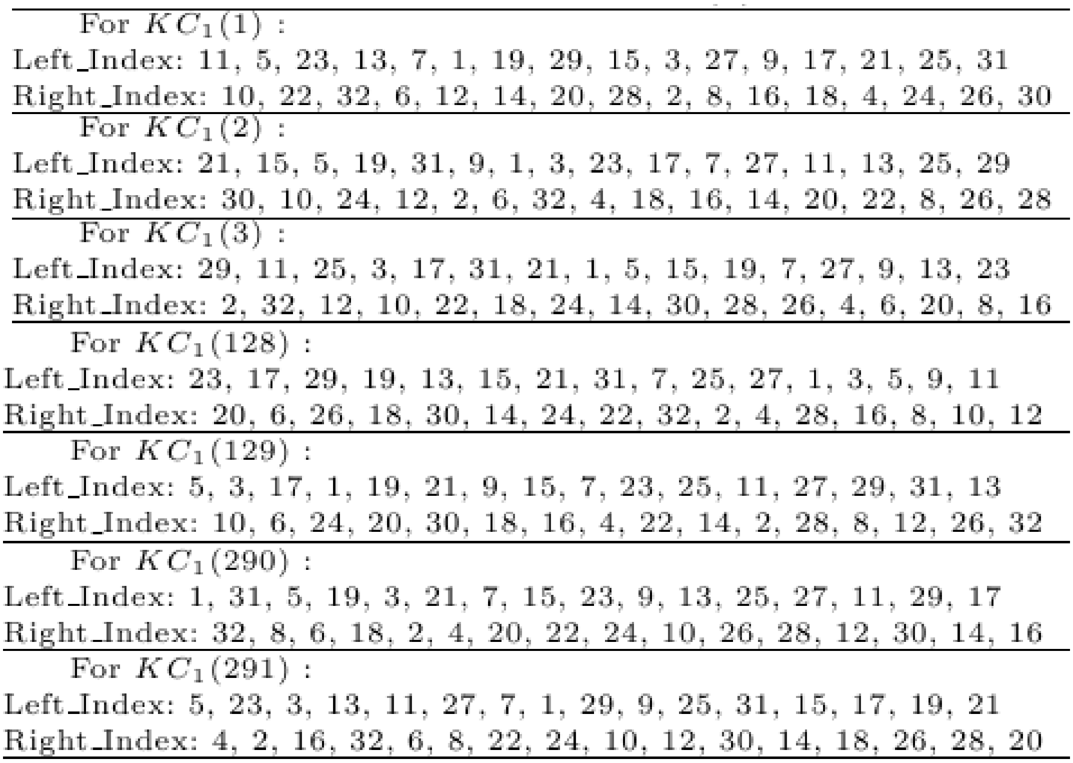
Input: KC (32 bytes key chain for a block)

Output: Left_Index[16] and Right-Index[16]

1: Set *Value[32]=0, Left_Index[16]=0 and Right_Index[16]=0*;

```
Set P1 = 0 and P2 = 0;
for $I$ = 1 to 32 do
    while 1 do
        $t = KC[i]\%32$;
        if $Value[t]$ = 0 then
            break;
        end if
        $KC[i] = KC[i] + 1$;
    end while
    if t%2 = 0 then
        Set $Left\_Index[P1] = t+1$;
        $P1 = P1 + 1$;
    else
        Set Right_Index[P2]= t+1;
        $P2 = P2+1$;
    end if
    Set $Value[t] = 1$;
end for
```

In Figure 5.3, some shuffling indexes are shown for different *KC(j)* where the initial key is K_{y1}. Here, each *KC(j)* is used to generated two shuffling indexes of j^{th} block.



```
    For KC_1(1) :
Left_Index: 11, 5, 23, 13, 7, 1, 19, 29, 15, 3, 27, 9, 17, 21, 25, 31
Right_Index: 10, 22, 32, 6, 12, 14, 20, 28, 2, 8, 16, 18, 4, 24, 26, 30
    For KC_1(2) :
Left_Index: 21, 15, 5, 19, 31, 9, 1, 3, 23, 17, 7, 27, 11, 13, 25, 29
Right_Index: 30, 10, 24, 12, 2, 6, 32, 4, 18, 16, 14, 20, 22, 8, 26, 28
    For KC_1(3) :
Left_Index: 29, 11, 25, 3, 17, 31, 21, 1, 5, 15, 19, 7, 27, 9, 13, 23
Right_Index: 2, 32, 12, 10, 22, 18, 24, 14, 30, 28, 26, 4, 6, 20, 8, 16
    For KC_1(128) :
Left_Index: 23, 17, 29, 19, 13, 15, 21, 31, 7, 25, 27, 1, 3, 5, 9, 11
Right_Index: 20, 6, 26, 18, 30, 14, 24, 22, 32, 2, 4, 28, 16, 8, 10, 12
    For KC_1(129) :
Left_Index: 5, 3, 17, 1, 19, 21, 9, 15, 7, 23, 25, 11, 27, 29, 31, 13
Right_Index: 10, 6, 24, 20, 30, 18, 16, 4, 22, 14, 2, 28, 8, 12, 26, 32
    For KC_1(290) :
Left_Index: 1, 31, 5, 19, 3, 21, 7, 15, 23, 9, 13, 25, 27, 11, 29, 17
Right_Index: 32, 8, 6, 18, 2, 4, 20, 22, 24, 10, 26, 28, 12, 30, 14, 16
    For KC_1(291) :
Left_Index: 5, 23, 3, 13, 11, 27, 7, 1, 29, 9, 25, 31, 15, 17, 19, 21
Right_Index: 4, 2, 16, 32, 6, 8, 22, 24, 10, 12, 30, 14, 18, 26, 28, 20
```

Figure. 5.3: Channel Shuffling Pair Generation Using K_{y1}

5.3. Audio Encryption & Decryption Scheme

The algorithm takes the audio file as input and produces an encrypted audio of the same size as the output. It is based on block ciphering followed by channel shuffling. The stepwise encryption scheme is described as follows:

(1) The entire secret data *(S)* is divided into number of blocks *(NoB)* and each *NoB* has 32 bytes of secret data *(V).*

(2) For a *NoB*, a new set of key chain *(KC)* is generated using subsection-5.2.1.

(3) Next, a *KC* is used to generate $x_0$using equation-6, where K_i is replaced by K'_i.

(4) Now, a DNA replacement rule is generated using equation-7, where K_i is replaced by K'_i and it is shown in equation-10. The value of C' is K'_{i-1} and the initial value of C' is K'_1.

$$Rule_i = \lfloor x_i \times (K'^2_i + C') \rfloor \%8 \qquad (10)$$

where x_0 and x_i are the initial value and the i^{th} value of equation-4 respectively.

(5) Next, *Rule*$_i$ with DNA encoding is used to replace V_i and K'_i to generate DV_i and DK_i respectively.

(6) Now, a new ciphered byte V_i' is generated by applying bitwise XOR-operation between DV_i and *DKi*, i.e. $V_k' = DV_i \oplus DK_i$.

(7) Repeat step-4 to step-6 for remaining 31 times to generate a block of 32 ciphered bytes.

(8) Next, the channel (left, right) shuffling indexes are generated using algorithm-1 along with a *KC*. This generates two sets of 16 bytes values namely *Left_Index[16]* and *Right_Index[16]*.

(9) Now, the two channel values of the 32 ciphered bytes are interchanged between the *Left_Index* and the *Right_Index*.

(10) Repeat step-2 to step-9 for remaining (NoB-1) times to generate the completely encrypted audio data.

The detailed encryption algorithm is shown in the form of a pseudo code in algorithm-2. The decryption phase is the reverse of the encryption phase, where channel shuffling operation is followed by DNA encoding. The reconstruction of secret byte *(Vr$_i$)* is done by applying reverse DNA replacement of V_i'', where $V_i'' = V_i' \oplus DK_i$ and the generated Vr_i is completely same as to that of V_i.

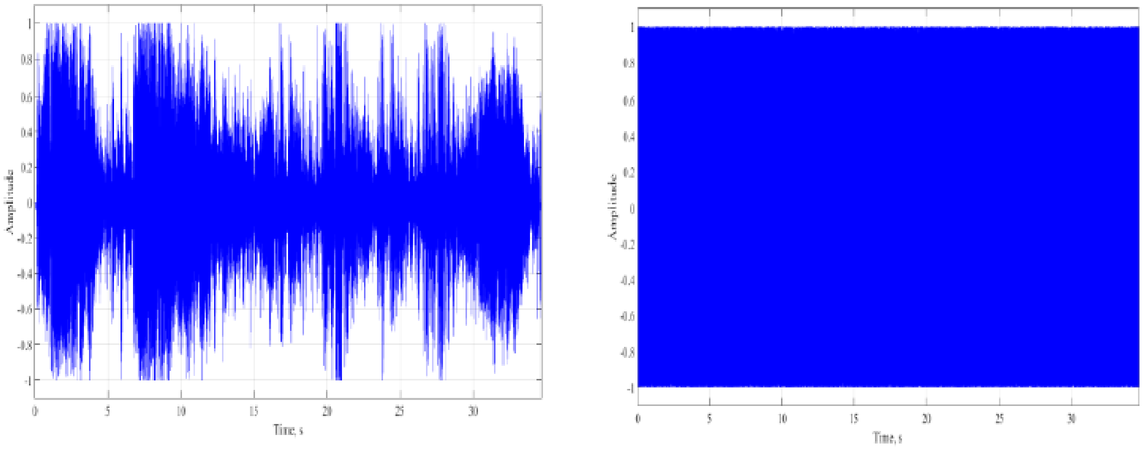
Algorithm 2. Audio Encryption Scheme

```
    Input: An audio (.wav) file of size S bytes and $K_y$
    Output: An encrypted audio of size S bytes, say E
Calculate $NoB = S / 32$;
for $I = 1$ to $NoB$ do
    Read_Buffer ← Read 32 bytes data from S;
    KC ← Key Chain Generation using $K_y$ (Sect.5.2.1);
    Set $x_0$ ← Generate initial value of eq.4 (using eq.6 and KC);
    Set $x_n \leftarrow x_0$;
    Set $C \leftarrow KC[1]$;
    for $J = 1$ to 32 do
        Set $V \leftarrow Read\_Buffer[J]$;
        Calculate $x_{n1} \leftarrow x_n \times 3.999 \times (1 - x_n)$;
        Set $x_n \leftarrow x_{n1}$;
        Calculate $Rule_j \leftarrow \lfloor x_i \times (KC[j] \times KC[J] + C) \rfloor \%8$;
        Set $C \leftarrow KC\ [J]$;
        Calculate $DV \leftarrow$ DNA Replacement of V using $Rule_i$;
        Calculate $DK \leftarrow$ DNA Replacement of KC [J] using $Rule_i$;
        Calculate $V \leftarrow DV \oplus DK$;
        Set Read_Buffer[J] ← V;
    end for
    Fill (Left_Index[ ] and Right _Index[ ])← Using Algo.1 with KC (Channel Shuffling indexes generation);
    for $K = 1$ to 16 do
        Set Lft ← Left_Index[K];
        Set Rgt ← RightJndex[K];
        Interchange Read_Buffer[Lft] & Read_Buffer[Rgt];
    end for
    Write Read_Buffer in Encrypted Audio E;
end for
```

Chapter 6

Experimental Results

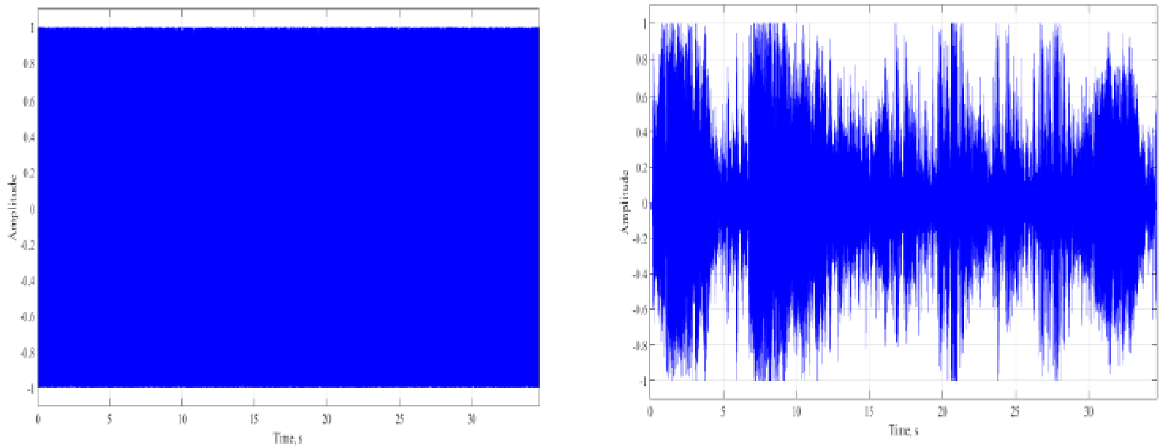
Figure 6.1 shows the encryption and decryption of Audio-1. Figure 6.1.a shows an original audio of size (2.11MB), which is encrypted using K_{y1} to generate an encrypted noisy audio EAudio-1 (Figure 6.1.b). Figure 6.1.c shows a noisy audio construction using single byte difference key K_{y2}, whereas Figure 6.1.d shows a lossless audio reconstruction DAudio-1 using the key K_{y1}.



a. Audio-1 (2.11 MB)

b. Encrypted audio EAudio-1 (2.11MB) using K_{y1}

(I)



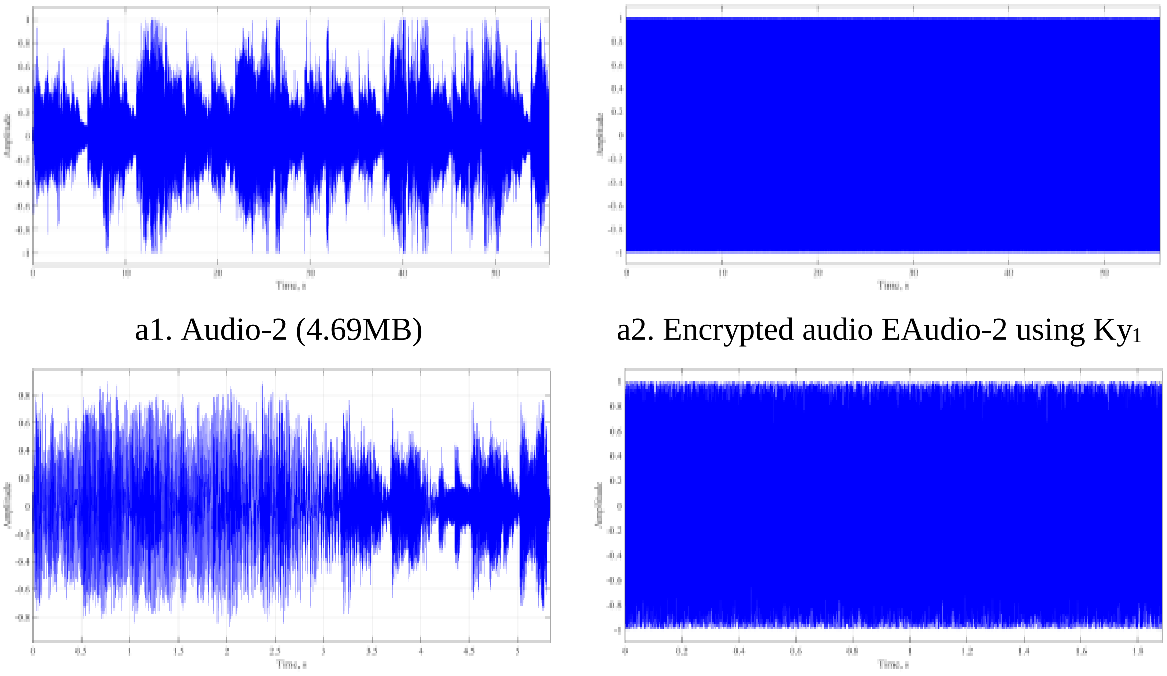
c. Noisy audio construction using K_{y2}.

d. Lossless audio reconstruction DAudio-1 (2.11MB) using K_{y1}.

(II)

Figure.6.1 (I-II): Encryption and Decryption of Audio-1 Using K_y

Nine other experimental results are also shown in Figure 6.2 for different audios with different sizes and different encryption keys.



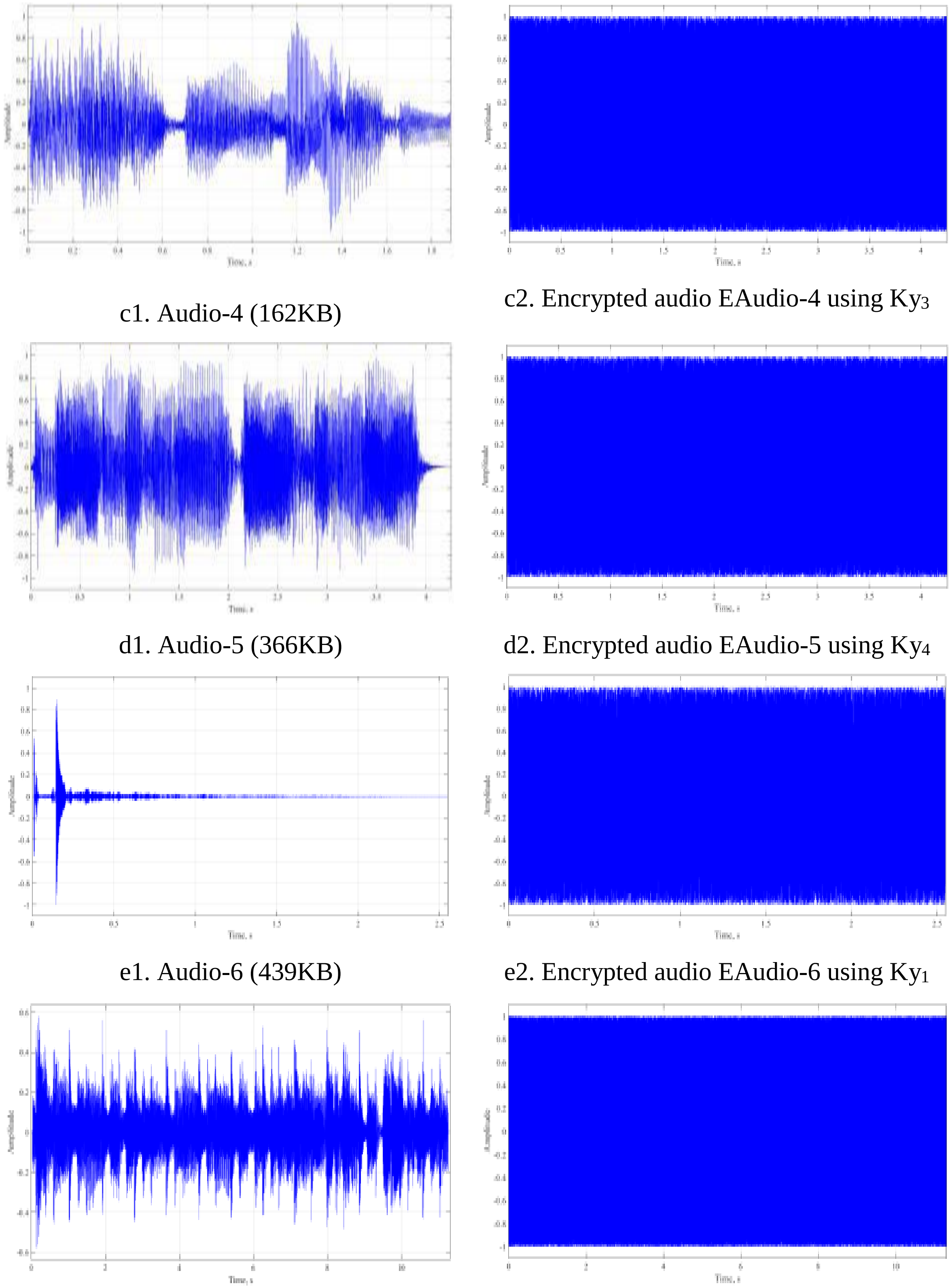
a1. Audio-2 (4.69MB)

a2. Encrypted audio EAudio-2 using Ky_1

b1. Audio-3 (918KB)

b2. Encrypted audio EAudio-3 using Ky_2

(I)



c1. Audio-4 (162KB)

c2. Encrypted audio EAudio-4 using Ky_3

d1. Audio-5 (366KB)

d2. Encrypted audio EAudio-5 using Ky_4

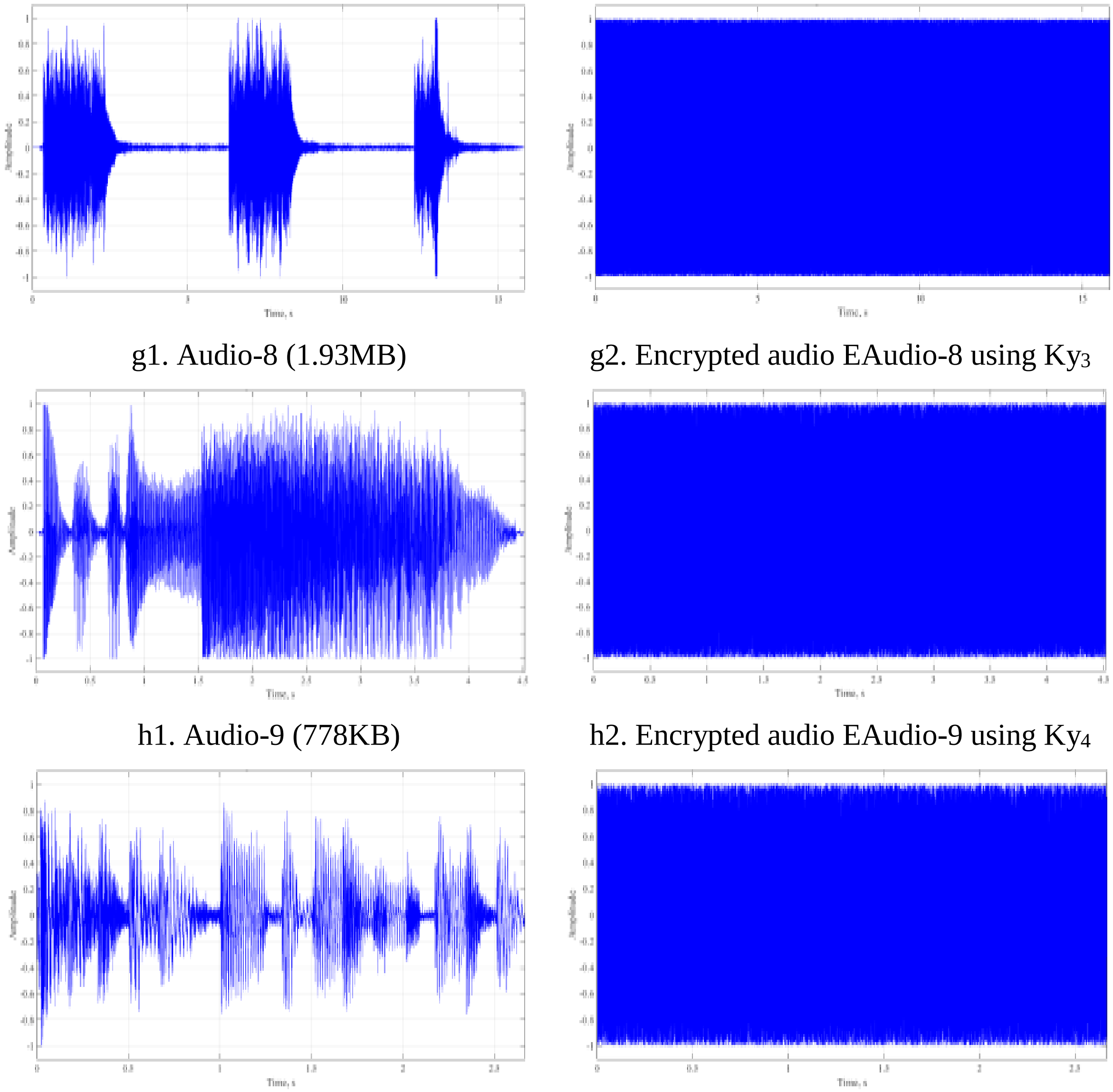
e1. Audio-6 (439KB)

e2. Encrypted audio EAudio-6 using Ky_1

f1. Audio-7 (1.9MB)

f2. Encrypted audio EAudio-7 using Ky_2

(II)



g1. Audio-8 (1.93MB)

g2. Encrypted audio EAudio-8 using Ky3

h1. Audio-9 (778KB)

h2. Encrypted audio EAudio-9 using Ky4

i1. Audio-10 (459KB)

i2. Encrypted audio EAudio-10 using Ky1

(III)

Figure 6.2 (I-II-III): Encryption of Nine Different Audios.

Chapter 7

Strength & Security Analysis

A secure multimedia encryption algorithm should be robust against all types of attacks such as cryptanalytic, statistical and brute-force attacks. Here, we discuss the security analysis of the proposed algorithm by using key space & key sensitivity analysis and statistical analysis. The resistance against different types of attacks is a useful measure for the performance of a cryptosystem. Some security analysis results are incorporated in the following section to establish the strength of our proposed scheme.

7.1 Key Space & Sensitivity Analysis

A good cryptosystem should have sufficiently large key space to make brute-force attacks infeasible. Key spaces imply the size of keys used in a cryptosystem. The proposed algorithm uses a key of size 32 bytes (*Ky*). Therefore, the total key space of ***Ky*** is = (2^{32x8}) = 2^{256}, which is large enough for brute-force attacks according to the present available computational speed.

Moreover, a new set of 32 bytes *KC* is generated for ciphering of every block. Each *KC* is highly sensitive to the initial key ***Ky***, change of a single bit/byte in the ***Ky*** produces an altogether different *KC*s as well as different encrypted audios.

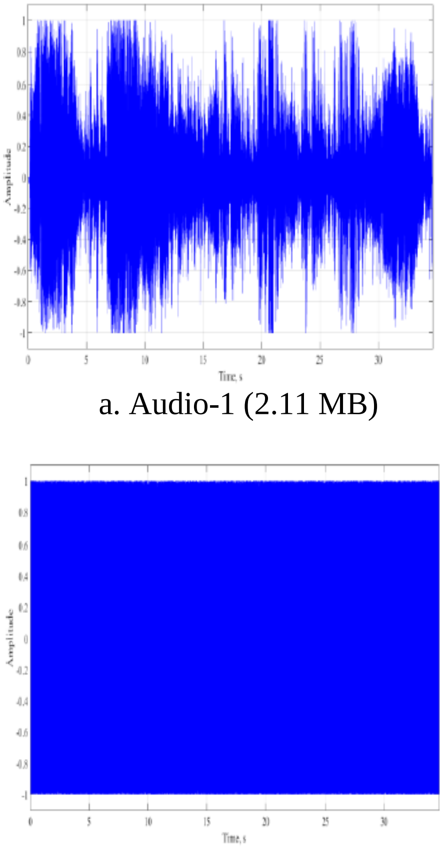
Table-7.1 shows the correlation value (using equation-11) between different key pairs. The correlation value between $KC_i(j)$ and Ky_i is close to zero, which indicates each $KC_i(j)$ is completely different from Ky_i. Moreover, single byte difference $\mathrm{Ky}_{2,3,4}$ from Ky_i gives the correlation value as 1, which proves that the key chain generation is secured and it is highly sensitive to the initial key.

Table 7.1: Correlation between Different Key Pairs

Key pair	***Correlation***	***Key pair***	***Correlation***
Ky_1 & Ky_2	1.0000	Ky_1&Ky_3	1.0000
Ky_1 & Ky_4	1.0000		
Ky_1 & $KC_1(1)$	0.1445	$KC_1(1)$ & $KC_1(2)$	0.0598
KC_1(2) & KC_1 (3)	0.2653	$KC_1(128)$ & $KC_1(129)$	0.2774
$KC_1(290)$ & $KC_1(291)$	0.0683	Ky_2& $KC_2(1)$	0.1103
$KC_2(1)$ & $KC_2(2)$	0.0459	$KC_2(2)$ & $KC_2(3)$	0.2657
Ky_3 & $KC_3(1)$	-0.0342	$KC_3(1)$ & $KC_3(2)$	-0.1074
$KC_3(2)$ & $KC_3(3)$	0.2765	Ky_4& $KC_4(1)$	-0.0810
$KC_4(1)$ & KC_4(2)	0.0231	$KC_4(2)$ & $KC_4(3)$	0.1477

Fig.6.1.a shows a secret audio (Audio-1), which is encrypted using Ky_i to generate EAudio-1 (Figure 6.1.b). Once again the Audio-1 is encrypted to generate EAudio-1.1 (Figure 7.1.b) using Ky_2, which has a single byte difference from Ky_i.

The correlation and number of signal change rate between EAudio-1 (Figure 6.1.b) and EAudio-1.1 (Figure 7.1.b) are 0.000224 and 99.9989 (≌ 100) respectively. These two values indicate that there is no statistical similarity between the two audios. Therefore, it proves the high key sensitivity of the proposed scheme.



a. Audio-1 (2.11 MB)

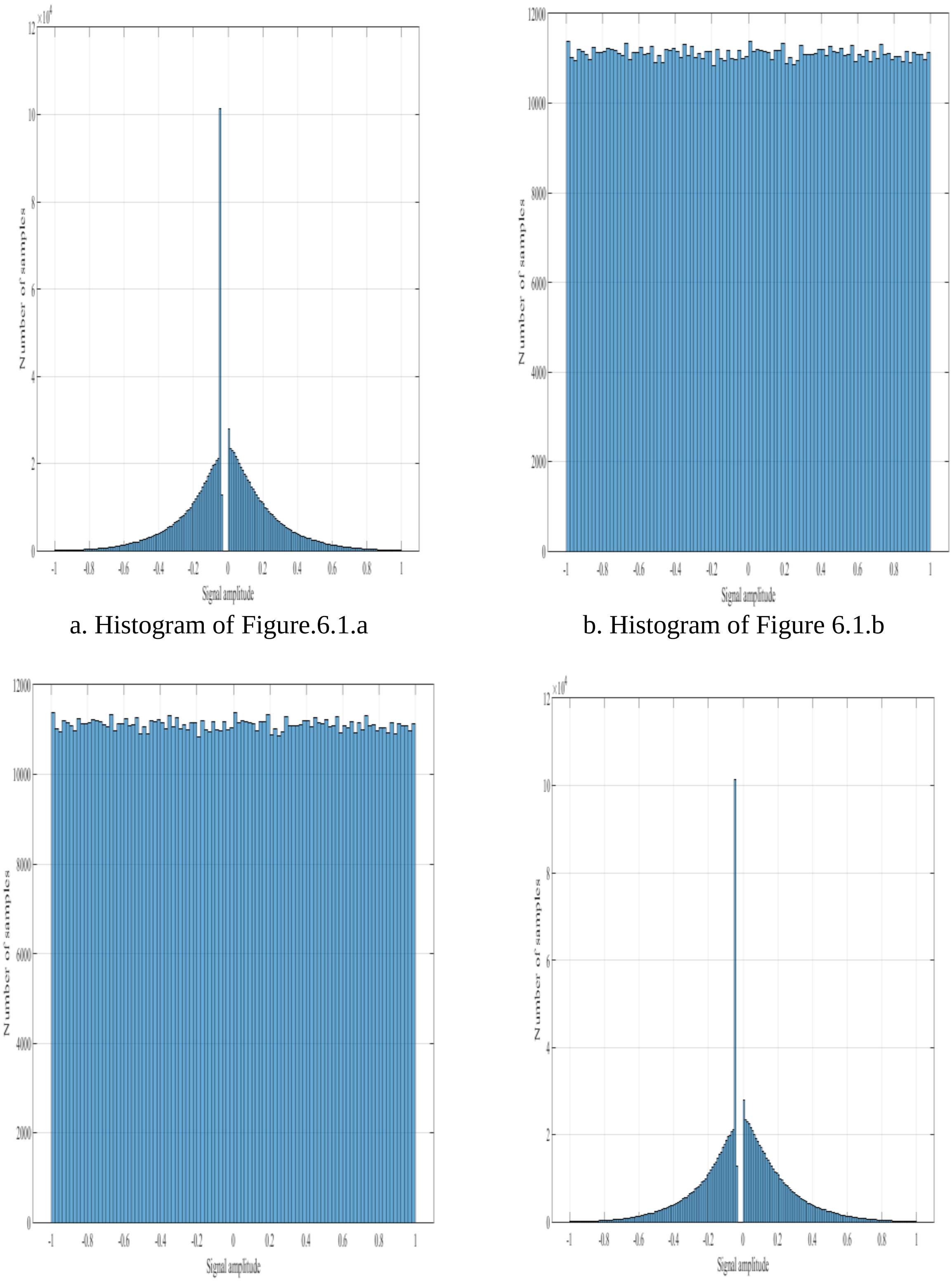
b. Encrypted audio EAudio-1.1 using Ky$_2$

Figure. 7.1: Encryption of Audio-1 Using K_{y2}

7.2 Histogram Analysis

The shape of the histogram clarifies the distribution of the signals of an audio. The histogram of Audio-1 (Figure 6.1.a) is shown in Figure 7.2.a and the uniformly distributed histogram of the encrypted audio EAudio-1 (Figure 6.1.b) is shown in figure 7.2.b. Thus, Figure 7.2.b represents the completely noisy audio signal and there is no statistical similarity between the two audios.

The histogram shape of the decrypted audio (Figure 6.1.d) is completely same as to that of Figure 7.2.a, which proves lossless reconstruction of Audio-1. Fig.7.2.c also represents the histogram of the noisy audio (Figure 6.1.c), which is constructed using a single byte difference key.



a. Histogram of Figure.6.1.a

b. Histogram of Figure 6.1.b

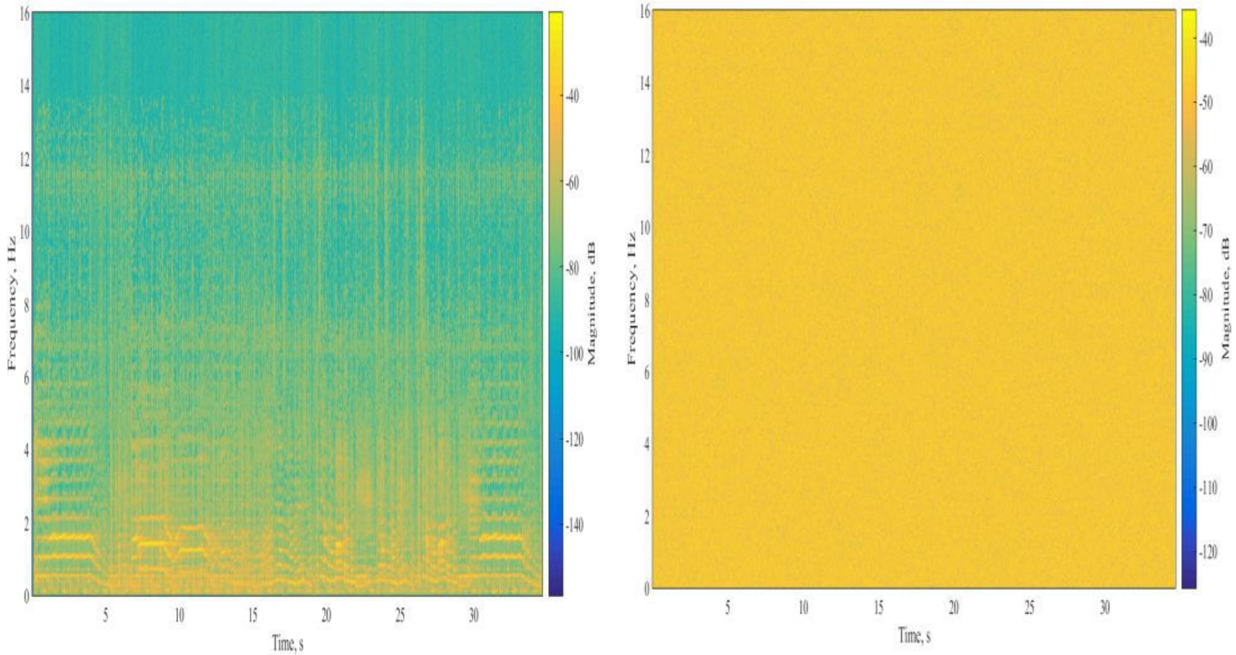
c. Histogram of Figure 6.1.c

d. Histogram of Figure 6.1.d

Figure. 7.2: Histogram Analysis of Fig.6.1

7.3 Spectrogram Analysis

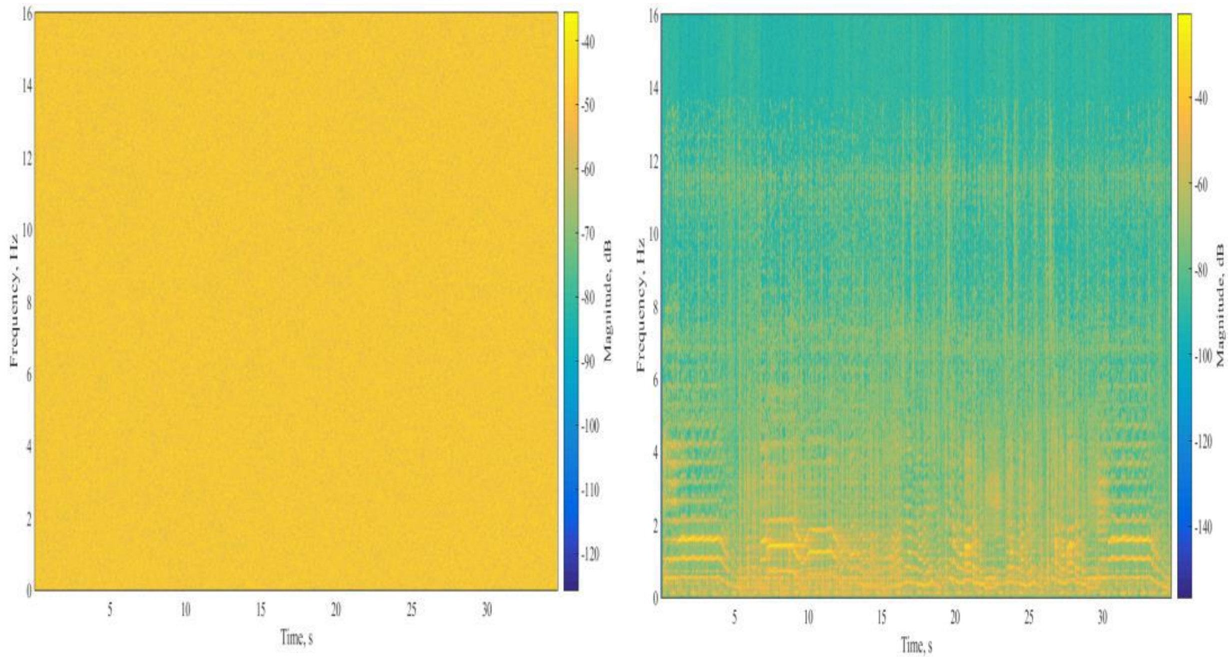
A spectrogram is a visual representation of the spectrum of frequencies of an audio as they vary with time and it is generated using Fourier transformation. Two geometric dimensions- time and frequency are used to represent the spectrogram graph. Here, spectrograms are created from a time-domain signal using Fourier transform. Digitally sampled data, in the time domain, is broken up into chunks which usually overlap and is Fourier transformed to calculate the magnitude of the frequency spectrum for each chunk. The spectrogram of Figure 6.1 is shown in Figure7.3. Figure 7.3.a and Figure 7.3.b show the spectrogram of Audio-1 and EAudio-1 respectively. This proves that there are no similarities between Audio-1 and EAudio-1. Moreover, Figure 7.3.a and Figure 7.3.d are completely same, which proves the lossless reconstruction of the original audio.



a. Spectrogram of Figure 6.1.a

b. Spectrogram of Figure 6.1.b

(I)



c. Spectrogram of Figure 6.1.c

d. Spectrogram of Figure 6.1.d

(II)

Figure 7.3 (I-II): Spectrogram Analysis of Figure 6.1

7.4 Correlation Test

A useful measure to assess the encryption quality of any cryptosystem is the correlation coefficient between the similar segments in the original audio signal and the encrypted audio signal. It is calculated using equation-11.

$$r_{pq} = \frac{cov(p,q)}{\sqrt{D(p)}\times\sqrt{D(q)}} \qquad (11)$$

where *cov(p, q)* is the covariance between the original signal *p* and the encrypted signal q. *D(p)* and *D(q)* are the variances of the signals p and q. *D(p)* and *cov(p,q)* are calculated using equation-12, equation-13 and equation-14 as-

$$D(p) = \frac{1}{N}\sum_{i=1}^{N}(p(i) - E(p))^2 \qquad (12)$$

$$E(p) = \frac{1}{N}\sum_{i=1}^{N} p(i) \tag{13}$$

$$cov(p, q) = \frac{1}{N}\sum_{i=1}^{N}\big(p(i) - E(p)\big)\,(q(i) - E(q)) \tag{14}$$

where N is the length of the audio (or block). The low value of the correlation coefficient r_{pq} shows an encryption of good quality.

Table-7.2 shows the correlation between two consecutive blocks of original audio and encrypted audios. The *2nd*-column shows the correlation between two consecutive blocks of original audio (Audio-1) and the *4th*-column shows the correlation between two consecutive blocks of encrypted audio (EAudio-1). The correlation between two blocks of original and encrypted audios is also shown in the *6th*-column. The results confirm that the correlation between the original and ciphered blocks is close to zero and the correlation between two consecutive ciphered blocks is also close to zero, which conforms the acceptability of proposed scheme.

Table 7.2: Correlation between Consecutive Blocks

Block	***C.V***	***Block***	***C.V***	***Block***	***C.V***
OB1, OB2	1.0000	EB1, EB2	0.0353	OB1, EB2	0.0393
OB2, OB3	1.0000	EB1, EB2	-0.1001	OB2, EB2	-0.0192
OB3, OB4	0.5365	EB1, EB2	0.0233	OB3, EB3	0.0121
…	…	…	…	…	…
OB50, OB51	0.2365	EB50, EB51	0.0034	OB50,EB50	0.0235
OB51, OB52	0.4565	EB51, EB52	0.0125	OB51,EB51	0.0135
OB52, OB53	0.2963	EB52, EB53	0.0223	OB52,EB52	0.1015
OB53, OB54	0.4365	EB53, EB54	0.0115	OB53,EB53	0.0350
OB54, OB55	0.3965	EB54, EB55	0.0013	OB54,EB54	0.0154
OB55, OB56	0.2165	EB55, EB56	0.0203	OB55,EB55	0.0212
…	…	…	…	…	…
OB200, OB201	0.3364	EB200, EB201	0.0027	OB200, EB200	0.0013
OB201, OB202	0.3212	EB201, EB202	0.0113	OB201, EB202	0.0025

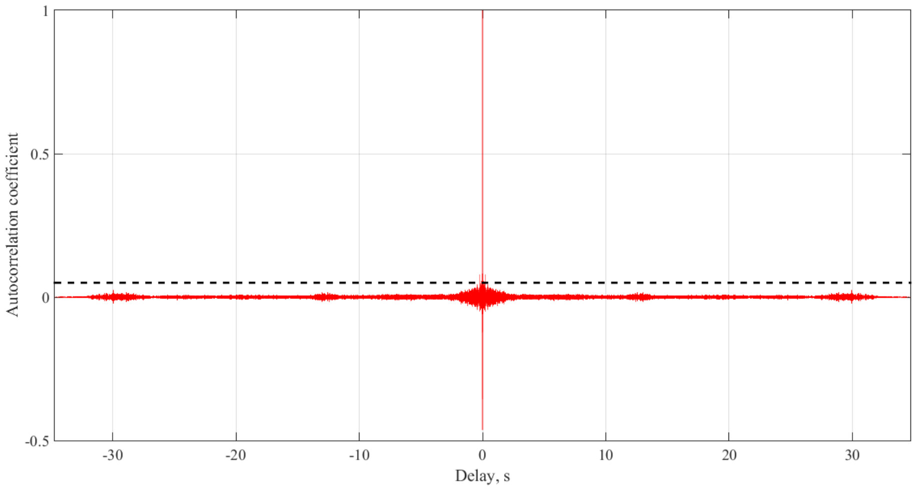
OB = Original block; EB = Encrypted block; C.V. = Correlation value

Table-7.3 shows the correlation values amongst the original, encrypted and decrypted audios. The correlation values between the original and the encrypted audios are close to zero, which proves that there is no statistical similarity between the two audios. Moreover, the correlation value between the original and the decrypted audios (last row) is one, which proves lossless decryption of the secret audio.

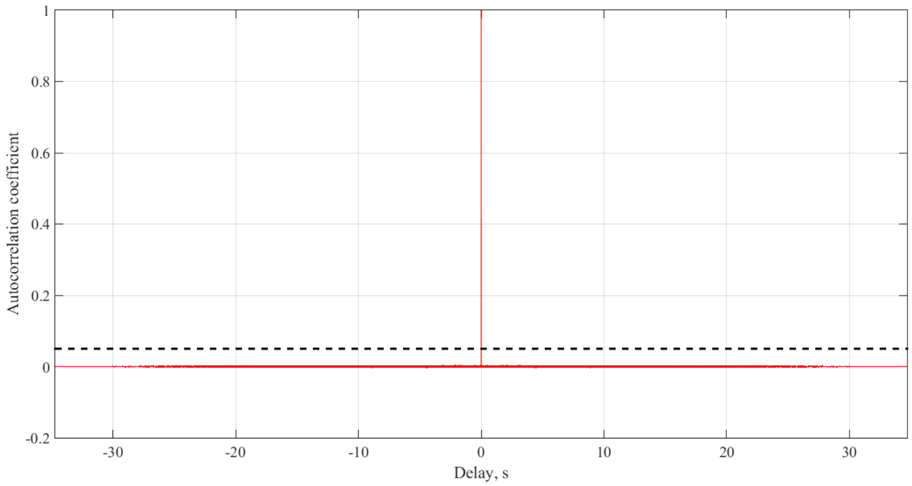
Table 7.3: Correlation Values among Original, Encrypted & Decrypted Audios

Original & Encrypted Audios	***Correlation value***
Audio-1 & EAudio-1	0.000852
Audio-2 & EAudio-2	0.000207
Audio-3 & EAudio-3	0.00108
Audio-4 & EAudio-4	-0.00216
Audio-5 & EAudio-5	-0.00230
Audio-6 & EAudio-6	-0.00083
Audio-7 & EAudio-7	-0.00023
Audio-8 & EAudio-8	-0.00154
Audio-9 & EAudio-9	-0.000208
Audio-10 & EAudio-10	-0.00034
Original & Decrypted Audios	
Audio-1 & DAudio-1	1.000

The autocorrelation coefficient analysis is shown in Figure 7.4 between Audio-1(Figure 6.1.a) and EAudio-1(Figure 6.1.b), which shows the correlation coefficients of the encrypted audio are completely different from the original audio.



a. Autocorrelation coefficient of Figure 6.1.a



b. Autocorrelation coefficient of Figure 6.1.b

Figure7.4 (a-b): Autocorrelation Coefficient Analysis of Figure.6.1.a & Figure.6.1.b

7.5 Number of Samples Change Rate (NSCR)

The major requirement of all the encryption techniques is that the encrypted data should be greatly different from its original form. The encrypted audio signals are compared by the number of samples change rate (NSCR) from the original audio, which is defined in equation-15.

$$NSCR = \frac{\sum_i D_i}{L} \times 100\% \qquad (15)$$

where *L* represents the length of the audio vector and *Di* is determined according to the rule

$$D_i = \begin{cases} 0, A_i = A_i' \\ 1, A_i \neq A_i' \end{cases} \qquad (16)$$

The NSCR values between two audios using equation-15 are shown in table- 7.4. The high value of NSCR means that the signal values are dramatically randomized, which indicates that the original audio and the encrypted audio are significantly different from one another, so the proposed algorithm is highly resistive against differential attacks. The last row of table-7.4 shows the NSCR value as zero between the original audio and the decrypted audio, which proves lossless reconstruction of the secret audio.

Table 7.4: NSCR Values among Original, Encrypted & Decrypted Audios

Original & Encrypted Audios	***NSCR value***
Audio-1 & EAudio-1	99.9977
Audio-2 & EAudio-2	99.9989
Audio-3 & EAudio-3	99.9987
Audio-4 & EAudio-4	99.9986
Audio-5 & EAudio-5	99.9994
Audio-6 & EAudio-6	99.9982
Audio-7 & EAudio-7	99.9980
Audio-8 & EAudio-8	99.9983
Audio-9 & EAudio-9	99.9985
Audio-10 & EAudio-10	99.9988
Original & Decrypted Audios	
Audio-1 & DAudio-1	0.000

7.6 Root Mean Square (RMS) & Crest Factor (CF) value

Root mean square (RMS) value is measured as the average amplitude level of an audio signal. The RMS is equivalent to standard deviation when the input signal has zero mean and it is calculated using equation-17 for an audio signal A of length N.

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|A_i|^2} \quad (17)$$

Crest factor (CF) is a parameter of a waveform, such as that of alternating current or sound, showing the ratio of peak values to the effective value. In other words, crest factor indicates how extreme the peaks are in a waveform. The CF is calculated using equation-18.

$$CF = 20\log_{10}\frac{|A_{Peak}|}{A_{RMS}} \quad (18)$$

Table 7.5 shows the RMS and CF values for different audios and it shows that the RMS and CF values for all of the encrypted audios are close to 0.6 and 4.8 respectively, irrespective of the original audios. Figure 7.5 is also used to demonstrate the RMS and CF values for different audios. This result proves that there are no statistical relationship between the original and the encrypted audios. Moreover, the 3rd-row of Table 7.5 indicates lossless decryption of secret audios because the generated values are completely same as to the original one.

Table 7.5: RMS Value of Original and Encrypted Audio

Audio	***Size***	***RMS(db)***	***CF(db)***
Audio-1	2.11MB	0.25332	11.9251
EAudio-1	2.11MB	0.57731	4.7715
DAudio-1	2.11MB	0.25332	11.9251
Audio-2	4.69MB	0.19403	14.2411
EAudio-2	4.69MB	0.57688	4.778
Audio-3	918KB	0.24892	11.0925
EAudio-3	918KB	0.5786	4.7621
Audio-4	162KB	0.21442	12.9095
EAudio-4	162KB	0.57821	4.775
Audio-5	366KB	0.2840	10.9332
EAudio-5	366KB	0.57689	4.779
Audio-6	439KB	0.04721	25.5492
EAudio-6	439KB	0.57826	4.7673
Audio-7	1.9MB	0.09485	15.6862
EAudio-7	1.9MB	0.57743	4.7699
Audio-8	1.93MB	0.14083	17.0259
EAudio-8	1.93MB	0.57730	4.7717
Audio-9	778KB	0.34666	9.2017
EAudio-9	778KB	0.57743	4.7698
Audio-10	459KB	0.20778	12.4846
EAudio-10	459KB	0.57775	4.7750

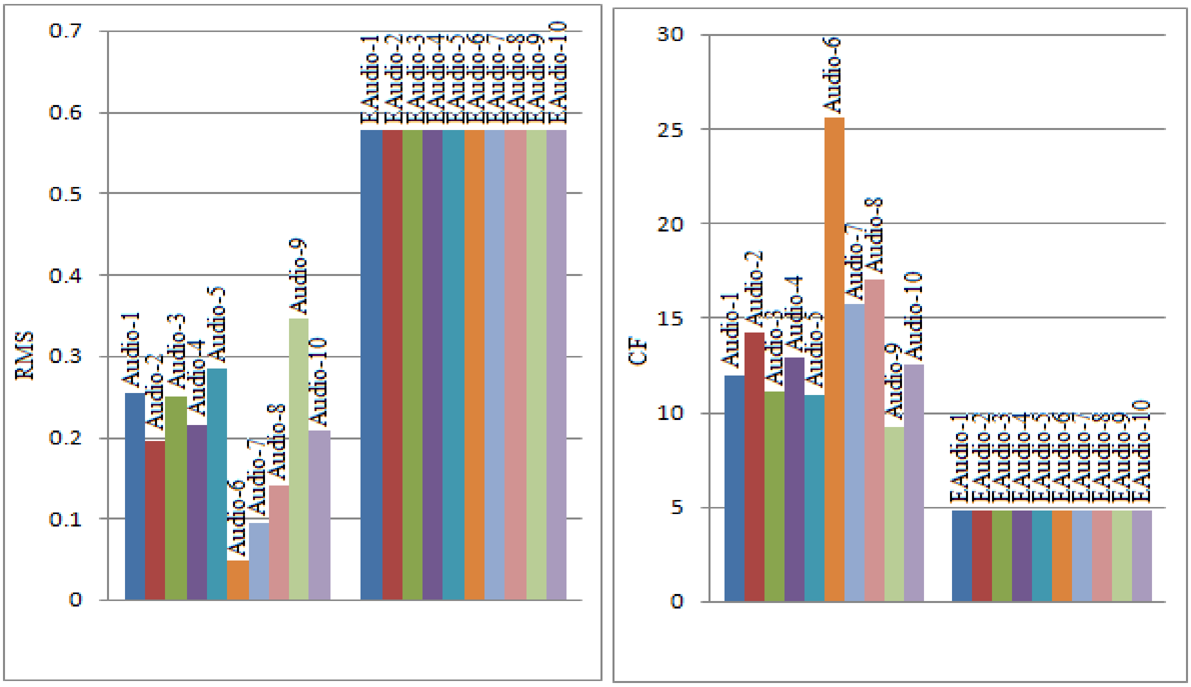


Figure 7.5: RMS & CF Values for Different Audios

7.7 Comparison with Previous Schemes

Performance analysis with three other audio encryption schemes namely Lima and Neto [18], Sathiyamurthi & Ramakrishnan [26] and Liu et al.[102] along with our proposed scheme is depicted in table-7.6.

Here, Audio-2 is selected as the secret audio and EAudio-2 is the corresponding encrypted audio, which are tested using four functionalities like correlation, NSCR, CF and lossless reconstruction.

Table 7.6: Comparisons of the related Audio Encryption Schemes

Functionality	***Ref. [30]***	***Ref. [36]***	***Ref. [98]***	***Proposed Scheme***
Correlation	0.0021	0.000358	0.0119	0.000207
NSCR	99.9992	99.9978	99.9996	99.9989
CF value(dB)	≅ 4.8	≅ 4.8	≅ 4.8	≅ 4.8
Lossless Reconstruction	Yes	Yes	No	Yes

Table-7.6 clearly indicates that the proposed scheme has either better or similar effect as compared to the above mentioned schemes.

Chapter 8

Application on Image

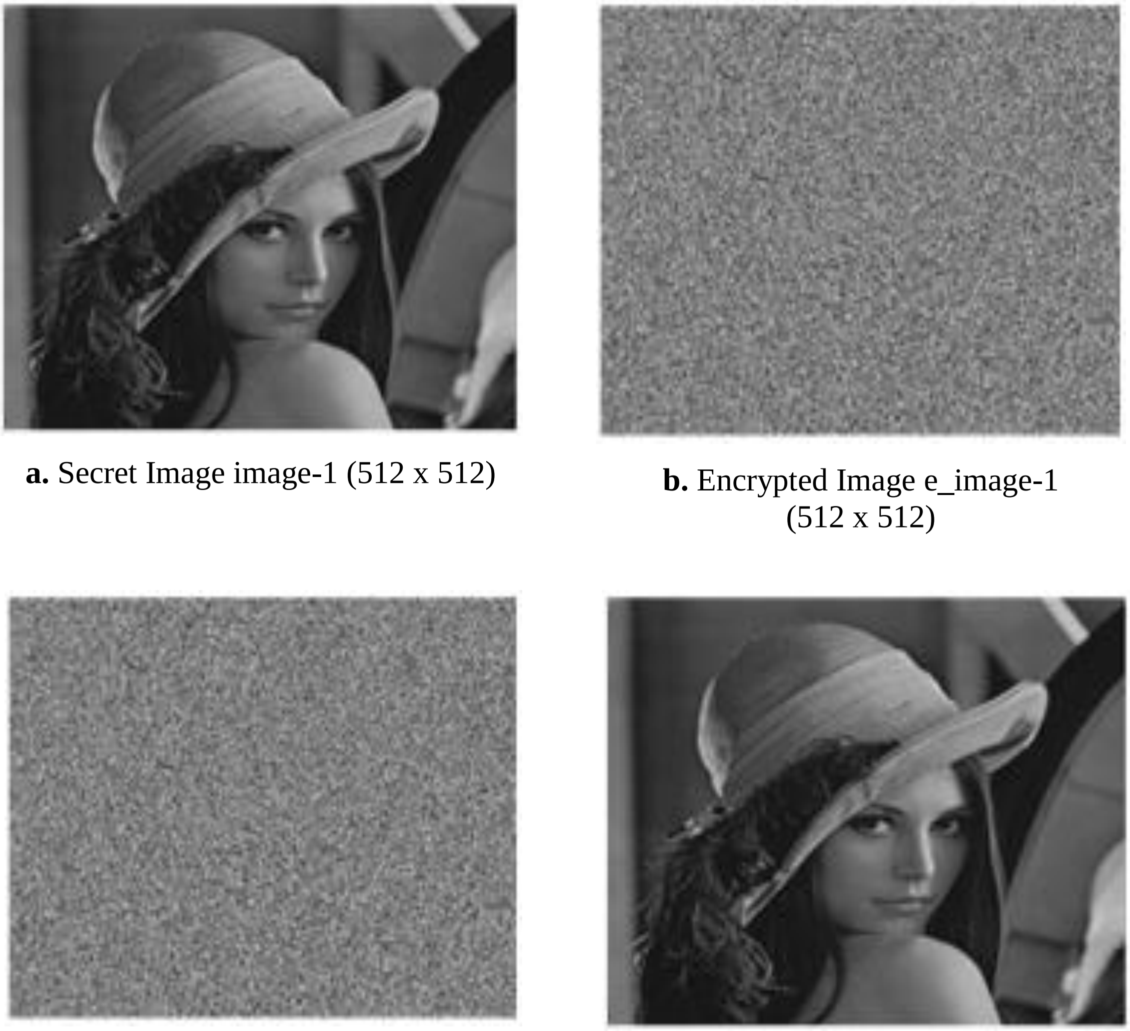
We already shown that all experimental result and different statistical analysis for applying our proposed scheme on audio. Now we will see the result for applying our proposed scheme on image. Nowadays, safeguard of a secret image is an important issue. Image encryption is completely different from text encryption, because an image has some intrinsic features, such as bulk data capacity and high correlation among pixels. As a result, traditional encryption algorithm such as DES, TDEA, AES and RSA are not suitable to encrypt digital images. Although the traditional encryption algorithms necessitate longer time to directly encrypt a secret image. In literature, various image encryption techniques are present to encrypt and decrypt data, but there is no solitary encryption algorithm to satisfy the different image formats. Generally, the values of the neighboring pixels of an image are highly correlated.

8.1 Experimental Result

Experimental result of encryption of a gray scale image for applying our previously mentioned proposed scheme (which we already applied for ten different audios), are shown in

Figure 8.1. Figure 8.1.a shows secret gray scale image-1 of size (512 x 512), which is encrypted using proper key to generate an encrypted image e_image-1 (Figure 8.1.b). Figure 8.1.c shows a decrypted image d_image-1, construction using wrong key, whereas Figure 8.1.d shows lossless decrypted image, d_image-1 using proper key.

Two other experimental results are also shown in Figure 8.2 for different images with different sizes and different encryption keys.



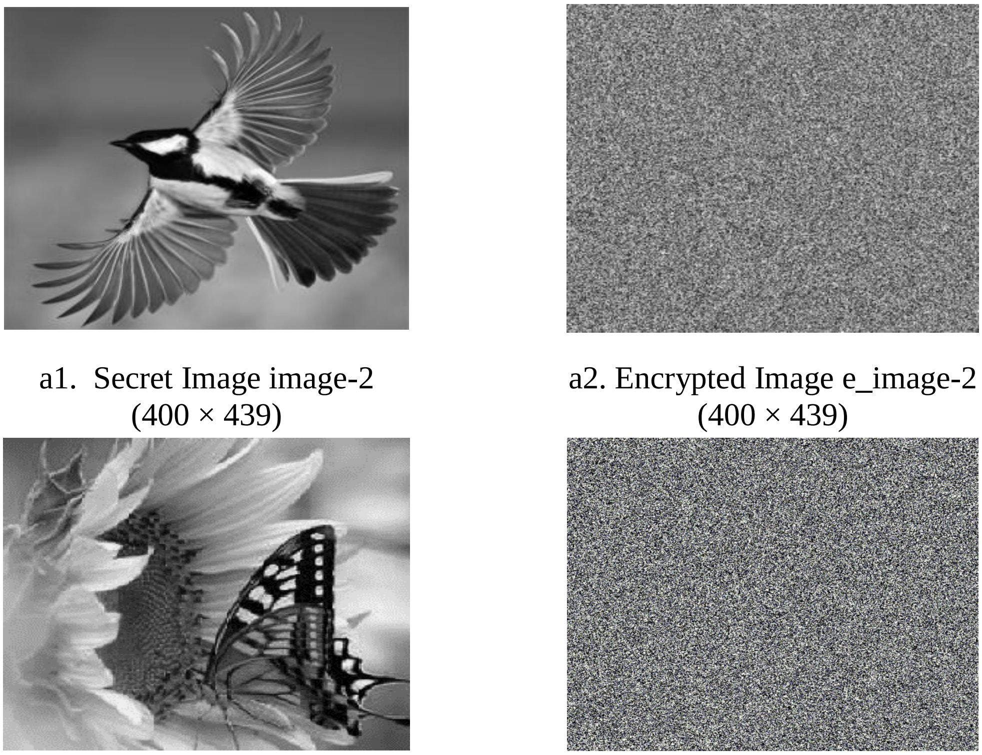
a. Secret Image image-1 (512 x 512)

b. Encrypted Image e_image-1 (512 x 512)

c. Decrypted Image d_ image-1 using wrong key (512 x 512)

d. Decrypted Image d_ image-1 using proper key (512 x 512)

Figure 8.1: Gray Image Encryption & Decryption



a1. Secret Image image-2 (400 × 439)

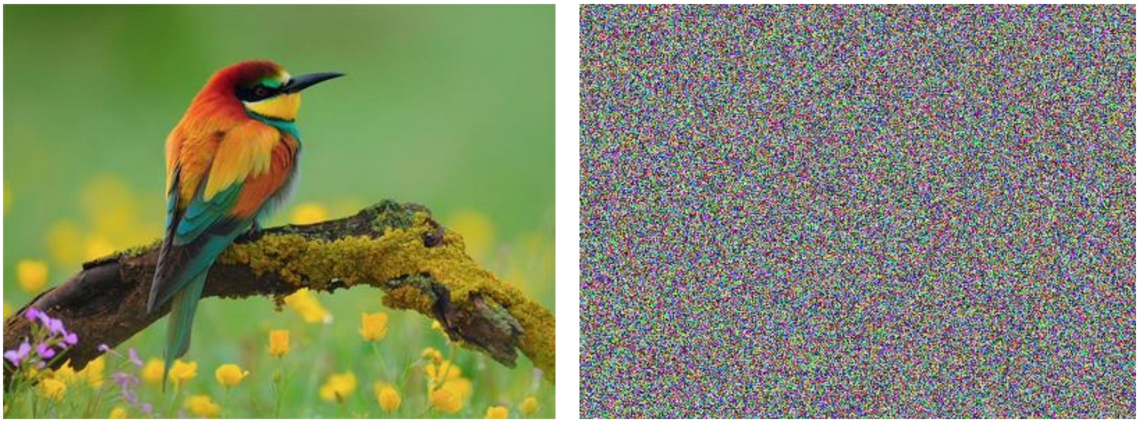
a2. Encrypted Image e_image-2 (400 × 439)

b1. Secret Image image-3 (453 × 395)

b2. Encrypted Image e_image-3 (453 × 395)

Figure 8.2: Encryption of Two Different Gray Image.

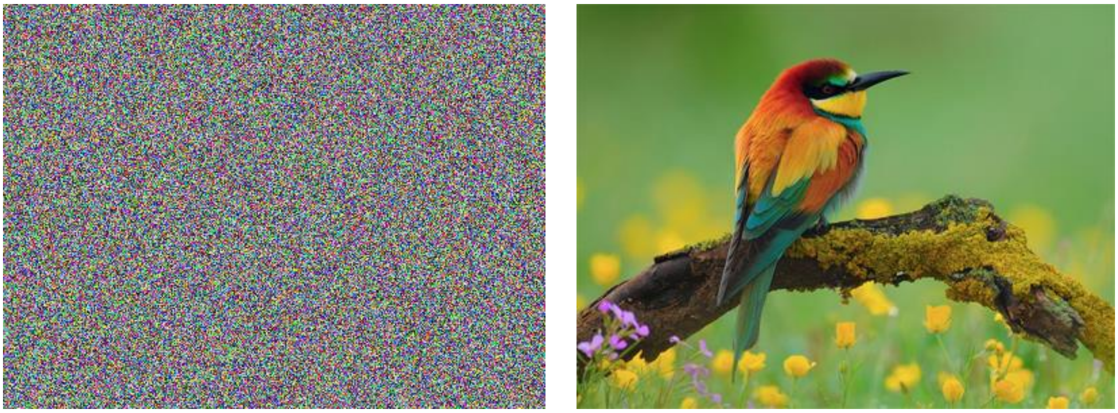
Now we will see experimental result for RGB image, which is shown in Figure 8.3.



a. Secret Image image-1 (400 x 300)

b. Encrypted Image e_image-1 (400 x 300)

(I)



c. Noise Image n_image-1 (400 x 300)

d. Decrypted Image d_image-1 (400 x 300)

(II)

Figure 8.3 (I-II): RGB Image Encryption & Decryption

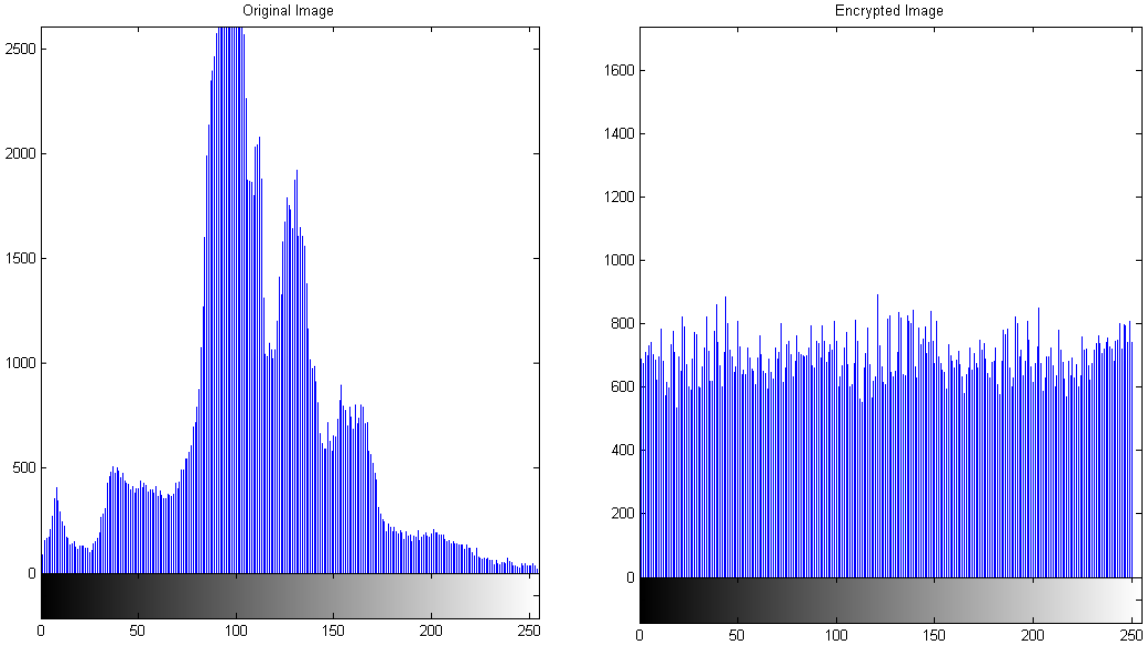
8.2 Strength and Security Analysis

8.2.1 Histogram Analysis

Now we will discuss corresponding histogram analysis of gray image and RGB image for applying same proposed scheme which we have already applied in audio encryption. Here the histogram analysis clarifies how pixels in an image are distributed by plotting the number of pixels at each intensity level.

The histogram of encrypted image has uniform distribution which is significantly different from original image and has no statistical similarity in appearance.

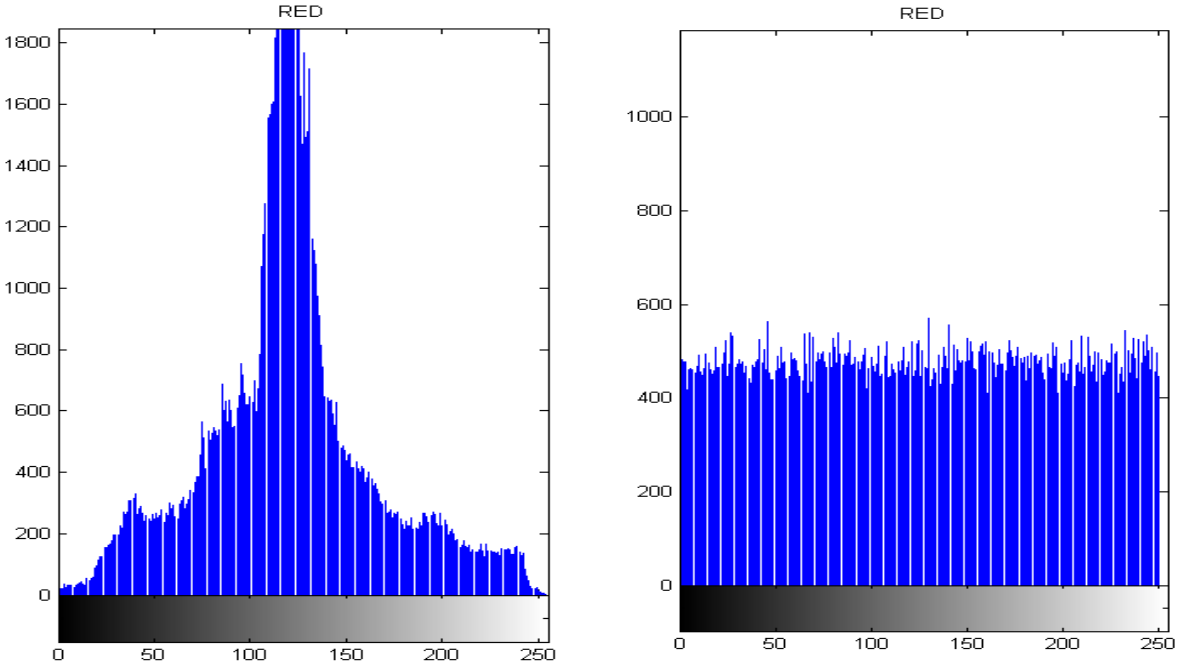
Figure 8.4 shows histogram of gray secret image and encrypted image, where Figure 8.4 (a) shows the histogram of original gray secret image Figure 8.1(a) and Figure 8.4 (b) shows the histogram of gray encrypted image Figure 8.1(b). Figure 8.5 shows histogram of RGB secret image and encrypted image, where figure 8.5 (a) shows the histogram of red plane of Figure 8.3 (a), Figure 8.5 (b) shows the histogram of red plane of Figure 8.3 (b), figure 8.5 (c) shows the histogram of green plane of Figure 8.3 (a), Figure 8.5 (d) shows the histogram of green plane of Figure 8.3 (b), Figure 8.5 (e) shows the histogram of blue plane of Figure 8.3 (a), Figure 8.5 (f) shows the histogram of blue plane of Figure 8.3 (b).



a. Histogram of gray original secret image (Figure 8.1.a)

b. Histogram of gray encrypted image (Figure 8.1.b)

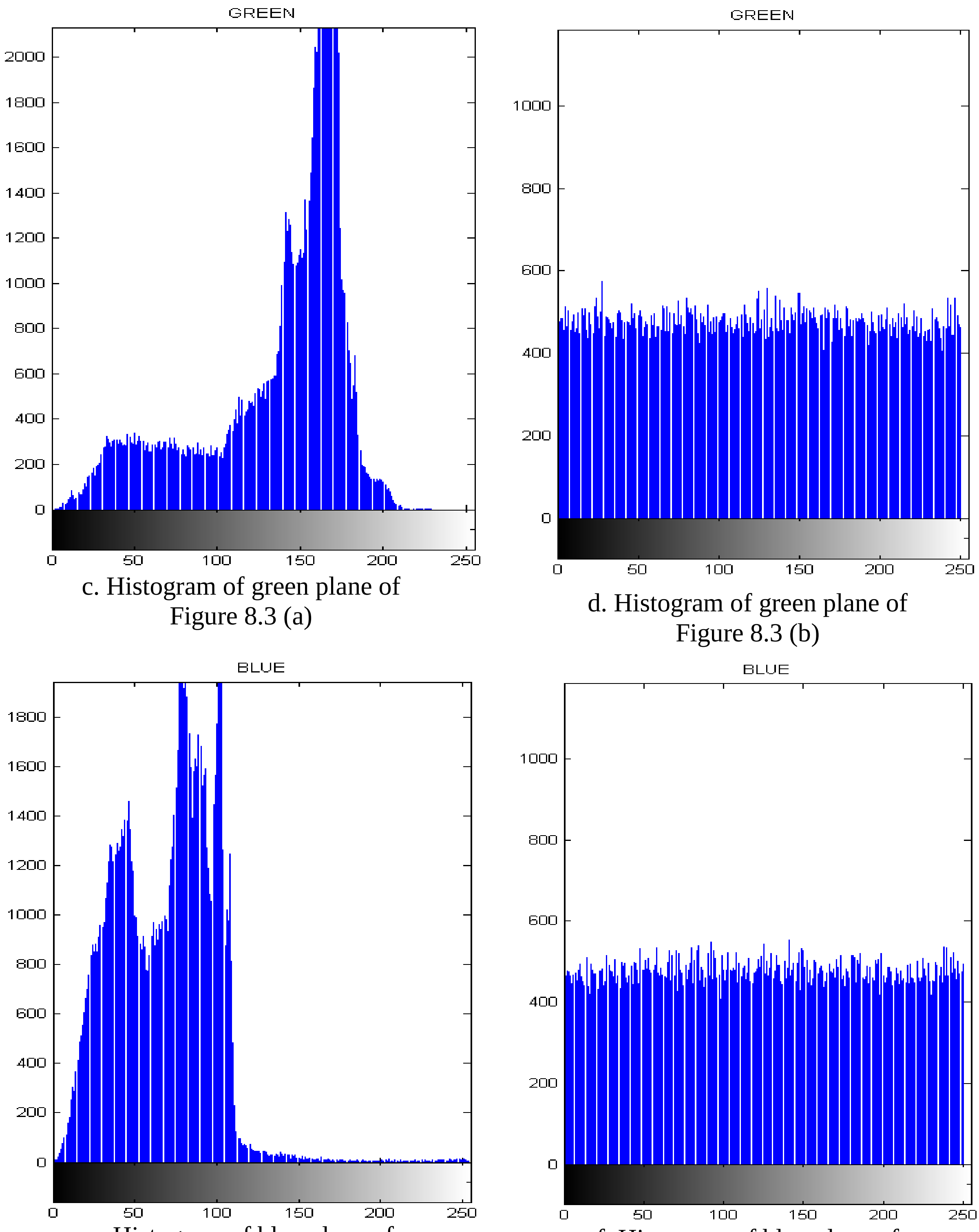
Figure 8.4 Histogram of Original Gray Secret Image and it's Encrypted Image



a. Histogram of red plane of Figure 8.3 (a)

b. Histogram of red plane of Figure 8.3 (b)

(I)



c. Histogram of green plane of Figure 8.3 (a)

d. Histogram of green plane of Figure 8.3 (b)

e. Histogram of blue plane of Figure 8.3 (a)

f. Histogram of blue plane of Figure 8.3 (b)

(II)

Figure 8.5(I-II): Histogram of Different Plane (Red, Green, Blue) of Original RGB Secret Image and its Corresponding Encrypted Image

8.2.2 Chi-square test

In security analysis, chi-square test is used to examine the non-homogeneity between the original and encrypted image. This statistical test is used to examine the variations of data from the expected value. It is defined as:

$$\chi^2 = \sum_{i=0}^{255} \frac{(O_i - E_i)^2}{E_i} \qquad (18)$$

where *i* is the number of gray level, O_i and E_i are observed and expected frequency of each gray level (0 to 255), respectively. The values of χ^2 for original image and encrypted images are depicted in Table 2. Result shows the low value of χ^2 for encrypted images compared to the original one. These lower values of χ^2 indicate that the proposed scheme provides fairly good encryption effect.

Table 8.1: Chi-Square Values for Original and Encrypted Images

Original image	***χ2***	***Encrypted image***	***χ2***
Figure 8.1(a)	244,688.97	Figure 8.1(b)	313.35
Figure 8.2(a1)	562,348.17	Figure 8.2(a2)	224.78
Figure 8.2(b1)	161,973.18	Figure 8.2(b2)	307.16

8.2.3 Differential Attack

The major requirement of all the encryption techniques is that the encrypted image should be greatly different from its original form. Two measures are adopted to quantify this requirement. They are number of pixel change rate (NPCR) and unified average changing intensity (UACI). The NPCR is used to measure the number of pixels in difference of gray level in two images. Let $P(i, j)$ and $P'(i, j)$ be the *i*th row and *j*th column pixel of two images *P* and *P′*, respectively, the NPCR can be defined as

$$NPCR = \frac{\sum_{i,j} C(i,j)}{N} \times 100 \qquad (19)$$

where N is the total number of pixels in the image and C(i, j) is defined as

$$C(i,j) = \begin{cases} 0 & P(i,j)=P'(i,j) \\ 1 & P(i,j)\neq P'(i,j) \end{cases}$$

the NPCR values for various images using equation-20 are shown in table-8.2 The high value of NPCR means the pixel values are dramatically randomised. This result indicates that the original image and the encrypted image are significantly different from one another, so the proposed algorithm is highly resistive against differential attack. Another quantity, UACI measures the average intensity differences between the two images. It can be defined as

$$UACI = \frac{1}{N}\left[\sum_{i,j}\frac{|P(i,j)-P'(i,j)|}{255}\right] \times 100 \qquad (20)$$

Table 8.2: NPCR and UACI Value for Different Images

Original vs. Encrypted	***NPCR***	***UACI***
Figure 8.1(a) and Figure 8.1(b)	99.529	11.676
Figure 8.2(a1) and Figure 8.2(a2)	99.616	22.671
Figure 8.2(b1) and Figure 8.2(b2)	99.583	21.938
Original vs. Decrypted		
Figure 8.1(a) and Figure 8.1(d)	0.00	0.00

Two quantities, NPCR and UACI are calculated for various images using equation-19 and equation-20, respectively. The test results for NPCR and UACI are shown in table 8.2.

Higher value of NPCR and lower value of UACI indicate the absence of any probable statistical relationship between original image and encrypted image. It may be noted that the last row of table 8.2 shows that the result of NPCR and UACI are zero which proves lossless reconstruction of secret image.

8.2.4 Correlation Analysis

A secure encryption scheme should generate a completely noisy encrypted image independent of the original secret image. Therefore, they must have a very low correlation coefficient which should be very closer to zero. Here, we calculate the correlation between original and encrypted image using equation-21.

$$r = \frac{\sum_i \sum_j (M_{ij} - \bar{M})(N_{i,j} - \bar{N})}{\sqrt{\left(\sum_i \sum_j (M_{ij} - \bar{M})^2\right)\left(\sum_i \sum_j (N_{i,j} - \bar{N})^2\right)}} \tag{21}$$

where M and N are two images of same size and M and N indicate the mean of the images M and N, respectively. A low value of correlation coefficient shows that there is no straight relation between the original and encrypted images. Following table 8.3 shows the correlation value between gray scale images and encrypted images.

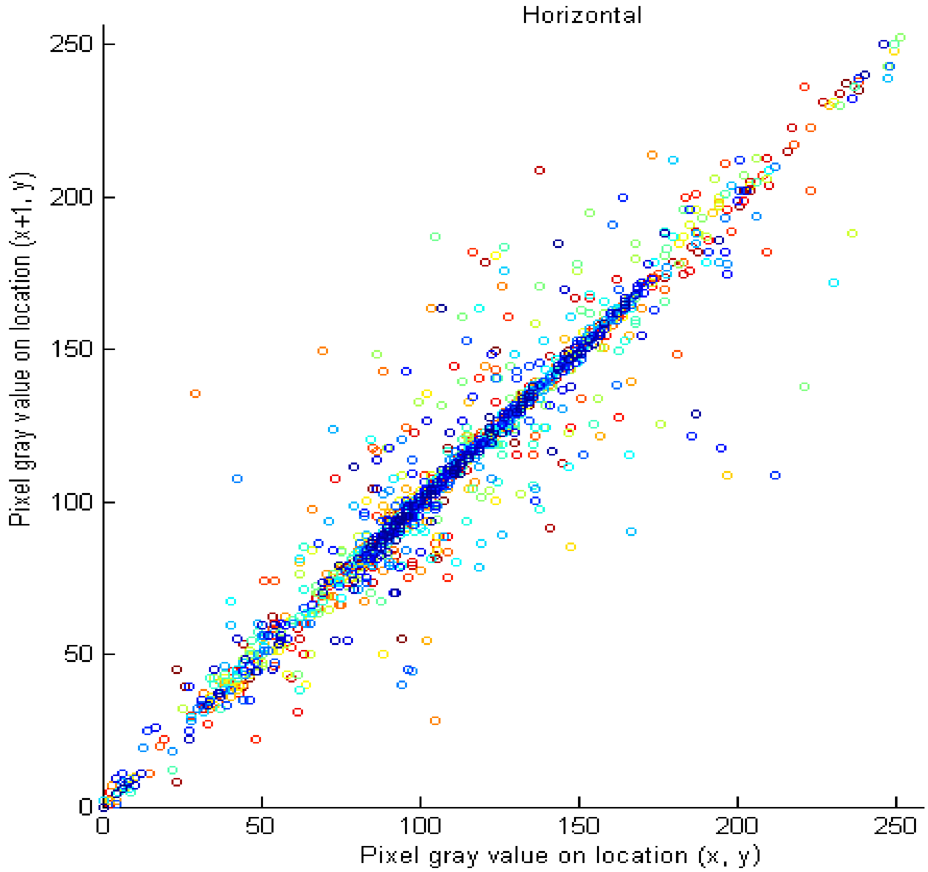
Table 8.3 Correlation Value for Different Images

Original vs. Encrypted	***Correlation value***
Figure 8.1(a) and Figure 8.1(b)	–0.0046
Figure 8.2(a1) and Figure 8.2(a2)	–0.0023
Figure 8.2(b1) and Figure 8.2(b2)	0.0018
Figure 8.1(a) and Figure 8.1(d)	1.0000

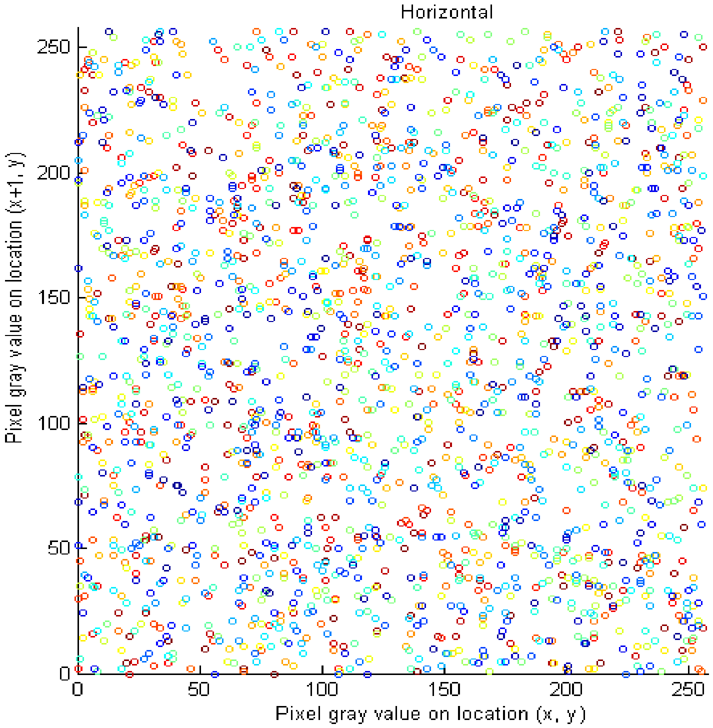
Table 8.4 Correlation Coefficients of Two Adjacent Pixels in Original and Encrypted Image

	Original image [Figure 8.1(a)]	***Encrypted image [Figure 8.1(b)]***	***Original image [Figure 8.2(a1)]***	***Encrypted image [Figure 8.2(b1)]***
Vertical	0.9533	–0.0018	0.9770	–0.0007
Horizontal	0.9503	–0.0003	0.9691	0.0024
Diagonal	0.9166	0.0010	0.9562	–0.0030

The correlation coefficients are used to measure the degree of similarities between two images. Higher correlation value implies better match between two images. Above result shows, that correlation value between original image and encrypted image is very low. Therefore, encrypted image is completely different from the original one. On the other hand, we get the correlation value one for original image [figure 8.1(a)] and decrypted image [figure 8.1(c)], which proves lossless reconstruction of the secret image. The correlation distributions of horizontally, vertically and diagonally adjacent pixels of original image as well as its encrypted image are shown in figure 8.6 (a1) & figure 8.6 (a2), figure 8.6 (b1) & figure 8.6 (b2) and Figure 8.6 (c1) & figure 8.6 (c2) respectively. Table 8.4 shows correlation coefficient of two adjacent pixels in original image and encrypted image.

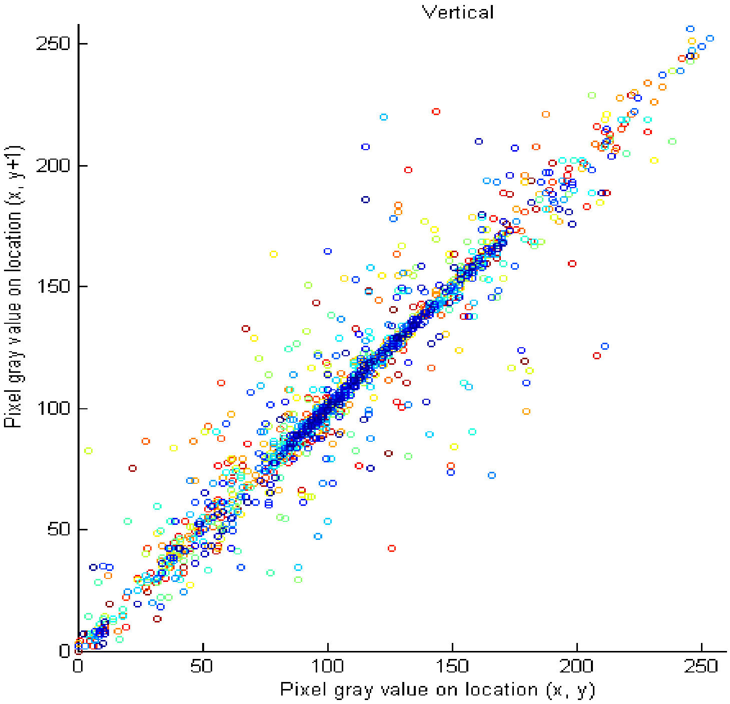


a1. Correlation distributions of horizontally pixels of figure 8.1 (a)

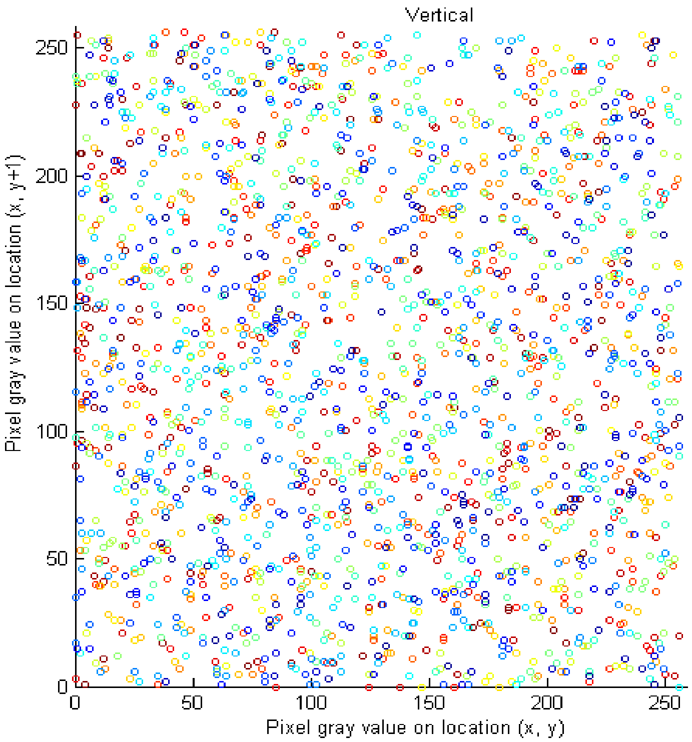


a2. Correlation distributions of horizontally pixels of figure 8.1 (b)

(a)

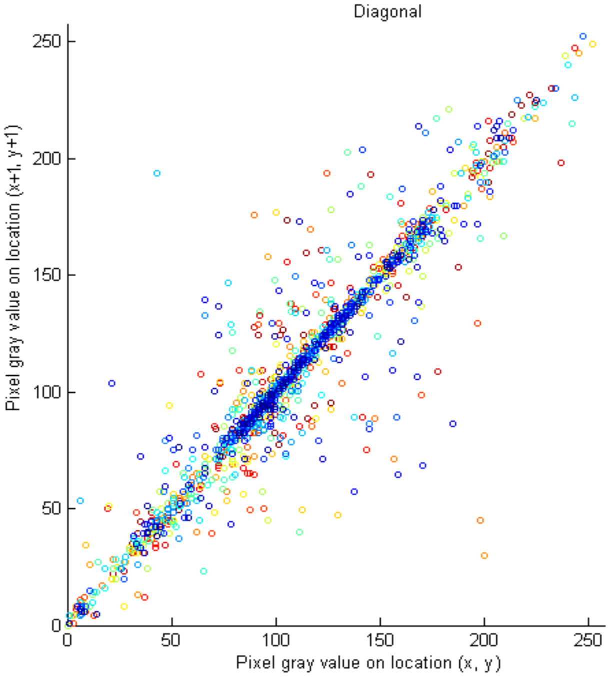


b1. Correlation distributions of vertical pixels of figure 8.1 (a)

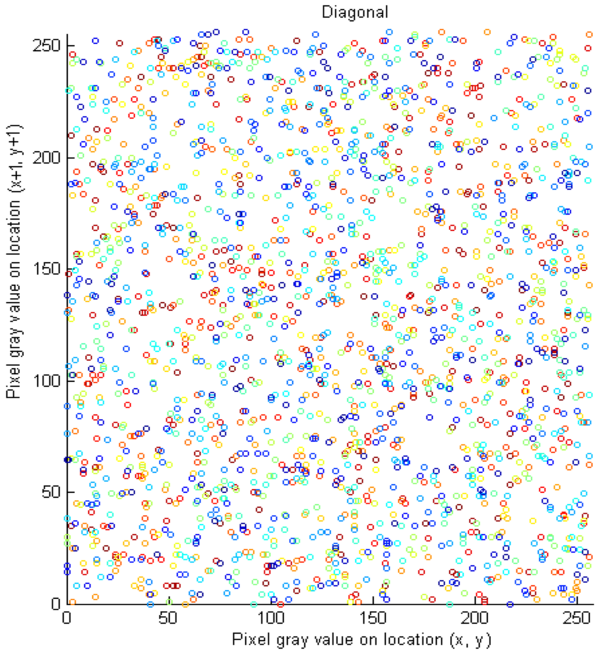


b2. Correlation distributions of vertical pixels of figure 8.1 (b)

(b)



c1. Correlation distributions of diagonal pixels of figure 8.1 (a)



c2. Correlation distributions of diagonal pixels of figure 8.1 (b)
(c)

Figure 8.6(a-b-c): Correlation Distributions of Horizontally, Vertically and Diagonally Adjacent Pixels of Original Image as well as its Encrypted Image

8.2.5 MSE and PSNR Measure

The MSE and PSNR for the proposed technique have been computed for different images. The high value of MSE and low value of PSNR causes the resulting encrypted image more robust. MSE is calculated using equation-22.

$$MSE = \frac{1}{MN} \sum_{i=1}^{N} \sum_{j=1}^{M} [C(i,j) - C'(i,j)]^2 \tag{22}$$

where $C(i, j)$ and C' (i, j) be the i^{th} row and j^{th} column pixel of two images C and C' , respectively. M and N are number of rows and columns of original image. PSNR can be computed by equation-23.

$$PSNR = 10 \times \log_{10} \left[\frac{R^2}{MSE}\right] \tag{23}$$

where value of *R* is 255 and gray image is used in this experiment. Calculated results of MSE and PSNR are tabulated in the following table.

Table 8.5: MSE and PSNR Value of Different Images

Original vs. Encrypted	***MSE***	***PSNR***
Figure 8.1(a) and Figure8.1(b)	230.116	24.511
Figure 8.2(a1) and Figure 8.2(a2)	234.110	24.436
Figure 8.2(b1) and Figure 8.2(b2)	234.378	24.431
Original vs. Decrypted		
Figure 8.1(a) and Figure 8.1(d)	0.00	∞

8.2.6 Entropy Analysis

Entropy analysis is an essential statistical measurement, which is used to test the robustness of an image encryption algorithm. Entropy measurement of a source *k* is defined as,

$$H(k) = \sum_i P(K_i) \log_2 \frac{1}{P(k_i)} \qquad (24)$$

where $P(k_i)$ represents the probability of the pixel value *xi*. For an image there could be 2^8 (00 to FF) symbols. If the probability of occurrence of each pixel value is same, according to the equation-24 entropy value will be $H(k) = 8$. This will be the maximum entropy for an image having truly uniform pixel distribution. An encrypted image will be considered robust if its entropy tends to the value 8. Therefore, higher the entropy value of an encrypted image betters the security.

Table 8.6: Chi-Square Values for Original and Encrypted Images

Original image	***Entropy***	***Encrypted image***	***Entropy***
Figure 8.1(a)	7.224	Figure 8.1(b)	7.963
Figure 8.2(a1)	7.087	Figure 8.2(a2)	7.991
Figure 8.2(b1)	7.280	Figure 8.2(b2)	7.990
Figure 8.3(a)	7.779	Figure 8.3(d)	7.989

Table 8.6 shows entropy of all encrypted images are close to 8, irrespective of entropy of original images, thus the proposed algorithm is robust enough.

8.2.7 Comparison with Other Image Encryption Schemes

Performance comparison of three most recent schemes (Zhang and Wang, 2015; Zhou et al., 2014; Sathiskumar and Bagan, 2011) along with our scheme is depicted in Table 8. Here, Figure 8.2(a1) is taken as secret image and Figure 8.2(a2) is the corresponding encrypted image. Table 8.7 clearly indicates that the proposed scheme has either similar or better effect compared to the recently proposed schemes.

Table 8.7: Comparison with Other Image Encryption Schemes

	Ref. [103]	***Ref. [104]***	***Ref. [105]***	***Our scheme***
Horizontal	0.0024	0.0006	0.0028	0.0024
Vertical	–0.0006	0.0005	0.0062	–0.0007
Diagonal	0.0012	0.0025	0.0058	–0.0030
NPCR	99.78	99.62	99.63	99.616
UACI	33.57	13.89	13.58	22.671
Entropy	7.987	7.997	7.972	7.991

Chapter 9

Conclusion

This scheme is a secured digital content encryption technique that depends on block ciphering followed by channel shuffling, where both the size of the key and block is 32 bytes. The block ciphering is based on the concept of DNA encoding with logistic chaotic map. Next the concept of channel shuffling is used to make the cipher more secure against cryptanalysis. Moreover, the entire ciphering and shuffling operations are based on 32 bytes key chain, which is changed during the encryption of each block of secret digital content, which provides additional protection to the secret digital content. Here, different statistical analysis are performed on the scheme by taking different sized digital contents. The results obtained are found to be satisfactory which proves the strength of this scheme against cryptanalysis. It may be noted that correlation between original and ciphered block is close to zero and NSCR value is close to 100. Again correlation between two consecutive ciphered blocks is also close to zero. Moreover, key sensitivity analysis, histogram analysis and statistical analysis are performed to establish the strength of my algorithm against cryptanalysis.

The presented algorithm is applicable mainly for off-line digital content (audio & image) although nowadays there is a demand for live encrypted digital content streaming.

Thus, in future, effort may be given to make this algorithm faster so as to extend this application for live digital content streaming.

References

1. T. Rajani Devi. *"Importance of Cryptography in Network Security"*. International Conference on Communication Systems and Network Technologies, pages: 462 – 467, 2013.

2. L. Liu, Q. Zhang, X. Wei. *"A RGB image encryption algorithm based on DNA encoding and chaos map"*Computers and Electrical Engineering, vol.: 38, issue: 5, pages: 1240–1248, 2012.

3. M. Kaur and Ms. S. Kaur. *"Survey of Various Encryption Techniques for Audio Data"*. International Journal of Advanced Research in Computer Science and Software Engineering, vol.: 4, issue: 5, pages: 1314-1317, 2014.

4. S. Rajanarayanan and A. Pushparaghavan. *"Recent Developments in Signal Encryption – A Critical Survey".* International Journal of Computational Intelligence and Informatics,vol.: 1, no.: 3, pages: 208-2015, 2011.

5. M. Mathur, A. Kesarwani. *"Comparison between DES, 3DES, RC2, RC6, BLOWFISH AND AES"*. Proceedings of National Conference on New Horizons in IT – NCNHIT, pages: 143-148, 2013.

6. Ganesh Babu.S, Ilango.P. *"Higher dimensional chaos for audio encryption"*. IEEE, pages: 52-58, 2013.

7. M. Arnold. *"Audio watermarking: features, applications and algorithms"*. IEEE, vol.: 3, pages: 1013-1016, 2000.

8. B. Hamdi, S. Hassene. *"A new Approach Combining speech Chaotic Encryption with fragile Image Watermarking For audio securing and intrusion detection"*. ICEESA, pages: 1-6, 2013.

9. Sruthi B. Asok , P. Karthigaikumar, Sandhya R, Naveen Jarold K and N.M Siva Mangai.*"A secure cryptographic scheme for audio signals"*. International conference on Communication and Signal Processing,pages: 748-752, 2013.

10. H. Wang, M. Hempel, D. Peng, W. Wang, H.Sharif and H.H. Chen. *"Index-based selective encryption for wireless multimedia networks".* IEEE,vol.: 12, issue: 3, 2010.

11. A. A. Tamimi and A. M. Abdalla. *"An Audio Shuffle-Encryption Algorithm"*. World Congress on Engineering and Computer Science (WCECS), vol.: I, 2014.

12. N. J. Thorwirth, P. Horvatic, R. Weis, and J. Zhao. *"Security methods for mp3 music delivery".* Thirty-Fourth Asilomar Conference on Signal, Systems and Computers, pages: 1831-1835, 2002.

13. M. Grangetto, E. Magli, and G. Olmo. *"Multimedia selective encryption by means of randomized arithmetic coding",* IEEE Transactions on Multimedia, vol.: 8, no.: 5, pages: 905-917, 2006.

14. W.Q. Yan., W.G. Fu, and M. S. Kankanhalli. *"Progressive audio scrambling in compressed domain".* IEEE Transaction on Multimedia, vol.:10, no.: 6, pages: 960-968, 2008.

15. J. Zhou and O. C. Au. *"Security and efficiency analysis of progressive audio scrambling in compressed domain".* IEEE International Conference on Acoustics, Speech and Signal Processing, pages 1802-1805, 2010.

16. M. A. Yakubu, N. C. Maddage, and P. K. Atrey. *"Audio secret management scheme using shamir's secret sharing".* International Conference on Multimedia Modeling. LNC, pages: 396-407, 2015.

17. P. K. Naskar, H. N. Khan, U. Roy, A. Chaudhuri, and A. Chaudhuri. *"Shared cryptography with embedded session key for secret audio".* International Journal of Computer Applications,vol.: 26, no.: 8, pages: 5-9, 2011.

18. J. B. Lima and E. F. da Silva Nero. *"Audio encryption based on the cosine number transform",* Multimedia Tools and Applications, vol.: 75, issue: 14, pages: 8403-8418, 2016.

19. N. K. Pareek, V. Patidar and K. K. Sud. *"Image encryption using chaotic logistic map"*.Image and Vision Computing, vol.: 24, issue: 9, pages: 926-934, 2006.

20. A. Kanso and N. Smaoui. *"Logistic chaotic maps for binary numbers generations".* Chaos, Solitons and Fractals, vol.:40, issue: 5, pages: 2557-2568, 2009.

21. P. K. Naskar and A. Chaudhuri. *"A robust image encryption technique using dual chaotic map"*. International Journal of Electronic Security and Digital Forensics, Inderscience, vol.: 7, issue: 4, pages: 358-380, 2015.

22. H. Gao, Y. Zhang, S. Liang, and D. Li. *"A new chaotic algorithm for image encryption".* Chaos, Solitons and Fractals, Elsevier, vol.: 29, issue: 2, pages: 393-399, 2006.

23. A. Kanso and M. Ghebleh. *"A novel image encryption algorithm based on a 3d chaotic map".* Communications in Nonlinear Science and Numerical Simulation,vol.: 17, issue: 7, pages: 2943-2959, 2012.

24. Y. Zhou, L. Bao, and C. L. P. Chen. *"A new 1d chaotic system for image encryption".*Signal Processing, vol.: 97, pages: 172-182, 2014.

25. E. Mosa, N. W. Messiha, O. Zahran, and E. A. El-Samie. *"Chaotic encryption of speech signals",* International Journal of Speech Technology, vol.: 14, issue: 4, pages: 285-296, 2011.

26. P. Sathiyamurthi and S. Ramakrishnan. *"Speech encryption using chaotic shift keying for secured speech communication"*. EURASIP Journal on Audio, Speech, and Music Processing, 20, pages: 1-11, 2017.

27. L. M. Adleman. *"Molecular computation of solutions to combinatorial problems".* American Association for the Advancement of Science, vol.:266, no.: 5187, pages: 1021-1024, 1994.

28. L. M. Adleman. *"Computing with DNA"*. Scientific American, pages: 54-61, 1998.

29. S. H. Jiao and R. Goutte. *"Code for encryption hiding data into genomic dna of living organisms".* In Proceedings of the 9th International Conference on Signal Processing (ICSP'08), China, pages.: 2166-2169, 2008.

30. R. Enayatifar, A. H. Addula, and I. F. Isnin. *"Chaos-based image encryption using a hybrid genetic algorithm and a dna sequence".* Optics and Lasers in Engineering, Elsevier, vol.: 56, pages: 83-93, 2014.

31. P. K. Naskar and A. Chaudhuri. *"Secured secret sharing technique based on chaotic map and dna encoding with application on secret image".* The Imaging Science Journal, Taylor & Francis, vol.: 94, issue: 8, pages: 460-470, 2016.

32. M. Shyam, N. Kiran, V.Maheswaran. *"A novel encryption scheme based on DNA Computing"*. TSWJ,vol.: 1, pages: 1-10, 2007.

33. Q. Zhang, L. Guo, X. Xue, X. Wei.*"An image encryption algorithm based on DNA sequence addition operation"*. Fourth International Conference on Bio-Inspired Computing - Beijing, China, pages: 1–5, 2009.

34. L. Liu, Q. Zhang, X. Wei. *"A RGB image encryption algorithm based on DNA encoding and chaos map"*Computers and Electrical Engineering, vol.: 38, issue: 5, pages: 1240–1248, 2012.

35. X. Wei, L. Guo, Q. Zhang, J. Zhang and S. Lian. *“A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system”.*Journal of Systems and Software, vol.: 85, issue: 2, pages: 290–299, 2012.

36. R. Enayatifar, A. H. Abdullah, I. F.Isnin. “*Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence”*. Optics and Lasers Engineering, vol.: 56, pages: 83–93, 2014.

37. R. Guesmi, M. B. Farah, A. Kachouri and M. Samet. *“A novel chaos-based image encryption using DNA sequence operation and Secure Hash Algorithm SHA-2”.* Nonlinear Dynamics,vol.: 83, issue: 3, pages: 1123–1136, 2016.

38. H. Liu, X. Wang, A. Kadir. *“Image encryption using DNA complementary rule and chaotic maps”,*Applied Soft Computing, vol.: 12, issue:5, pages: 1457–1466, 2012.

39. G. Chen, Y. Mao and C. K. Chui. *“A symmetric image encryption scheme based on 3D chaotic cat maps”*. Chaos, Solitons & Fractals, vol.: 21, issue: 3, pages: 749–76, 2004.

40. H. Liu, X. Wang, A. Kadir. *“Color image encryption using choquet fuzzy integral and hyper chaotic system”*.Optik, vol.: 124, issue: 18, pages: 3527–33, 2013.

41. J. Fridrich. *“Symmetric ciphers based on two-dimensional chaotic maps”*. International Journal of Bifurcation and Chaos, vol.: 8, no.: 6, pages: 1259–1284, 1998.

42. V. Patidar, N. K. Pareek, K. K. Sud. *“A new substitution–diffusion based image cipher using chaotic standard and logistic maps”*.Commun Nonlinear Sci Numer Simulat,vol.: 14, issue: 7, pages: 3056–3075, 2009.

43. H. Liu, X. Wang. *“Color image encryption based on one-time keys and robust chaotic maps”.*Computers and Mathematics with Applications, vol.: 59, issue: 10, pages: 3320–3327, 2010.

44. G. Chen, Y. Mao, C. K. Chui. *“A symmetric image encryption scheme based on 3D chaotic cat maps”.*Chaos, Solitons & Fractals, vol. 21, issue: 3, pages: 749–761, 2004.

45. H. Liu, X. Wang, A. Kadir. *“Color image encryption using choquet fuzzy integral and hyper chaotic system”.* Optik, vol.:124, issue: 8, pages: 3527–3533, 2013.

46. M. Ghebleh, A. Kanso, H. Noura. *“An image encryption scheme based on irregularly decimated chaotic maps”.* Signal Processing: Image Communication, vol.: 29, issue: 5, pages: 618–27, 2013.

47. G. Zhang, A. Liu.*“A novel image encryption method based on total shuffling scheme”.*Optics Communications, vol.: 284, issue: 12, pages: 2775–2780, 2011.

48. Y. Zheng, J. Jin.*"A novel image encryption scheme based on henon map and compound spatiotemporal chaos"*.Multimedia Tools and Applications, vol.: 74, issue: 18, pages: 7803-7820, 2014.

49. T. Gao, Z. Chen. *"Image encryption based on a new total shuffling algorithm".*Chaos, Solitons and Fractals, vol.: 38, issue: 1, pages: 213–220, 2008.

50. Y. Wang, K-W. Wong, X. Liao, T. Xiang, G. Chen. *"A chaos-based image encryption algorithm with variable control parameters".*Chaos, Solitons and Fractals, vol.: 41, issue: 1, pages: 1773-1783.

51. S. M. Seyedzadeh, S. Mirzakuchaki. *"A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map".*Signal Processing, vol.: 92, issue: 5, pages: 1202–1215, 2012.

52. R-J. Chen, S-J. Horng. *"Novel SCAN-CA-based image security system using SCAN and 2-D von Neumann cellular automata"*. Signal Processing: Image Communication, vol.: 25, issue: 6, pages: 413–426, 2010.

53. R-J. Chen, J-N. Lai. *"Image security system using recursive cellular automata substitution".* Pattern Recognition, vol.: 40, issue: 5, pages: 1621–1631, 2007.

54. A. P. Tafti, S. Janosepah. *"Digital Images Encryption in Frequency Domain Based on DCT and One Dimensional Cellular Automata".*Informatics Engineering and Information Science, vol.: 24, pages: 421–427, 2011.

55. C. Peng, Y. Li. *"A New Algorithm for Image Encryption based on Couple Chaotic System and Cellular Automata".* International Conference on Mechatron Sci Electr Eng Comput (MEC), Shenyang, China, pages: 1645–1648, 2013.

56. S. Zamani, M. Javanmard, N. Jafarzadeh, M. Zamani. *"A Novel Image Encryption Scheme Based on Hyper Chaotic Systems and Fuzzy Cellular Automata".* The 22nd Iranian Conference on Electrical Engineering (ICEE 2014), Shahid Beheshti University, pages: 1136–1141, 2014.

57. S. S. Maniccam, N. G. Bourbakis. *"Lossless image compression and encryption using SCAN".* Pattern Recognition, vol.: 34, issue.: 6, pages: 1229–1245, 2001.

58. G. Bhatnagar, Q. M. J. Wu. *"Selective image encryption based on pixels of interest and singular value decomposition".* Digital Signal Processing,vol.: 22, issue: 4, pages: 648–663, 2012.

59. G. Bhatnagar, Q. M. J. Wu, B. Raman. *"Image and video encryption based on dual space-filling curves"*. The Computer Journal Advance Access, vol.: 55, issue: 6, pages: 667–685, 2012.

60. N. Pareek, V. Patidar, K. Sud. *“Colour Image Encryption Scheme Based on Permutation and Substitution Techniques”*. Communications in Computer and Information Science,vol.: 3, pages: 413–27, 2011.

61. C. Shi and B. Bhargava. *“A Fast MPEG Video Encryption Algorithm”*. Proceedings of the 6th ACM International Conference on Multimedia, New York, USA, pages: 81 – 88, 1989.

62. C. Shi and B. Bhargava, *“MPEG Video Encryption In Real-Time Using Secret Key Cryptography”*, available on: http://www.ncc.org.in/download.php?f=NCC2003/F-2.pdf.

63. W. Zeng. *“Efficient Frequency Domain Selective Scrambling of Digital Video”.* IEEE Transactions on Multimedia,vol.: 5, issue: 1, pages: 118 – 129, 2003.

64. L. Tang. *“Methods for Encrypting and Decrypting MPEG Video Data Efficiently”.*Proceeding 4th ACM International Multimedia Conference, pages: 219 – 229, 1996.

65. L. Qiao and K. Nahrstedt. *“Comparison of MPEG Encryption Algorithms”.* International Journal on Computer and Graphics, Special Issue on Data Security in Image Communications and Network, vol.: 22, issue: 3, Permagon Publisher, 1998.

66. G. L. Hobbs, *“Video Scrambling”,* U.S. Patent No. 5 815 572, Sept. 29, 1998.

67. D. Zeidler and J. Griffin, *“Method and Apparatus for Television Signal Scrambling Using Block Shuffling”,* U.S. Patent No. 5 321 748, June 14, 1994.

68. W. Zeng. *“Efficient Frequency Domain Selective Scrambling of Digital Video”.*IEEE Transactions on Multimedia, vol.: 5, issue: 1, pages: 118 – 129, 2003.

69. R. Tao, X.-Y. Meng, Y. Wang.*“Image Encryption with Multi-Orders of Fractional Fourier Transforms”.* IEEE transactions on Information Forensics and Security, vol.: 5, issue: 4, pages: 734 – 738, 2010.

70. Z. Liu, L. Xu, J. Dai, S. Liu.*“Image encryption by using local random phase encoding in fractional Fourier transform domains”*. Elsevier, Optic,vol.: 123, pages: 428 – 432, 2012.

71. Z. Yu, Z. Zhe, Y. Haibing, P. Wenjie, Z. Yunpeng. *"A ChaosBased Image Encryption Algorithm Using Wavelet Transform"*. IEEE, 2010.

72. A. Agarwal. *“Secret Key Encryption Algorithm Using Genetic Algorithm”.* International Journal of Advanced Research in Computer Science and Software Engineering, vol.: 2, issue: 4, pages: 216 – 218, 2012.

73. S. Bhowmik and S. Acharyya. *“Image Cryptography: The Genetic Algorithm Approach”.* IEEE International Conferenceon Computer Science and Automation Engineering (CSAE), pages: 232 – 227, 2011.

74. J. Kumar and S. Nirmala. *"Encryption of Images Based on Genetic Algorithm – A New Approach"*. Advances in Intelligent Systems and Computing, vol.: 167, pages: 783 – 791, 2012.

75. A. Soni and S. Agrawal. *"Using Genetic Algorithm for Symmetric key Generation in Image Encryption".* International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol.: 1, issue: 10, pages: 137 – 140, 2012.

76. A. Abdelsalam, K. Anil, A. Ibrahim and E. Nasreddin, *"A New Approach for Data Encryption Using Genetic Algorithms"*, available on: https://uqu.edu.sa/files2/tiny_mce/plugins/filemanager/files/30/papers/f162.pdf

77. V. Srikanth, U. Asati, V. Natarajan, T.P. Kumar, T. Mullapudi and N.Ch.S.N.Iyengar. *"Bit-Level Encryption of Images Using Genetic Algorithm".*International Journal of Computing Science and Communication Technologies, vol.: 3, no.: 1, pages: 546 – 550, 2010.

78. M. Ahmed, D. Robert and A. Shawki, *"An Adaptive Encryption Based Genetic Algorithms for Medical Images".* IEEE International Workshop on Machine Learning for Signal Processing, Southampton, UK, pages: 22–25, 2013.

79. S. Dilbag, R. Pooja and K. Rajesh. *"To Design a Genetic Algorithm for Cryptography to Enhance the Security".* International Journal of Innovations in Engineering and Technology (IJIET), vol.: 2, issue: 2, pages: 380 – 385, 2013.

80. S. E. E. Khamy, M. Lotfy, and A. H. Ali. *"A New Color Image Encryption Technique Utilizing Fuzzy Pseudo-Rsadom Bit Generator".* 22nd URSI NRSC Conference, Cairo, Egypt, pages: 185 – 194, 2005.

81. R. D. Srinivasa, M. Seetha and P. Krishna. *"Quality Assessment of Pixel – Level Image Fusion Using Fuzzy Logic".* International Journal on Soft Computing (IJSC), vol.: 3, issue: 1, pages: 13 – 25, 2012.

82. B. Maneckshaw and K. Kumar. *"Multiple FG Mapping Technique for Image Encryption",* International Journal of Engineering Research and Applications (IJERA), vol.: 3, issue: 1, pages: 811 – 815, 2013.

83. B. K. ShreyamshaKumar and C. R. Patil. *"JPEG Image Encryption Using Fuzzy PN Sequences".* Signal, Image and Video Processing, vol.: 4, issue: 4, pages: 419 – 427, 2010.

84. H. UYSAL, S. KURT and T. YILDIRIM. *"Automatic Decryption of Images through Artificial Neural Networks".* Trends in Innovative Computing Intelligent Systems Design, pages: 187 – 191, 2012.

85. I. A. Ismail, G. H. G. Edeen, S. Khattab, and M. A. E. M. E. Bahtity. *“Satellite Image Encryption Using Neural Networks Backpropagation”.* (ICCTA), Alexandria, Egypt, pages:148 - 152 , 2012.

86. R. Enayatifar and A. H. Abdullah. *“Image Security Via Genetic Algorithm”.* International Conference on Computer and Software Modeling (IPCSIT), vol:14, pages: 198 – 203, 2011.

87. M. K. Ghose, A. K. Ghose. *“Overview of Information Security Using Genetic Algorithm and Chaos”,* Information Security Journal: A Global Perspective, vol: 18, issue: 6, pages: 306 – 315, 2009.

88. S. P. Nichat and S.S. Sikchi. *“Image Encryption Using Hybrid Genetic Algorithm”.* International Journal of Advanced Research in Computer Science and Software Engineering, vol: 3, issue: 1, pages: 421 – 431, 2013.

89. B. Nooshin, F. Yousef and A. Karim. *“A Novel Image Encryption/Decryption Scheme Based on Chaotic Neural Networks”.* Engineering Applications of Artificial Intelligence, vol: 25, pages: 753–765, 2012.

90. M. Hamri , J. Mikram and F. Zinoun. *“A digital Image Encryption Algorithm Based on Chaotic Logistic Maps Using A Fuzzy Controller”.* International Journal of Computer Science and Information Security, vol: 9, issue: 3 pages: 39 – 44, 2011.

91. W.M. A. Etaiwi, (2014). “*Encryption Algorithm Using Graph Theory”*. Journal of Scientific Research & Reports, vol. 3, issue: 19, pages: 2519–2527, 2014.

92. Y. Ravinath , V. Mangla , A. Bhattacharya and P. Ohri. *“Graph Theory Application in Selective Encryption Mechanism for Wireless Ad Hoc Network”.* International Journal of Emerging Trends and Technology in Computer Science (IJETTCS), vol. 2, issue: 2, pages: 363–365, 2013.

93. L.D. Singh, K.M.Singh. *“Implementation of text encryption using elliptic curve cryptography”*. Procedia Computer Science, vol. 54, page: 73–82, 2015.

94. Masmoudi, W. Puech, M. S. Bouhlel. “*A new joint lossless compression and encryption scheme combining a binary arithmetic coding with a pseudo random bit generator”*. International Journal of Computer Science and Information Security, IJCSIS, vol.: *8*, issue: 1, pages: 170–175, 2010.

95. W. Ditto, T. Munakata. *“Principles and Applications of Chaotic System”*. Communications of the ACM, vol.: 38, issue: 11, pages: 96-102, 1995.

96. J. M. Blackledge,. Cryptography using Chaos. Available online:http://konwersatorium.pw.edu.pl/wyklady/2010_VLZ7_02_wyklad.pdf (accessed on 17 March 2015).

97. S. Paul, P. Dasgupta, P. K. Naskar and A. Chaudhuri. *"Secured image encryption scheme based on DNA encoding and chaotic map"*. Review of Computer Engineering Studies, IIETA, vol. 4, no. 2, pp. 70-75, June 2017.

98. M. Xu and Z. Tian. *"Security analysis of a novel fusion encryption algorithmbased on dna sequence operation and hyper-chaotic system"*. Optik, vol. 134, page: 45–52, January 2017.

98. A. Akhavan, A. Samsudin and A. Akhshani. *"Cryptanalysis of an image encryption algorithm based on DNA encoding"*. Optics and Laser Technology, vol. 95, page: 94–99, April 2017.

99. X. Wang, S. Wang, Y. Zhang and K. Guo."*A novel image encryption algorithm based on chaotic shuffling method*". Information Security Journal: A Global Perspective, Taylor & Francis, vol. 64, no.8, page: 460-470, February 2017.

100. A. U. Rehman, X .Liao, R. Ashraf, S. Ullah and H. Wang. *"A color image encryption technique using exclusive-OR with DNA complementary rules based on chaos theory and SHA-2"*, Optik, vol. 159, page: 348-367, January 2018.

101. A. Kotb, K. E. Zoiros. *"Performance analysis of all-optical XOR gate with photonic crystal semiconductor optical amplifier-assisted Mach–Zehnder interferometer at 160 Gb/s"*. Optics Communications, vol.: 402, pages: 512-517, 2017.

102. H. Liu, A. Kadir and Y. Li. *"Audio encryption scheme by confusion and diffusion based on multi-scroll chaotic system and one-time keys"*. Optik - International Journal for Light and Electron Optics,vol.:127, issue:19, pages: 7431-7438, 2016.

103. Y. Q. Zhang and X.Y. Wang. "A new image encryption algorithm based on non-adjacent coupled map lattices". *Applied Soft Computing*, vol.: 26, page: 10–20, 2015.

104. Y**.** Zhou, L. Bao and C.L.P. Chen. "A new 1D chaotic system for image encryption". *Signal Processing*, vol.: 97, page: 172–182, 2014.

105. G. A. Sathiskumar and K. B. Bagan. "A novel image encryption algorithm using pixel shuffling and base 64 encoding based chaotic block cipher (IMPSBEC)", *WSEAS Transactions on Computers*, vol.: 10, no. 6, page: 169–178, 2011.