Design and Implementation of Authentication Technique in Higher Region of Convergence of Z Transformation

A Thesis Submitted in the partial fulfilment of the requirements of the Degree for Master of Technology (M.Tech) in Computer Science & Engineering (CSE) at University Of Kalyani, Nadia, West Bengal

By

SUMAN MAHAPATRA

Registration No : 0081461 of Session 2013-2014

Roll No: 90/CSE/130017

UNDER THE SUPERVISION OF

Prof.(Dr.) JYOTSNA KUMAR MANDAL

Professor, Department of Computer Science & Engineering

University of Kalyani

Kalyani-741235, Nadia, West Bengal

January 2015

UNIVERSITY OF KALYANI

Department of Computer Science & Engineering
Faculty of Engineering, Management & Technology

Certificate of Approval

This is to certify that the project report entitled "Design and Implementation of Authentication Technique in Higher Region of Convergence of Z Transformation" a record of bona fide work, submitted to University of Kalyani in partial fulfilment of the requirement for the award of the degree of Master of Technology (M.Tech) in Computer Science & Engineering, Part-II 1st Semester is carried out by Mr. Suman Mahapatra (Registration No : 0081461 of Session 2013-2014, Roll No: 90/CSE/130017) under my Supervision and guidance . In my opinion, the report in the present form is in compliance with the requirement, as specified by the Dept. of Computer Science & Engineering and as per the regulation of University of Kalyani.

To the best of my knowledge, the results included in the report are original and worthy of incorporation in the present version of the report for the M.Tech Program in Computer Science and Engineering.

Dr. Kalyani Mali
HOD & Associate Professor
Dept of CSE
University of Kalyani

Dr. Jyotsna Kumar Mandal
Supervisor & Professor
Dept of CSE
University of Kalyani

External Examiner(s)

ACKNOWLEDGEMENTS

Working for a project work as a requirement is no doubt a scientific step-ahead that has enormously helped me to gain immense experience to enrich the pathway towards the acquisition of knowledge. In the preparation of this work and its formation it is undoubtedly inevitable that there are many to whom I owe my sincere gratitude.

I would like to express my sincere gratitude to my project supervisor Dr. Jyotsna Kumar Mandal, Professor, Dept of CSE, University of Kalyani for his valuable advice, guidance and support right from the project idea was conceived.

We are also thankful to all the Professors of CSE Dept, University of Kalyani for their constant support and valuable suggestions and constant inspiration during the tenure of this work. We would like to convey our sincere thankfulness and cordial respect to all the Professors of CSE Dept, University of Kalyani for this work.

We are also indebted to all the staffs of for their extended arm of help to us all the time and grateful to my friends for their endless motivation, without which this project work could not have taken its shape. I am grateful to my parents and other family members for moral support and encouragement without which this project would have never completed.

Signature of the Student

(SUMAN MAHAPATRA)

Design and Implementation of Athentication Technique in Higher Region of Convergence of Z Transformation

Suman Mahapatra

Contents

Chapter 1

Introduction

In the present digital world, there is enormous increase of multimedia content transmission over the internet such as image, video, audio contents etc. Overwhelming growth in the frequency and sophistication of image processing softwares are becoming the concern behind the security, authentication and integrity of digital images and videos. The problem of sharing large amount of multimedia contents over the internet creates a concern among the scientists and researchers regarding the security of the transmitted data. One cannot deny the importance of security in data communication and networking. Security in networking is based on cryptography, the science and art of transforming messages to make them secure and immune to attack.

1.1 Cryptography

Cryptography, a word with Greek origins, means secret writing. However, we use the term to refer to the science and art of transforming messages to make them secure and immune to attacks. Fig. 1.1 shows the components involved in cryptography.

Plaintext & Ciphertext

The original message, before being transformed, is called plaintext. After the message is transformed, it is called ciphertext.

An encryption algorithm transforms the plaintext into ciphertext. A decryption algorithm transforms the ciphertext back into plaintext. The sender

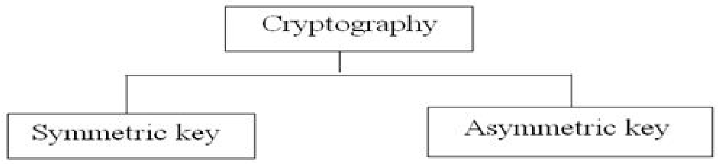


Figure 1.1: Components of Cryptography

uses an encryption algorithm, and the receiver uses a decryption algorithm.

Key

A key is a number (or a set of numbers) that the cipher, as an algorithm, operates on. To encrypt message, we need an encryption algorithm, an encryption key, and the plaintext which create ciphertext. To decrypt a message, we need a decryption algorithm, a decryption key, and the ciphertext which reveal the original plaintext.

1.1.1 Types of Cryptography

We can divide all the cryptographic algorithms into two groups: symmetric key (or secret key) cryptographic algorithms and asymmetric (or public key) cryptographic algorithms.

Symmetric Key Cryptography

In symmetric key cryptography, the same key is used by both parties. The sender uses this key and encryption algorithm to encrypt data. The receiver uses the same key and the corresponding decryption algorithm to decrypt the data (Fig. 1.2).

Assymetric Key Cryptography

In asymmetric 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.Here, the public key is used for encryption is different from the private key that is used for decryption (Fig. 1.3)

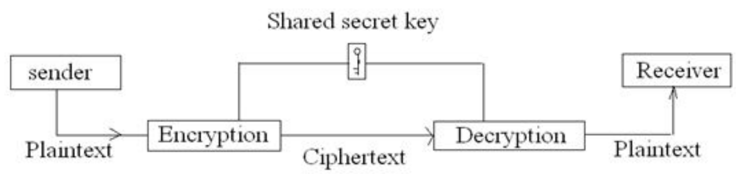


Figure 1.2: Block diagram of Symetric Key Cryptography

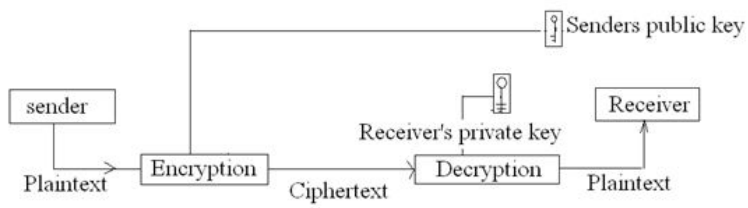


Figure 1.3: Block diagram of Assymetric Key Cryptography

1.1.2 Goals of Cryptography

There are several goals of cryptography. Some of them are as follows:

1. Message Confidentiality

Message confidentiality or privacy mans that the sender and the receiver expect confidentiality. The transmitted message must make sense to only the intended receiver. To all others, the message must be garbage. When a customer communicates with her banks, she expects that the communication is totally confidential.

2. Message Integrity

Message integrity means that the data must arrive at the receiver exactly as they were sent. There must be no changes during the transmission, neither accidentally nor maliciously. As more and more monetary exchanges occur over the internet, integrity is crucial.

3. Message Authentication

Message Authentication is a service beyond message integrity. In message authentication the receiver needs to be sure of the senders identity and the an imposter has not sent the message.

4. Message Nonrepudiation

Message nonrepudiation means that a sender must not be able to deny sending a message that he or she, in fact, had sent. The burden of proof falls on the receiver.

5. Entity Authentication

In entity authentication the entity or user is verified prior to access to the system resources.

1.1.3 Advantages of Cryptography

- The biggest advantage of public key cryptography is the secure nature of the private key. In fact, it never needs to be transmitted or revealed to anyone.
- It enables the use of digital certificates and digital timestamps which is a very secure technique of signature authorization.

1.1.4 Disadvantages of Cryptography

- The size of keys (both public and private) must be significantly larger than symmetric cryptography to achieve the same level of protection.
- Transmission time for documents encrypted using public key cryptography are significantly slower then symmetric cryptography.
- Public key cryptography is susceptible to impersonation attacks. Overhead always there have been for cryptography, to ease the heftiness another field was invented is Steganography.

1.2 Steganography

Steganography means covered writing in greek, is the art and science of hiding information by embedding messages within the other.It may be felt is the mirror image cryptography as its goal is to hide not to be understood by intruders. Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message.

1.2.1 History of Steganography

The word steganography came from a 15th century work called Steganographia by a German abbot named Trithemius. On the face of it, the three books were about magic, but they were also contained an encrypted treatise on cryptography so Steganographia was itself a case of steganography.

Other Historical Examples

There are some other examples of old steganographic methods which were basically the basis of modern steganographic schemes.

1. An ancient Greek named Histaiaeus was fomenting revolt against the king of Persia and needed to pass along a message secretly. He shaved the head of a slave, tattooed the message on his scalp, then sent him on his way when his hair grew back in. Recipients of the message shaved his head again to read the alert. The Greeks used the same trick shaving and writing on the belly of a rabbit.

2. Sometime in the 5th century B.C., an exiled Greek named Demaratus wrote a warning that the Persians planned to attack Sparta. He wrote the message on the wooden backing for a wax tablet, then hid it by filling in the wood frame with wax so it looked like a tablet containing no writing at all. The wife of the Spartan king divined that there was a message behind the wax, so they scraped it off and got the warning in time to set up a desperate defence at Thermopylae, incidentally giving modern screenwriters the plot for the movie The 300.

3. During World War II, microdots - miniaturized photos that can be hidden in plain sight, then read using magnifiers were used by spies to carry data out of enemy countries. Here the microdot circled in red piggybacks on a watch face. Blown up, it reveals a message written in German. In modern printers also, this type of microdots are used to authenticate its manufacturer and make.

4. When the USA Pueblo was captured by North Korea in 1968, the crew was forced to pose for propaganda photos to demonstrate they were being well treated. Their finger gestures are a form of steganography that sends a message Americans could decrypt right away, the North Koreans, not so quickly.

1.2.2 Objectives of Steganography

There are several objectives of steganography for which steganography is used in various applications.They are as follows :

Invisibility

The first and most important requirement of steganography is invisibility. The strength of the steganographic algorithm lies in ability to be unnoticed by human eyes, as soon as it is noticed by the desired algorithm.

Payload Capacity

Visible watermark has only a small amount of information for copyright or authentication and it can also be easily traceable. Sufficient amount of information cannot be embedded/watermarked using visible watermarking process. The total number of bits inserted per considered block should be increased to obtain a considerable better payload value.

Robustness Against Image Manipulation

As the image or any other type of multimedia contents rely on the transmission media, so during transmission there can be various types of attacks on the transmitted media file. The image can undergo different types of distortion due to several malicious or non-malicious attacks. The algorithm should be such that it can manage those type of distortions.

Robustness Against Statistical Attacks

There are several types of statistical attacks which can easily detect any types of visually perceptible embedding and also certain strong statistical methods can detect non-perceptible authenticating bits. So the proposed stego algorithm should be strong enough to withstand such types of statistical attacks without disclossing the embedded pattern.

1.2.3 Types of Steganography

There are several types of steganography based on various applications. They are as follows:

Text Steganography

When steganographic algorithm is applied on a text file then that type of steganography is called Text Steganography. Hiding information in plain text can be done in many different ways. For example, it can be decided that in which position of the word the secret data can be hidden depending on the number of characters present in the word.

Audio Steganography

When steganographic technique is applied on digital sound, that type of steganography is called Audio Steganography. Secret messages are embedded in digital sound. The secret message is embedded by slightly altering the binary bit sequence of a sound file. Embedding secret messages in digital sound is usually a more difficult process than embedding messages in other media contents, such as digital images. In order to conceal secret messages successfully, a variety of methods for embedding information bits in digital audio have been introduced.

Image Steganography

When steganography is applied over image, that type of steganography is called Image Steganography. In this case secret messages are embedded in digital image. In order to conceal secret messages successfully, a variety of methods for embedding information bits in digital images have been introduced.

1.2.4 Performance Metrics for Steganographic Applications

To judge the performance of the steganographic algorithms and to check the visual imperceptibility and the traceability of the embedded information bits there are some performance metrics used in various steganographic literatures. They are as follows :

Mean Square Error (MSE)

In statistical analysis, MSE is used to quantify the difference values between the original pixel values and the embedded or modified pixel values divided by the total number of pixel components of the image. MSE is a risk function, corresponding to the modified value of the squared error loss or quadratic loss. The mathematical definition of MSE is given by the following Eq.(1.1)

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I_1(i,j) - I_2(i,j)|^2 \tag{1.1}$$

where $I_1(i,j)$ and $I_2(i,j)$ are the gray level intensities of the pixels at the i^{th} row and j^{th} column of two images of size $M \times N$ respectively.

Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio, abbreviated as PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. As many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel representation. Basically PSNR is an approximation to human perception of reconstruction quality. In case of cryptogrphic applications, as we need to have a good encryption and does not intend to have similarity, so PSNR should be considerably low in that case. But in Steganography, as we intend to have a visually imperceptible media content, so PSNR must have to be considerably high (at least greater that 35 dB). The mathematical definition of PSNR is based on the MSE value as defined by the following Eq.(1.2):

$$PSNR = 10 * log_{10}(\frac{Max^2}{MSE}) = 20 * log_{10}(\frac{Max}{\sqrt{MSE}}) \tag{1.2}$$

where Max is the maximum value of the media component under consideration. In case of a grayscale image, this Max is considered to be 255.

Image Fidelity (IF)

Image Fidelity, abbreviated as IF, is a similarity measure between the original media content components (pixels in case of an image) and the corresponding encrypted or embedded media content components. In the IF measurement, first of all the dissimilarity value between the original and embedded or encrypted pixels (for images) over all the pixels present in the digital image. Then that fractional dissimilarity value is subtracted from 1 to get the similarity measure. The mathematical definition of IF is given by the following Eq.(1.3) as follows:

$$IF = 1 - \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I_1(i,j) - I_2(i,j)| \tag{1.3}$$

1.2.5 Applications of Steganography

Usage in Modern Printers

Steganography is used by some modern printers, including HP and Xerox brand colour laser printers. Tiny yellow dots are added to each page. The dots are barely visible and contain encoded printer serial numbers and sometimes other important informations, as well as date and timestamps.

Usage in Legal Documents

Steganography can be used for digital watermarking, where an information or a message (being simply an identifier) is hidden in an image so that its source can be tracked or verified. Also for copyright protection, bank draft, cheque etc. steganography is used for document authentication.

Usage in Mobile Phones

As we have discussed earlier that steganography is used in audio files to hide some information bitstream. Audio Steganography is used in mobile phones.

1.2.6 Steganography in Spatial Domain

In different spatial domain applications, steganographic schemes have been implemented. Some of them are discussed below:

LSB Replacement

LSB refers to Least Significant Bit.It is used to embed bits of secret messages to LSB positions which results in an imperceptible embedding scheme. LSB insertion is a common and simple approach to embed information in an image file. In this method the LSB of a byte is replaced with a single bit from the embedding bitstream.This technique works good for image,audio and video steganography. To human eyes, the resulting image will look identical to the original image.

Pixel Value Difference (PVD) Method

In order to improve the capacity of the hidden secret data and to provide an imperceptible stego-image quality, a method Pixel Value Difference (PVD) method is presented.The method exploits the difference value of two consecutive pixels to estimate how many secret bits will be embedded into the two pixels. Pixels located in the edge areas are embedded by a bit LSB substitution method with a larger value of than that of the pixels located in smooth areas. The range of difference values is adaptively divided into lower level, middle level, and higher level. However, the value is adaptive and is decided by the level which the difference value belongs to. In order to remain at the same level where the difference value of two consecutive pixels belongs, before and after embedding, a delicate readjusting phase is used.

Tri-Way Pixel Value Difference (TPVD) Method

The Tri-way Pixel Value Difference (TPVD) method is a modification over the previously discussed PVD as a spatial domain steganography . The following diagram is basically an illustrated example of the TPVD in a diagramatric representation:

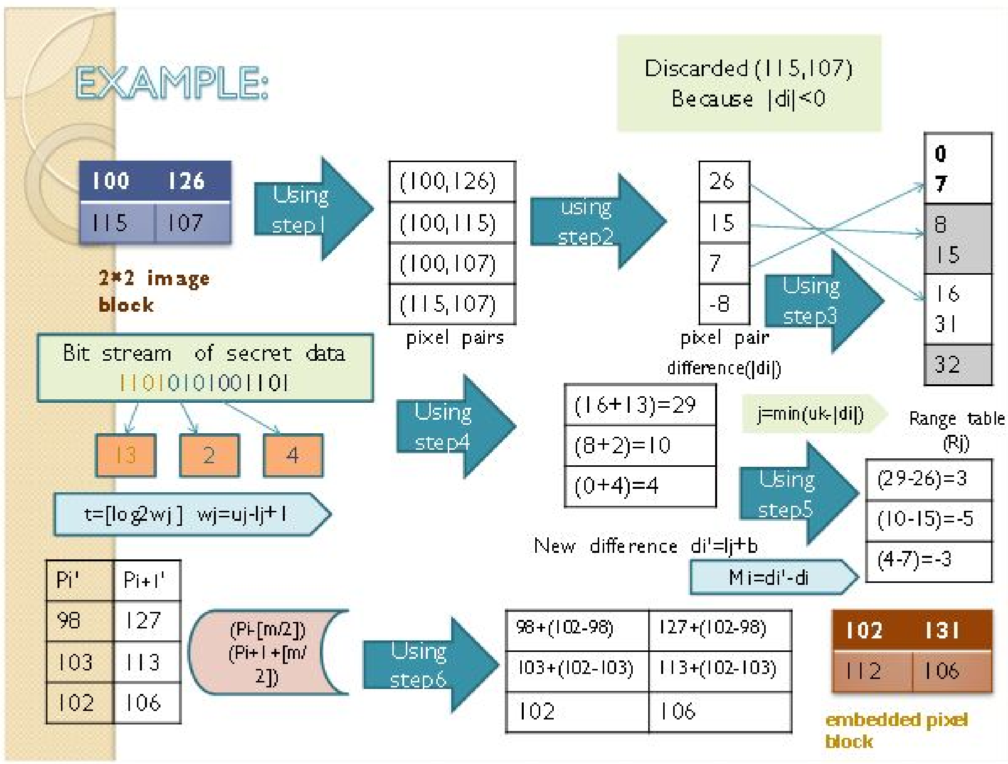


Figure 1.4: Diagramatic Representation of Tri-way Pixel Value Difference (TPVD)

Chapter 2

Frequency Domain Transformation

2.1 What is frequency domain transformation ?

An image is basically a collection of pixels in a 2-dimensional space. The total number of pixels in an image is called the resolution of the image, viz. for a $M \times N$ image, the resolution will be MN. Each pixel is assigned a value based on the underlying grayscale (0 to 255 for a 8-bit representation). There are many types of frequency domain transformation techniques such as Discrete Cosine Transformation (DCT), Discrete Fourier Transformation (DFT), Discrete Wavelet Transformation (DWT) etc. which have been implemented in present digital world techniques such as steganography, data compression etc.

2.2 Discrete Fourier Transformatin(DFT)

In mathematics, Discrete Fourier Transform (DFT), sometimes called the finite Fourier transform, is a Fourier transformation widely employed in signal processing and related fields to analyze the frequencies contained in a sampled signal. It can be said to convert the sampled function from its original domain to the frequency domain.

The DFT is the most important discrete transform, used to perform

Fourier analysis in many practical applications.

Basic DFT Formula

Mathematically, Discrete Cosine Transform (DFT) can be expressed in two ways:

- One-dimensional (1-D) DFT Formula
- Two-dimensional (2-D) DFT Formula

2.2.1 One-dimensional (1-D) Formula

Mathematically, one-dimensional (1-D) DFT can be expressed by the following equations:

1-D Forward Transformation

The one-dimensional forward transformation is denoted by the Eq.(2.1).

$$F(u) = \sum_{x=0}^{M-1} f(x).e^{-j2\pi(\frac{ux}{M})} \tag{2.1}$$

where,x is a spatial domain variable, $f(x)$ is a matrix of size M and u is the corresponding frequency domain variable and $F(u)$ is corresponding frequency domain matrix of size M.

1-D Inverse Transformation

Conversely, the one-dimensional inverse transformation is denoted by the Eq.(2.2).

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u).e^{j2\pi(\frac{ux}{M})} \tag{2.2}$$

Illustrative Example

Let us consider a spatial domain 2×2 matrix as follows:

$$f(x) = \begin{bmatrix} 112 & 110 \\ 90 & 120 \end{bmatrix}$$

Now using the Eq.(2.1) for 1-D forward transformation on the spatial domain matrix $f(x)$ and considering $M = 2$, we get the corresponding DFT co-efficients as follows:

$F(u=0) = 112 \times e^{-j2\pi(\frac{0\times0}{4})} + 110 \times e^{-j2\pi(\frac{0\times1}{4})} + 90 \times e^{-j2\pi(\frac{0\times2}{4})} + 120 \times e^{-j2\pi(\frac{0\times3}{4})}$
$= 112 \times (1) + 110 \times (1) + 90 \times (1) + 120 \times (1) = 432$

$F(u=1) = 112 \times e^{-j2\pi(\frac{1\times0}{4})} + 110 \times e^{-j2\pi(\frac{1\times1}{4})} + 90 \times e^{-j2\pi(\frac{1\times2}{4})} + 120 \times e^{-j2\pi(\frac{1\times3}{4})}$
$= 112 \times (1) + 110 \times e^{-j\frac{\pi}{2}} + 90 \times e^{-j\pi} + 120 \times e^{-j\frac{3\pi}{2}}$
$= 112 \times (1) + 110 \times (-j) + 90 \times (-1) + 120 \times (j) = 22 + 10j$

$F(u=2) = 112 \times e^{-j2\pi(\frac{2\times0}{4})} + 110 \times e^{-j2\pi(\frac{2\times1}{4})} + 90 \times e^{-j2\pi(\frac{2\times2}{4})} + 120 \times e^{-j2\pi(\frac{2\times3}{4})}$
$= 112 \times (1) + 110 \times e^{-j\pi} + 90 \times e^{-j2\pi} + 120 \times e^{-j3\pi}$
$= 112 \times (1) + 110 \times (-1) + 90 \times (1) + 120 \times (-1) = -28$

$F(u=3) = 112 \times e^{-j2\pi(\frac{3\times0}{4})} + 110 \times e^{-j2\pi(\frac{3\times1}{4})} + 90 \times e^{-j2\pi(\frac{3\times2}{4})} + 120 \times e^{-j2\pi(\frac{3\times3}{4})}$
$= 112 \times (1) + 110 \times e^{-j\frac{3\pi}{2}} + 90 \times e^{-j3\pi} + 120 \times e^{-j\frac{9\pi}{2}}$
$= 112 \times (1) + 110 \times (j) + 90 \times (-1) + 120 \times (-j) = 22 - 10j$

So the 2×2 corresponding forward DFT co-efficients are as follows:

$$F(u) = \begin{bmatrix} 432 & 22 + 10j \\ -28 & 22 - 10j \end{bmatrix}$$

Now, inverse transformation of the DFT co-efficients in $F(u)$ is aplied using the Eq.(2.2) to get back the corresponding spatial domain components.

$f(x=0) = \frac{1}{4}[432 \times e^{j2\pi(\frac{0\times0}{4})} + (22+10j) \times e^{j2\pi(\frac{1\times0}{4})} + (-28) \times e^{j2\pi(\frac{2\times0}{4})} +$

$(22-10j)\times e^{j2\pi(\frac{3\times 0}{4})}]$
$=\frac{1}{4}[432\times(1)+(22+10j)\times(1)+(-28)\times(1)+(22-10j)\times(1)]=112$

$f(x=1)=\frac{1}{4}[432\times e^{j2\pi(\frac{0\times 1}{4})}+(22+10j)\times e^{j2\pi(\frac{1\times 1}{4})}+(-28)\times e^{j2\pi(\frac{2\times 1}{4})}+$
$(22-10j)\times e^{j2\pi(\frac{3\times 1}{4})}]$
$=\frac{1}{4}[432\times(1)+(22+10j)\times e^{j\frac{\pi}{2}}+(-28)\times e^{j\pi}+(22-10j)\times e^{j\frac{3\pi}{2}}]$
$=\frac{1}{4}[432\times(1)+(22+10j)\times(j)+(-28)\times(-1)+(22-10j)\times(-j)]$
$=\frac{1}{4}[432+22j-10+28-22j-10]=110$

$f(x=2)=\frac{1}{4}[432\times e^{j2\pi(\frac{0\times 2}{4})}+(22+10j)\times e^{j2\pi(\frac{1\times 2}{4})}+(-28)\times e^{j2\pi(\frac{2\times 2}{4})}+$
$(22-10j)\times e^{j2\pi(\frac{3\times 2}{4})}]$
$=\frac{1}{4}[432\times(1)+(22+10j)\times e^{j\pi}+(-28)\times e^{j2\pi}+(22-10j)\times e^{j3\pi}]$
$=\frac{1}{4}[432\times(1)+(22+10j)\times(-1)+(-28)\times(1)+(22-10j)\times(-1)]$
$=\frac{1}{4}[432-22-10j-28-22+10j]=90$

$f(x=3)=\frac{1}{4}[432\times e^{j2\pi(\frac{0\times 3}{4})}+(22+10j)\times e^{j2\pi(\frac{1\times 3}{4})}+(-28)\times e^{j2\pi(\frac{2\times 3}{4})}+$
$(22-10j)\times e^{j2\pi(\frac{3\times 3}{4})}]$
$=\frac{1}{4}[432\times(1)+(22+10j)\times e^{j\frac{3\pi}{2}}+(-28)\times e^{j3\pi}+(22-10j)\times e^{j\frac{9\pi}{2}}]$
$=\frac{1}{4}[432\times(1)+(22+10j)\times(-j)+(-28)\times(-1)+(22-10j)\times(j)]$
$=\frac{1}{4}[432-22j+10+28+22j+10]=120$

So, from the inverse DFT, we get back the original spatial domain 2×2 matrix as follows:

$$f(x)=\begin{bmatrix}112 & 110\\ 90 & 120\end{bmatrix}$$

2.2.2 Two-dimensional (2-D) Formula

Two-dimensional (2-D) DFT can be mathematically expressed by the following equations:

2-D Forward Transformation

The two-dimensional forward transformation is denoted by the Eq.(2.3).

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y).e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})} \tag{2.3}$$

where, x & y are spatial domain variables, $f(x,y)$ is a matrix of size $M \times N$ and u & v are the corresponding frequency domain variables.

2-D Inverse Transformation

Conversely, the two-dimensional inverse transformation is expresses by the Eq.(2.4).

$$f(x,y) = \frac{1}{M \times N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v).e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})} \tag{2.4}$$

Conversion To An Easy Form

For a 2×2 spatial domain matrix under consideration, the generalized equation for two-dimensional DFT can be reduced to a rather easy short form. Mathematically this conversion in a generalized form is illustrated as follows:

$F(u = 0, v = 0) = f(0,0) \times e^{-j2\pi(\frac{0\times0}{2}+\frac{0\times0}{2})} + f(0,1) \times e^{-j2\pi(\frac{0\times0}{2}+\frac{0\times1}{2})} + f(1,0) \times e^{-j2\pi(\frac{0\times1}{2}+\frac{0\times0}{2})} + f(1,1) \times e^{-j2\pi(\frac{0\times1}{2}+\frac{0\times1}{2})}$
$F(0,0) = f(0,0) \times (1) + f(0,1) \times (1) + f(1,0) \times (1) + f(1,1) \times (1)$
$F(0,0) = f(0,0) + f(0,1) + f(1,0) + f(1,1)$

$F(u = 0, v = 1) = f(0,0) \times e^{-j2\pi(\frac{0\times0}{2}+\frac{1\times0}{2})} + f(0,1) \times e^{-j2\pi(\frac{0\times0}{2}+\frac{1\times1}{2})} + f(1,0) \times e^{-j2\pi(\frac{0\times1}{2}+\frac{1\times0}{2})} + f(1,1) \times e^{-j2\pi(\frac{0\times1}{2}+\frac{1\times1}{2})}$
$F(0,1) = f(0,0) \times (1) + f(0,1) \times e^{-j\pi} + f(1,0) \times (1) + f(1,1) \times e^{-j\pi}$
$F(0,1) = f(0,0) - f(0,1) + f(1,0) - f(1,1)$

$F(u = 1, v = 0) = f(0,0) \times e^{-j2\pi(\frac{1\times0}{2}+\frac{0\times0}{2})} + f(0,1) \times e^{-j2\pi(\frac{1\times0}{2}+\frac{0\times1}{2})} + f(1,0) \times e^{-j2\pi(\frac{1\times1}{2}+\frac{0\times0}{2})} + f(1,1) \times e^{-j2\pi(\frac{1\times1}{2}+\frac{0\times1}{2})}$
$F(1,0) = f(0,0) \times (1) + f(0,1) \times (1) + f(1,0) \times e^{-j\pi} + f(1,1) \times e^{-j\pi}$
$F(1,0) = f(0,0) + f(0,1) - f(1,0) - f(1,1)$

$F(u=1,v=1) = f(0,0) \times e^{-j2\pi(\frac{1\times 0}{2}+\frac{1\times 0}{2})} + f(0,1) \times e^{-j2\pi(\frac{1\times 0}{2}+\frac{1\times 1}{2})} + f(1,0) \times e^{-j2\pi(\frac{1\times 1}{2}+\frac{1\times 0}{2})} + f(1,1) \times e^{-j2\pi(\frac{1\times 1}{2}+\frac{1\times 1}{2})}$
$F(1,1) = f(0,0) \times (1) + f(0,1) \times e^{-j\pi} + f(1,0) \times e^{-j\pi} + f(1,1) \times e^{-j2\pi}$
$F(1,1) = f(0,0) - f(0,1) - f(1,0) + f(1,1)$

Similarly, the generalized inverse DFT equation can be converted to its equivalent reduced form as illustrated below:

$f(x=0,y=0) = \frac{1}{2\times 2}[F(0,0) \times e^{j2\pi(\frac{0\times 0}{2}+\frac{0\times 0}{2})} + F(0,1) \times e^{j2\pi(\frac{0\times 0}{2}+\frac{1\times 0}{2})} + F(1,0) \times e^{j2\pi(\frac{1\times 0}{2}+\frac{0\times 0}{2})} + F(1,1) \times e^{j2\pi(\frac{1\times 0}{2}+\frac{1\times 0}{2})}]$
$f(0,0) = \frac{1}{4}[F(0,0) \times (1) + F(0,1) \times (1) + F(1,0) \times (1) + F(1,1) \times (1)]$
$f(0,0) = \frac{1}{4}[F(0,0) + F(0,1) + F(1,0) + F(1,1)]$

$f(x=0,y=1) = \frac{1}{2\times 2}[F(0,0) \times e^{j2\pi(\frac{0\times 1}{2}+\frac{1\times 0}{2})} + F(0,1) \times e^{j2\pi(\frac{0\times 0}{2}+\frac{1\times 1}{2})} + F(1,0) \times e^{j2\pi(\frac{0\times 1}{2}+\frac{0\times 1}{2})} + F(1,1) \times e^{j2\pi(\frac{0\times 1}{2}+\frac{1\times 1}{2})}]$
$f(0,1) = \frac{1}{4}[F(0,0) \times (1) + F(0,1) \times e^{j\pi} + F(1,0) \times (1) + F(1,1) \times e^{j\pi}]$
$f(0,1) = \frac{1}{4}[F(0,0) - F(0,1) + F(1,0) - F(1,1)]$

$F(x=1,y=0) = \frac{1}{2\times 2}[F(0,0) \times e^{j2\pi(\frac{0\times 1}{2}+\frac{0\times 0}{2})} + f(0,1) \times e^{j2\pi(\frac{0\times 1}{2}+\frac{1\times 0}{2})} + f(1,0) \times e^{j2\pi(\frac{1\times 1}{2}+\frac{0\times 0}{2})} + f(1,1) \times e^{j2\pi(\frac{1\times 1}{2}+\frac{1\times 0}{2})}]$
$f(1,0) = \frac{1}{4}[F(0,0) \times (1) + F(0,1) \times (1) + F(1,0) \times e^{j\pi} + F(1,1) \times e^{j\pi}]$
$f(1,0) = \frac{1}{4}[F(0,0) + F(0,1) - F(1,0) - F(1,1)]$

$F(x=1,y=1) = \frac{1}{2\times 2}[F(0,0) \times e^{j2\pi(\frac{1\times 0}{2}+\frac{1\times 0}{2})} + f(0,1) \times e^{j2\pi(\frac{1\times 0}{2}+\frac{1\times 1}{2})} + f(1,0) \times e^{j2\pi(\frac{1\times 1}{2}+\frac{1\times 0}{2})} + f(1,1) \times e^{j2\pi(\frac{1\times 1}{2}+\frac{1\times 1}{2})}]$
$f(1,1) = \frac{1}{4}[F(0,0) \times (1) + F(0,1) \times e^{j\pi} + F(1,0) \times e^{j\pi} + F(1,1) \times e^{j2\pi}]$
$f(1,1) = \frac{1}{4}[F(0,0) - F(0,1) - F(1,0) + F(1,1)]$

Illustrated Example

Let us now consider a spatial domain 2×2 sample matrix for the application of 2-D forward DFT as follows:

$$f(x,y) = \begin{bmatrix} 200 & 100 \\ 250 & 230 \end{bmatrix}$$

Using the above reduced equations for a 2×2 sample matrix, the following results are obtained:

$F(0,0) = f(0,0) + f(0,1) + f(1,0) + f(1,1)$
$F(0,0) = [200 + 100 + 250 + 230] = 780$

$F(0,1) = f(0,0) - f(0,1) + f(1,0) - f(1,1)$
$F(0,0) = [200 - 100 + 250 - 230] = 120$

$F(1,0) = f(0,0) + f(0,1) - f(1,0) - f(1,1)$
$F(1,0) = [200 + 100 - 250 - 230] = -180$

$F(1,1) = f(0,0) - f(0,1) - f(1,0) + f(1,1)$
$F(1,1) = [200 - 100 - 250 + 230] = 80$

So the 2×2 corresponding DFT co-efficients are as follows:

$$F(u,v) = \begin{bmatrix} 780 & 120 \\ -180 & 80 \end{bmatrix}$$

Now, for inverse transformation, the reduced equations are used as follows to get back the spatial domain components:

$f(0,0) = \frac{1}{4}[F(0,0) + F(0,1) + F(1,0) + F(1,1)]$
$f(0,0) = \frac{1}{4}[780 + 120 + (-180) + 80] = \frac{1}{4}[780 + 120 - 180 + 80] = 200$

$f(0,1) = \frac{1}{4}[F(0,0) - F(0,1) + F(1,0) - F(1,1)]$
$f(0,1) = \frac{1}{4}[780 - 120 + (-180) - 80] = \frac{1}{4}[780 - 120 - 180 - 80] = 100$

$f(1,0) = \frac{1}{4}[F(0,0) + F(0,1) - F(1,0) - F(1,1)]$
$f(1,0) = \frac{1}{4}[780 + 120 - (-180) - 80] = \frac{1}{4}[780 + 120 + 180 - 80] = 250$

$f(1,1) = \frac{1}{4}[F(0,0) - F(0,1) - F(1,0) + F(1,1)]$
$F(1,1) = \frac{1}{4}[780 - 120 - (-180) + 80] = \frac{1}{4}[780 - 120 + 180 + 80] = 230$

So, the spatial domain 2×2 sample matrix is obtained back as follows:

$$f(x,y) = \begin{bmatrix} 200 & 100 \\ 250 & 230 \end{bmatrix}$$

2.3 Properties of Fourier Transformation

Due to its properties, Fourier Transformation helps us to manipulate the representation of signals, which helps us in making decision useful for different applications. The properties are discussed below:

For the calculation purpose from the equation Eq.(2.1) for fourier transformation, an assumption is made as $e^{-j\frac{2\pi}{N}} = W_N$

1. Linear Property

Let, $DFT[x(n)] = \sum_{n=0}^{N-1} x(n)W_N^{nk} = X(k)$ and similarly $DFT[g(n)] = G(k)$.

where $x(n)$ and $g(n)$ both are finite sequence of N samples, so as $X(k)$ and $G(k)$, then
$DFT[ax(n) + bg(n)] = aX(k) + bG(k)$

Now,
$DFT[ax(n) + bg(n)] = \sum_{n=0}^{N-1}[ax(n) + bg(n)]W_N^{nk}$
$= a\sum_{n=0}^{N-1} x(n)W_N^{nk} + b\sum_{n=0}^{N-1} g(n)W_N^{nk}$
$= aX(k) + bG(k)$

2. Periodic Property

If $x(n) = x(n + N)$ where $x(n)$ is a signal with N samples, then after N samples the same finite value repeats.

$X(k + N) = DFT[x(n)]$
$= \sum_{n=0}^{N-1} x(n)W_N^{n(k+N)}$
$= \sum_{n=0}^{N-1} x(n)W_N^{nk}W_N^{nN}$
$= \sum_{n=0}^{N-1} x(n)W_N^{nk}$ [As $W_N^{nN} = e^{-j\frac{2\pi\times nN}{N}} = e^{-j2\pi n} = 1$]

It shows that in frequency domain the signal repeats after N samples.

3. Time Shifting Property

Let, the time domain signal $x(n)$ is lagged or shifted by a lag of n_0.

$$DFT[x(n - n_0)] = \sum n = 0^{N-1}x(n - n_0)W_N^{nk}$$

Let, $(n - n_0) = m$
$= \sum m = -n_0{}^{N-1-n_0} x(m) W_N^{(m+n_0)k}$
$= \sum_{m=0}^{N-1} x(m) W_N^{n_0 k} W_N^{mk}$
$= W_N^{n_0 k}.DFT[x(m)]$
$= W_N^{n_0 k} X(k)$

4. Frequency Shifting Property

Let, the frequency is shifted by k_0.

$X(k - k_0) = \sum_{n=0}^{N-1} x(n) W_N^{n(k-k_0)}$
$= \sum_{n=0}^{N-1} x(n) W_N^{nk} W_N^{-nk_0}$
$= W_N^{-nk_0}.DFT[x(n)] = W_N^{-nk_0} X(k)$
The magnitude is not changed only phase is shifted.

5. Conjugation Property

$DFT[x(n)] = \sum_{n=0}^{N-1} x(n) W_N^{nk}$
So,
$DFT[x^*(n)] = \sum_{n=0}^{N-1} x^*(n) W_N^{nk}$
$= \sum_{n=0}^{N-1} x^*(n) [W_N^{-nk}]^*$
$= \sum_{n=0}^{N-1} [x(n) W_N^{-nk}]^*$
$= \sum_{n=0}^{N-1} [x(n) W_N^{nN} W_N^{-nk}]^*$
$= \sum_{n=0}^{N-1} [x(n) W_N^{n(N-k)}]^* = X(N - k)$

6. Convolution Property

Convolution makes out the area or intersection of two functions by inverting anyone of them.
Let, $x(n) * g(n) = \sum_{m=-\infty}^{\infty} x(m) g(n - m) = y(n)$
Alternatively, this property can be expressed as follows:
$x(n) * g(n) = X(k)G(k)$ and $X(k) * G(k) = x(n)g(n)$
$Y(k) = \sum_{k=0}^{N-1} y(n) W_N^{nk}$
$Now, Y(k) = \sum_{k=0}^{N-1} \sum_{m=0}^{M-1} x(m) g(k - m) W_N^{nk}$
$= \sum_{m=0}^{M-1} x(m) \sum_{n=0}^{N-1} g(k - m) W_N^{n(k-m)} W_N^{nm}$
$= \sum_{m=0}^{M-1} x(m) W_N^{nm} DFT[g(k - m)]$
With periodicity property,

$= \sum_{m=0}^{M-1} x(m) W_N^{nm} DFT[g(k)]$
$= G(k) \sum_{m=0}^{M-1} x(m) W_N^{nm}$
$= G(k) DFT[x(k)] = G(k) X(k)$

2.4 Applications of DFT

The DFT is the most important discrete transform, used to perform Fourier analysis in many practical applications.

- In Digital Signal Processing, the function is any quantity or signal that varies over time, such as the pressure of a sound wave, a radio signal, or daily temperature readings, sampled over a finite time interval.
- In Image Processing, the samples can be the values of pixels along a row or column of raster images.
- DFT is also used to efficiently solve partial differential equations and to perform other operations such as convolutions or multiplying large integers.
- Easier to remove undesirable frequencies.
- Perform certain operations faster in the frequency domain than in the spatial domain.
- Cosine/Sine signals are easy to define and interpret.

2.5 Discrete Cosine Transformation(DCT)

Though due to its computational efficiency Discrete Fourier Transform (DFT) is quite popular, but it has some disadvantages as it deals with complex analysis and it has a poor energy compaction. In this scenario the concept of Discrete Cosine Transform (DCT) comes. DCT has the ability to pack spatial sequence energy into as few frequency coefficients as possible which is very important for image compression.

Discrete Cosine Transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions associated with different frequencies. DCT deals with only real analysis unlike DFT which deals with complex analysis.

DCT is very useful in analyzing the amplitude sprectrum of the signal as it concentrates signal energy in less frequency region which helps us to store amplitudes for those frequencies which will be enough to reconstruct the signal with minimum error.

Basic DCT Formula

Discrete Cosine Transform (DCT) can be mathematically expressed in two ways:

- One-dimensional (1-D) DCT Formula
- Two-dimensional (2-D) DCT Formula

2.5.1 One-dimensional (1-D) Formula

Mathematically, one-dimensional (1-D) DCT can be represented by the following equations:

1-D Forward Transformation

The one-dimensional forward transformation is denoted by the Eq.(2.5).

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x).cos[\frac{\pi(2x+1)u}{2N}] \tag{2.5}$$

where,

$\alpha(u) = \sqrt{\frac{1}{N}} \quad if u = 0$

$\alpha(u) = \sqrt{\frac{2}{N}} \quad if u \neq 0$

x is a spatial domain variable, $f(x)$ is a matrix of size N and u is a frequency domain variable and $C(u)$ is corresponding frequency domain matrix of size N.

1-D Inverse Transformation

The one-dimensional inverse transformation is denoted by the Eq.(2.6).

$$f(x) = \sum_{u=0}^{N-1} \alpha(u).C(u).cos[\frac{\pi(2x+1)u}{2N}] \tag{2.6}$$

Illustrative Example

Let us consider a spatial domain 1-D 1×2 matrix as follows:

$$f(x) = \begin{bmatrix} 110 & 120 \end{bmatrix}$$

Now using the Eq.(2.5) for 1-D forward transformation on the spatial domain matrix $f(x)$, we get the corresponding DCT co-efficients as follows:

$C(u=0) = \sqrt{\frac{1}{2}} \times [110 \times cos(0) + 120 \times cos(0)] = \frac{1}{\sqrt{2}} \times 230 = 115\sqrt{2}$

$C(u=1) = \sqrt{\frac{2}{2}} \times [110 \times cos(\frac{\pi}{4}) + 120 \times cos(\frac{3\pi}{4})] = 110 \times \frac{1}{\sqrt{2}} + 120 \times (-\frac{1}{\sqrt{2}})$
$= (55\sqrt{2} - 60\sqrt{2}) = -5\sqrt{2}$

So, the corresponding frequency domain 1×2 matrix is as follows:

$$C(u) = \begin{bmatrix} 115\sqrt{2} & -5\sqrt{2} \end{bmatrix}$$

Let us now try to find out the inverse of DCT co-efficients in $C(u)$ using the Eq.(2.6) to get back the corresponding spatial domain components.

$f(x=0) = \sqrt{\frac{1}{2}} \times (115\sqrt{2}) \times cos(0) + \sqrt{\frac{2}{2}} \times (-5\sqrt{2}) \times cos(\frac{\pi}{4})$
$= (\frac{1}{\sqrt{2}} \times 115\sqrt{2} \times 1) - (5\sqrt{2} \times \frac{1}{\sqrt{2}})$
$= (115 - 5) = 110$

$f(x=1) = \sqrt{\frac{1}{2}} \times (115\sqrt{2}) \times cos(0) + \sqrt{\frac{2}{2}} \times (-5\sqrt{2}) \times cos(\frac{3\pi}{4})$
$= (\frac{1}{\sqrt{2}} \times 115\sqrt{2} \times 1) - (5\sqrt{2}) \times (\frac{1}{\sqrt{2}})$
$= (115 + 5) = 120$

So, from the inverse DCT, we get back the original spatial domain 1×2 matrix as follows:

$f(x) = \begin{bmatrix} 110 & 120 \end{bmatrix}$

2.5.2 Two-dimensional (2-D) Formula

Two-dimensional (2-D) DCT can be mathematically expressed by the following equations:

2-D Forward Transformation

The two-dimensional forward transformation is denoted by the Eq.(2.7).

$$c(u,v) = \alpha(u).\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y).cos[\frac{\pi(2x+1)u}{2N}].cos[\frac{\pi(2y+1)v}{2N}] \quad (2.7)$$

where,
$\alpha(u) = \sqrt{\frac{1}{N}} \quad if u = 0$
$\alpha(u) = \sqrt{\frac{2}{N}} \quad if u \neq 0$

x & y are spatial domain variables, $f(x,y)$ is a matrix of size $N \times N$ and u & v are frequency domain variables.

2-D Inverse Transformation

The two-dimensional inverse transformation is expressed by the Eq.(2.8).

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u).\alpha(v).c(u,v).cos[\frac{\pi(2x+1)u}{2N}].cos[\frac{\pi(2y+1)v}{2N}] \quad (2.8)$$

DCT is completely reversible, but as it deals only with *cosine* functions so it may be sometimes lossy.

Illustrative Example

Let us now consider a spatial domain 2×2 sample matrix for the application of 2-D forward DCT as follows:

$$f(x,y) = \begin{bmatrix} 200 & 100 \\ 250 & 255 \end{bmatrix}$$

Now using the Eq.(2.7) for 2-D forward transformation on the matrix $f(x,y)$, we get the corresponding 2-D DCT co-efficients as follows:

$C(u=0,v=0) = \sqrt{\frac{1}{2}} \times \sqrt{\frac{1}{2}} \times [200 \times cos(0) \times cos(0) + 100 \times cos(0) \times cos(0) + 250 \times cos(0) \times cos(0) + 255 \times cos(0) \times cos(0)]$
$= \frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}} \times [200 + 100 + 250 + 255] = \frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}} \times 805 = 402.5$

$C(u=0,v=1) = \sqrt{\frac{1}{2}} \times \sqrt{\frac{2}{2}} \times [200 \times cos(0) \times cos(\frac{\pi}{4}) + 100 \times cos(0) \times cos(\frac{3\pi}{4}) + 250 \times cos(0) \times cos(\frac{\pi}{4}) + 255 \times cos(0) \times cos(\frac{3\pi}{4})]$
$= \frac{1}{\sqrt{2}} \times [200 \times 1 \times \frac{1}{\sqrt{2}} + 100 \times 1 \times (-\frac{1}{\sqrt{2}}) + 250 \times 1 \times \frac{1}{\sqrt{2}} + 255 \times 1 \times (-\frac{1}{\sqrt{2}})] = 47.5$

$C(u=1,v=0) = \sqrt{\frac{2}{2}} \times \sqrt{\frac{1}{2}} \times [200 \times cos(\frac{\pi}{4}) \times cos(0) + 100 \times cos(\frac{\pi}{4}) \times cos(0) + 250 \times cos(\frac{3\pi}{4}) \times cos(0) + 255 \times cos(\frac{3\pi}{4}) \times cos(0)]$
$= \frac{1}{\sqrt{2}} \times [200 \times \frac{1}{\sqrt{2} \times 1} + 100 \times \frac{1}{\sqrt{2}} \times 1 + 250 \times (-\frac{1}{\sqrt{2}}) \times 1 + 255 \times (-\frac{1}{\sqrt{2}}) \times 1] = -102.5$

$C(u=1,v=1) = \sqrt{\frac{2}{2}} \times \sqrt{\frac{2}{2}} \times [200 \times cos(\frac{\pi}{4}) \times cos(\frac{\pi}{4}) + 100 \times cos(\frac{\pi}{4}) \times cos(\frac{3\pi}{4}) + 250 \times cos(\frac{3\pi}{4}) \times cos(\frac{\pi}{4}) + 255 \times cos(\frac{3\pi}{4}) \times cos(\frac{3\pi}{4})]$
$= [200 \times \frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}} + 100 \times \frac{1}{\sqrt{2}} \times (-\frac{1}{\sqrt{2}}) + 250 \times (-\frac{1}{\sqrt{2}}) \times \frac{1}{\sqrt{2}} + 255 \times (-\frac{1}{\sqrt{2}}) \times (-\frac{1}{\sqrt{2}})] = 52.5$

So the 2×2 corresponding DCT co-efficients are as follows:

$$C(u,v) = \begin{bmatrix} 402.5 & 47.5 \\ -102.5 & 52.5 \end{bmatrix}$$

Now, inverse transformation of the DCT co-efficients in $C(u,v)$ is aplied using the Eq.(2.8) to get back the corresponding spatial domain components.

$f(x=0,y=0) = [\sqrt{\frac{1}{2}} \times \sqrt{\frac{1}{2}} \times 402.5 \times cos(0) \times cos(0) + \sqrt{\frac{1}{2}} \times \sqrt{\frac{2}{2}} \times$

$47.5 \times cos(0) \times cos(\frac{\pi}{4}) + \sqrt{\frac{2}{2}} \times \sqrt{\frac{1}{2}} \times (-102.5) \times cos(\frac{\pi}{4}) \times cos(0) + \sqrt{\frac{2}{2}} \times \sqrt{\frac{2}{2}} \times 52.5 \times cos(\frac{\pi}{4}) \times cos(\frac{\pi}{4})]$
$= [\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}} \times 402.5 \times 1 \times 1 + \frac{1}{\sqrt{2}} \times 47.5 \times 1 \times \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \times (-102.5) \times \frac{1}{\sqrt{2}} \times 1 + 52.5 \times \frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}] = 200$

$f(x = 0, y = 1) = [\sqrt{\frac{1}{2}} \times \sqrt{\frac{1}{2}} \times 402.5 \times cos(0) \times cos(0) + \sqrt{\frac{1}{2}} \times \sqrt{\frac{2}{2}} \times 47.5 \times cos(0) \times cos(\frac{3\pi}{4}) + \sqrt{\frac{2}{2}} \times \sqrt{\frac{1}{2}} \times (-102.5) \times cos(\frac{\pi}{4}) \times cos(0) + \sqrt{\frac{2}{2}} \times \sqrt{\frac{2}{2}} \times 52.5 \times cos(\frac{\pi}{4}) \times cos(\frac{3\pi}{4})]$
$= [\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}} \times 402.5 \times 1 \times 1 + \frac{1}{\sqrt{2}} \times 47.5 \times 1 \times (-\frac{1}{\sqrt{2}}) + \frac{1}{\sqrt{2}} \times (-102.5) \times \frac{1}{\sqrt{2}} \times 1 + 52.5 \times \frac{1}{\sqrt{2}} \times (-\frac{1}{\sqrt{2}})] = 100$

$f(x = 1, y = 0) = [\sqrt{\frac{1}{2}} \times \sqrt{\frac{1}{2}} \times 402.5 \times cos(0) \times cos(0) + \sqrt{\frac{1}{2}} \times \sqrt{\frac{2}{2}} \times 47.5 \times cos(0) \times cos(\frac{\pi}{4}) + \sqrt{\frac{2}{2}} \times \sqrt{\frac{1}{2}} \times (-102.5) \times cos(\frac{3\pi}{4}) \times cos(0) + \sqrt{\frac{2}{2}} \times \sqrt{\frac{2}{2}} \times 52.5 \times cos(\frac{3\pi}{4}) \times cos(\frac{\pi}{4})]$
$= [\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}} \times 402.5 \times 1 \times 1 + \frac{1}{\sqrt{2}} \times 47.5 \times 1 \times \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \times (-102.5) \times (-\frac{1}{\sqrt{2}}) \times 1 + 52.5 \times (-\frac{1}{\sqrt{2}}) \times \frac{1}{\sqrt{2}}] = 250$

$f(x = 1, y = 1) = [\sqrt{\frac{1}{2}} \times \sqrt{\frac{1}{2}} \times 402.5 \times cos(0) \times cos(0) + \sqrt{\frac{1}{2}} \times \sqrt{\frac{2}{2}} \times 47.5 \times cos(0) \times cos(3\frac{\pi}{4}) + \sqrt{\frac{2}{2}} \times \sqrt{\frac{1}{2}} \times (-102.5) \times cos(3\frac{3\pi}{4}) \times cos(0) + \sqrt{\frac{2}{2}} \times \sqrt{\frac{2}{2}} \times 52.5 \times cos(\frac{3\pi}{4}) \times cos(\frac{3\pi}{4})]$
$= [\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}} \times 402.5 \times 1 \times 1 + \frac{1}{\sqrt{2}} \times 47.5 \times 1 \times (-\frac{1}{\sqrt{2}}) + \frac{1}{\sqrt{2}} \times (-102.5) \times (-\frac{1}{\sqrt{2}}) \times 1 + 52.5 \times (-\frac{1}{\sqrt{2}}) \times (-\frac{1}{\sqrt{2}})] = 255$

Thus, we get back the original spatial domain 2×2 matrix $f(x, y)$ by inverse transformation as follows:

$$f(x, y) = \begin{bmatrix} 200 & 100 \\ 250 & 255 \end{bmatrix}$$

2.5.3 Application of Discrete Cosine Transformation

Depending upon the features, there are some applications where DCT is applied. They are as follows:

One-dimensional DCT/IDCT

- Dolby AC2 & AC3.
- Speech information compression.
- Biomedical signals like EEG, ECG.

Two-dimensional DCT/IDCT

- MPEG-1 & MPEG-2.
- Image acquisition, image recognition & pattern recognition.
- JPEG Encoding, JPEG Compression.

Again for steganographic application, DCT encoding is sometimes used with Huffman encoding for its good compressing nature. Here DCT is used on cover image blocks whereas Huffman encoding is applied over the authenticating bits/messages/image.

Chapter 3

Z Transformation Theory & Proposed Scheme

3.1 What is Z transformation ?

In signal processing, Z transformation basically converts a discrete time domain signal into complx frequency domain representation consisting of both real and imaginary components. Z transformation is mathematically derived from the Laplace transformation.

3.2 The Transformation Technique

Mathematically the Region of Convergence (ROC) is the set of all the points in the complex plane for which the Z transform summation converges. The set of all the complex points z such that the Eq.(3.1) holds.

$$\left| \sum_{n=-\infty}^{\infty} x[n] z^{-n} \right| < \infty \tag{3.1}$$

The definition of Generalized Forward Z Transformation (GFZT) is given by the Eq.(3.2).

$$z(re^{j\omega}) = \sum_{h=0}^{\infty} g(h).r^{-h}.e^{-j\omega h} \tag{3.2}$$

where r is the magnitude of z or the ROC of z, j is the imaginary unit, ω is the angle in radians and h is the positional indicator. The proposed technique considered $\omega = 0, \frac{\Pi}{2}, \Pi, \frac{3\Pi}{2}$ and $h = 0, 1, 2, 3$. g(h) depicts the spatial domain matrix elements.

Now, the Generalized Inverse Z Transformation (GIZT) with the ROC of z as r is defined by the Eq.(3.3).

$$g(re^{j\omega}) = \frac{r^h}{M} . \sum_{h=0}^{\infty} z(h).r^h.e^{j\omega h} \tag{3.3}$$

where M is the total number of frequency domain components. In this case, for a 2×2 mask, frequency domain components of same dimensions are generated and thus here the value of $M = 4$.

3.3 Z transformation properties

As a specific frequency domain transformation technique, Z transformation possesses some special properties. They are discussed in details as follows:

- for One-dimensional (1-D) Z transformation
- for two-dimensional (2-D) Z transformation

3.3.1 Properties for one-dimensional (1-D) transformation

Using the following tabular representation, the properties of one-dimensional (1-D) Z transformation based on the signal sequence is provided and also the corresponding Region of Convergence (ROC) is enlisted in the table 3.1.

3.3.2 Properties for two-dimensional (2-D) transformation

Again, using the following tabular representation, the properties of two-dimensional (2-D) Z transformation based on the signal sequence is provided and also the corresponding Region of Convergence (ROC) is enlisted in the

Table 3.1: Look-up Table for 1-D Z Transformation Properties

Property	**Sequence**	**Z Domain**	**ROC**
	$g[n]$	$G(Z)$	R_g
	$h[n]$	$H(Z)$	R_h
Linearity	$\alpha g[n] + \beta h[n]$	$\alpha G(Z) + \beta H(Z)$	$R_g \bigcap R_h$
Conjugation	$g^*[n]$	$G^*(Z^*)$	R_g
Time Reversal	$g[-n]$	$G(\frac{1}{Z})$	$\frac{1}{R_g}$
Time Shifting	$g[n - n_0]$	$Z^{-n_0}G(Z)$	R_g excluding $Z = 0$ or $Z = \infty$
Differentiation	$ng[n]$	$-Z\frac{d}{dZ}G(Z)$	R_g excluding $Z = 0$ or $Z = \infty$
Convolution	$g[n] * h[n]$	$G(Z)H(Z)$	$R_g \bigcap R_h$
Multiplication of Exponential	$\alpha^n g[n]$	$G(\frac{Z}{\alpha})$	$\|\alpha\| R_g$

table 3.2.

The mask generation and the embedding procedure of the generated embedding mask is discussed as follows:

3.4 Mask Generation

In this technique the authenticating image pixels in row major order are converted to equivalent binary value and for a hidden authenticating image of size 192×128 equivalent binary authenticating mask of size 192×1024 is obtained. A chaotic mask of same size is generated by using the Skew Tent map [6] with the initial conditions and system parameter (α) values which act as the key values. A bitwise XOR operation is performed between the authenticating mask and chaotic mask to generate the embedding mask.

Table 3.2: Look-up Table for 2-D Z Transformation Properties

Property	**Sequence**	**Z Domain**	**ROC**
	$g[n][m]$	$G(Z1)(Z2)$	R_g
	$h[n][m]$	$H(Z1)(Z2)$	R_h
Linearity	$\alpha g[n][m] + \beta h[n][m]$	$\alpha G(Z1)(Z2) + \beta H(Z1)(Z2)$	$R_g \bigcap R_h$
Conjugation	$g^*[n][m]$	$G^*(Z1^*)(Z2^*)$	R_g
Time Reversal	$g[-n][-m]$	$G(\frac{1}{Z1})(\frac{1}{Z2})$	$\frac{1}{R_g}$
Time Shifting	$g[n-n_0][m-m_0]$	$Z1^{-n_0}Z2^{-m_0}G(Z)$	R_g excluding either $Z1=0$ or $Z1=\infty$ or $Z2=0$ or $Z2=\infty$
Differentiation	$nmg[n][m]$	$Z1Z2\frac{d}{dZ1}\frac{d}{dZ2}G(Z1)(Z2)$	R_g excluding either $Z1=0$ or $Z1=\infty$ or $Z2=0$ or $Z2=\infty$
Convolution	$g[n][m] * h[n][m]$	$G(Z1)(Z2)H(Z1)(Z2)$	$R_g \bigcap R_h$
Multiplication of Exponential	$\alpha^n\beta^m g[n][m]$	$G(\frac{Z1}{\alpha})(\frac{Z2}{\beta})$	$\|\alpha\|\|\beta\|R_g$

3.5 Embedding

The technique uses a grayscale image of size 512×512 as the carrier/cover image. The pixel components of the cover image undergo a pre-adjustment stage, if necessary. A forward Z domain transformation is applied on the 2×2 mask from the cover image in the row major order. 3 bits from the embedding mask is embedded in the frequency domain components of each 2×2 mask converted into Z domain. The process of embedding is given in algorithm 1.

Algorithm 1.

Input: A cover image , say I_{Cov} of size 512×512 , a hidden embedding mask (E) of size 192×128.

Output: An embedded image of size 512×512.

Method: The Generalized Forward Z Transformation (GFZT) converts I_{Cov} from spatial domain to frequency domain. The transform domain components are utilized to fabricate E (authenticating image bits XORed with the chaotic mask).

1: *A pre-adjustment method is applied on each pixel component p of the spatial domain grayscale image to attain the pixel values positive and less than, or equal to 255 even after the embedding process.*

If $p \leq 4$, *then* $p = (p + 4)$

If $p \geq 251$, *then* $p = (p - 4)$

2: *Cover image mask of size* 2×2 *from pre-adjusted image matrix is considered in a sliding window manner in row major order and GFZT is applied on the* 2×2 *mask to obtain the corresponding frequency domain components.*

3: *Sequencially* 3 *bits from the embedding mask are considered and consequently, the LSB of the imaginary integer part of the HF component is replaced by the first* 1 *bit from the mask and also for proper computation the same bit is embedded in the LSB of the imaginary part of the complex conjugate of the HF component. The LSB and* $(LSB+1)^{th}$ *bit of the real integer part of the VF component are replaced by the rest* 2 *bits.*

4: *A Fidelity Adjustment (FA) is applied, if needed.*

5: *GIZT is applied on the embedded frequency domain* 2×2 *mask to get back the spatial domain mask of size* 2×2.

6: *Fraction handling is performed on the embedded spatial domain* 2×2 *mask to avoid precision loss of the frequency domain coefficients and helps in proper decoding at the receiver end.*

7: *Steps* 2 *to* 6 *are repeated for the whole cover image until the last* 2×2 *mask is processed.*

3.6 Fidelity Adjustment

Adjustment of fidelity in bit-level is an added step which minimizes the changes in the image even after embedding the hidden data without affecting the bits inserted. Two embedded bit positions are LSB and (LSB+1). Let us consider C_{iOld} and C_{iNew} as the input pixel value and embedded pixel value of the real integer part of the VF component respectively. The detailed steps of the FA phase are as follows.

1: *The difference* $D_i = (C_{iNew} - C_{iOld})$ *is calculated.*
2: *If* $D_i = 3$*, then from* $(LSB + 2)^{nd}$ *position towards the MSB the first occurrence of bit value* 1 *is checked and it is flipped to bit value* 0*. The bits from the bit position toward the LSB excepting LSB and* $(LSB+1)^{th}$ *bit are flipped to* 1*.*
3: *If* $D_i = -3$*, then from* $(LSB + 2)^{nd}$ *position towards the MSB the first occurrence of bit value* 0 *is checked and it is flipped to bit value* 1*. All the bits from the bit position towards the LSB excepting LSB and* $(LSB+1)^{th}$ *bit are flipped to* 0*.*
4: *If* $|D_i| < 3$*, then no action is needed.*

In all the above cases, the LSB and (LSB+1)th bit value must remain unchanged for proper decoding.

Fidelity Adjustment - An Illustration

Positive Difference

Let, a pixel under consideration has a pixel value **200** and let the 2 bits to be embedded is **11**.

Now, the corresponding binary equivalent is : **200 = 11001000**.

After embedding the 2 bits the binary value becomes : **11001011 = 203**.

So, difference becomes = **(203 - 200) = 3**

After fidelity adjustment, we get binary value as : **11000111 = 199**, which is obviously quite close to the original value.

Negative Difference

Let, a pixel under consideration has a pixel value **135** and let the 2 bits to be embedded is **00**.

Now, the corresponding binary equivalent is : **135 = 10000111**.

After embedding the 2 bits the binary value becomes : **11000100 = 132**.

So, difference becomes = **(132 - 135) = -3**

After fidelity adjustment, we get binary value as : **11001000 = 136**, which is obviously quite close to the original value.

3.7 Handling with Fraction

Handling with Fraction is an essential step to avoid any type of precision loss after converting the frequency domain components into spatial domain. The detailed steps for fraction handling are as follows.

1: *The fractional part of the* $(0,0)^{th}$ *and* $(0,1)^{th}$ *element from the* 2×2 *mask converted from frequency domain to spatial domain after embedding and FA, if needed is taken. Let the fractional parts be* $Frac_{(0,0)}$ *and* $Frac_{(0,1)}$ *respectively.*

2: $Frac_{(0,0)}$ *and* $Frac_{(0,1)}$ *are added to get* $Frac_{tot}$*.*

If $Frac_{tot} \geq 1$*, then* $Frac_{tot} = (Frac_{tot} - 1)$*.*

3: $Frac_{tot}$ *is multiplied by* 4 *to obtain* $Frac_{comp}$*.*

$$Frac_{comp} = (Frac_{tot} \times 4)$$

4: *The* $(1,1)^{th}$ *element in the* 2×2 *embedded mask is compared with the* $(1,1)^{th}$ *element in the* 2×2 *original cover image mask. Let the two elements be denoted by* $I_{new}(1,1)$ *and* $I_{new}(1,1)$ *respectively.*

If $I_{new}(1,1) < I_{old}(1,1)$*, then* $Frac_{comp}$ *is added to the real part of LF component of the corresponding* 2×2 *frequency domain mask. Otherwise,* $Frac_{comp}$ *is subtracted from the real part of the LF component.*

5: *GIZT is performed on the modified frequency domain mask to get back the modified spatial domain components.*

6: *If* $Frac_{(0,0)} > 0$*, then the steps* 2 *to* 5 *are repeated, Otherwise stop.*

Now using the Eq.(2.7) for 2-D forward transformation on the matrix $f(x,y)$, we get the corresponding 2-D DCT co-efficients as follows:

3.8 Decoding

The embedded image is received in spatial domain at the receiver end. The GFZT is applied on the 2×2 mask from the embedded image in row major order and consequently the embedded bits are extracted. The algorithm is given as algorithm 2.

Algorithm 2.
Input: The embedded image , say I_{Emb} of size 512×512 in spatial domain.
Output: The extracted mask of size 192×128.
Method: GFZT is applied on the embedded image to extract the embedding mask (authentication image bits XORed with chaotic mask) by con-

verting the image from spatial domain to transform domain. Successive extracted bits re-generate the embedding mask. The detailed steps of extraction are as follows.

1: *The embedded image matrix is taken under consideration.*
2: *Mask of size* 2×2 *from* I_{Emb} *is considered in sliding manner in row major order and GFZT is applied on the* 2×2 *mask to get the frequency domain components.*
3: 1 *bit from either the LSB of the integer part of the imaginary HF component or the LSB of the integer imaginary part of the complex conjugate of the HF component is extracted. Consequently* 2 *bits are extracted from the LSB and (LSB+1)*th *bit of the real integer part of the VF component.*
4: *Successively the bits are extracted from each* 2×2 *mask and the steps* 2 *to* 3 *are repeated until all the* 2×2 *masks are processed.*

3.9 Authentication

For authentication, the extracted mask is bitwise XORed with the chaotic mask generated at the receiver using the same initial conditions and system parameter as used during the formation of embedding mask at the sender. From the extracted binary mask the binary authenticating mask is formed. For each consecutive 8 bits from the extracted authenticating mask in row major order the decimal equivalent is obtained. Now the extracted hidden image is compared with the original hidden image for authentication.

3.10 A Demonstrative Example

At Sender End

Let us now consider a spatial domain 2×2 sample matrix for the application of 1-D forward Z as follows:

$$g(h) = \begin{bmatrix} 56 & 112 \\ 192 & 212 \end{bmatrix}$$

Now using the Eq.(3.2) for 1-D forward transformation on the matrix $g(h)$ and considering $r = 2$, we get the corresponding 1-D Z co-efficients as follows:

$F(0) = 56 \times 2^{-0} \times (cos(0 \times 0) - jsin(0 \times 0)) + 112 \times 2^{-1} \times (cos(0 \times 1) - jsin(0 \times 1)) + 192 \times 2^{-2} \times (cos(0 \times 2) - jsin(0 \times 2)) + 212 \times 2^{-3} \times (cos(0 \times 3) - jsin(0 \times 3))$
$= [56 \times (1) + 112 \times \frac{1}{2} \times (1) + 192 \times \frac{1}{4} \times (1) + 212 \times \frac{1}{8} \times (1)] = 186.50$

$F(\frac{\pi}{2}) = 56 \times 2^{-0} \times (cos(\frac{\pi}{2} \times 0) - jsin(\frac{\pi}{2} \times 0)) + 112 \times 2^{-1} \times (cos(\frac{\pi}{2} \times 1) - jsin(\frac{\pi}{2} \times 1)) + 192 \times 2^{-2} \times (cos(\frac{\pi}{2} \times 2) - jsin(\frac{\pi}{2} \times 2)) + 212 \times 2^{-3} \times (cos(\frac{\pi}{2} \times 3) - jsin(\frac{\pi}{2} \times 3))$
$= [56 \times (1) + 112 \times \frac{1}{2} \times (-j) + 192 \times \frac{1}{4} \times (-1) + 212 \times \frac{1}{8} \times (j)] = 8.00 - 29.50j$

$F(\pi) = 56 \times 2^{-0} \times (cos(\pi \times 0) - jsin(\pi \times 0)) + 112 \times 2^{-1} \times (cos(\pi \times 1) - jsin(\pi \times 1)) + 192 \times 2^{-2} \times (cos(\pi \times 2) - jsin(\pi \times 2)) + 212 \times 2^{-3} \times (cos(\pi \times 3) - jsin(\pi \times 3))$
$= [56 \times (1) + 112 \times \frac{1}{2} \times (-1) + 192 \times \frac{1}{4} \times (1) + 212 \times \frac{1}{8} \times (-1)] = 21.50$

$F(\frac{3\pi}{2}) = 56 \times 2^{-0} \times (cos(\frac{3\pi}{2} \times 0) - jsin(\frac{3\pi}{2} \times 0)) + 112 \times 2^{-1} \times (cos(\frac{3\pi}{2} \times 1) - jsin(\frac{3\pi}{2} \times 1)) + 192 \times 2^{-2} \times (cos(\frac{3\pi}{2} \times 2) - jsin(\frac{3\pi}{2} \times 2)) + 212 \times 2^{-3} \times (cos(\frac{3\pi}{2} \times 3) - jsin(\frac{3\pi}{2} \times 3))$
$= [56 \times (1) + 112 \times \frac{1}{2} \times (j) + 192 \times \frac{1}{4} \times (-1) + 212 \times \frac{1}{8} \times (-j)] = 8.00 + 29.50j$

So, the Z domain components after using forward Z transformation are as follows:

$$F(\omega) = \begin{bmatrix} 186.50 & 8.00 - 29.50j \\ 21.50 & 8.00 + 29.50j \end{bmatrix}$$

Now, as discussed 3 bits from the embedding mask is embedded in the 2×2 block, so let the bitstream to be embedded is 110.

After embedding 0 in the imaginary part of the HF component and the imaginary part of the complex conjugate of HF component and 11 in the real part of the VF component, we are getting the embedded frequency domain matrix as follows:

$$F(\omega) = \begin{bmatrix} 186.50 & 8.00 - 28.50j \\ 23.50 & 8.00 + 28.50j \end{bmatrix}$$

Now using the Eq.(3.3) on the Z domain co-efficients, the spatial domain embedded values are as follows:

$g(0) = \frac{2^0}{4}[186.50 \times (cos(0 \times 0) + jsin(0 \times 0)) + (8.00 - 28.50j) \times (cos(0 \times 1) + jsin(0 \times 1)) + 23.50 \times (cos(0 \times 2) + jsin(0 \times 2)) + (8.00 + 28.50j) \times (cos(0 \times 3) + jsin(0 \times 3))]$
$= \frac{1}{4}[186.50 \times (1) + (8.00 - 28.50j) \times (1) + 23.50 \times (1) + (8.00 - 28.50j) \times (1)] = 56.50$

$g(1) = \frac{2^1}{4}[186.50 \times (cos(\frac{\pi}{2} \times 0) + jsin(\frac{\pi}{2} \times 0)) + (8.00 - 28.50j) \times (cos(\frac{\pi}{2} \times 1) + jsin(\frac{\pi}{2} \times 1)) + 23.50 \times (cos(\frac{\pi}{2} \times 2) + jsin(\frac{\pi}{2} \times 2)) + (8.00 + 28.50j) \times (cos(\frac{\pi}{2} \times 3) + jsin(\frac{\pi}{2} \times 3))]$
$= \frac{2}{4}[186.50 \times (1) + (8.00 - 28.50j) \times (j) + 23.50 \times (-1) + (8.00 - 28.50j) \times (-j)] = 110.00$

$g(2) = \frac{2^2}{4}[186.50 \times (cos(\pi \times 0) + jsin(\pi \times 0)) + (8.00 - 28.50j) \times (cos(\pi \times 1) + jsin(\pi \times 1)) + 23.50 \times (cos(\pi \times 2) + jsin(\pi \times 2)) + (8.00 + 28.50j) \times (cos(\pi \times 3) + jsin(\pi \times 3))]$
$= \frac{4}{4}[186.50 \times (1) + (8.00 - 28.50j) \times (-1) + 23.50 \times (1) + (8.00 - 28.50j) \times (-1)] = 194.00$

$g(3) = \frac{2^3}{4}[186.50 \times (cos(3\frac{\pi}{2} \times 0) + jsin(\frac{3\pi}{2} \times 0)) + (8.00 - 28.50j) \times (cos(\frac{3\pi}{2} \times 1) + jsin(\frac{3\pi}{2} \times 1)) + 23.50 \times (cos(\frac{3\pi}{2} \times 2) + jsin(\frac{3\pi}{2} \times 2)) + (8.00 + 28.50j) \times (cos(\frac{3\pi}{2} \times 3) + jsin(\frac{3\pi}{2} \times 3))]$
$= \frac{8}{4}[186.50 \times (1) + (8.00 - 28.50j) \times (-j) + 23.50 \times (-1) + (8.00 - 28.50j) \times (j)] = 212.00$

So, the spatial domain 2×2 embedded matrix after inverse transformation is as follows:

$$g(h) = \begin{bmatrix} 56.50 & 110.00 \\ 194.00 & 212.00 \end{bmatrix}$$

But, here is a problem with the embedded 2×2 matrix. In spatial domain, a pixel cannot have a value with fractional part. So, the fraction must be handled.

Now, using the fraction handling process we get $Frac_{0,0} = 0.50 \& Frac_{0,1} = 0.00$. So, we have to add $[(0.50 + 0.00) \times 4] = 2.00$ with the g(0) value to get the result $F(0) = [F(0) + 2.00] = [186.50 + 2.00] = 188.50$

Now again using the Eq.(3.3) on the modified Z domain co-efficients, the spatial domain embedded values are as follows:

$g(0) = \frac{2^0}{4}[188.50 \times (cos(0 \times 0) + jsin(0 \times 0)) + (8.00 - 28.50j) \times (cos(0 \times 1) + jsin(0 \times 1)) + 23.50 \times (cos(0 \times 2) + jsin(0 \times 2)) + (8.00 + 28.50j) \times (cos(0 \times 3) + jsin(0 \times 3))]$
$= \frac{1}{4}[188.50 \times (1) + (8.00 - 28.50j) \times (1) + 23.50 \times (1) + (8.00 - 28.50j) \times (1)] = 57$

$g(1) = \frac{2^1}{4}[188.50 \times (cos(\frac{\pi}{2} \times 0) + jsin(\frac{\pi}{2} \times 0)) + (8.00 - 28.50j) \times (cos(\frac{\pi}{2} \times 1) + jsin(\frac{\pi}{2} \times 1)) + 23.50 \times (cos(\frac{\pi}{2} \times 2) + jsin(\frac{\pi}{2} \times 2)) + (8.00 + 28.50j) \times (cos(\frac{\pi}{2} \times 3) + jsin(\frac{\pi}{2} \times 3))]$
$= \frac{2}{4}[188.50 \times (1) + (8.00 - 28.50j) \times (j) + 23.50 \times (-1) + (8.00 - 28.50j) \times (-j)] = 111$

$g(2) = \frac{2^2}{4}[188.50 \times (cos(\pi \times 0) + jsin(\pi \times 0)) + (8.00 - 28.50j) \times (cos(\pi \times 1) + jsin(\pi \times 1)) + 23.50 \times (cos(\pi \times 2) + jsin(\pi \times 2)) + (8.00 + 28.50j) \times (cos(\pi \times 3) + jsin(\pi \times 3))]$
$= \frac{4}{4}[188.50 \times (1) + (8.00 - 28.50j) \times (-1) + 23.50 \times (1) + (8.00 - 28.50j) \times (-1)] = 196$

$g(3) = \frac{2^3}{4}[188.50 \times (cos(3\frac{\pi}{2} \times 0) + jsin(\frac{3\pi}{2} \times 0)) + (8.00 - 28.50j) \times (cos(\frac{3\pi}{2} \times 1) + jsin(\frac{3\pi}{2} \times 1)) + 23.50 \times (cos(\frac{3\pi}{2} \times 2) + jsin(\frac{3\pi}{2} \times 2)) + (8.00 + 28.50j) \times (cos(\frac{3\pi}{2} \times 3) + jsin(\frac{3\pi}{2} \times 3))]$
$= \frac{8}{4}[188.50 \times (1) + (8.00 - 28.50j) \times (-j) + 23.50 \times (-1) + (8.00 - 28.50j) \times (j)] = 216$

So, the embedded spatial domain 2×2 matrix after inverse transformation and handling with fraction is as follows:

$$g(h) = \begin{bmatrix} 57 & 111 \\ 196 & 216 \end{bmatrix}$$

At Receiver End

At the receiver terminal, the embedded matrix is obtained after transmission. So in this end, for decoding the authenticating bits the forward Z transformation is applied on the embedded 2×2 matrix using Eq.(3.2) as follows:

$F(0) = 57\times2^{-0}\times(cos(0\times0)-jsin(0\times0))+111\times2^{-1}\times(cos(0\times1)-jsin(0\times 1))+196\times2^{-2}\times(cos(0\times2)-jsin(0\times2))+216\times2^{-3}\times(cos(0\times3)-jsin(0\times3))$
$= [57 \times (1) + 111 \times \frac{1}{2} \times (1) + 196 \times \frac{1}{4} \times (1) + 216 \times \frac{1}{8} \times (1)] = 188.50$

$F(\frac{\pi}{2}) = 57 \times 2^{-0} \times (cos(\frac{\pi}{2} \times 0) - jsin(\frac{\pi}{2} \times 0)) + 111 \times 2^{-1} \times (cos(\frac{\pi}{2} \times 1) - jsin(\frac{\pi}{2} \times 1)) + 196 \times 2^{-2} \times (cos(\frac{\pi}{2} \times 2) - jsin(\frac{\pi}{2} \times 2)) + 216 \times 2^{-3} \times (cos(\frac{\pi}{2} \times 3) - jsin(\frac{\pi}{2} \times 3))$
$= [57 \times (1) + 111 \times \frac{1}{2} \times (-j) + 196 \times \frac{1}{4} \times (-1) + 216 \times \frac{1}{8} \times (j)] = 8.00 - 28.50j$

$F(\pi) = 57 \times 2^{-0} \times (cos(\pi \times 0) - jsin(\pi \times 0)) + 111 \times 2^{-1} \times (cos(\pi \times 1) - jsin(\pi \times 1)) + 196 \times 2^{-2} \times (cos(\pi \times 2) - jsin(\pi \times 2)) + 216 \times 2^{-3} \times (cos(\pi \times 3) - jsin(\pi \times 3))$
$= [57 \times (1) + 111 \times \frac{1}{2} \times (-1) + 196 \times \frac{1}{4} \times (1) + 216 \times \frac{1}{8} \times (-1)] = 23.50$

$F(\frac{3\pi}{2}) = 57 \times 2^{-0} \times (cos(\frac{3\pi}{2} \times 0) - jsin(\frac{3\pi}{2} \times 0)) + 111 \times 2^{-1} \times (cos(\frac{3\pi}{2} \times 1) - jsin(\frac{3\pi}{2} \times 1)) + 196 \times 2^{-2} \times (cos(\frac{3\pi}{2} \times 2) - jsin(\frac{3\pi}{2} \times 2)) + 216 \times 2^{-3} \times (cos(\frac{3\pi}{2} \times 3) - jsin(\frac{3\pi}{2} \times 3))$
$= [57 \times (1) + 111 \times \frac{1}{2} \times (j) + 196 \times \frac{1}{4} \times (-1) + 216 \times \frac{1}{8} \times (-j)] = 8.00 + 28.50j$

So, the Z domain components after using forward Z transformation are as follows:

$$F(\omega) = \begin{bmatrix} 188.50 & 8.00 - 28.50j \\ 23.50 & 8.00 + 28.50j \end{bmatrix}$$

Now from the LSB and $(LSB + 1)th$ position of the real part of VF the bits extracted are 11. 23 = 000101**11** and from the LSB position of the imaginary part of HF component (or imaginary part of complex conjugate of HF component), 0 is extracted as 28 = 0001110**0**. Thus, we get back the authenticating bits 110.

Chapter 4

Results & Comparisons

For the purpose of embedding mask generation 1-D skew tent map [6] has been taken under consideration which has been used to formulate 2-D chaotic map by cross-coupling of two piecewise linear 1-D skew tent maps.

$$x(i+1) = \frac{x(i)}{\alpha}; 0 \leq x(i) < \alpha$$
$$x(i+1) = \frac{1-x(i)}{1-\alpha}; \alpha < x(i) \leq 1$$

To ensure the chaotic nature of the map, two sensitivity curves of chaotic solution of the skew tent map on system parameter (α) and initial condition are shown in Fig. 4.1 and Fig. 4.2 respectively.

To formulate the results, an extensive analysis has been made on benchmark (PGM) [5] images of size 512×512. The proposed technique is applied on the cover images of the database under consideration.

The authenticating source image is given in the Fig. 4.3

The original cover (PGM) images and the embedded images with initial parameters $x_0 = 0.4417689$, $y_0 = 0.754193089$ and system parameter $\alpha = 0.49493$ are given in Fig. 4.4(a) and Fig. 4.4(b) respectively as follows:

MSE and PSNR between the embedded images and original cover images are calculated. The MSE, PSNR and IF values obtained for various images in the proposed scheme is provided in Table 4.1.

From Table 4.1 it is seen that, the average MSE value (5.10459099) is

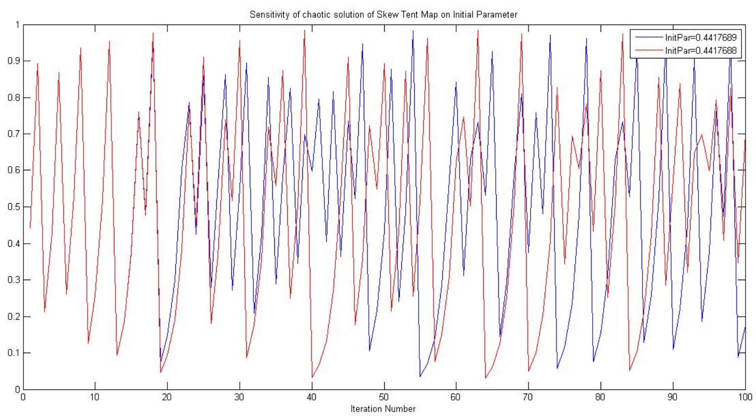


Figure 4.1: Sensitivity of Skew Tent Chaotic Map on Initial Parameter

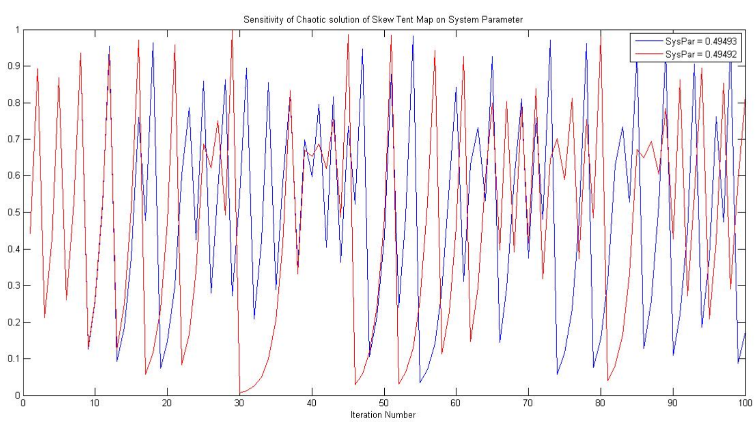


Figure 4.2: Sensitivity of Skew Tent Chaotic Map on System Parameter

Figure 4.3: Authenticating Source Image ‘Gold Coin’

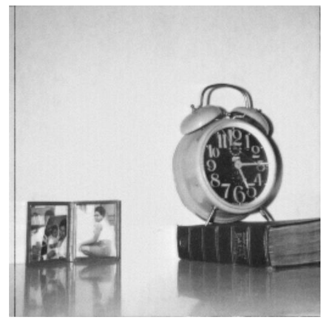
Table 4.1: MSE, PSNR and IF values obtained for several images

Image	MSE	PSNR(dB)	IF
Baboon	5.12048721	41.03769075	0.99972354
Elaine	5.11608887	41.04142281	0.99972992
Clock	5.11590195	41.04158149	0.99975231
Boat	5.09545135	41.05897701	0.99968587
Couple	5.03861999	41.10768755	0.99986659
Jet	5.10183334	41.05354093	0.99983618
Map	5.13410568	41.02615557	0.99971782
Space	5.10458374	41.05120028	0.99958029
Tank	5.05083084	41.09717537	0.99970333
Truck	5.16796494	40.99760803	0.99984834
Average	***5.10459099***	***41.05130398***	***0.99974442***

quite low and thus the average PSNR value (41.05130398) is considerably high and the minimum PSNR value is 40.99760803. The minimum IF value is 0.99958029. The average IF value (0.99974442) with close proximity to 1 ensures the imperceptibility of the embedded image through visual perception.

A graphical comparison has been drawn in Fig. 4.5 based on the average PSNR values in different existing methods. In AINCDCT (2011) [4], if the payload is 0.5 bpB then the average PSNR value is 45.61 whereas if the payload is 1 bpB the PSNR decreases to 32.53. In Luo's Method (2011) [7], for payload of 0.3 bpB the average PSNR value is 41.16. Again, in SCDFT

(a)

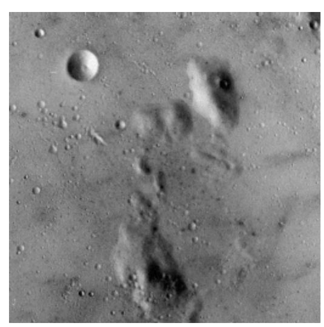


(b)

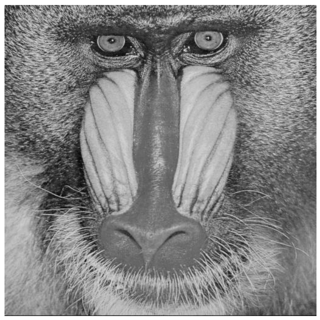
(a)

(b)

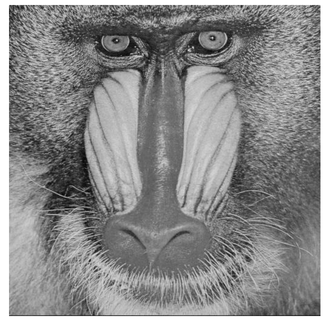
(a)



(b)



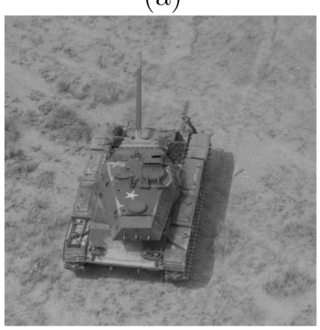
(a)



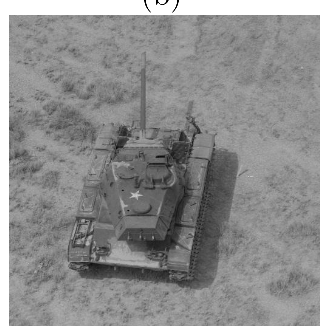
(b)

(a)

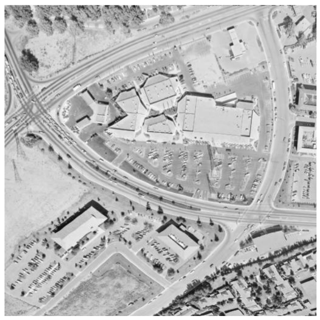
(b)



(a)



(b)



(a)

(b)

(a)

(b)

(a)

(b)

(a) (b)

Figure 4.4: Original Cover Image & Embedded Image

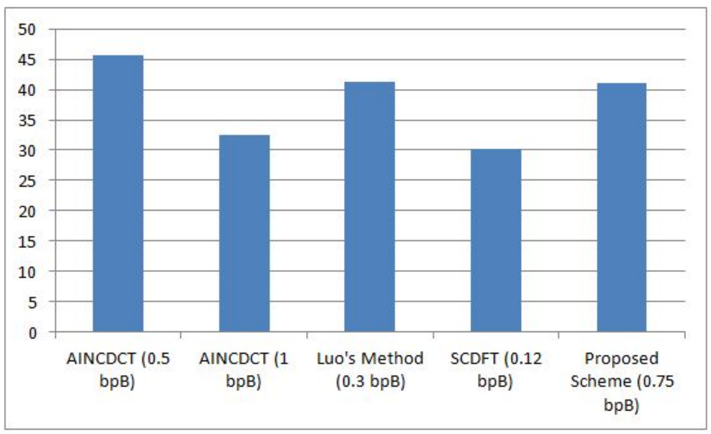


Figure 4.5: Comparison Graph of PSNR(dB) Values of Different Methods

(2008) [8], the average PSNR value is 30.10 for the payload of 0.12 bpB. The proposed AHRocZ gives a better average PSNR value of 41.05 for the payload of 0.75 bpB.

Chapter 5

Conclusion

The scheme proposed here has made an effort into the transform domain which gives an another layer of security to secret image, along with the Cover image to cover the source image which is used for authentication purpose. The fact is that having a considerably small MSE value and consequently a pretty good PSNR value ensures a high security of the hidden image.

The proposed scheme is a novel authentication technique which is used for image authentication in frequency domain through Z transformation with higher region of convergence ($r = 2$). Here though the payload is low (0.75 bpB) but the scheme provides a better security in terms of robustness and sensitivity of Z transform domain in higher magnitude. Experimental results show that the proposed method gives better performance than AINCDCT [4], Luo's Method [7], SCDFT [8] etc.

Here also for attacking the image or decoding the bitstream, the stego-analyser must have to know the value of the Region of Convergence (ROC) value which makes this steganographic algorithm a quite complex one. Thus, it is evident that if various radius or ROC value is used depending upon a typically good hash function can make this steganographic approach even more tough to be broken.

Chapter 6

Future Scope

In this proposed scheme, 3 bits from the authenticating embedding mask are embedded in an image subblock of size 2×2 providing a payload value of $0.75bpB$. In the future approaches, it will be taken under consideration if the payload value can be somehow increased, i.e. more number of bits from the authenticating bitstream must be embedded within a single image subblock. For that purpose, optimization techniques such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) etc. are under cosideration.

Another thing is that in the proposed approach, the bits are basically embedded in the LSB or $(LSB{+}1)th$ position. So, it also has been considered to use a proper and perfect hash function for selecting the bit positions for embdding so that the total scheme becomes more secure and more robust.

Bibliography

[1] Sanjit Kr. Mitra, "DIGITAL SIGNAL PROCESSING", 3^{rd} Edition.

[2] Nasir Ahmed, "DISCRETE TIME SIGNAL".

[3] Gonzalez & Woods, "DIGITAL IMAGE PROCESSING", 2^{nd} Edition

[4] Madhumita Sengupta, J. K. Mandal, "Authentication of Images through Non Con-voluted DCT (AINCDCT)", first International Conference on Communication and Industrial Application (ICCIA 2011), Dec, 26-28, 2011, IEEE, pp- 1-4, DOI: 10.1109/ICCIndA.2011.6146672.

[5] Allan G. Weber, The USC-SIPI Image Database: Version 5, Original release, October, 1997, Signal and Image Processing Institute, University of Southern California, Department of Electrical Engineering. http://sipi.usc.edu/database/

[6] Narendra K Pareek, Vinod Patidar, Krishan K Sud, "A Random Bit Generator Using Chaotic Maps", International Journal of Network Security, Vol.10, No.1, pp. 32-38, Jan, 2010.

[7] [Luo, 2011] Hao Luo, Fa-Xin Yu and et al. "Reversible data hiding based on block median preservation", Information Sciences, vol 181, pp - 308328, (2011), doi:10.1016/j.ins.2010.09.022.

[8] T. T. Tsui, X. P. Zhang, D. Androutsos, "Color Image Watermarking Using Multidimensional Fourier Transformation", IEEE Trans. on Info. Forensics and Security, vol.3, No.1, pp. 16-28, 2008.

[9] N. Ghoshal, J. K. Mandal, " A Bit Level Image Authentication/Secret Message Transmission Technique (BLIA/SMTT)", Association for the Advancement of Modelling and Simulation Technique in Enterprise

(AMSE), AMSE Journal of Signal Processing and Pattern Recognition, Vol. 51, No. 4, pp. 1-13, France, 2008.

[10] Nameer N. El Emam, "Hiding A LArge Amount of Data With High Security Using Steganography Algorithm", Journal of Computer Science, ISSN 1549-3636, vol. 3, no. 4, pp. 223-232, 2007.

[11] S. Dumitrescu, W. Xiaolin, Z. Wang, "Detection of LSB Steganography via Sample Pair Analysis", IEEE Trans. on Signal Processing, Vol. 51, no. 7, pp. 1995-2007, 2003.

[12] R. Chandramouli, N. Memon, "Analysis of LSB Based Image Steganography Techniques", Proc. of ICIP, Thissaloniki, pp. 1019-1022, Greece, 2001.