IMAGE SECURITY USING WATERMARKING AND ENCRYPTION TECHNIQUE

A Thesis

Submitted For the Degree of

Master of Technology

in Software Technology

by

Bikash Chandra Swain

ROLL NO: 2K13/SWT/07

Under the guidance of

Asst Prof. Anil Singh Parihar

Computer Science and Engg.

Delhi Technological University

DELHI – 110 042

JULY 2016

Student Declaration

I hereby undertake and declare that this submission is my original work and to the best of my knowledge and believe, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of any Institute or other University of higher learning, except where due acknowledgement has been made in the text. Project work and published paper associated to the chapters are well discussed and improved under the guide supervision.

Date :

Signature :

Bikash Chandra Swain

Roll No: 2K13/SWT/07

Registration No: DTU/13/M.Tech/484

Certificate

This is to certify that the thesis entitled, **Image Security Using Watermarking and Encryption Technique**, is a bona fide work done by **Mr. Bikash Chandra Swain** in partial fulfilment of requirements for the award of Master of Technology Degree in software technology at Delhi Technological University (New Delhi) is an authentic work carried out by him under my supervision and guidance. The matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma to the best of my knowledge.

Date :

Signature :

Prof. Anil Singh Parihar
Department Of Computer Science And Engineering
Delhi Technological University

Acknowledgements

First, my special thanks goes to my supervisor, **Asst Prof. Anil Singh Parihar**, who encouraged and guiding me to the successful completion of this work. I always remember all his guidance, support, kindness, and friendliness.

I am especially grateful to my parents for their continuous love and support gave me strength for pursuing my dream. My special thanks goes to all my friends and colleagues who have been a source of encouragement and inspiration throughout the duration of this thesis. My deepest thanks and love to my colleagues Deepak, Puja, Pankaj, Nitin for their support to complete my thesis. I am also thankful to the Samsung who has provided me opportunity to enrol in the M.Tech Programme and to gain knowledge through this programme.

Finally but certainly not least, I am grateful to my family especial my mother for her unlimited and unconditional love, patience and support in my whole life and to my wife Rinakshee for all support, encouragement, and patience during my study.

Bikash Chandra Swain

Roll No: 2K13/SWT/07

Registration No: DTU/13/M.Tech/484

Abstract

Watermarking algorithm typically used to identify ownership of image data but not provide any security to the image data where as encryption algorithm provides security to image data, but not the ownership. In my proposed algorithm, I have combined both watermarking and encryption techniques for security as well as authentication of an image data. For image encryption and decryption i have used Blowfish and Advance Encryption Standard (AES) which was based on block cipher. To get watermarked image for owner's authentication, we have been using the modified least significant bit (LSB) watermark algorithm. In first step of the algorithm, original image and watermark logo is embedded using watermark embedding process. In second step we encrypted watermarked image using encrption algorithm i.e blowfish or Advance Encryption Standard (AES) to produced encrypted image. In third step we decrypted the encrypted image using same blowfish or Advance Encryption Standard (AES) decrypted algorithm. In fourth step we get the original image from decrypted image by using watermark extraction process. This method is highly advisable when image sends from sender to reciever through a digital network system. The encryption technique will provided the security to image data when it transfers data in a network and watermarking provides ownership of image data at reciever end. We have also calculated PSNR and MSE value for original image and watermarked image. We also measures time taken for different phases like image embedding process, image encryption process, image decryption process and image extraction process.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The digital network system means the digital data (i.e image, audio, video, text file etc) can be copied and share it easily to large number of people without any cost. To protect such digital media files from unauthenticated users,there are many techniques which hides information. Digital image watermarking, image encryption and decryption, steganography and anonymity were developed to sloved this problem. In watermarking technology we hide a message, text, logo or signature into an image, audio file, video or any other work of media. Image data should be secure when it traveled from sender end to reciever end through a digital network. To protected the image data we have apply different encryption algorithm so that no one can change the image data without the permission of sender. For this purpose we have used both encryption and watermarking algorithm to secure the image data.

1.1 Motivation

Most of algorithms are designed either in form of image watermarking or image encryption to protect the image data. Watermarking algorithms are verify the identity of it's owner. Image encryption provides security to image data so that no users can change the image data when it transfers from sender to reciver. Now a days most of data (image, video etc) are communicated through internet which can easily access by

unauthorized users. The motivation behind my research is to implement both watermarking and encryption technique so that it will provide ownership as well as security for digital image files. In this thesis, i have used block cipher encryption algorithm i.e BLOWFISH and AES (Advanced Encryption Standard) for image encryption and decryption process. Block cipher encryption algorithm generally reduced the relationship through image elements by increasing entropy value to the encrypted image.

1.2 Project Goals

As watermarking methods are suitable to detect ownership of an image and Encryption provides only security. In this thesis, we have proposed a new method by combing both method to provides ownership as well as security for an image. This combined technique takes input as original image and produced these output images:

1. Watermarked image, using the least significant bit(LSB) watermarking embedding algorithm.

2. Encrypted image, using the blowfish or advanced encryption standarda(AES) image encryption algorithm.

3. Decrypted image, using the blowfish or advanced encryption standarda(AES) image decryption algorithm.

4. Original image and watermark logo, using the least significant bit(LSB) watermarking extraction algorithm.

1.3 Thesis Organization

The thesis is organized as follows. In Chapter 2, we discuss about current research in watermarking and encryption algorithm. We explain the concept of image watermarking and image encryption technique. In Chapter 3, The basic terms of image watermarking and encryption are explained. In addition, we discuss the structure of watermarking and

encryption system. We also briefly discussed about different types of image encryption and watermarking algorithm. In Chapter 4, we discussed briefly about the proposed work. In addition, we explain details about blowfish and advance encryption standard (AES) algorithm.We also expain how least significant bit watermark algorithm implemented. In Chapter 5, we analysis the result of different images which are used in our proposed algorithm. We present the summary and the future work in Chapter 6.

Chapter 2

Literature Review

Dalel Bouslimi, Michel Cozic and Christian Roux [1] proposed a paper "A Joint Encryption/Watermarking System for Verifying the Reliability of Medical Images" in which both encryption and watermarking algorithm was used to protect the medical images. In this paper RC4 and AES was used for image encryption. In first stage watermark embedding and encryption works together. Decryption and watermarking extraction works independently in final stages. G. Boato, N. Conci, V. Conotter, F.G.B. De Natale and C. Fontanari [2] proposed a paper "Multimedia asymmetric watermarking and encryption" in which a private key is used for watermark embedding and a public key is used for watermark extraction process. This both embedding and extraction process jointly work with a encryption technique to achieve two level security for digital data. Sangita Zope-Chaudhari, Parvatham Venkatachalam, and Krishna Mohan Buddhiraju [3] proposed a paper "Secure Dissemination and Protection of Multispectral Images Using Crypto-Watermarking" in which crypto-watermarking scheme was used for giving more security to multispectral images. In this paper it shows result for multispectral images in the field of large data hiding capacity, filtering and noise. Stefan Katzenbeisser, Aweke Lemma, Mehmet Utku Celik, Michiel van der Veen, and Martijn Maas [4] proposed a paper "A BuyerSeller Watermarking Protocol Based on Secure Embedding" in which the rules and protocol that are present now make the use of a homomorphic public key cipher to encrypt the watermark as well as to embed under encryption. But what is

different in a forensic watermarking architecture is that there are protocols that hide the watermark secret form the buyer. Thus there can not be any false allegations by the seller about his watermarking secrets. B. Subramanyan, Vivek M. Chhabria, T. G. Sankar Babu [5] proposed a paper "Image Encryption Based on AES Key Expansion" in which image was encryption using AES (advanced encryption standard) encryption algorithm. This paper also proposed how the AES key expansion happens by using XOR operator on image pixel and 128 bit key. The keys are produced independently for encryption and decryption process. This algorithm was working good when image was attacked by brute force method. Abdullah Bamatraf, Rosziati Ibrahim and Mohd. Najib B. Mohd Salleh [6] presented a paper "Digital watermarking algorithm using LSB" in which watermarked image was created by embedding two bits with third and fourth bit of original image. This paper also shows very good quality of the watermarked image. A. V. Subramanyam, Sabu Emmanuel and Mohan S. Kankanhalli [7] presented a paper "Robust Watermarking of Compressed and Encrypted JPEG2000 Images" in which watermark algorithm implemented for a compressed encrypted digital data. This paper also explains how watermark embedding apply for a compressed-encrypted domain and extraction for decrypted domain. A. Mousa [8] proposed a paper "Data encryption performance based on Blowfish" in which speed and security was checked for any type of data i.e image, text, video file. It also checked the performance by changing the key length and file size. In this paper it shows how performance affected by changing the file size rather than key length.

2.1 Digital Images

A digital image is consists a finite set of individual pixel values which represents in a matrix form. Each pixel has represented by it's own RGB value. The set of pixels in an image is known as bitmap. For example if we take an image of size 300 X 300 pixels, then it will produced a data of 90000 pixel size . In watermarking and encryption algorithm we change the pixel value to secure the image data from other users.

2.2 Image Watermarking

A watermarking system might be described as a structure that contains two parts: an embedding part and a extraction part. In embedding process we take two images as input i.e one is the original image and other one is watermark logo. By using watermark embedding process we can generated watermarked image by giving stream data of original image and watermark logo. The watermarked stream image data can be further extracted by using the watermark extraction process, which determines whether the watermark exists or not in an image data. Digital watermarking is used to offer ownership security, including identification of the copyright owner and protection.

- **Embedding:** The process in which inserting both original image and watermark logo to prduced watermarked image.
- **Extraction:** The process in which original image and watermark logo extracted using watermarked image.
- **Watermarking:** is the process in which both embedding and extraction process works.

2.3 Image Encryption

Image Encryption is a process in which we change the pixel value of an image so that only authorized users can read the image data. In image encryption, pixel value was changed by it's place so that we can protect the image information for future use. Mainly image encryption techniques are used key mapping or hiding of image fusion at pixel lable.means changing the pixel values of image so that no user can read the image data. There are two types of encryption used to encrypt or decrypt the data.

- **Symmetric key encryption:** is a process in which we used same key for both encryption and decryption.

- **Public key encryption:** is a process in which the encryption key is known to all users where as decryption is only known to receiving party.

In this thesis we mainly used Blowfish and Advanced Encryption Standard(AES) algorithm for image encryption and decryption process.

Chapter 3

Watermarking and Encryption Technology

3.1 Classification of Watermark Algorithms

In this section we have discussed about different types of watermarking algorithms and how it works on digital image data.

3.1.1 Taxonomy of Watermarking

First on the basis of data, watermarking algorithms are classified into four categories [10,11].

- Text watermarking
- Image watermarking
- Video watermarking
- Audio watermarking

Further on the basis of human eye recognition, watermarking algorithms are further divided into two categories.

- Visible watermarking.
- Invisible watermarking.

In visible watermarking, the watermarked image can easily detected by human eye without extraction of watermarked image. Generally visible watermarking is used in papers and video with watermark logo to protect the ownership of digital image data. In this method original image is embedded with watermark logo to produced watermarked image. In this thesis we have used visible watermarking algorithm.

In invisible watermarking, the watermarked image cann't be easily detected by human eye and it can extract the original image who have the rights for that. In this case it changes the pixel bit value so that original image can show the changes. Mainly we use this method to protect the copyright of image for owner.

Third, on the basis of information detection, watermarking algorithms are classified into following categories. They are as follows.

- Blind or public watermarking: In public watermarking, there is no need for original signal during the detection processing to detect the watermark. In this watermarking only secret key is used. For example, in image blind watermarking we do not need the original image.
- Non-blind or private watermarking: In non-blind or private watermark, To detect the watermarked data original signal is required.
- Semi-blind watermarking: In semi-blind watermarking, sometimes we may need some extra information for detecting the watermark. Some watermarking require access to the original signal just after adding the watermarking,is known as published watermarked signal. This form of watermarking is called semi-blind watermarking. We further divided the watermarkin algorithm on the basis of processing domain are as follows:

- Spatial domain: It is a watermarking technique in which pixel values are changed. As image data consist of large number of pixels, so when you change the pixel value it will automatically change the image. In this method there was very small change in watermarked image. The best example of spatial domain is least significant bit watermarking algorithm in which only bits value was changed in pixel data. In spatial domain we also merge watermark logo and original pixels bit to get watermarked image.

- Transform domain: It is a watermarking technique in which transform coecients are changed in watermark embedding process. Transform domain is also known as frequency domain because the value of frequency are changed. Discrete cosine transform (DCT) and Discrete Wavelet Transform (DWT) are the watermark techniques which are used frequency domain.

Additionally, classication can be based on the robustness feature. Dierent techniques of this category are as follows.

- Robust watermark: One of the properties of the digital watermarking is robustness. We call a watermark algorithm robust if it can survive after common signal processing operations such as ltering and lossy compression.

- Fragile watermark: A fragile watermark should be able to be detected after any change in signal and also possible to identify the signal before modication. This kind of watermark is used more for the verication or authenticity of original content.

- Semi-fragile watermark: Semi-fragile watermark is sensitive to some degree of the change to a watermarked image.

Furthermore, from application point of view, watermark techniques can be grouped as source based or destination based. In source based, all copies of a particular data have a unique watermark, which identies the owner of that data, while in the destination based, each distributed copy is embedded using a unique watermark data, which identies

a particular destination. Figure 2.2 depicts dierent classication for digital watermarking algorithms. In the rest of the section, we discuss processing-domain, spatial and transform in more detail.

3.1.2 Spatial Domain Watermarking

Spatial domain watermark algorithms insert watermark data directly into pixels of an image. For example, some algorithm insert pseudo-random noise to image pixels.

Other techniques modify the Least Significant Bit (LSB) of the image pixels. The

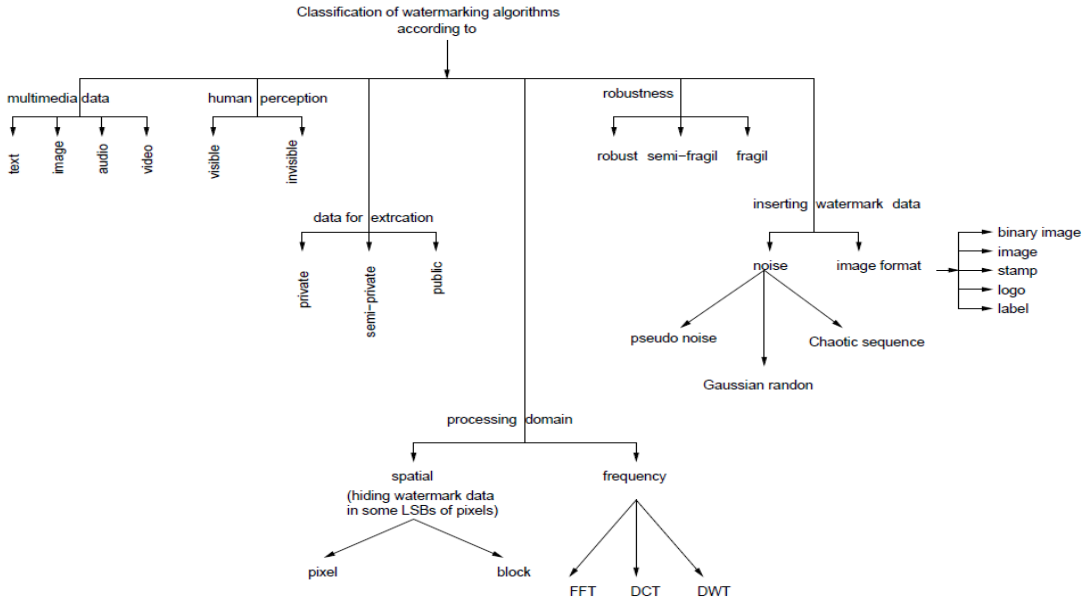


Figure 3.1: Classification of watermarking techniques.

invisibility of the watermark data is obtained on the assumption that the LSB bits are visually insignificant. There are two ways of doing an LSB modification. There are some methods to change the LSB bits. The LSB of each pixel can be replaced with the secret message or image pixels may be chosen randomly according to a secret key. Here is an example of modifying the LSBs, suppose we have three R, G, and B component in an image. Their value for a chosen pixel is green (R,G,B) = (0, 255, 0). If a watermark algorithm wants to hide the bit value 1 in R component then the new pixel value has components (R,G,B) = (1, 255, 0). As this modification is so small, the new image is to

the human eye indistinguishable from the original one.

Although this spatial domain techniques can be easily used on almost every image, they have the following drawbacks. These techniques are highly sensitive to signal processing operations and can be easily damaged. For example, lossy compression could completely defeat the watermark. In other words, words, watermarking in the spatial domain is easy to destroy using some attacks such as low-pass filtering. As a result, transform domain watermarking algorithms are used.

3.1.3 Transform Domain Watermarking

Transform domain watermarking embed watermark data into the transformed image. Transform domain algorithms have many advantages over spatial domain algorithms [9].

For example, Figure 3.2 depicts common processing operations on a watermarked

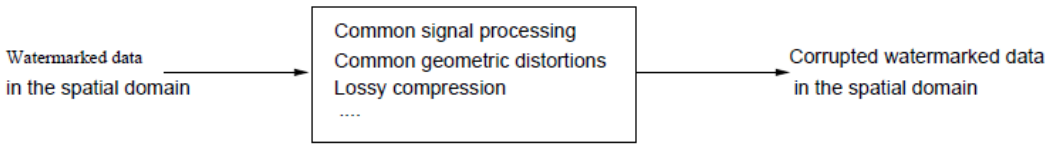


Figure 3.2: Common processing operations on a watermarked data in the spatial domain.

data in the spatial domain. Common signal processing includes operations such as up-sampling, downsampling, quantization, and requantization. Rotation, translation, and scaling are common geometric operations. Lossy operation is an operation to remove some unimportant parts of the data. Most of the processing for this category takes place in the transform domain and eliminates high-frequency values. As can be seen in Figure 3.2, those operations corrupts the watermark data, which has been embedded into the original data.

In addition, spatial domain techniques are not very usefull for lossy image compression. Compare to transform domain, the spatial doman are very low bit capacity. Even a noise can remove the watermark from the image. As another example, a watermark data placed in the high-frequency values can be easily eliminated with little degradation of

the image by any low-pass filtering.

On the other hand, Spatial domain techniques are less robust in cmpare to transform domain watermarking techniques. This is because the transform domain does not use the original image for embedding the watermark data. In addition, a transform domain algorithm spreads the watermark data over all part of the image. Additionally, in frequency domain-based techniques more bits are chnaged when it converted original image nto watermarked image. So it will be more robust towards image attack. Furthermore, most of the images are avaliable in the transform domain. Some transforms such as DCT and DWT are used for watermarking in the frequency domain. Most DCT-based techniques work with 88 blocks.

3.2 Single and Multiple Bit Watermarking Algorithms

Different watermarking algorithms can also be divided into two groups namely, single bit and multiple bits algorithms [12]. In the single bit watermark algorithms, the watermark pattern consists of the integers randomly selected from 1, 0, 1. The watermark pattern is created using a secret key. This secret key is used as an input key to the random number generator. When adding this type of watermark to the image, the care should be taken to uniformly distribute the energy in the pattern. Equation (3.1) shows how the 1-bit watermark is added to the image.

$$I^{'}(x,y) = I(x,y) + kw(x,y). \tag{3.1}$$

where k is the watermark gain factor and a number between 0 and 1 ($0 < k < 1$), w(x, y) is the watermark pattern, I(x, y) is the original image and $I^{'}$(x, y) is the watermarked image. Single bit watermarking can also be applied in the transform domain to the coefficients of the transformed image. Equation (3.2) indicates this type of watermark embedding.

$$F^{'}(x,y) = F(x,y)(1 + kw(x,y)). \tag{3.2}$$

where $F^{'}$(x, y) is the watermarked coefficient, and F(x, y) is the original transform coefficients of the image. During the detection in these types of watermarking, either the watermark is detected (logic-1) or it is not (logic-0).

On the other hand, in multiple bit watermarking methods string of bits b_1, b_2, . . . b_M are embedded to an image. They first divide the image into M sub-images I_1, I_2, . . . I_M of size mn. Then a random watermark pattern having the same size as the sub-image, is added to each sub-image I_i. This is done by modulating the pattern according to the corresponding bit value b_i. The modulation can be done in several different ways. The simplest way is adding the random pattern of size mn to the sub-image if the watermark bit is 1, and leaving the sub-image unchanged if the watermark bit is -1. Another method for modulating watermark bits is generating a pseudorandom pattern of 1, 1 for each bit of the watermark to be embedded. This pattern generation is similar to the sequence generation method used in Code Division Multiple Access (CDMA) spread spectrum. In this technique, if the watermark is b_1, b_2, . . . , b_M, then M stochastically independent pseudorandom patterns v1v2vM having the same size as the image are created. Each pattern, v_i, is modulated by its corresponding bit, b_i. The sum of all random patterns v_i constructs the watermark. To apply this technique to 2D images, we replace the image with the m n blocks and the watermark vectors with random 1 and 1.

3.3 Structure of Watermarking System

The process of digital watermarking is divided into two basic steps. They are watermark embedding and watermark extraction process. The watermark embedding and extraction processes are described in the following subsections :

- Embedding process
- Extraction Process

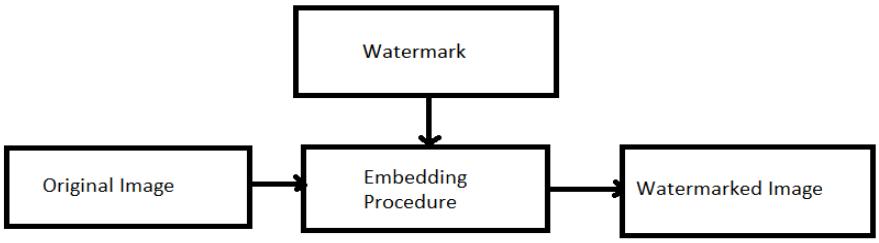


Figure 3.3: Watermark Embedding Diagram

3.3.1 Embedding Process

Let us defined an original image by I, a watermark logo by W and the watermarked imageI_w. The embedding function E_{mb} takes the original image I and watermark logo W as an input to generate a new watermarked image, denoted as I_w.First original image converted into either spatial domain or frequency domain. The domain selection mainly based on what type of watermarking algorithm was used for image data. In frequency domain we apply inverse transform to get watermarked image. For spatial domain algorithm, we defined the Mathematically expressed for embedding process as follows:

$$E_{mb}(I, W) = I_w. \tag{3.3}$$

3.3.2 Extraction Process

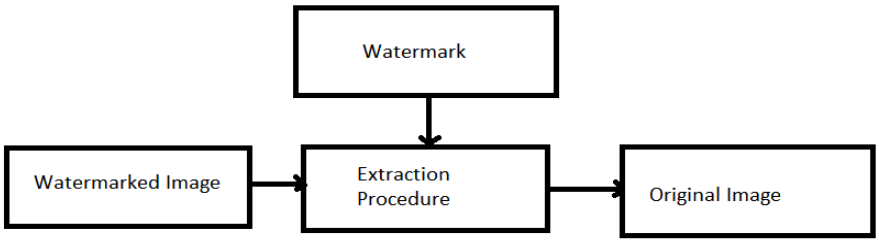


Figure 3.4: Watermark Extraction Diagram

A extraction function D_{tc} takes a watermarked image I_r whose ownership is to be

determined. The extraction function either recovers a watermark W_e from the watermarked image or checks the presence of the watermark W in a given watermarked image I_r. Mathematically, the extraction procedure can be expressed as follows:

$$D_{tc}(I_r, W_e) = I \tag{3.4}$$

3.4 Image Encryption and Decryption Algorithms

Image encryption is a process in which we can change the image data into unreadable format so that no user can modified or read actual image data. Image decryption is a reverse process in which we decrypt the encrypted image using the encryption key. Image encryption and decryption are mainly used to secure the image data from unauthorised users.

The important image encryption algorithms are as follows:

- Data Encryption Standard (DES) Algorithm
- Blowfish Algorithm
- Advanced Encryption Standard (AES) Algorithm
- Twofish Algorithm

3.4.1 Data Encryption Standard (DES) Algorithm

The DES, also known as Data Encryption Standard is an outdated method of data encryption based on symmetric key. DES uses the same key to encrypt and decrypt a message. Thus, the same private key must be known and used by the sender and the receiver both. At one time, it was the most widely used symmetric key algorithm for encryption of electronic data. But now it has been replaced by the AES (Advanced Encryption Standard) algorithm which is more secure and robust. In the early 1970s, it was originally designed by researchers at IBM. In 1977, it was adopted as an official Federal

Information Processing Standard (FIPS) by the U.S. government, for the encryption of commercial and sensitive yet unclassified government computer data. It was the first instance of any such encryption algorithm being approved for public disclosure by the US government, thus ensuring that DES was quick to be adopted by industries such as financial services. This so happened because in these cases there is a high need for strong encryption. DES is used in a variety of network devices requiring encryption like modems, set-top boxes and routers, SIM cards, smart cards and embedded systems. But now, DES is considered to be insecure for many applications because the 56-bit key size that it uses is considered too small. in January 1999, the Electronic Frontier Foundation collaborated with distributed.net to publicly break a DES key in about 22 hours and 15 minutes . Some analytical results were also there demonstrating various theoretical weaknesses in the cipher. Now DES has reached the end of its useful life. Nonetheless, its arrival has enough well worked to promote two things:

- The study of cryptography
- The development of new encryption algorithms.

Until DES happened, the cryptography was known to be dark art confined to the realms of government intelligence organizations and military. The open nature of DES made it possible for anyone interested in security , be it academics, mathematicians or anyone else; could study how the algorithm worked and try to crack it.

3.4.2 Blowfish Algorithm

Blowfish is a symmetric block ciper algorithm which was designed in 1993 by Bruce Scheier. It is very fast to existing DES encryption algorithm and encrypts 64-bits block data at a time. Blowfish algorithm can run in low memory. Blowfish algorithm encrypts data on large 32-bit microprocessors at a rate of 26 clock cycles per byte. It uses XOR, addition and lookup table with 32-bit operands. The key used in Blowfish algorithm is veritable in length and can be in the range of 32 448 bits. The default key length is 128

bits. This algorithm is very useful where key length is fixed. Blowfish uses the same key for both encryption and decryption process.

3.4.3 Advanced Encryption Standard (AES) Algorithm

The Advanced Encryption Standard (AES) which is also known as Rijndael, is used for encryption of electronic data established by U.S National Institute of Standards and Technology in 2001. As the key size of DES algorithm was small, to replace DES algorithm Advanced Encryption Standard (AES) was designed which is six times faster than triple DES. AES also uses symmetric key i.e same key is used for both encryption and decryption process. It divided the data into a block size of 128 bits. AES algorithm computes on bytes rather than bits. For example if a data of size 128 bits, it will treat as 16 bytes. These 16 bytes are represented in a matrix form i.e four rows and columns. AES rounds mainly depends on it's key length i.e for 128 bits key it required 10 rounds, for 192 bits key it required 12 rounds and for 256 bits key it required 14 rounds. It required low ram and speed of this algorithm is very fast. AES required 18 clock cycle to encrypt a byte. In present day, Advanced Encryption Standard (AES) algorithm is widely used for data encryption and decryption for both hardware and software. AES will give more secure to data if it follows good key management and correctly implemented.

3.4.4 Twofish Algorithm

In cryptography, Twofish is simillar to blowfish algorithm in which data are send in a block cipher format. Twofish also used symmetric key technique for encryption and decryption. It transfered original data into a block size of 128 bits and used a key having size upto 256 bits. Its distinctive features are as following:

- The use of pre-computed key-dependent S-boxes
- A relatively complex key schedule.

The two halves of an n-bit key are used as the actual encryption key and used to modify the encryption algorithm, respectively. This algorithm was designed by Bruce Schneier,

John Kelsey, Doug Whiting, David Wagner, Chris Hall, and Niels Ferguson. They belonged to the "extended Twofish team". They performed further cryptanalysis of Twofish. Stefan Lucks, Tadayoshi Kohno, and Mike Stay were the other AES entrants who should be given due credit. On most software platforms Rijndael (the chosen algorithm for Advanced Encryption Standard) is slightly faster than Twofish for 128-bit keys, but it is slower than Twofish in case of 256-bit keys. This cipher has not been patented .This means the Twofish algorithm is free for anyone to use without any restrictions because its reference implementation has been placed in the public domain.

3.5 Structure of Image Encryption and Decryption System

The process of digital image security is divided into two basic steps. They are image encryption and image decryption process. The image encryption and image decryption process are described in the following subsections :

- Encryption Process
- Decryption Process

3.5.1 Encryption Process

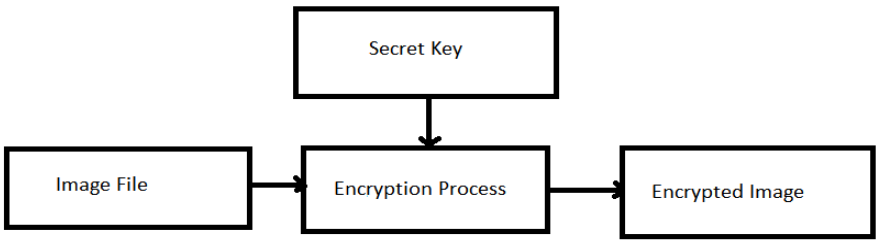


Figure 3.5: Image Encryption Diagram

Encryption is a process in which we encrypt the data using a key so that no user can read or modified the data when it travel in a network system. In image encryption,

image cann't be read until it was decrypted using same encryption key. Encryption process hides the data using the key. To prevent from unauthorized user, we encrypted the data so that no user can read or modified the user's personal data. In encryption process a plain data converts in cipher data.

3.5.2 Decryption Process

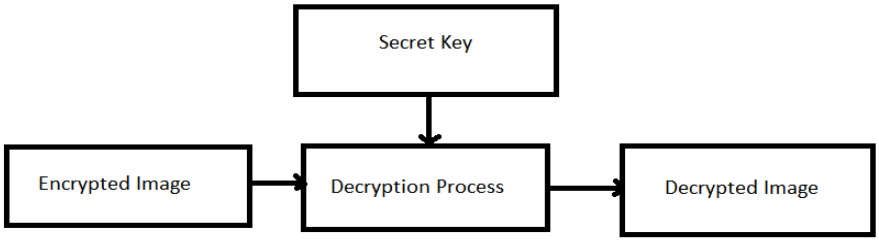


Figure 3.6: Image decryption Diagram

Image decryption is a process in which we decrypt the encrypted image. In general the encryption applied on sender's side and decryption applied at reciever's end so that no other user can accessed the image data. For good encryption technique it was very difficult to decrypt the data without knowing the key.When key length are more longer, it will more difficult to crack the data using permutation method.

Chapter 4

Proposed Work

4.1 Introduction

The ubiquitous nature of digital network systems means that digital data can be copied and share it in a social network without taking the permision of actual data owner. People can modified the data also. To protect the data, we generally used encryption technique so that no unauthorized access can happen to the data. There are a number of technologies that will provide protection from illegal copying. Digital watermarking algorithms were developed to solve this problem. Watermarking algorithms embed digital signatures or digital data to prove the owners identity and stop copyright infringement. Watermarking helps in recognizing or identifying any unauthorized access. A watermark will be visible on the data if someone uses without authorization or the owner can recognize the unauthorized use by using watermarking.

4.2 Proposed Method

Watermarking algorithm provides ownership of a user but not giving any security to image data. Also encryption technique provides security but not giving any authority to users. So by combined these two method to give user both ownership as well as security. In our proposed algorithm we combined both watermarking and encryption technique to

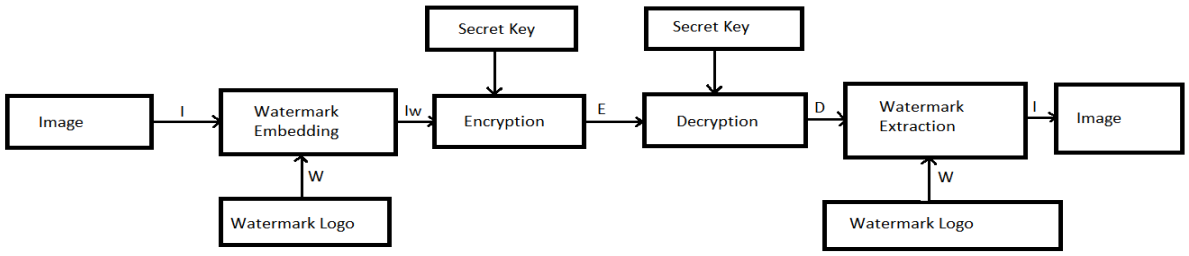


Figure 4.1: System Block Diagram

give more security to digital image data. In our proposed method, we divided the whole process into four parts :

- **Watermark Embedding Process:** In this process we first convert the original image (I) and watermark logo (W) into pixel matrix form. Then XOR operation is carried out between the RGB value in each index of the original image (I) matrix with the RGB value in corresponding index of watermark logo image (W) matrix to produced a watermarked image (I_w).

- **Image Encryption Process:** In this process the watermarked image (I_w) will encrypted using either blowfish encryption algorithm or advanced encryption standard (AES) encryption algorithm to produced a enccrypted watermarked image (E).

- **Image Decryption Process:** In this process we use same encrypted key for decrypt the encrypted watermark image (E) to produced decrypted watermark image (D). We use same blowfish and Advanced Encryption Standard (AES) algorithm to decrypt the image. We use same key for encryption and decryption of image data because we are using symmetric key algorithms.

- **Watermark Extraction Process:** In this process we again convert the decrypted watermark image watermark logo into pixel matrix form. Then XOR operation is carried out between the RGB value of each index of the decrypted watermark image (D) with the RGB value in corresponding index of watermark logo image (W) matrix to produced original image (I).

For the above proposed method, we are using least significant bit algorithm for water-marking and blowfish, advanced encryption standard (AES) for image encryption and decryption. These algorithms are explained as follows:

4.3 Least Significant Bit Watermark Algorithm

The most common method of watermark embedding is to embed the watermark into the leastsignificant-bits of the cover object. Generally colour image pixels are represented in 24 bits with RGB format. Each eight bits represented for intensity of each colour value. Suppose we take red part of a pixel, it has 256 different colour values ranging from (00000000 through 11111111). Simillarly eight bits for green value and eight bits for blue value to store the complete pixel information. The diagram Fig 4.2 shows how

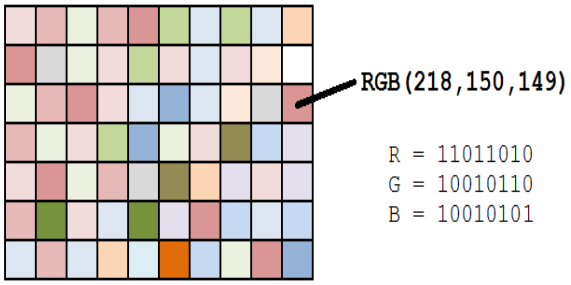


Figure 4.2: Image RGB Value Diagram

(a) Original Image (b) Watermark Logo

Figure 4.3: Block Diagram for Image Pixel

each pixel of an image can be described by three 8-bit binary numbers. The diagram Fig

4.3 show pixel value of original image and watermark logo. In my proposed LSB(Least Significant Bit) algorithm i take RGB value of original image and watermark logo and XOR it to get the visible watermarked image useing watermark embedding process. Finally XOR the RGB value of decrypted watermarked image and watermark logo to get the original image. In least significant bit watermark algorithm we changed the original image pixels values.

4.3.1 LSB Watermarking Embedding Code

```
public static void embeddingImage(BufferedImage originalImage,
        BufferedImage logoImage, int i) {
    // get image width and height
    int width = originalImage.getWidth();
    int height = originalImage.getHeight();
    File waterMarked = null;

    // For loop for pixel matrix of image
    for (int y = 0; y < height; y++) {
       for (int x = 0; x < width; x++) {
          int p1 = originalImage.getRGB(x, y);
          int p2 = logoImage.getRGB(x, y);

          // RGB value of original image
          int a1 = (p1 >> 24) & 0xff;
          int r1 = (p1 >> 16) & 0xff;
          int g1 = (p1 >> 8) & 0xff;
          int b1 = p1 & 0xff;
          // RGB value of logoImage image
          int a2 = (p2 >> 24) & 0xff;
          int r2 = (p2 >> 16) & 0xff;
          int g2 = (p2 >> 8) & 0xff;
```

```
            int b2 = p2 & 0xff;

            // XOR on RGB value of originalimage and logoimage
            if (r2 != 255 && g2 != 255 && b2 != 255)
               p1 = (a1 << 24) | (r1 << 16) | (g1 << 8) | (b1 ^ 31);

            originalImage.setRGB(x, y, p1);
         }
      }

      // write image
      try {
         waterMarked = new File("D:/MTECH_PROGRAM/BLOWFISH/waterMarked" + i
               + ".png");
         ImageIO.write(originalImage, "png", waterMarked);
      } catch (IOException e) {
         System.out.println(e);
      }
   }
```

4.3.2 LSB Watermarking Extraction Code

```
public static void extraction(BufferedImage waterMarkImage,BufferedImage
    logoImage,int imageCount){
      //get image width and height
       int width = waterMarkImage.getWidth();
       int height = waterMarkImage.getHeight();
       File dewaterMarked = null;

       //For loop for pixel matrix of image
```

```
for(int y = 0; y < height; y++){
  for(int x = 0; x < width; x++){
    int p1 = waterMarkImage.getRGB(x,y);
    int p2 = logoImage.getRGB(x,y);

    //RGB value of original image
    int a1 = (p1>>24)&0xff;
    int r1 = (p1>>16)&0xff;
    int g1 = (p1>>8)&0xff;
    int b1 = p1&0xff;

    //RGB value of watermark logo image
    int a2 = (p2>>24)&0xff;
    int r2 = (p2>>16)&0xff;
    int g2 = (p2>>8)&0xff;
    int b2 = p2&0xff;

    if (r2 != 255 && g2 != 255 && b2 != 255)
     p1 = (a1<<24) | (r1<<16) | (g1<<8) | (b1^31 );


    waterMarkImage.setRGB(x, y, p1);
  }
}

//write image
try{
  dewaterMarked = new
      File("D:/MTECH_PROGRAM/BLOWFISH/dewaterMarked"+imageCount+".png");
  ImageIO.write(waterMarkImage, "png", dewaterMarked);
}catch(IOException e){
```

```
        System.out.println(e);
      }
  }
```

4.4 Blowfish Algorithm

Blowfish algorithm is a symmetric encryption algorithm which is used same key for both encryption and decryption. Blowfish is also used block cipher which divides a data inti fixed size of blocks when it encrypt and decrypt a data. The size of block which is used in blowfish algorithm are 64 bits. Blowfish divides into two parts:

- Key-expansion
- Data Encryption

4.4.1 Key-expansion

Blowfish algorithm convert it's key into several subkey arrays having total size of 4168 bytes. Blowfish generally used very large number of subkeys for it's data encryption. Before the encryption and decryption process, we first generated the key for it. For blowfish algorithm we used eighteen P-array which consits of 32 bits subkeys for each.

$$P_1, P_2, ...P_{18} \tag{4.1}$$

For blowfish algorithm we used four 32-bit R-boxes which consits of 256 for each entry:

$$S_10, S_11, ...S_1255 \tag{4.2}$$

$$S_20, S_21, ...S_2255 \tag{4.3}$$

$$S_30, S_31, ...S_3255 \tag{4.4}$$

$$S_40, S_41, ...S_4255 \tag{4.5}$$

Generating the Subkeys: In Blowfish algorithm, the subkeys are calculated using following steps:

1. We have to start by initializing the P-array and the four S boxes, with a xed string. This should be done in order. The xed string contains hexadecimal digits of P_i (less the initial 3): P_1 = 0x243f6a88, P_2 = 0x85a308d3, P_3 = 0x13198a2e, P_4 = 0x03707344, etc.

2. Now, XOR operation is carried out between P_1 and rst 32 bits of the key.Similar operation is repeated for P_2 and next 32 bits of the key and so forth(estimates up to P_{14}). This cycle is to be repeated until XOR operation ha sbeen carried out for the entire P- array. (one think that is to be remebered is that every short key has an equivalent longer key,e.g. if B is a 64 bit key, it's equivalent keys wuld be BB,BBB, etc.)

3. All - zero string must be encrypted using the methods mentioned in points 1 and 2, following the Blowfish algorithm.

4. P_1 and P_2 are replaced with the output of step 3.

5. The output of step 3 is encrypted with the modified sukeys using Blowfish algorithm.

6. P_3 and P_4 are replaced with the output of step 5.

7. This process is continued ,replacing all P array entries alongwith all four S-boxes with the outputs of the Blowfish algorithm that is continuously changing.

To generate all possible subkeys, blowfish algorithm performed 521 iterations. In blowfish algorithm all the subkeys should be stored in a memory so that it can reduced the iterative key derivation process.

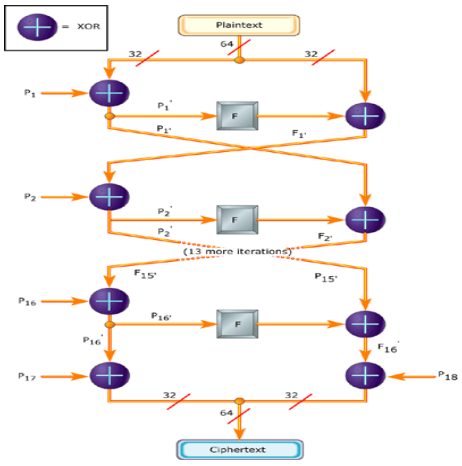


Figure 4.4: Blowfish algorithm

4.4.2 Data Encryption

For blowfish algorithm, data encrypt and decrypt using a fixed block size of 64 bits. Blowfish required 5KB memory to execute both encryption and decryption process. Blowfish encrypt or decrypt a 32 bits processor using a 64 bits block data in 12 clock cycles are required. For example, if you have a data of size 256 bits then it will take a time (4x12) clock cycles as 256 bits data divided into four 64 bits data block. First 64-bit block data are divided into two parts (i.e XL and XR) having each part 32-bits. The first element of P-array P_1 and XL has XORed to produced a value say ${P_1}^1$ and run this ${P_1}^1$ value using transformation function F. Then XR and value of transformation function F will again XORed to produced a new value called F^1. Then F^1 will assigned to XL and ${P_1}^1$ will assigned to XR value and same above process will repeated upto 15 more times with P-array value to produced corresponding P^1 and F^1 values. In last entry of P-array (i.e for 17 and 18 index), the resulting value P^1 and F^1 are recombined to produced a 64 bit cipher data. In figure 4.4 the above blowfish algorithm flows are shown.

Divide x into two 32-bit halves: XL, XR

For i = 1to 16:

XL = XL XOR P_i

XR = F(XL) XOR xR

Swap XL and XR

Swap XL and XR (Undo the last swap.)

XR = XR XOR P_{17}

XL = XL XOR P_{18}

Recombine XL and XR

Figure 4.5 representation of blowfish F function. In this function a 32 bit input is de-

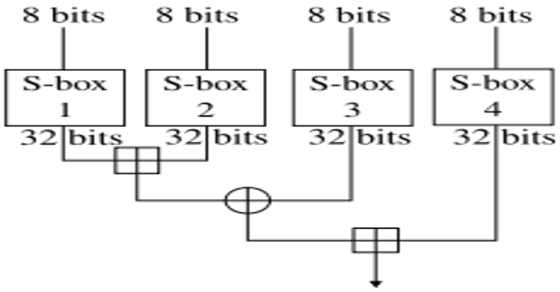


Figure 4.5: Graphic representation of F

vided into four bytes and used in four S-boxes. The lookup results are then added and XORed to produced the final output in a F function.

As blowfish is based on symmetric key, the same key is used for both encryption as well as decryption but main difference is for encryption it used plain data where as for decryption it used cipher data. User key precomputed the values of P-array and S-array. If the user key does not change then there is no need to recomput the P-array and S-array values.

4.5 AES (Advance Encryption Standard) Algorithm

AES algorithm follows the rules of being a symmetric key cipher as a single key is used by both the sender and the receiver for the purpose of encryption and decryption. The length varies and can be of 128,192 or 256 bits.Here 128 is set as the data block length. The AES also goes by the rules of iterative algorithms and hence is one.Here, total number of rounds can be 10,12 or 14;the key length being 128,192 or 1256 repectively where each round signifies iteration. Each 128 bit data block can be divided into smaller

chunks of 16 bytes which are further mapped to a 4x4 array(State). It is this array on which all the internal operations of AES algorithms are carried out.

Table 4.1: AES Parameters Diagram

Algorithm	Key length(Nk words)	Block size(Nb words)	Number of rounds(Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

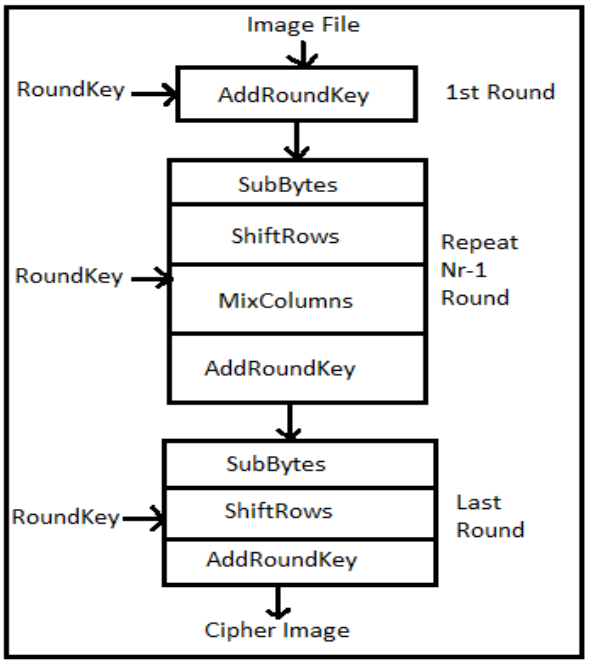


Figure 4.6: AES Encryption Algorithm Block Diagram

- **Subbyte Transformation:** It is a byte subsdtitution method, non linear in nature. It used a substation table(s-box) ,which has been created by multiplicative inverse and Ane transformation.

- **Shift rows transformation:** It is a byte transposition method,quite simple in nature. Here the bytes in the state's last three rows are shifted in a cyclic manner.The group of bytes to be shifted can be of length from one to three bytes.

- **Mix columns transformation:** It is somewhat like a matrix multiplication carried out on the columns of the state.Here xed is multiplied to every column vector.One impportnat thing about this method is that the bytes are not treated as numbers,but as polynomials.

- **Add round key transformation:** It refers to a simple XOR operation,which is actually its own inverse, carried out between the working state and the round key.

- **Expansion key:** AES algorithm follows the rules of being a symmetric key cipher as a single key is used by both the sender and the receiver for the purpose of encryption and decryption. It is very robust and secure as it is almost impossible to determine the key.Even if someone gets to know the plain text and the cipher text e can not determine the key.The AES algorithm makes use of the following key sizes: AES-128,AES-196 and AES-256. Each of them uses key sizes 256 bit (32 bytes, 8 words), 196 bit (24 bytes, 6 words) and 128 bit (16 bytes, 4 words) respectively. It is much more secure and robust than DES as it safeguards each and every key value.Therefore, every encryption is equally secure. The expansion of the keys take place using a key expansion routine. This routine can be performed wholly or in buts as and when they are needed.

Chapter 5

Result and Analysis

In this chapter we discuss and presend our experimental results for different images. Generally speaking, the combined watermarking and encryption method gives a better security and authentication to the image data.

5.1 Quality Measurements

Two methods are generally used to measure the error quantity between original image and watermarked image knowns as Mean Square Error (MSR) and Peak Signal to Noise Ratio (PSNR). Their equations are as follows.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{5.1}$$

In Mean Square Error (MSR) equation 5.1, i takes the value from 0 to image pixel row size m and j takes the value from 0 to image pixel column size n. To get the Mean Square Error (MSR) value of original image and wtermarked image, we first calculate the sum over of difference between original image and watermarked image for all image pixels and divided it by product of image pixel size.

$$PSNR = 10. \log_{10} (\frac{R^2}{MSE}) \tag{5.2}$$

In peak signal-to-noise ratio (PSNR) equation 5.2, R is defined as maximum fluctuation in a inputed image data. If the input image is a double-precision floating-point data type then R value is 1 and if it has 8-bit grayscale image then R value is 255. If we have a image of B bits, we can calculate the value of R is 2^B-1. The peak signal-to-noise ratio (PSNR) is defined as the ratio between maximum possible signal value and distorting noise value which affects the quality of image. Higher the PSNR values indicates that higher the reconstruction quality. Increasing PSNR represents increasing delity of compression. If image peak signal-to-noise ratio (PSNR) value is greater or equal to 40 dB then images cann't be indistinguishable by human eye i.e the original image and watermarked image was same.

5.2 Images Database

For our experimental result, we have take six different colored images which are provided at Fabien Petitcolas database [] for the purpose of watermarking and encryption process. We take 300x300 image size for our experiment process. We experiment both watermarking as well as encryption technique on all images and find out the results.

5.3 Experimental Setup

In this thesis, we focused on visible watermark data as well as image encryption. We used the equation 5.1 to measure the Mean Square Error (MSR) and equation 5.2 to measure peak signal-to-noise ratio (PSNR) value of original image and watermarked image. In our implementations, we have used the Matlab tools and Java JDK 7.0. We performed two groups of experiments. These experiments are based on Least Significant Bit-Blowfish and Least Significant Bit-AES for image watermarking and encryption. First of all, each picture is watermarked using the methods described in the previous chapter, then an encryption is applied. Next we try to decrypted and extract the watermark to compute the amount of damage done to the watermark.

5.4 Experimental Results

We take six different colored images and run the proposed algorithm LSB-AES and LSB-BLOWFISH to find out watermarked, encrypted watermarked, decrypted watermarked and finally the original image. For LSB-AES and LSB-BLOWFISH algorithm we have written the code on java JDK 7.0 and Matlab R2015a. This method is genarated a image on first step i.e (c) watermarked image was generated at watermark embedding process by using stream data of (a) original image and (b) watermark logo. In second step, encryption process used the (c) watermarked image stream to produced the (d) encrypted watermarked image. In third step, decryption process used the (d) encrypted watermarked image stream data to produced (e) decrypted watermarked image. In last step, watermark extraction process take the (e) decrypted watermarked image stream data to get the (a) original image using (b) watermark logo.

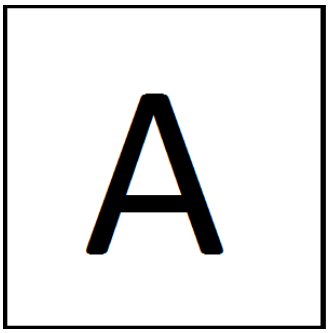
Table 5.1: PERFORMANCE OF LSB-BLOWFISH ALGORITHM

Image Name	Embedding Time	Encryption Time	Decryption Time	Extraction Time	PSNR original image and watermark image	MSE original image and watermark image
Image1	0.325	0.055	0.040	0.105	28.3263	95.5978
Image2	0.082	0.011	0.012	0.070	29.2551	77.1913
Image3	0.056	0.010	0.011	0.062	28.8527	84.6865
Image4	0.054	0.004	0.004	0.063	32.3273	38.0493
Image5	0.117	0.014	0.014	0.100	30.4211	59.0158
Image6	0.055	0.006	0.008	0.056	26.3805	149.6356

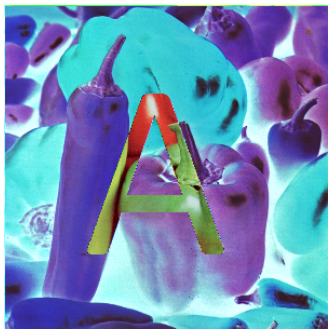
Table 5.2: PERFORMANCE OF LSB-AES ALGORITHM

Image Name	Embedding Time	Encryption Time	Decryption Time	Extraction Time	PSNR original image and watermark image	MSE original image and watermark image
Image1	0.321	0.078	0.042	0.102	28.3263	95.5978
Image2	0.078	0.007	0.007	0.065	29.2551	77.1913
Image3	0.057	0.007	0.007	0.059	28.8527	84.6865
Image4	0.053	0.003	0.003	0.065	32.3273	38.0493
Image5	0.050	0.008	0.006	0.068	30.4211	59.0158
Image6	0.058	0.005	0.005	0.056	26.3805	149.6356

(a) Original Image



(b) Watermark Logo



(c) Watermarked Image

(d) Encrypted Image

(e) Decrypted Image

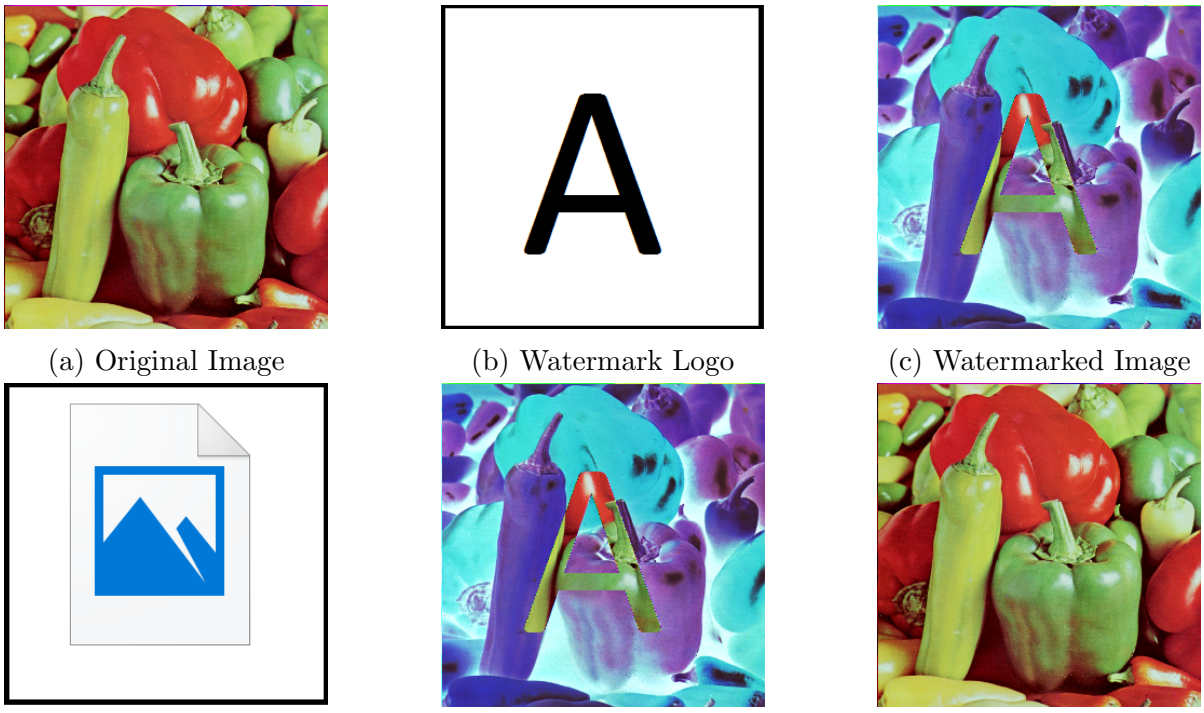
(f) Extracted Image

Figure 5.1: Results of LSB-AES Algorithm

Figure 5.1 shows the experimental results for LSB-AES analgorithm where as figure 5.2 shows the experimental results for LSB-BLOWFISH algorithm. For both method, we used same watermarking algorithm for embedding and extraction process but different encryption technique to encrypt and decrypt the image data.

We have shown the performance metrics like time required for watermark embedding process, encryption process, decryption process and extraction process. Table 5.1 shows the performace metrics for LSB-BLOWFISH algorithm where as Table 5.2 shows the performace metrics for LSB-AES algorithm.

From the performance metrics table, it is seen both encryption and decryption process is taking nearly same time where as the embedding and extraction process time significantly differ for different images. For both algorithm, PSNR value of watermarked



(a) Original Image (b) Watermark Logo (c) Watermarked Image

(d) Encrypted Image (e) Decrypted Image (f) Extracted Image

Figure 5.2: Results of LSB-BLOWFISH Algorithm

image and original image is acceptable. We have repeated these four phases for all images. We have found that the performance metrics is totally depending upon the pixel numbers which is present in a image. In my experimented results the PSNR values are in the range +25 to 35dB. Both blowfish and advanced encryption standard (AES) are giving better results as compared to RC4 encryption algorithm. To get less computational complexity we have used a look-up table method to calculate complexity. In my experimental result shows variation in encryption and decryption is due to variation in total number of pixel in different images.

Chapter 6

Conclusion and Future Work

In watermarking process data is embedded in the image in such a way that the quality of image remains the same and the modification is not very much perceptible, and the changes may or may not be visible in the actual image. Image encryption is done to ensure that the image is not tempered with while it is in transition from source to destination. In today's scenario when users are moving to mobile devices with low power the amount of CPU required by any technique should be less and should be fast and use less memory techniques.

6.1 Summary

In my proposed work we have combined both watermarking and encryption technique which gives image authority as well as image security. For watermarking algorithm we used spatial domain i.e least significant bit watermarking algorithm and Blowfish, Aes for image encryption. We have implemented two combined algorithm LSB-BLOWFISH and LSB-AES and compared their experimental results. It is found that both the algorithms was works acceptably well for image encryption security and watermarking scheme, as compared with performance of other techniques[12]. Also the PSNR values are in the range +25 to +35db which is acceptable for watermark algorithm.

6.2 Future Work

In our proposed method, we used spatial watermarking algorithm which is not giving good results against major attacks as well as in a compressed domain. Also spatial domain watermarking algorithms are relative low-bit capacity. In our current model lossy image compression techniques are not very much resistant. Future work can be carried out by working on and analyzing the transform domain watermarking algorithms. We can analysis the results of both spatial domain and transform domain with encryption technique.

References

[1] Dalel Bouslimi, "A Joint Encryption/Watermarking System for Verifying the Reliability of Medical Images", IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE, VOL. 16, NO. 5, SEPTEMBER 2012

[2] G. Boato, N. Conci, V. Conotter, F.G.B. De Natale and C. Fontanari, "Multimedia asymmetric watermarking and encryption", IEEE Trans. April 2008

[3] Sangita Zope-Chaudhari, Parvatham Venkatachalam, and Krishna Mohan Buddhiraju, "Secure Dissemination and Protection of Multispectral Images Using Crypto-Watermarking", IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, VOL. 8, NO. 11, NOVEMBER 2015

[4] Stefan Katzenbeisser, Aweke Lemma, Mehmet Utku Celik, Michiel van der Veen, and Martijn Maas, "A BuyerSeller Watermarking Protocol Based on Secure Embedding", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 3, NO. 4, DECEMBER 2008

[5] B. Subramanyan, Vivek M. Chhabria, T. G. Sankar Babu, "Image Encryption Based on AES Key Expansion", IEEE Feb. 2011

[6] Abdullah Bamatraf, Rosziati Ibrahim and Mohd. Najib B. Mohd Salleh, "Digital watermarking algorithm using LSB", IEEE Dec. 2010

[7] A. V. Subramanyam, Sabu Emmanue and Mohan S. Kankanhalli, "Robust Watermarking of Compressed and Encrypted JPEG2000 Images", IEEE Transaction on Multimedia, VOL 14, NO. 4, pp 703-716,JUNE 2012.

[8] A. Mousa Data, "encryption performance based on Blowfish", IEEE June 2005

[9] M. S. Hsieh, D. C. Tseng, and Y. H. Huang, Hiding Digital Watermarks using Multiresolution Wavelet Transform, IEEE Trans. on Industrial Electronics 48 (2006), no. 5, 875882.

[10] S. J. lee and S. H. Jung, A Survey of WAtermarking Techniques Applied to Multimedia, Proc. IEEE Int. Symp. on Industrial Electronics, June 2001.

[11] Y. Yusof and O. O. Khalifa, Digital Watermarking For Digital Images Using Wavelet Transform, Proc. IEEE Int. Conf. on Telecomunicatios, May 2007.

[12] M. Steinebach, S. Zmudzinski, and F. Chen, The Digital Watermarking Container: Secure and Ecient Embedding, Proc. ACM Multimedia and Security Workshop, Sebtember 2004.

[13] Shilpa P. Metkar, Milind V. Lichade, Digital image security improvement by integrating watermarking and encryption technique, IEEE, Sept. 2013

[14] Ashwak Alabaichi, Faudziah Ahmad, Ramlan Mahmod, Security analysis of blowfish algorithm, IEEE, Sept. 2013