

Secure Watermarking Techniques for Copyright Protection and Authentication of Digital Images and Audio Signals

Thesis Submitted by

Chinmay Maiti

Doctor of Philosophy (Engineering)

Department of Information Technology
Faculty Council of Engineering & Technology
Jadavpur University
Kolkata, India
2024

INDEX NO. 107/17/E

1. Title of the Thesis: Secure Watermarking Techniques for Copyright Protection and Authentication of Digital Images and Audio Signals

2. Name, Designation & Institute of the Supervisor/s:

Prof. Bibhas Chandra Dhara

Professor,

Department of Information Technology,

Jadavpur University

Kolkata-700032

3. List of publications:

(a) Journal Publications:

- 1. Chinmay Maiti and Bibhas Chandra Dhara**, “A blind audio watermarking based on singular value decomposition and quantization”, International Journal of Speech Technology (IJST), Springer, Vol. 25, pp. 759-771, 2022.
- 2. Chinmay Maiti, Bibhas Chandra Dhara, Saiyed Umer and Vijayan Asari**, “An Efficient and Secure Method of Plaintext-based Image Encryption Using Fibonacci and Tribonacci Transformations”, IEEE Access, Vol. 11, pp. 48421-48440, 2023.

(b) Conference Publications:

- 3. Chinmay Maiti and Bibhas Chandra Dhara**, “A Robust Binary Watermarking Scheme using BTC-PF Technique”, In Proc. of International Conference on Eco-friendly Computing and Communication Systems, LNCS, Springer, CCIS 305, pp. 178-185, 2012.

4. **Chinmay Maiti and Bibhas Chandra Dhara**, “A Binary Watermarking Scheme using Quantization Levels of BTC-PF Method”, IEEE International Conference on Communications, Devices and Intelligent Systems (CODIS), pp. 604-607, 2012.
5. **Chinmay Maiti and Bibhas Chandra Dhara**, “A Grayscale Watermarking Technique using Sub-sampling and Singular Value Decomposition”, IEEE International Conferences on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 1511-1516, 2016.
6. **Chinmay Maiti and Bibhas Chandra Dhara**, “A New Audio Watermarking Technique based on Empirical Mode Decomposition and Quantization”, 5th International Conference on Computing, Communication and Sensor Network (CCSN), pp. 177-181, 2016.
7. **Chinmay Maiti and Bibhas Chandra Dhara**, “Image encryption with a new fibonacci transform”, in IEEE 5th International Conference on Emerging Applications of Information Technology, pp.1-4, 2018.

4. **List of Patents:** None

5. **List of Presentations in International/National Conferences:**

1. **Chinmay Maiti and Bibhas Chandra Dhara**, “A Binary Watermarking Scheme using Quantization Levels of BTC-PF Method”, IEEE International Conference on Communications, Devices and Intelligent Systems (CODIS), pp. 604-607, 2012.
2. **Chinmay Maiti and Bibhas Chandra Dhara**, “A Grayscale Watermarking Technique using Sub-sampling and Singular Value

Decomposition”, IEEE International Conferences on Wireless Communications, Signal Processing and Networking (WiSP-NET), pp. 1511-1516, 2016.

3. **Chinmay Maiti and Bibhas Chandra Dhara**, “A New Audio Watermarking Technique based on Empirical Mode Decomposition and Quantization”, 5th International Conference on Computing, Communication and Sensor Network (CCSN), pp. 177-181, 2016.
4. **Chinmay Maiti and Bibhas Chandra Dhara**, “Image encryption with a new fibonacci transform”, in IEEE 5th International Conference on Emerging Applications of Information Technology, pp.1-4, 2018.

“Statement of Originality”

I **Chinmay Maiti** (Index No. 107/17/E) registered on February 23, 2017 do hereby declare that this thesis entitled “Secure Watermarking Techniques for Copyright Protection and Authentication of Digital Images and Audio Signals” contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred to materials and results that are not original to this work.

I also declare that I have checked this thesis as per the “Policy on Anti Plagiarism, Jadavpur University, 2019”, and the level of similarity as checked by iThenticate software is 8%.

Chinmay Maiti

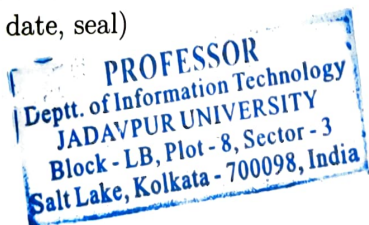
Signature of Candidate

Date: *20/02/2024*

Bohan 20.02.24

Certified by Supervisor:

(Signature with date, seal)



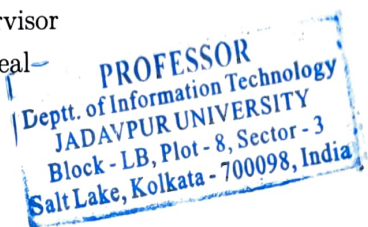
CERTIFICATE FROM THE SUPERVISOR/S

This is to certify that the thesis entitled "Secure Watermarking Techniques for Copyright Protection and Authentication of Digital Images and Audio Signals" submitted by Shri Chinmay Maiti, who got his name registered on February 23, 2017 for the award of Ph.D. (Engg.) degree of Jadavpur University is absolutely based upon his own work under the supervision of me and that neither his thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.


.....

Signature of the Supervisor

and date with Office seal-



Acknowledgements

I am deeply indebted to many people who have contributed in making the completion of this PhD dissertation work possible.

Undertaking this Ph.D. has been a truly life-changing experience for me and it would not have been possible without the constant support and encouragement from my supervisor Dr. Bibhas Chandra Dhara. From topic selection to paper writing and finally to thesis preparation he has guided me constantly within his busy schedule. I would like to express earnest gratitude and respect to him from the bottom of my heart.

I do wish to gratefully acknowledge the continuous cooperation and support provided by Dr. Uttam Roy, Dr. Parama Bhowmik, Dr. Sujit Das, Dr. Shyamalendu Kandar, Mr. Alenrex Maity and all other research fellows of the Department of Information Technology, Jadavpur University, Salt Lake Campus, Kolkata, India. The meaningful suggestions, constructive criticisms provided by them related to this research work has enriched the quality of the work. My heart-full thanks to Susanta da and all the other technical assistants, office assistants of Department of Information Technology, Jadavpur University for providing support in many technical and official issues towards successful completion of this thesis.

My sincere thanks to all my colleagues in the department of Computer Science & Engineering, Kolaghat and all the technical assistants and office staff for their continuous support. A special thanks to Dr. Prabhas Bose, and Dr. Suman Bhowmik for their constructive suggestions and encouragements.

I am deeply indebted to my parents Mr. Gobinda Maiti and Mrs. Sumati Maiti who are the main inspiration of my life. I wish to express my gratitude to my parents-in-laws Late Mr. Madhusudan Dutta, and Mrs Subhra Dutta for their constant support.

Last but not the least, I feel thankful to my beloved wife Mrs. Baisakhi Maiti , my beloved daughter Miss Senha Maiti and my son Mr. Ishan Maiti for their presence in my life. Their full-hearted support has made the journey of this thesis towards completeness.

Abstract

In recent years, due to the advancement in multimedia technology and Internet technology, a large volume of digital media such as images, audio, and video are generated, stored, and shared on various open platforms. This data may be crucial to an individual or an organization as it contains sensitive and useful information like personal information, medical information, business secrets, military information, etc. The illegal copying, modification, and forgery of digital media is a growing problem. There is a high risk of misuse of that information due to unauthorized access to the media data. As a result, it can harm personal reputation and national security, and huge financial loss in business, etc. So, issues concerning the security of these digital media are crucial.

The common approaches for information security of digital information are cryptography, steganography, watermarking, and secret sharing. In this thesis, we have focused on the security of images and audio signals. Our primary goal is to design secure watermarking techniques for images and audio signals. The watermarking techniques involve embedding the specific information (i.e., a watermark) into the digital media without perceptual degradation of the media. The ownership of media or the illegal use of the data can be established by the extracted watermark. The main target of this thesis is to use watermarking techniques for copyright protection and authentication of digital images and audio signals.

In this work, we have proposed two image watermarking techniques to achieve the goal of the thesis. The first technique is a binary image watermarking scheme, a semi-blind and robust scheme, based on the BTC-PF method where a binary image is used as a watermark. The second technique is a blind grayscale image watermarking scheme based on SVD where a grayscale image is embedded as a watermark. Both the proposed techniques are secure, provide good quality watermarked images, and are applicable for copyright protection or authentication of images. The performance of the proposed image watermarking techniques is similar to or better compared to state-of-the-art (SoA) methods.

Digital audio is an important media data and utilization of audio data is increasing day-by-day. So, it becomes mandatory to safeguard the ownership as well as the integrity of audio data. In this thesis, we have proposed an audio watermarking technique for copyright protection and authentication of audio signals. The proposed audio watermarking technique is based on SVD and quantization of the largest singular value to embed a binary watermark. The experimental output of the proposed audio watermarking technique is comparable to or better than SoA methods.

To make our proposed watermarking methods secure, the watermark images are encrypted before embedding. The proposed image encryption method is very efficient, plaintext sensitive, robust against different attacks, and gives results similar to the SoA methods. The proposed robust encryption method is further extended to achieve fragile image encryption. This is done using the concept of triple encryption: i) first encryption using a key key_1 , ii) then decrypt using another key key_2 , and iii) finally, the intermediate result is encrypted further using the key key_1 , where $key_1 \neq key_2$. To achieve robust/fragile watermarking to verify copyright information and authentication of data, the watermark image needs to be encrypted by robust/fragile image encryption method, respectively.

Table of contents

List of figures	xv
List of tables	xix
1 Introduction	1
1.1 Methods for Information Security	1
1.1.1 Cryptography	2
1.1.2 Steganography	3
1.1.3 Watermarking	5
1.1.4 Secret Sharing	5
1.1.5 Comparison of the security methods	6
1.2 Basics of watermarking and related issues	8
1.2.1 Terminologies in watermarking	8
1.2.2 Features of watermarking techniques	9
1.2.3 Types of watermarking	12
1.2.4 Evaluation of watermarking techniques	13
1.2.5 Applications of watermarking	15
1.3 Objective of the thesis	16
1.4 Contribution of the thesis	17
1.5 Organization of the thesis	18
2 Secure Watermarking Techniques for Copyright Protection and Authentication of Images	19
2.1 Introduction	19
2.2 Literature review	20
2.2.1 Spatial domain-based watermarking	21
2.2.2 Transform domain-based watermarking	22
2.3 Background of the watermarking methods	23

2.3.1	BTC-PF method	23
2.3.2	Singular value decomposition	24
2.4	Proposed Binary Image Watermarking Method	28
2.4.1	Experimental Results of BWBTC-PF	30
2.5	Proposed Grayscale Image Watermarking Method	37
2.5.1	Embedding process of GWSVD	42
2.5.2	Extraction process of GWSVD	44
2.5.3	Experimental Results of GWSVD	46
2.6	Conclusions	60
3	Secure Watermarking Technique for Copyright Protection and Authentication of Audio Signals	61
3.1	Introduction	61
3.2	Literature Survey	62
3.3	Proposed Audio Watermarking Method	65
3.3.1	Embedding process of AWSVD method	65
3.3.2	Extraction process of AWSVD method	69
3.4	Experimental Results and Performance Analysis of AWSVD	69
3.4.1	Imperceptibility and Payload	73
3.4.2	Time complexity	76
3.4.3	Robustness test	76
3.4.4	Quantization parameter	85
3.5	Conclusions	85
4	Image Encryption for Secure and Robust-Fragile Watermarking	87
4.1	Features to Analysis the Image Encryption Method	89
4.2	Literature Survey	95
4.3	Mathematical Background	98
4.3.1	Fibonacci Numbers	98
4.3.2	Tribonacci Numbers	100
4.4	Proposed Image Encryption Method	101
4.4.1	Key generation	102
4.4.2	The Confusion phase	103
4.4.3	The Diffusion phase	104
4.4.4	Image encryption	108
4.4.5	Experimental Results	109
4.5	Security analysis	112

4.5.1	Key space analysis	114
4.5.2	Complexity and Execution time	114
4.5.3	Randomness analysis	116
4.5.4	Pixel disparity analysis	123
4.5.5	Key sensitivity analysis of encryption process	124
4.5.6	Key sensitivity of the decryption process	125
4.5.7	Plaintext sensitivity analysis	128
4.5.8	Differential analysis	128
4.5.9	Chosen-plaintext and Known-plaintext attacks	130
4.5.10	Cropping attack	130
4.5.11	Noise attack	133
4.6	Extended FTTIE method	133
4.7	Conclusions	137
5	Conclusions and Future Scopes	141
5.1	Conclusions	141
5.2	Future scope of improvement	144
	Bibliography	147

List of figures

1.1	Block diagram of cryptography technique	2
1.2	Diagram of general image encryption scheme	3
1.3	Block diagram of steganography technique	4
1.4	Block diagram of watermarking technique.	5
1.5	Block diagram of (4, 5) secret sharing technique.	6
1.6	Trade-off among imperceptibility, capacity, and robustness	11
1.7	Example of visible and invisible image watermarking	12
2.1	Example of BTC-PF method with one 4×4 block and eight 2-level patterns: (a) original block, (b)-(i) represent the error due to encoding of the block for the corresponding pattern in the figure, (j) reconstructed block, the best-fit pattern is P_1 (as E_1 is minimum)	25
2.2	Block diagram of the proposed BWBTC-PF method: (a) Embedding process, (b) Extraction process	28
2.3	Test images (including both host images and watermark images) used in these experiments.	31
2.4	Quality of the watermarked images of BWBTC-PF method	33
2.5	Payload vs. PSNR of different methods	34
2.6	Block diagram of the embedding process of the proposed GWSVD method	44
2.7	Block diagram of the extraction process of the proposed GWSVD method	45
2.8	Quality of the watermarked images of GWSVD method with embedding strength $T=50$	48
2.9	Average PSNR of the watermarked images with varying embedding strength (T)	49
2.10	Average SSIM of the watermarked images with varying embedding strength (T)	50
2.11	Average PSNR of the extracted watermark images with varying embedding strength (T)	53

2.12	Average SSIM of the extracted watermark images with varying embedding strength (T)	54
3.1	Block diagram of the embedding process of AWSVD	66
3.2	Partition of the range of $\alpha_{k,1}$ with interval length Δ	66
3.3	Watermark bit embedding when watermark bit is '0' and h_k is odd: (a) $h_k=1$, (b) $h_k = h_{max}$, (c) $1 < h_k < q$	67
3.4	Block diagram of the extraction process of AWSVD	69
3.5	Host audio signals	71
3.6	Binary watermark images: original, encrypted by FTTIE, and encrypted by FTTIE _{ext}	71
3.7	Performance of the proposed method AWSVD on the signal 'Jazz'.	73
3.8	Payload vs. PSNR of different methods	75
3.9	SNR (dB) value of the watermarked image of AWSVD method with varying quantization steps	83
3.10	BER (%) of extracted watermark of AWSVD method with varying quantization steps under AN, LP, EA, DE attacks	84
4.1	Histogram of the grayscale images: (a) Original image, (b) Encrypted image	90
4.2	Block diagram of the proposed FTTIE method.	102
4.3	Simplified diagram of the FTTIE method which shows the looping between the confusion and diffusion phases.	105
4.4	Test images used in this experiment: grayscale and corresponding binary images	110
4.5	Encrypted images using FTTIE method	111
4.6	Decipher images using FTTIE method with correct decryption key	112
4.7	Decipher images using FTTIE method with wrong decryption key	113
4.8	Histogram of the images: (a) Original images, (b) Encrypted images.	118
4.9	Scatter diagrams of some original and corresponding encrypted images.	122
4.10	Cropped cipher images with different degrees of cropping (a - d, a'-d') and corresponding decipher images (e - f, e'-f')	131
4.11	Cipher images with different degree of Salt & Pepper noise (a - d, a'-d') and corresponding decipher images (e - f, e'-f')	132
4.12	Block diagram of the FTTIE _{ext} method.	133
4.13	Fragility output of FTTIE _{ext} when cipher images (a - f) are modified at (100, 100) location and corresponding decipher images are in (a' - f').	135

4.14	Output of FTTIE and FTTIE _{ext} methods under cropping attack with 50% and 25% cropping of the cipher image ‘Lena’	138
4.15	Output of FTTIE and FTTIE _{ext} methods under salt & pepper noise with noise density 0.01 and 0.03 on the cipher image ‘Lena’	139

List of tables

1.1	A comparison of different security methods.	9
2.1	Comparative study of the payload and the quality of the watermarked images given by different methods.	32
2.2	Average execution time (seconds) of the proposed BWBTC-PF method and SoA methods	36
2.3	Different attacks on watermarked image	37
2.4	Robustness and Fragility of BWBTC-PF method: host image ‘Air-plane’, watermark image ‘Logo’	38
2.5	Robustness and Fragility of BWBTC-PF method: host image ‘Lena’, watermark image ‘Logo’	39
2.6	Robustness and Fragility of BWBTC-PF method: host image ‘Air-plane’, watermark image ‘Flower’	40
2.7	Robustness and Fragility of BWBTC-PF method: host image ‘Lena’, watermark image ‘Flower’	41
2.8	Average NC (%) and SSIM of different techniques under attacks ‘Logo’ as watermark	42
2.9	Average NC(%) and SSIM of different techniques under attacks ‘Flower’ as watermark	43
2.10	Comparative study of the payload and the quality of the watermarked images given by different methods.	51
2.11	Average execution time (seconds) of the proposed GWSVD method and SoA methods	51
2.12	Robustness and Fragility of GWSVD method in terms of PSNR and SSIM (given in sideways): host image ‘Zelda’, watermark image ‘Cam-eraman’	55

2.13	Robustness and Fragility of GWSVD method in terms of PSNR and SSIM (given in sideways): host image ‘Splash’, watermark image ‘Cameraman’	56
2.14	Robustness and Fragility of GWSVD method in terms of PSNR and SSIM (given in sideways): host image ‘Zelda’, watermark image ‘Baboon’	57
2.15	Robustness and Fragility of GWSVD method in terms of PSNR and SSIM (given in sideways): host image ‘Splash’, watermark image ‘Baboon’	58
2.16	Average PSNR and SSIM of different techniques under attacks ‘Cameraman’ and ‘Baboon’ as watermark with embedding strength $T=50$. .	59
3.1	SNR versus payload given by AWSVD method with ‘Logo’ as watermark	74
3.2	SNR versus payload given by AWSVD method with ‘Flower’ as watermark	74
3.3	Comparative study of the performance of the proposed method with SoA methods	76
3.4	Average execution time (seconds) of the proposed AWSVD method and SoA methods	76
3.5	Different attacks on watermarked audio signal	77
3.6	Performance of AWSVD against attacks: host signal ‘Jazz’ and watermark images ‘Logo’ and ‘Flower’	79
3.7	Quality of the extracted watermark from watermarked audio signals with ‘Logo’ as watermark under attacks	80
3.8	Quality of the extracted watermark from watermarked audio signals with ‘Flower’ as watermark under attacks	80
3.9	Robustness performance of different audio watermarking techniques on the ‘Classical’ audio signal with watermark ‘Logo’.	81
3.10	Robustness performance of different audio watermarking techniques on the ‘Classical’ audio signal with watermark ‘Flower’.	82
4.1	Fibonacci and Tribonacci numbers	100
4.2	Techniques and encryption process used in FTTIE and SoA methods. .	115
4.3	Average execution time (seconds)	116
4.4	Comparative study with respect to the execution time (seconds)	117
4.5	Distribution of 0’s and 1’s in binary images (in %).	117
4.6	Analysis of randomness with respect to entropy	119
4.7	The achievement of entropy of the proposed method in compare with SoA methods	119
4.8	Local Shannon entropy of the cipher images.	119

4.9	Correlation coefficients of the images	120
4.10	Comparative study of randomness analysis with respect to the correlation coefficients	121
4.11	Comparative study of randomness analysis with respect to the χ^2 value	123
4.12	Pixel difference between the plain and cipher images	124
4.13	Test cases of key sensitivity analysis.	124
4.14	NPCR measure for key sensitivity analysis in % (for grayscale images) .	125
4.15	UACI measure for key sensitivity analysis in % (for grayscale images) .	125
4.16	The key sensitivity of the proposed method and SoA methods in terms of NPCR %	126
4.17	NPCR/UACI/MAE measure for key sensitivity analysis in % (for binary images)	126
4.18	Key sensitivity measure of the decryption process in % (for grayscale images)	127
4.19	Key sensitivity measure of the decryption process in % (for binary images)	127
4.20	Test cases of differential attacks.	128
4.21	NPCR value (%) due to modified plain image (grayscale)	129
4.22	UACI value (%) due to modified plain image (grayscale)	129
4.23	NPCR/UACI value (%) due to modified plain image (binary)	129
4.24	The performance of the proposed method and SoA methods against differential attack in % (for grayscale image).	130
4.25	Fragility of the FTTIE _{ext} method, only modified the pixel at (100, 100) of the cipher image.	136

List of Abbreviations

AWSVD	Audio Watermarking using SVD
BER	Bit Error Rate
bpp	bit per pixel
bps	bit per second
BTC	Block Truncation Coding
BTC-PF	Block Truncation Coding with Pattern Fitting
BWBTC-PF	Binary image Watermarking using BTC-PF
EFW	Encryption for Fragile Watermarking
ERW	Encryption for Robust Watermarking
FT	Fibonacci Transformation
FTTIE	Fibonacci and Tribonacci Transformation based Image Encryption
GWSVD	Grayscale image Watermarking using SVD
IFPI	International Federation of the Phonographic Industry standard
ITT	Inverse Tribonacci Transformation
LSB	Least Significant Bit
LSV	Largest Singular Value
MAE	Mean Absolute Error
MSE	Mean Square Error
NC	Normalize Correlation
NPCR	Number of Pixel Change Rate
PB	Patternbook
PSNR	Peak Signal-to-Noise-Ratio
SHA	Secure Hash Algorithm
SNR	Signal-to-Noise-Ratio
SoA	State-of-the-Art
SSIM	Structural Similarity Index Measure
SVD	Singular Value Decomposition
TT	Tribonacci Transformation
UACI	Unified Average Changing Intensity

Chapter 1

Introduction

In this age of the digital revolution, digital information is increasing at an exponential rate and is easily available to everyone and everywhere through the Internet. The information may be very crucial to an individual or an organization. This information can be modified and duplicated easily by using freely available multimedia processing tools [1]. In recent years, the emergence of modern multimedia devices such as smartphones, tablets, camcorders, cameras, and state-of-the-art communication technology are essential parts of our everyday life. Due to this advancement, a large volume of digital media such as images, audio, and video are stored and shared on open platforms [2] like Facebook, Twitter, LinkedIn, Orkut, Flickr, etc. These digital media contain sensitive and useful information like personal information, medical information, confidential data and trade secrets, military information, etc. When the information is transmitted through communication channels, there is a high risk of misuse of information. As a result, it harms personal reputation and national security, and huge financial loss in business, etc. Easy access to multimedia gives rise to issues such as content authentication, security, copyright protection, and ownership identification. Various methods have been developed for information security like Cryptography, Steganography, Watermarking, and Secret Sharing. In this thesis we use the terms ‘digital information’ and ‘information’ synonymously, i.e., information means digital information. In the following section, we have studied basic techniques of information security.

1.1 Methods for Information Security

Generally, information is transmitted through public channels, which is not secure and hence we need to protect information from the above threats. There are four

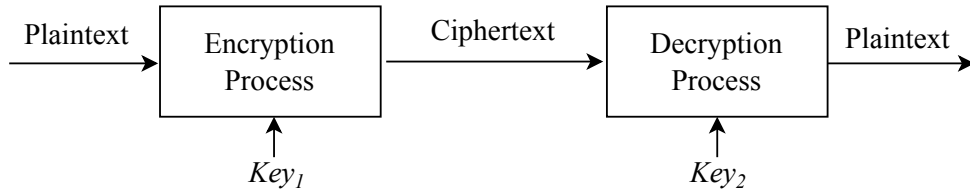


Figure 1.1: Block diagram of cryptography technique

widely used methods for information security: i) Cryptography, ii) Steganography, iii) Watermarking, and iv) Secret Sharing. These four methods are briefly discussed in the following subsections.

1.1.1 Cryptography

Cryptography is the art of transforming a message into an unreadable form, using mathematics so that unintended recipients cannot understand the transformed message. Cryptography is widely and frequently used technology to ensure the security (privacy) of the information and its importance has been growing with the advancement of technology. A cryptographic method is a combination of encryption and decryption processes. Encryption is a process by which we can convert a plaintext into a meaningless ciphertext with the help of a key. The encrypted message is transmitted through the communication channel and the receiver receives a cipher message. To decipher the message, the receiver decrypts the encrypted message using the decryption process with the corresponding key. The security of the encrypted message solely depends on the key. A block diagram of the cryptography system is shown in Fig. 1.1.

A cryptographic method is either symmetric (private key) where the same key is used for both encryption and decryption or asymmetric (public key) where the public key is used to encrypt the message and the corresponding private key is used to decrypt the message. For example, Data Encryption Standard (DES) [3], Advanced Encryption Standard (AES) [4] are symmetric key cryptography whereas RSA [5], ECC [6], DH [7] are asymmetric key cryptography. Asymmetric key cryptography is obtained at the cost of computational complexity; whereas, symmetric key cryptography is faster and less complex. To enhance the security level, both approaches are needed and sometimes these are combined to fulfill the requirement, such technique is known as hybrid cryptosystem [8–10] and they are extensively used in many applications such as transport layer security (TLS) protocol and secure shell (SSH) protocol. To protect

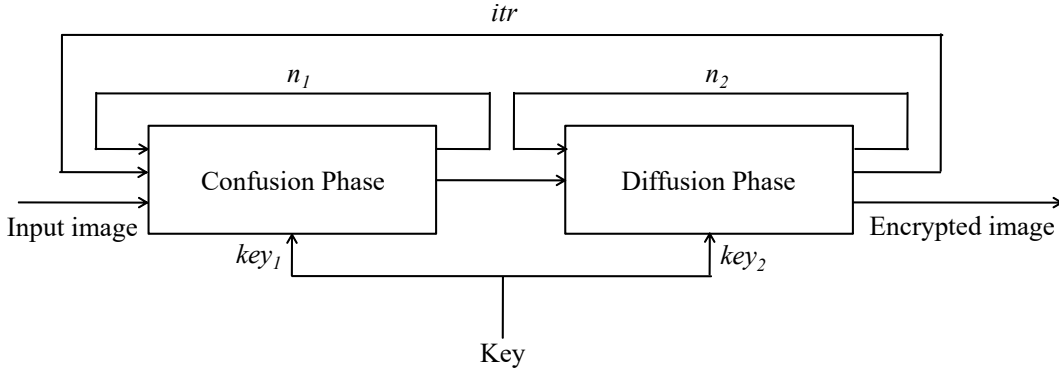


Figure 1.2: Diagram of general image encryption scheme

the digital data, techniques like image encryption [11, 12], audio encryption [13, 14], and video encryption [15, 16] are required which are in separate class other than the classical encryption techniques due to certain specific features of multimedia data like a huge volume of the data, high redundancy in data, strong spatial correlation in data etc.

Image encryption is a special kind of encryption technique that can handle the bulky data of an image efficiently. In the present multimedia age, image plays an important role. So, image encryption is also an important kind of technique to provide security to multimedia data. The general block diagram of the image encryption is given in Fig. 1.2. An image encryption based on two phases: a) confusion phase, and b) diffusion phase. In the confusion phase, the position of pixels is changed without modifying the intensity value. In diffusion phase, pixel's intensity value is changed. An image encryption method is said to be robust, if we can retrieve the original image (it may be a noisy version, but the original image is easily recognizable) under any attack. An image encryption method is said to be fragile, if we cannot obtain any information for the plain image from the decipher image, i.e., the deciphered image is noisy.

1.1.2 Steganography

“Steganography is the practice of undetectably altering a work to embed a secret message” [17], i.e., it is a process to hide a secret message into another media like image, audio, or video. The media in which the message is embedded is called cover media, and the cover media with the embedded message is known as stego media. The block diagram of the steganography method is given in Fig. 1.3. Steganography system has different components like secret message, cover media, and an embedding algorithm

(with a key, which is optional). Finally, the stego data is transmitted through a communication channel. The process of embedding has designed so that the change in the stego is not visible to unauthorized persons. Commonly used steganography methods are classified depending on the cover medium. The steganography methods are image steganography, audio steganography, and video steganography.

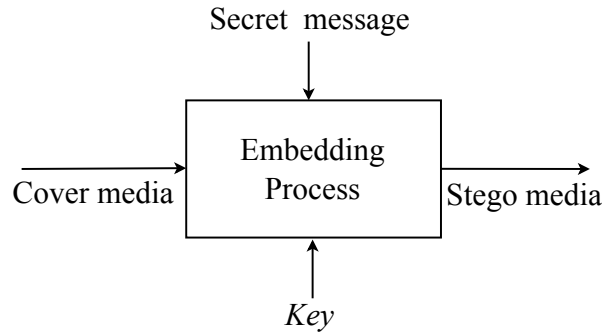


Figure 1.3: Block diagram of steganography technique

Among these methods image steganography is widely used. According to the domain, image steganography techniques are divided into two classes: Spatial domain methods and Transform domain methods.

1. Spatial domain methods: In the spatial domain steganography algorithm, a secret message is concealed in the cover image by modifying the pixel values directly. The simplest and easiest technique in the spatial domain is the least significant bit (LSB) modification. The LSB of pixels of the cover image are replaced by the bits of the secret message. Different techniques have been developed in the spatial domain such as LSB-based techniques [18–20], Gray Level Modification (GLM) [21], Pixel Value Difference (PVD) [22, 23], Histogram based techniques [24, 25], Pixel Pair Matching (PPM) [26, 27], etc.
2. Transform domain methods: In the transform domain steganography algorithm, a secret message is concealed in the cover image by modifying the frequency components of the cover image. The transform-based steganography algorithm is more complex than a spatial-based algorithm. However, transform domain approaches are more secure, than spatial domain approaches, against different stego attacks like cropping, compression, etc. Transform-based techniques can be designed using different methods such as Discrete Cosine Transform (DCT) [28, 29], Discrete Fourier Transform (DFT) [30], Discrete Wavelet Transform (DWT) [31, 32].

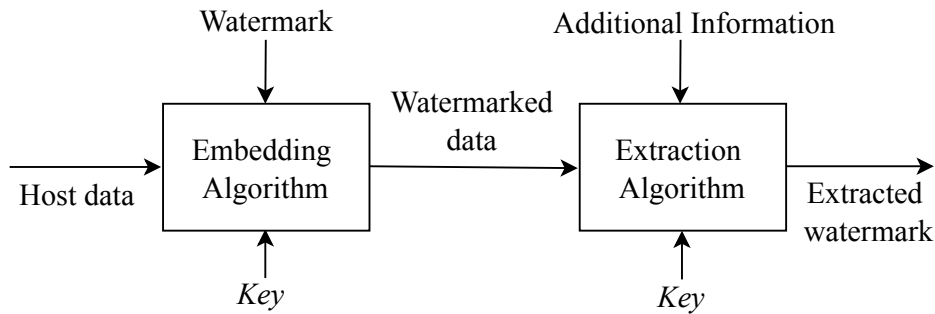


Figure 1.4: Block diagram of watermarking technique.

An effective image steganography technique must consider certain features like i) visual quality of the stego image, (ii) amount of secret message that can be embedded, iii) robustness against attacks, and iv) blindness during the extraction of secret message, etc.

1.1.3 Watermarking

“Digital watermarking is the practice of imperceptibly altering a Work to embed a message about that Work” [17]. It is another important technique to provide the security to the confidential data. This technique is used to verify content authentication and ownership detection (copyright protection) of multimedia data. Watermarking is a method in which ownership data (may be a symbol) or copyright information is embedded into the host media while maintaining minimum degradation of the host. The ownership of data or the illegal use of the secret data can be established by the extracted watermark. In authentication, if watermarked data is modified, then nothing can be guessed from the extracted watermark, i.e., a noisy data will be extracted that proves that the watermarked image has been altered. The block diagram of the watermarking technique is shown in Fig. 1.4. For copyright protection and authentication of digital media, many technique have been proposed by researchers [33, 34].

1.1.4 Secret Sharing

Secret sharing is an important technique where a piece of secret information is shared among the authorized parties after splitting it into smaller pieces or shares. Each share is useless on its own; but when all the shares or a certain number of shares are combined then the original secret can be reconstructed. The goal of the secret sharing is to design a cryptographic protocol that requires interaction among entities.

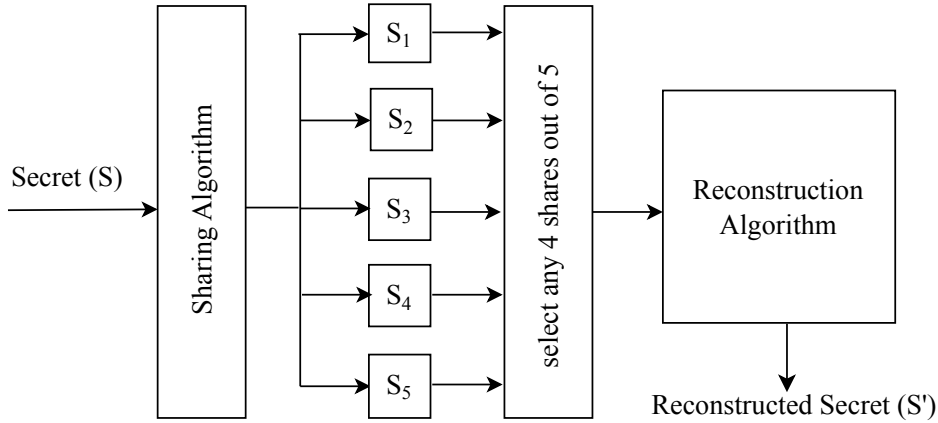


Figure 1.5: Block diagram of $(4, 5)$ secret sharing technique.

A (k, n) , $k \leq n$, threshold secret sharing scheme implies that it divides the secret ‘ S ’ into n -shares and distributes among n -participants and the original secret ‘ S ’ can be recovered only if k or more number of correct shares are combined. A diagram of $(4, 5)$ secret sharing scheme is shown in Fig. 1.5.

Secret sharing technique is independently formulated in the field of information security by Shamir [35] and Blakley [36] in 1979. In [35], Shamir proposed a (k, n) threshold sharing scheme based on Lagrange polynomial interpolation. In [36], the secret sharing technique is based on the concept of hyperplane geometry. Initially, a secret sharing technique was developed to safeguard cryptographic keys. These techniques have been used to share strategic resources, for example, missile activation codes, etc. In the secret sharing technique, different entities like i) a dealer, who computes the shares from the given secret and then distributes them to the participants, ii) participants who hold the shares, iii) an entity called a combiner, who collects the necessary shares from the participants and reconstructs the secret. The secret sharing process becomes successful if all the entities involved in this process are trusted and honest.

Some of the entities may be dishonest, and so to detect this, different types of secret sharing schemes have been proposed like verifiable secret sharing [37, 38], cheater identification [39, 38] and cheating detection [40, 41], etc.

1.1.5 Comparison of the security methods

A study of the different security mechanisms are given in Table 1.1. From the above discussion, it has been understood that in (k, n) secret sharing scheme, if $n - k + 1$ or more shares are corrupted or altered then the secret cannot be reconstructed; other-

wise, the secret can be recovered. Moreover, we may note that other than the secret sharing scheme, all methods have the problem of single point failure, i.e., if the ciphertext, stego data, or watermarked image is corrupted or altered then there is a chance that the original secret message may not be recovered (in case of cryptography or steganography) or may not be possible to prove the ownership (in case of watermarking).

So, from this discussion, we may consider secret sharing as a particular class of security mechanism, whereas other security techniques belong to another class. In the cryptography technique, the plaintext is transformed into a meaningless form. Though both the steganography and watermarking techniques embed an information into a media, there are certain differences between steganography and watermarking, which are given below.

1. In watermarking, a pattern is embedded into digital media to claim ownership or verify the content's authenticity. In the steganography process, a piece of secret information is concealed within another piece of data, like an image or audio file.
2. Watermarking is visible or detectable, intending to deter illegal use or copying, whereas steganography has designed to be invisible and hide data covertly within other media.
3. Steganography is used for secret communication, whereas watermarking is useful for copyright management, and content authentication etc.

In cryptography, a message is converted into a format that is not understandable by anyone, and it can be understood only after the decoding process. So, after decoding, the message is not secured anymore and it can be manipulated, copied, or distributed by anyone. This is the major disadvantage of the cryptography system. Moreover, once an unauthorized user cracks the cryptography key in some way, then it is under the control of the cracker. The limitations of cryptography are solved by steganography. In the steganography process, the presence of a hidden message cannot be detected by the attacker, only authorized users can identify the existence of the hidden message. So, the main idea of steganography is secret communication between sender and receiver, and the stego itself is completely undetectable [42]. The main problem in steganography is that if attackers manipulate the media, and as a result the message may become either obliterated or unrecovered. The digital watermarking technique solves the drawbacks of both cryptography and steganography [43, 44]. In comparison with steganography, watermarking is more flexible to survive against

some manipulations, and therefore, watermarking can be used as a promising solution for the protection of digital data. To protect the media, the watermarking process can be used without significantly degrading its quality. So, watermarking techniques overcome the weaknesses that exist in cryptography and steganography. Among these techniques, watermarking has been accepted as an effective and practical technique for many applications as compared to cryptography and steganography methods.

Though digital watermarking is very important in multimedia activities, still people are using paper watermarking to identify owners and also for authentication of the document. For example, university certificates, currency, passports, etc., are authenticated with the watermark. This thesis is focused on the copyright protection and authentication of digital images and audio signals. For this purpose, in this thesis, we have proposed secure image watermarking and audio watermarking techniques where an image is used as the watermark. Here, the embedding principles are simple, and the security of the proposed systems dependent on the encryption of the watermark image. To achieve both secure and robust/fragile watermarking system, in this research, we have proposed robust and fragile image encryption techniques.

The rest of this chapter is arranged as follows. In Section 1.2, we present the basics of the watermarking technique. The objective and contribution of this thesis are presented in Section 1.3 and 1.4, respectively. Finally, the organization of the thesis is given in Section 1.5

1.2 Basics of watermarking and related issues

In this section, we have discussed the different parameters related to the watermark like commonly used terminologies, different features of watermarking, types of watermarking, evaluation of watermarking methods, and application of watermarking, etc.

1.2.1 Terminologies in watermarking

Commonly used terminologies in the field of watermarking are given below.

1. Watermark: The information that will be embedded.
2. Host data: The content in which the watermark will be embedded.
3. Watermarked data: The host data along with the embedded watermark.
4. Embedding: A process to embed the watermark into the host data.

Table 1.1: A comparison of different security methods.

Criteria	Cryptography	Steganography	Watermarking	Secret Sharing
Definition	It is a technique to convert data into incomprehensible form	It is a technique to hide the existence of communication	It is an art and science of hiding information	It is a process of distributing the secret among a group
Services offered	Confidentiality, data integrity	Confidentiality, authentication	Copyright protection, authentication	Confidentiality, reliability
Carrier	-	Any digital data	Any digital data	-
Perception	Visible but unreadable	Invisible/ Inaudible	Visible/ Invisible	Visible but incomprehensible
Key	Necessary requirement	Optional, but offers more security if it is used	Optional, but offers more security if it is used	Necessary
Concern	Robustness against attacks	Embedding capacity and recovery of the hidden message	Imperceptibility, Robustness and embedding capacity	Reconstruction of the secret message
Challenges	If one possesses the decryption key, system fails	Once the presence of a secret message is discovered, anyone can use the secret data	If removed/destroyed the watermark, system fails to serve the purpose	If shares are modified, the secret cannot be generated

5. Extraction: A process to extract the watermark from the watermarked data.
6. Watermarking: It includes both the embedding and the extraction processes of the watermark.
7. Attack: It refers to the intentional modification of the watermarked data that may impair the detection of the watermark.

1.2.2 Features of watermarking techniques

To understand a watermarking system and evaluate the effectiveness of the algorithm for different applications, we need to study the properties of the watermarking technique. Some features are computed on the watermarked data and some are computed

on the extracted watermark. Those important features associated with the watermarking system are listed below [45, 46].

1. Imperceptibility: The watermarking process should maintain the perceptual quality of the watermarked data similar to the host data. Otherwise, the distortions in the watermarked data caused by the embedding process would reduce its creative value. Besides, it may cause doubt and threaten the watermark security. There are different metrics to measure the imperceptible performance of a watermark system such as Peak Signal-to-Noise Ratio (PSNR), Signal-to-Noise Ratio (SNR), Structural Similarity Index Measure (SSIM), etc.
2. Capacity: The amount of information embedded into the host data is called data payload [46, 47] and the maximum amount of data (depends on the embedding/encoding process) that can be embedded into the host media is the watermark capacity. Depending on the application, the data payload may vary. Many researchers have been attracted to studying the capacity of information that can be embedded into the host data [2, 48].
3. Robustness: It refers to its ability to withstand non-malicious attacks. It means the extracted watermark must be identifiable even after modification of the watermarked data. The various attacks include common multimedia processing, geometrical transforms, compression, etc. For example, a watermarking algorithm is robust against compression if the watermark is extracted/detected successfully after the compression of the watermarked data. It is desirable that a robust watermarking system must show robustness against all types of attack. However, real-life applications may require robustness only on a subset of the attacks. A robust watermarking does not mean that it will be able to detect watermarks under extreme conditions (a severe degradation of the watermarked data).
4. Fragility: A watermarking system is designed to protect the embedded watermark from any type of attack. A fragile watermarking scheme is used for the complete authentication of data, where unaltered data is considered authenticated. If the watermarked data is modified, the watermark information should be destroyed. This feature can be used to determine whether the watermarked image has been modified or not. So, the fragile watermarking technique can be used to check the authenticity.
5. Blindness of detection: It refers to extracting the watermark without additional information, other than the key. A non-blind watermark system requires

- the host data, and a semi-blind watermark system requires the original watermark/some extra information to extract the watermark. The major problem of non-blind/semi-blind methods is that the host data or the original watermark may not be readily available to the receiver.
6. **Security:** This means an unauthorized user cannot remove/extract the watermark from the watermarked data. According to Kerckhoff's principle, the embedding algorithm is publicly available, and security depends on the key used in the embedding process. Authorized user, who has the correct key can disclose the watermark information in its original form. In this regard, to enhance the security level, we can pass the watermark data through the encryption process before embedding. If watermark data is encrypted then the 'key' of the watermarking process includes both the 'embedding key' and 'encryption key'. It may be noted that robustness and security are different issues in the watermarking field. A robust watermarking method can resist common attacks, but may not be secure (i.e., if no 'key' is there any person can extract the watermark since the corresponding algorithm is public).
 7. **Computational cost:** It refers to the complexity/processing time for the embedding and extraction process. Researchers have given more priority to imperceptibility, robustness, fragility, and security than the complexity of the watermarking algorithm. However, complexity is also an important parameter in designing a watermarking system.

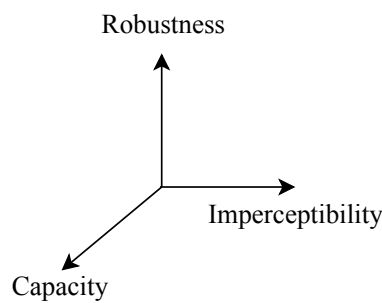


Figure 1.6: Trade-off among imperceptibility, capacity, and robustness

From the above, we may note that imperceptibility, robustness, and capacity are conflicting in nature. Due to these characteristics, for any watermarking technique, it is difficult to satisfy all three requirements simultaneously. If the imperceptibility increases then the other two properties will be decreased. On the other hand, if

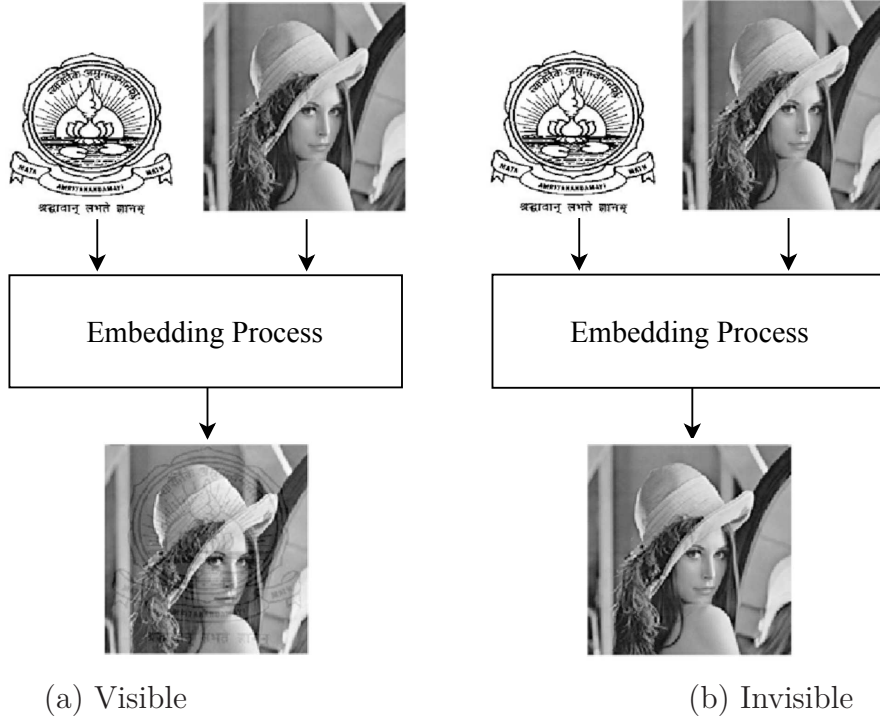


Figure 1.7: Example of visible and invisible image watermarking

capacity increases the robustness may be decreased. The trade-off among these three features is shown in Fig. 1.6. It is generally agreed that there is no “one size fits all” in watermarking. Despite the difficulty in designing robust watermarking that can defeat all possible attacks, it is desirable to tackle as many attacks as possible. Therefore, depending on the application, we have to compromise with certain features of the watermarking technique.

1.2.3 Types of watermarking

In general, digital watermark techniques are classified into five main groups:

1. Depending upon the type of host medium, techniques are classified as Audio watermarking, Image watermarking, and Video watermarking.
2. Based on domain, techniques can be of two types. The first one is the spatial domain-based method in which the watermark is embedded by changing the signal values directly [49–52]. The other one is the transform domain-based method where frequency coefficients are modified to embed the watermark [53–55].

3. Based on the visibility (in case of an audio signal, it is audibility) of the watermark, techniques are classified as visible [56], where the watermark is perceivable or invisible [57, 58], where the watermark is not perceivable. Examples of visible and invisible image watermarking are shown in Fig. 1.7.
4. Depending upon the requirement of additional information during extraction, techniques can be of three types.
 - (a) The first one is non-blind technique [59, 60] where host data is needed.
 - (b) The second one is known as the semi-blind method [61] where the original watermark or some other information is required.
 - (c) The third category is a blind watermarking technique, no information is required to extract the watermark [11, 62, 63].
5. One of the most widely adopted classifications is based on robustness against the attacks where watermark techniques can be classified into three types:
 - (a) Robust watermarking technique can resist non-malicious distortions. Watermark can be extracted even after an attack but watermark quality may be degraded [64–66].
 - (b) Fragile watermarking in which watermark is destroyed due to any sort of modification of watermarked data [67, 68], i.e., if a method is fragile then no information about the watermark can be obtained when the watermarked data is attacked.
 - (c) Semi-fragile watermarking where watermark can be destroyed by certain types of distortions, while resisting some minor changes [69, 70]

The characteristics of robust watermarking and fragile watermarking are opposite, i.e., a particular watermarking method cannot be robust as well as fragile. Concerning applications, robust watermarking schemes are suitable for copyright protection as they can resist common attacks. On the other hand, fragile watermarking methods can be used for the authentication of multimedia data as it is sensitive to modifications.

1.2.4 Evaluation of watermarking techniques

Watermarking techniques are normally evaluated concerning different features such as imperceptibility, robustness, capacity, etc. To compute numerical values for evaluating the features, different parameters are defined below.

- (i) Peak Signal-to-Noise Ratio (PSNR): It is applied to compute the quality of the watermarked image concerning the corresponding host image. It is measured in decibels, and the formula to compute PSNR is defined below.

$$\begin{aligned} PSNR &= 10 * \log_{10} \frac{(2^r - 1)^2}{MSE} \text{ dB} \\ MSE &= \frac{\sum_{i=1}^N \sum_{j=1}^N [f_h(i, j) - f_w(i, j)]}{N \times N} \end{aligned} \quad (1.1)$$

where r is the number of bits required to represent the pixel intensity, $f_h(i, j)$ and $f_w(i, j)$ denote the pixel value at (i, j) position of the host image and watermarked image, respectively, and $N \times N$ is the size of the images. The higher PSNR indicates smaller distortions due to the embedding of the watermark. It is used to judge the imperceptibility feature of the watermarking algorithm. A higher value means better imperceptibility. The PSNR is also used to evaluate the quality of the extracted watermark and compute the PSNR value between the original watermark and the extracted watermark.

- (ii) Signal-to-Noise Ratio (SNR): It is used to measure the quality of the watermarked audio signal to the corresponding host audio signal. It is defined as

$$SNR = 10 * \log_{10} \frac{\sum_{i=1}^L Z^2(i)}{\sum_{i=1}^L [Z(i) - Z_w(i)]^2} \text{ dB} \quad (1.2)$$

where Z and Z_w are host and watermarked audio signals. 'L' is the length of the audio signal. The higher value of SNR means less difference between the host signal and the watermarked signal, i.e., high imperceptibility.

- (iii) Structural Similarity Index Measure (SSIM): This quality assessment metric is used to compute the similarity between two images. The higher value of SSIM indicates higher similarity. The SSIM metric is defined by

$$SSIM = \frac{(2\mu_1\mu_2 + c_1)(2\sigma_{12} + c_2)}{(\mu_1^2 + \mu_2^2 + c_1)(\sigma_1^2 + \sigma_2^2 + c_2)} \quad (1.3)$$

Here, μ_1 and μ_2 are the mean of the first and second images, respectively. σ_1 and σ_2 are the variance of the first and second image, respectively. σ_{12} is the covariance between two images. $c_1=(k_1L)^2$, $c_2=(k_2L)^2$ have used to keep the stabilization of the decision with a weak denominator. L is the dynamic range

of the pixel intensity (for 8-bit images, $L=255$), and the default values of k_1 and k_2 are 0.001 and 0.03 respectively [71].

- (iv) Normalized Correlation (NC): Normalized correlation is used to measure the degree to which the algorithm can withstand attack. The NC is defined as follows:

$$NC = \frac{\sum_{i=1}^M \sum_{j=1}^N W(i, j) W'(i, j)}{\sqrt{\sum_{i=1}^M \sum_{j=1}^N W(i, j)^2} \sqrt{\sum_{i=1}^M \sum_{j=1}^N W'(i, j)^2}} \quad (1.4)$$

Here, W is the original watermark image, W' is the corresponding extracted watermark, and $M \times N$ is the size of the watermark. This parameter is defined to measure the quality of the binary image. A larger NC (≤ 1) value means a high similarity between two images. If $NC(W, W')$ is close to 1, then the similarity between W and W' is very high.

- (v) Bit Error Rate (BER): The BER is the measure of dissimilarity, and it is defined as

$$BER = \frac{\sum_{i=1}^M \sum_{j=1}^N W(i, j) \oplus W'(i, j)}{M \times N} \times 100\% \quad (1.5)$$

where \oplus is the exclusive or (XOR) operator and W and W' are the original and extracted watermarks, respectively. So, when the BER value is close to zero, then we may infer that both the watermarks (original and extracted) are close enough.

1.2.5 Applications of watermarking

Digital watermarking has been developed for many applications. Here, some important applications of the watermarking are mentioned below [72, 73].

1. Copyright protection: For copyright application, a watermark is embedded into the host media and it is used as copyright or ownership information. This copyright information allows owner of multimedia content to protect their possessions and exhibit their ownership in case of a controversy. A robust watermark method should resist different types of signal processing attacks which means, the watermark should be extracted with an acceptable quality from the attacked watermarked media to demonstrate the ownership or protect the copyright. Furthermore, it is difficult to remove or destroy the watermark from the watermarked media without substantial degradation of watermarked content.

2. Authentication: The multimedia content can be tampered with easily and authentication of media data must be carried out to make sure that no unauthorized modifications are performed. Images are important information and are used in many crucial applications. It is essential to ensure that image content remains intact as they have used for many sensitive purposes such as police investigations, medical applications, military uses, etc. Image watermarking techniques can be used to verify image authentication by embedding a watermark using a fragile watermarking technique. Any efforts to attack/tamper with the watermarked data by an adversary will destroy the watermark, i.e., the watermark symbol cannot be recovered; otherwise, the original watermark can be extracted.
3. Broadcast monitoring: It is used to monitor the broadcasting stream and to identify any unauthorized broadcast material. This can be useful for production companies to prevent unauthorized broadcasting activities.
4. Fingerprinting: It is useful for monitoring or tracing illegally distributed copies of the data. When customers purchase digital documents, it is individually marked with the watermark, which is unique to the customer. If it is misused, then the owner can discover who created the illegal copy.

1.3 Objective of the thesis

The main concern of the thesis is to develop secure watermarking algorithms for copyright protection and authentication of digital images and audio signals. Here, in particular, we consider the image as the watermark message. The objectives of this thesis are listed as follows.

Proposed some image and audio watermarking techniques, where the methods

- should be secure,
- should provide an acceptable quality of the watermarked data,
- should be used as a tool for copyright protection,
- should be considered as a means to verify the authenticity of the data,
- should be blind,
- should have high embedding capacity.

1.4 Contribution of the thesis

It is well understood from the previous discussion that watermarking technology is a promising and effective tool for the copyright protection and authentication of multimedia data. In this thesis, we have proposed secure watermarking techniques for copyright protection and authentication of images and audio signals. The security of the watermarking system is ensured by the encryption method. In this work, images are considered as the watermark, where we adopt the image encryption method as the pre-processing step of the proposed watermarking techniques. The contributions of this thesis are listed below.

1. In this thesis, we have proposed two secure image watermarking techniques for copyright protection and authentication of images.
 - (a) The first technique is binary image watermarking based on Block Truncation Coding with Pattern Fitting (BTC-PF). This method is semi-blind, and the perceptual quality of the watermarked image is satisfactory. The same embedding method is applicable for both copyright protection, and to verify authenticity. For copyright protection, the watermark image is encrypted by a robust image encryption method. To verify the authenticity, the watermark image is encrypted by a fragile image encryption method.
 - (b) The second technique is grayscale image watermarking based on SVD where a grayscale image is embedded as a watermark. The method is blind, secure, has good embedding capacity, and provides satisfactory quality of watermarked images. Moreover, the method is also applicable for copyright protection and to verify the authenticity of confidential data. Like previously, depending on the requirement the watermark image needs to be encrypted either by robust image encryption method or by fragile image encryption method.
2. We have proposed a secure watermarking algorithm for audio signals that is applicable for copyright protection and authentication. This algorithm uses singular value decomposition and quantization of the largest singular value to embed a binary image into the audio signals. This method is blind and secure. The embedding capacity and quality of the watermarked audio are also good. The proposed audio watermarking algorithm is also applicable for copyright protection and authentication checking of the digital data. As per the requirement, the

watermark image will be encrypted either by a robust image encryption method or by a fragile image encryption method.

3. To make the watermarking system secure, watermark images must be encrypted before embedding. Here, we have proposed an image encryption method based on the Fibonacci Transformation and Tribonacci Transformation (FTTIE), which is suitable for both binary image and grayscale image. The proposed FTTIE method is a robust image encryption method. This encryption technique can be used to encrypt the watermark image for copyright protection. We further extend the robust FTTIE method as $\text{FTTIE}_{\text{ext}}$ to accomplish fragile image encryption. The watermark image is encrypted by $\text{FTTIE}_{\text{ext}}$ to achieve the fragility of the watermarking method and will be applicable for the authentication of data.

All the implementation of this thesis is done using MATLAB Version: 9.12.0.1884302 (R2022a) in the platform of Microsoft Windows 10 Pro Version 10.0 with the system has Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz 4.00 GB (3.88 GB usable).

1.5 Organization of the thesis

In this thesis, we have designed secure watermarking algorithms for copyright protection and authentication of digital images and audio signals. For this, we have proposed methods for image watermarking, audio watermarking, and image encryption. Chapter 2 presents the image watermarking techniques, a binary image watermarking system, and a grayscale image watermarking system. The audio watermarking technique is described in Chapter 3. The Chapter 4 has focused on the image encryption technique, where a robust image encryption method (FTTIE) is presented, and then the method is extended to $\text{FTTIE}_{\text{ext}}$, which is a fragile image encryption method. The summary of the contributions of the thesis with the possible direction of the future scope of research is given in Chapter 5.

Chapter 2

Secure Watermarking Techniques for Copyright Protection and Authentication of Images

2.1 Introduction

We have already studied the importance of the multimedia data at the present age. Image is an important source of information and crucial for different applications. In this chapter, we have focused on image watermarking techniques to protect the copyright and verify the authenticity of the images. Here, we have proposed two types of image watermarking techniques: i) binary image watermarking (i.e., watermark image is binary); ii) grayscale image watermarking (i.e., watermark image is grayscale) and host images are all grayscale image. The proposed binary watermarking method is semi-blind, and the grayscale watermarking method is blind. The proposed methods are robust against different attacks. In this work, the watermark images are encrypted before embedding. This encryption ensures the security of the proposed techniques. Also, helps to achieve a robust/fragile watermarking system. According to the requirement, the watermarked image is encrypted by robust or fragile image encryption to protect the copyright information or to verify the authenticity of the images, respectively.

In binary image watermarking, the host image is encoded by the BTC-PF method [74, 75] concerning the watermark image. In the BTC-PF method, a multi-level patternbook is considered, and for each image block, the method finds the best-fit pattern from the patternbook. Each block is represented by the index of the best-fit pattern

from the patternbook and corresponding grayvalues. Simultaneously, two different patternbooks may be used to encode an image; depending on the certain conditions a patternbook will be selected to encode a block. Sometimes, based on the situation/application we need to modify the value of the block at the cost of quality of the reconstructed image. The above ideas have been considered to embed the binary watermark into a grayscale image. In the proposed method, we consider one patternbook and partition it into two sub-patternbooks, and the partition remains fixed for the entire experiment. Here, depending on the bit value of the watermark image a sub-patternbook is selected to encode the current block of the host image. This method is robust and semi-blind.

In the grayscale image watermarking method, the host image is divided into sub-images by the sub-sampling method, and the co-located blocks of the sub-images are decomposed by the SVD method. Here, we consider grayscale images as the watermark image, and before embedding, the watermark image is normalized into $[0, 1]$. In the embedding process, the largest singular value of the blocks is modified according to the value of the current pixel of the normalized watermark image with a certain embedding strength. This proposed method is robust and blind.

The rest of the chapter is organized as follows. A review of the existing image watermarking techniques is discussed in Section 2.2. In Section 2.3, the background of the proposed watermarking techniques is presented. The proposed binary image watermarking technique is presented in Section 2.4. The proposed grayscale image watermarking technique is described in Section 2.5. The conclusions of the chapter are discussed in Section 2.6.

2.2 Literature review

Watermarking is an important technique to provide the facilities for copyright protection and authentication of multimedia data. In this section, we present the state-of-the-art image watermarking technique. In the last few decades, a significant number of algorithms have been designed for digital image watermarking [76, 77]. Generally, a digital image is statistically redundant in many features [78], such as pixel intensity, color, etc. The image watermarking methods have been designed both in the spatial domain and the transform domain. The literature survey of spatial domain-based methods and transform domain-based methods are given in the following two subsections.

2.2.1 Spatial domain-based watermarking

In the early stage, watermarking techniques have been developed in the spatial domain which are easily implemented but too fragile to resist varieties of attacks.

Spatial domain-based methods focus on directly modifying the pixel values of the image. Many researchers have been attracted to developed algorithms in the spatial domain due to their low complexity, ease of implementation, and faster execution. The most famous method is LSB in which watermark information is embedded into an image by substituting the LSB of the pixels with the watermark bit. The LSB method provides simple embedding and extraction [79–82]. Different types of the LSB method have been proposed to improve the trade-off between imperceptibility and robustness [83–85]. Although LSB started the development of watermarking, it is gradually ignored in the field of digital watermarking due to its unsatisfactory robustness against different attacks.

For faster transmission, some researchers, have proposed watermarking algorithms in spatial-compression domain [86–89]. In compression-based watermarking, the watermark is embedded by using the features of the host image. Thus, it is very difficult to remove or destroy the watermark without degradation of the watermarked image. For this reason, watermarking in the compression domain has gained popularity. In [87], a watermarking in the spatial domain has been developed based on Block Truncation Coding (BTC). In this work, the watermark is embedded by generating the ownership share with the help of the host image and watermark. To extract the watermark, the ownership share is to be used. It shows a high visual quality of the watermarked image, but robustness is not good against attacks. Lin et al. [86] have proposed a binary watermarking technique based on BTC, where watermark information is embedded into high and low means by modifying the bit according to the watermark bit as well as into the bitmap by minimum distortion algorithm. Both the BTC-based schemes provide good visual quality of the watermarked image but watermark extraction is not robust against many attacks. This problem is addressed in [88], where order dither block truncation coding (ODBTC) is used to embed the watermark. This technique uses the void-and-cluster method with BTC and ordered dithering to embed watermark information. The visual quality of the watermarked image is improved as well as robustness is also improved compared to the purely BTC-based method. In [89], a novel image watermarking technique has been proposed based on BTC. The watermark bit is embedded based on the number of 1's in the bitmap is odd or even by modifying at most three pixels in the original image block so that the number of 1's (or

0's) in the new bit-plane to be even for watermark bit '0' or to be odd for watermark bit '1'.

2.2.2 Transform domain-based watermarking

The methods described in the spatial domain are simple, and easy to understand, but not robust against many attacks. For this reason, lots of techniques have been designed based on transform domains such as Discrete Cosine Transformation (DCT), Discrete Wavelet Transformation (DWT), Discrete Fourier Transformation (DFT), and so on. In these methods, the host image is transformed according to different mathematical basis functions and then the watermark information is inserted by modifying selected coefficients. The visual quality of the watermarked image can be maintained because the characteristics of the HVS [90, 91] are captured by the spectral coefficients and the information can be spread out to the entire image.

The DCT separates an image into its equivalent frequency coefficients. After DCT transformation, most of the energy of the image is concentrated in the upper-left corner coefficient, i.e., DC coefficient, the values of the coefficients from left-up to right-bottom decrease gradually. These characteristics make DCT the basis of image compression. At the beginning of the study on DCT-based watermarking, most of the methods embedded the watermark into DCT coefficients directly by using a fixed embedding strength for every block [63, 92]. There are some DCT-based image watermarking schemes, such as Dither Modulation (DM) [93], Quantization Index Modulation (QIM) [94], Differential Modulation (DM) [95] etc. Although the performance of DCT-based methods is much better than spatial-based methods, it is still possible that the performance can be further improved by considering visual models [96–100].

Discrete Wavelet Transform (DWT) has gained importance in the field of image watermarking due to its multi-resolution description of the image. The DWT decomposes the image into four sub-bands: LL, LH, HL, and HH. The LL sub-band grasps the maximum energy of the image and the other three sub-bands LH, HL, and HH contain the details of the image [101, 102]. The image could be decomposed into any number of levels by selecting any sub-bands. The more detailed information contained in the corresponding sub-bands increases the level of decomposition. Based on the level of decomposition and selection of sub-bands, diverse watermarking techniques have been designed and obtain significant performance [103–108]. Also, Diverse image watermarking algorithms have been designed based on Discrete Fourier Transform (DFT) [109, 110] to obtain better imperceptibility.

Due to the attractive properties of SVD, various watermarking techniques have been proposed based on SVD to obtain a trade-off of the performance between imperceptibility and robustness [11]. The slight modification of the singular values does not affect the visual quality of the image [111]. It has been found that an SVD-based watermarking algorithm combined with other transformation methods like DCT, DWT, DFT, etc to obtain improved visual quality and robustness [111–113].

Few researchers have designed grayscale watermarking techniques where they convert the watermark image (grayscale) into bit stream before embedding [114–116]. However, these methods have a serious disadvantage. If the pixel value has a one-bit error, the pixel value will be changed. For instance, the original pixel value is 8 (00001000), but the extracted pixel value is 138 (10001000) due to a change in MSB position. This will influence the visual quality of the watermark image and this problem is addressed in [117], which is a blind grayscale image watermarking algorithm, the experimental result shows the good visual quality of the image but, the robustness property can still be improved. Inspired by the above discussion, in this thesis, we have proposed a grayscale image watermarking technique in the transform domain.

2.3 Background of the watermarking methods

We have discussed earlier that the proposed techniques are based on BTC-PF and SVD methods. Here, we present the BTC-PF method and SVD method in the following two subsections.

2.3.1 BTC-PF method

The BTC-PF [74, 75, 118] is a lossy spatial domain-based image compression method that uses a t -level quantizer to quantize a local region of the image. This method encompasses a hybrid combination of BTC [119] and VQ [120] method. In this method, to encode an image, it is divided into blocks of size $w \times w$ ($w = 2^h$) and each block is quantized by t grayvalues. In BTC-PF, a collection of $w \times w$ patterns is considered and each pattern has t -levels. Let us refer to this collection of patterns as ‘patternbook’ (PB) and assume that there are G patterns in the PB , i.e., $PB = \{P_h : 0 \leq h \leq G-1\}$. The quantization levels of an image block are determined for the pattern, which gives the minimum error (i.e., best-fit pattern). To select the best-fit pattern, each pattern is fit to the block B_i , and an error is computed for this. The error (E_h) between the

block B_i to a pattern P_h is computed as given in Eq. (2.1).

$$\begin{aligned} E_h &= \sum_{r=0}^{t-1} e_r \\ e_r &= \sum_{P_h(x,y)=r} (B_i(x,y) - m_r)^2 \\ m_r &= \frac{1}{|n_r|} \sum_{P_h(x,y)=r} B_i(x,y) \end{aligned} \quad (2.1)$$

where n_r is the number of positions with level r in the pattern P_h . The pattern with minimum error is considered the best-fit pattern and the index of the best-fit pattern is

$$q = \arg\min\{E_{h,0 \leq h \leq G-1}\}$$

The block B_i is reconstructed as B'_i where $B'_i(x,y) = m_r$ if $P_q(x,y) = r$, $0 \leq r < t$. An example of the BTC-PF method with 2-level patterns for a 4×4 block is illustrated in Fig. 2.1.

2.3.2 Singular value decomposition

There are enormous applications of SVD [121, 122] in the field of image processing. The image can be viewed as a matrix of non-negative scalar elements. SVD decomposes an image A of size $n \times n$ as a product of three matrices: U , S , and V , where U and V are orthogonal matrices. The matrix $U=[u_1, u_2, \dots, u_n]$ is the collection of left singular vectors, the matrix $V=[v_1, v_2, \dots, v_n]$ is the collection of right singular vectors, and the matrix S is a diagonal matrix whose elements are called singular values. The singular values are denoted as $\alpha_i, i = 1, \dots, n$. If the rank of A is r ($\leq n$), elements of S satisfy the conditions $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_r > \underline{\alpha_{r+1} = \alpha_{r+2} = \dots = \alpha_n = 0}$. The formula for SVD of A is given in Eq.(2.2).

$$A = USV^T = \begin{bmatrix} u_1 & \dots & u_n \end{bmatrix} \times \begin{bmatrix} \alpha_1 & & \\ & \ddots & \\ & & \alpha_n \end{bmatrix} \times \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \quad (2.2)$$

The SVD of an image has been analyzed in [123]. The singular values represent the luminance of the image and the corresponding pair of singular vectors represent the geometry of the image.

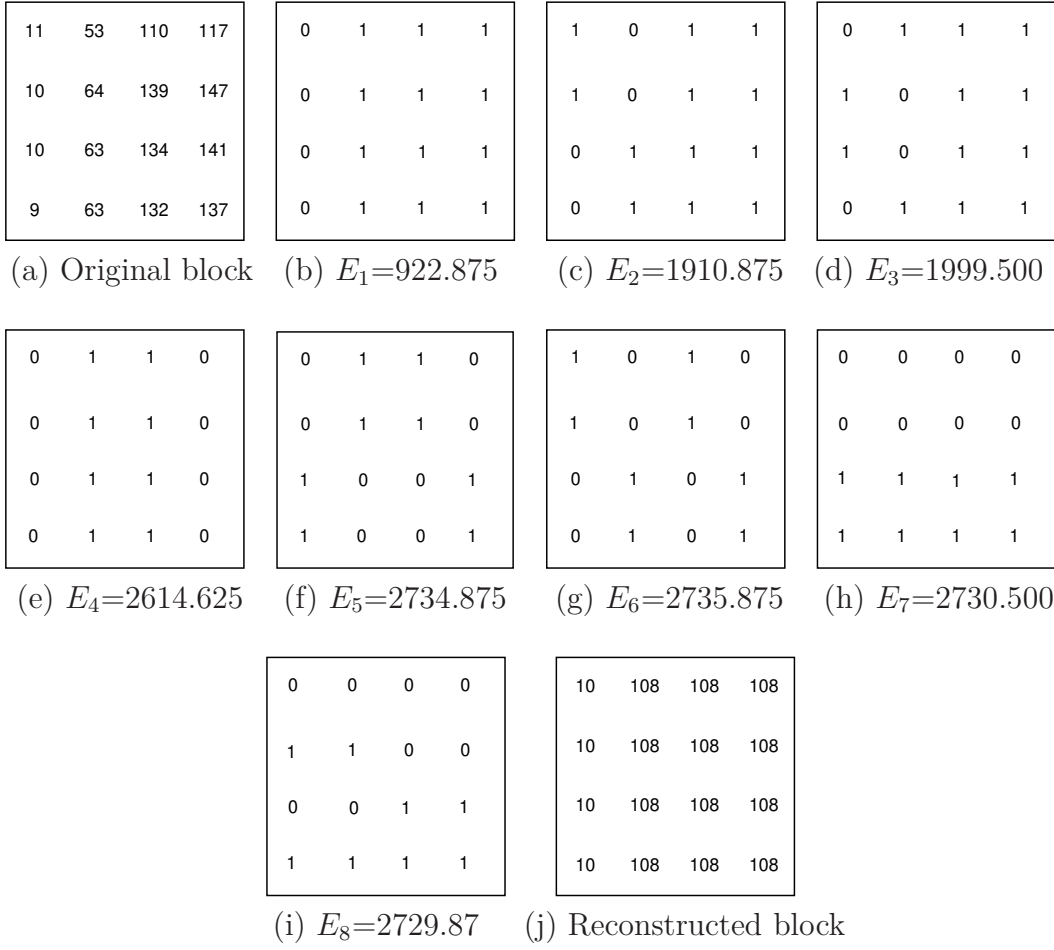


Figure 2.1: Example of BTC-PF method with one 4×4 block and eight 2-level patterns: (a) original block, (b)-(i) represent the error due to encoding of the block for the corresponding pattern in the figure, (j) reconstructed block, the best-fit pattern is P_1 (as E_1 is minimum)

Now, we illustrate the SVD with an example. We take a matrix 'A' of size 2×2 and the matrix to be factorized as the three matrices using SVD. The matrices U and V are computed from the eigenvectors of AA^T and $A^T A$ respectively. The matrix S is computed from the eigenvalues of AA^T or $A^T A$. The computation of the three matrices S , U , and V are as below.

$$A = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} \quad (2.3)$$

$$AA^T = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} \begin{bmatrix} 4 & 3 \\ 0 & -5 \end{bmatrix} = \begin{bmatrix} 16 & 12 \\ 12 & 34 \end{bmatrix} \quad (2.4)$$

$$A^T A = \begin{bmatrix} 4 & 3 \\ 0 & -5 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} = \begin{bmatrix} 25 & -15 \\ -15 & 25 \end{bmatrix} \quad (2.5)$$

The characteristic equation to compute the eigenvalues of AA^T by equating the determinant of $(AA^T - \lambda I) = 0$, i.e., $(16-\lambda)(34-\lambda) - 144 = \lambda^2 - 50\lambda + 400 = 0$. Therefore, the first and second eigenvalues of AA^T are $\lambda_1 = 40$ and $\lambda_2 = 10$ respectively. So, the first singular value (α_1) is $\sqrt{40}$ and second singular value (α_2) is $\sqrt{10}$. Now, the matrix S is given below.

$$S = \begin{bmatrix} \sqrt{40} & 0 \\ 0 & \sqrt{10} \end{bmatrix} \quad (2.6)$$

Now, we compute the left singular vectors (the column of 'U') by finding an orthonormal set of eigenvectors of AA^T . The calculations are as follows.

$$AA^T - \lambda I = \begin{bmatrix} 16 & 12 \\ 12 & 34 \end{bmatrix} - \begin{bmatrix} 40 & 0 \\ 0 & 40 \end{bmatrix} = \begin{bmatrix} -24 & 12 \\ 12 & -6 \end{bmatrix} \quad (2.7)$$

The matrix in 2.7 is now reduced by a series of elementary transformations and finally obtained the matrix as below.

$$\begin{bmatrix} -2 & 1 \\ 0 & 0 \end{bmatrix} \quad (2.8)$$

Now, we compute the eigenvector for eigenvalue 40 as below.

$$\begin{bmatrix} -2 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.9)$$

From 2.9, we have $-2u_1 + u_2 = 0$ and we get, $u_1 = -1$ and $u_2 = -2$. The eigenvector is given below.

$$\begin{bmatrix} -1 \\ -2 \end{bmatrix} \quad (2.10)$$

The vector in 2.10 is now converted into an orthonormal vector which is given below.

$$\begin{bmatrix} -1/\sqrt{5} \\ -2/\sqrt{5} \end{bmatrix} \quad (2.11)$$

Similarly, for the eigenvalue $\lambda_1 = 10$, we have obtained the eigenvector which is shown in 2.12.

$$\begin{bmatrix} 2 \\ -1 \end{bmatrix} \quad (2.12)$$

The vector in 2.12 is now converted into an orthonormal vector which is given below.

$$\begin{bmatrix} 2/\sqrt{5} \\ -1/\sqrt{5} \end{bmatrix} \quad (2.13)$$

Finally, we obtain the 'U' matrix which is given in 2.14.

$$U = \begin{bmatrix} -1/\sqrt{5} & 2/\sqrt{5} \\ -2/\sqrt{5} & -1/\sqrt{5} \end{bmatrix} \quad (2.14)$$

The right singular matrix 'V' is computed (like the 'U' matrix) from the eigenvalues of AA^T and the matrix 'V' is given in 2.15.

$$V = \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \quad (2.15)$$

The original matrix 'A' can be obtained by inverse transformation which is shown in 2.16.

$$USV^T = \begin{bmatrix} -1/\sqrt{5} & 2/\sqrt{5} \\ -2/\sqrt{5} & -1/\sqrt{5} \end{bmatrix} \begin{bmatrix} \sqrt{40} & 0 \\ 0 & \sqrt{10} \end{bmatrix} \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} \quad (2.16)$$

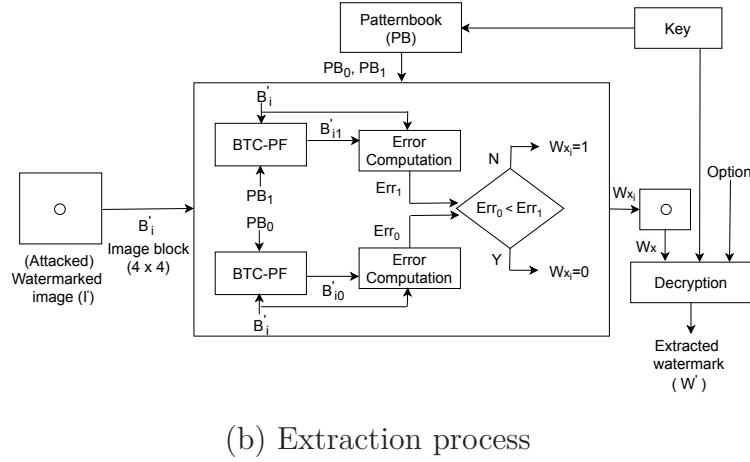
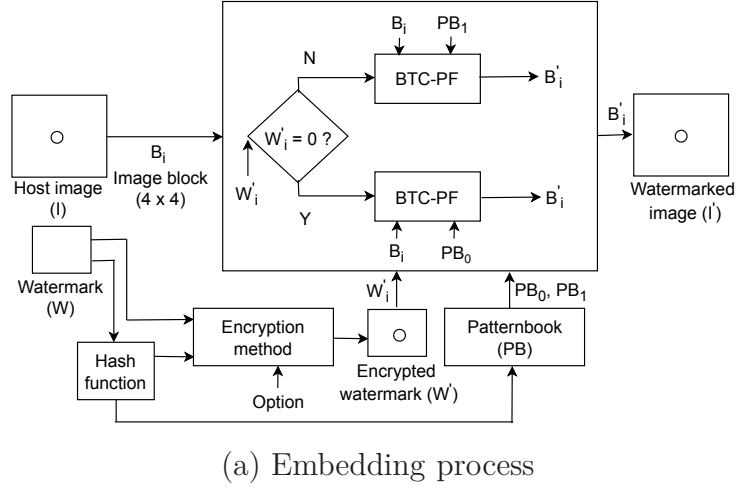


Figure 2.2: Block diagram of the proposed BWBTC-PF method: (a) Embedding process, (b) Extraction process

2.4 Proposed Binary Image Watermarking Method

Here, we are going to discuss the proposed binary watermarking technique. The present method has adopted the BTC-PF method to embed the watermark bits. BTC-PF method is a lossy image compression technique due to which the quality of the watermarked image is reduced. Though quality is reduced, still it is good enough for certain applications like news broadcasting, military applications, etc. [113, 124, 125].

The BTC-PF method encodes an image with respect to a multilevel patternbook. In this work, we consider a 2-level patternbook (PB) of 128 patterns, and each pattern is of size 4×4 . To define the patternbook, we follow the method as discussed in [75]. Here, the host image is divided into 4×4 blocks (as used in [75]). So, the size of the watermark image is $\frac{1}{16}$ -th of the host image to ensure that for each block there is a

Algorithm 1 : BWBTC-PF-EmbeddingProcess($I, W, key, 'opt', PB, I'$)

Input: Host image (I), Watermark image (W), hash value of W (key), Option ('opt'), PatternBook (PB)

Output: Watermarked image (I')

Step 1 I is divided into blocks of size 4×4 .

Step 2 $W' \leftarrow \text{Encrypt}(W, key, 'opt')$
Step 3 $\{PB_0, PB_1\} \leftarrow \text{Partition}(PB)$
Step 4 For each block B_i of I

 If ($W'_i == 0$) then $B'_i \leftarrow \text{BTC-PF}(B_i, PB_0)$

 Else $B'_i \leftarrow \text{BTC-PF}(B_i, PB_1)$
Step 5 Return I' .

watermark bit and vice-versa. We consider the size of the host image to be $M \times N$ and that of the watermark image is $\frac{M}{4} \times \frac{N}{4}$. Here, to embed the watermark we have applied a simple logic.

In the embedding process, first, we partition the patternbook, PB , into PB_0 and PB_1 , each one having 64 patterns and we also ensure that in the patternbook, no two patterns are the complement of each other. This partition is fixed for the entire embedding process, i.e., the same partition will be used to encode the entire host image. To embed each watermark bit, depending on the watermark bit either PB_0 or PB_1 will be selected and the block of the host image will be encoded concerning the selected sub-book. We have experimentally observed, that for any random partition of the patternbook during embedding more or less similar performance, and therefore, in our implementation, we partition PB into PB_0 and PB_1 and this partition is fixed for the entire experiment. Strictly speaking, in the proposed embedding method, the watermark bit is not embedded, instead, we encode the image block by the BTC-PF method with a sub-book selected depending on the watermark bit, i.e., the host image is encoded by the BTC-PF method with respect to the watermark image. Hereafter, we refer to the proposed method as Binary image Watermarking using BTC-PF (WBTC-PF) method. At the time of watermark extraction, the (attacked) watermarked image is further encoded by the BTC-PF method using PB . The best pattern of a block belongs to either PB_0 or PB_1 and accordingly, the watermark bit is fixed. During extraction, the patternbook PB is also used so the proposed

Algorithm 2 : BWBTC-PF-ExtractionProcess($I', key, 'opt', PB, W'$)

Input: Watermarked image (I'), Secret key (key), Option ('opt'), Patternbook (PB)

Output: Extracted watermark (W')

Step 1 Divide I' into blocks of size 4×4

Step 2 $\{PB_0, PB_1\} \leftarrow \text{Partition}(PB)$

Step 3 For each block B'_i do

Encode B'_i with respect to PB_0 , $Err_0 \leftarrow$ 'best fit error'

Encode B'_i with respect to PB_1 , $Err_1 \leftarrow$ 'best fit error'

If($Err_0 < Err_1$) then $W_{x_i} = 0$

Else $W_{x_i} = 1$

Step 4 $W' \leftarrow \text{Decrypt}(W_x, key, 'opt')$

step 5 Return W'

method is a semi-blind watermarking method. In the embedding process, instead of the original watermark image, we embed an encrypted version of the watermark as discussed earlier. Figure 2.2 shows the block diagram of the proposed BWBTC-PF system. The algorithmic flow of the watermark embedding process and extraction process are given in **Algorithm 1: BWBTC-PF-EmbeddingProcess()** and **Algorithm 2: BWBTC-PF-ExtractionProcess()**, respectively. In this experiment, the secret key (key) is the hash value of the watermark image (W), and the option ' opt ' represents the mode of encryption FTTIE or FTTIE_{ext}.

2.4.1 Experimental Results of BWBTC-PF

In this section, the proposed method is evaluated concerning different features to establish the effectiveness of the proposed technique.

To evaluate the effectiveness of the BWBTC-PF method, its performance is compared to existing watermarking techniques (Lin et. al. [86], Makbol et. al. [126], and Dhani et. al. [127]). For comparison purposes, we have implemented these SoA methods to the best of our understanding.

Lin et. al. [86] have proposed a watermarking scheme based on BTC to embed the watermark. In BTC, an image block is encoded by two quantization levels, low mean and high mean, and one bitmap. Lin et. al. proposed three different techniques

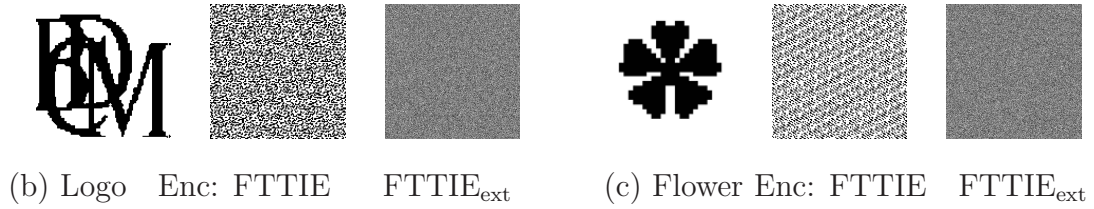
(a) Host images (Grayscale) each of size 512×512 (d) Cameraman Enc: FTTIE FTTIE_{ext} (e) Baboon Enc: FTTIE FTTIE_{ext}

Figure 2.3: Test images (including both host images and watermark images) used in these experiments.

to embed the watermark: i) by modifying the low mean of each block (BTC_L); ii) modifying the high mean of the blocks (BTC_H); iii) modifying the bitmap (BTC_B).

Makbol et. al. [126] have proposed a binary watermarking scheme in the transform domain. The DWT and SVD have been used to embed the watermark bits into significant image blocks selected by human visual system characteristics. During the embedding process, several coefficients of the ‘U’ matrix are modified according to the watermark bit.

Dhani et. al. [127] have proposed a binary watermarking method in the transform domain. In this method, discrete cosine transform is applied on the host image block

of size 8×8 . The watermark is embedded by modifying the selected middle-frequency coefficients of the image block.

In this experiment, a set of eight grayscale images: ‘Lena’, ‘Tiffany’, ‘Zelda’, ‘Splash’, ‘Airplane’, ‘Boat’, ‘Couple’ and ‘Man’ each of size 512×512 have been used as host images (shown in Fig. 2.3(a)) whereas two binary images ‘Logo’ and ‘Flower’ of size 128×128 are used as the watermark (shown in Fig. 2.3(b)-(c) along with their encrypted version). Here, we evaluate the applicability of the proposed method using different features (which are discussed in Section 1.2.2).

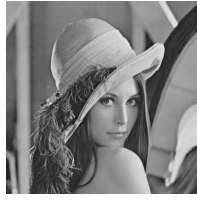
Imperceptibility and Capacity

Imperceptibility is one important property to evaluate the effectiveness of the watermarking method. Imperceptible measures the quality of the watermarked image. The result of the proposed method shows that the watermarked images are quite good (see Fig. 2.4). Here, we report the average quality of the watermarked images when the watermark images ‘Logo’ and ‘Flower’ are encrypted by FTTIE and FTTIE_{ext}.

From Fig. 2.4, we have observed that the PSNR value is greater than 30dB and SSIM is also above 0.97. Therefore, the proposed method is good enough for general applications [75]. Table 2.1 shows a comparison of the imperceptibility feature of the proposed method with SoA methods. For better understanding, we also report the payload (ratio between the size of the watermark and host image) of the individual method. Though the quality given by Dhani et. al. [127] is superior to the other methods; however, its payload is 25% of the payload of other methods. From the table, we can say that our proposed scheme performs better compared to the existing methods.

Table 2.1: Comparative study of the payload and the quality of the watermarked images given by different methods.

Watermarking Methods	Payload	Watermark ‘Logo’		Watermark ‘Flower’	
		PSNR	SSIM	PSNR	SSIM
BTC_L [86]	$\frac{128 \times 128}{512 \times 512} = 0.0625$	30.39	0.9657	30.48	0.9655
BTC_H [86]	$\frac{128 \times 128}{512 \times 512} = 0.0625$	30.52	0.9667	30.61	0.9685
BTC_B [86]	$\frac{128 \times 128}{512 \times 512} = 0.0625$	31.15	0.9739	31.25	0.9774
Makbol et. al. [126]	$\frac{128 \times 128}{512 \times 512} = 0.0625$	30.48	0.9543	30.21	0.9517
Dhani et. al. [127]	$\frac{64 \times 64}{512 \times 512} = 0.0156$	32.09	0.9783	32.05	0.9775
BWBTC-PF [128]	$\frac{128 \times 128}{512 \times 512} = 0.0625$	31.82	0.9792	32.15	0.9827



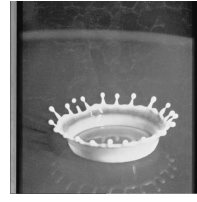
Lena
PSNR=31.87
SSIM=0.9817



Tiffany
PSNR=31.47
SSIM=0.9806



Zelda
PSNR=35.44
SSIM=0.9856



Splash
PSNR=32.62
SSIM=0.9835



Airplane
PSNR=31.90
SSIM=0.9819



Boat
PSNR=30.12
SSIM=0.9715

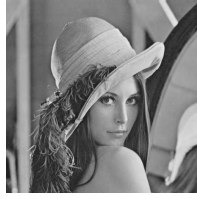


Couple
PSNR=31.10
SSIM=0.9775



Man
PSNR=30.01
SSIM=0.9712

(a) Watermarked images with 'Logo' as watermark



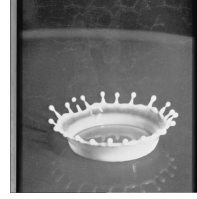
Lena
PSNR=32.19
SSIM=0.9833



Tiffany
PSNR=31.54
SSIM=0.9820



Zelda
PSNR=35.65
SSIM=0.9868



Splash
PSNR=32.78
SSIM=0.9848



Airplane
PSNR=31.30
SSIM=0.9833



Boat
PSNR=31.03
SSIM=0.9765



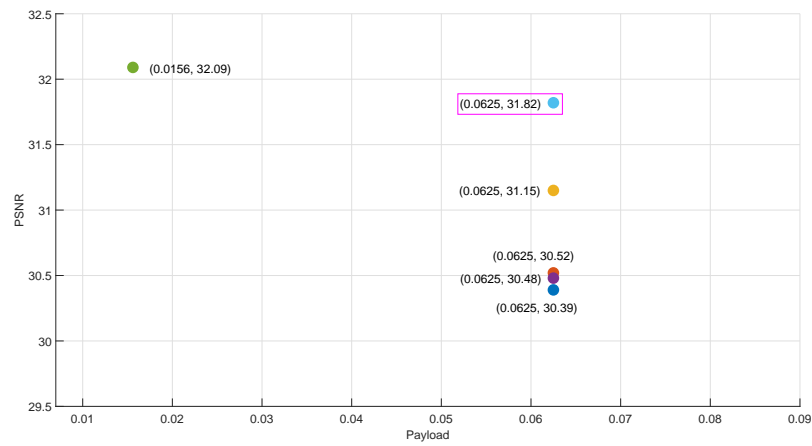
Couple
PSNR=31.27
SSIM=0.9805



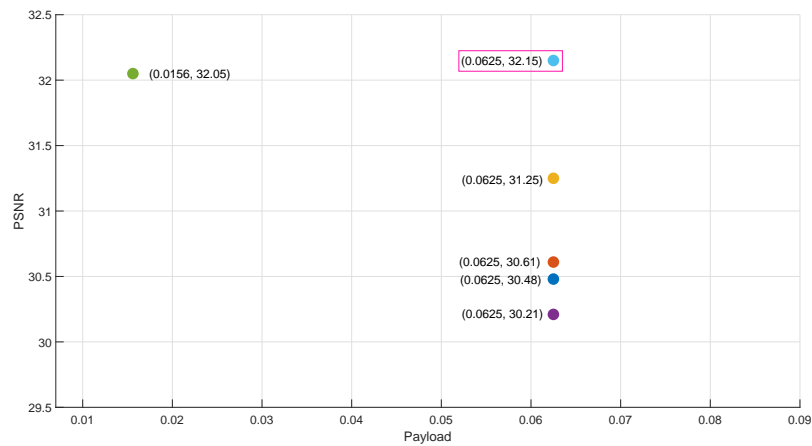
Man
PSNR=31.42
SSIM=0.9842

(b) Watermarked images with 'Flower' as watermark

Figure 2.4: Quality of the watermarked images of BWBTC-PF method



(a) Watermark as ‘Logo’



(b) Watermark as ‘Flower’

Figure 2.5: Payload vs. PSNR of different methods

The efficiency of a watermark system can not only be evaluated by the quality of the watermarked data. It is a fact that if the payload is low, then the quality of the watermarked data becomes high. So, to judge a method, we also need to consider payload. Hence, to test the overhead of a process, it is necessary to consider both the payload and quality of the watermarked data. To compare the overhead of the proposed method with SoA methods, we show the distribution of the points (payload vs. quality), and obviously, a point at the extreme right-top portion indicates its superiority. For this purpose, we show the overhead of the different methods in Fig. 2.5. From this figure, we observe that the point for our method is located at right-top position for both watermark symbols. Therefore, the proposed BWBTC-PF method is better than the SoA methods.

Time complexity

The complexity of the watermark embedding is another important criterion for judging the applicability of a watermark system. The time complexity of a method depends on the encoding method by which the watermark symbol is embedded. In our proposed method, we follow the BTC-PF method to embed the binary watermark. The BTC-PF method follows the same line as the vector quantization (VQ) method. The major disadvantage of the VQ-based method is the complexity of the encoding method. The proposed BWBTC-PF method also needs more time. Here, for a block, to determine the best pattern, we need to compare with all the available patterns. So, the execution time of the BWBTC-PF method is very high. A comparative study of the execution time of the proposed method along with the SoA methods is reported in Table 2.2. The table shows the execution time of the proposed method, which is very high, as expected, compared to the other methods. Since the watermark image will be embedded just once, and we need to extract the watermark many times for ownership detection or authentication checking, we may think the embedding process is offline. In this regard, more embedding time hardly matters for the intended applications.

Performance against attacks

This section examines the efficiency of the proposed method against various attacks. Here, we consider three cases in terms of the watermark image: i) plain watermark image (not encrypted), ii) encrypted watermark image using FTTIE, and iii) encrypted watermark image using FTTIE_{ext} method. To measure the efficiency of the method, we consider some attacks, on the watermarked image, given in Table 2.3. Due to space

Table 2.2: Average execution time (seconds) of the proposed BWBTC-PF method and SoA methods

Methods	Execution time	
	Watermark ‘Logo’	Watermark ‘Flower’
BTC_L [86]	11.630278	11.548927
BTC_H [86]	11.555244	11.522836
BTC_B [86]	4.138235	4.137638
Makbol et. al. [126]	14.022220	13.974880
Dhani et. al. [127]	6.497678	6.490971
BWBTC-PF [128]	499.620769	498.163776

issues, we include the output on two host images, say, ‘Airplane’ and ‘Lena’ for both the watermark ‘Logo’ and ‘Flower’. Table 2.4 - Table 2.7 provide the performance of the proposed method under various attacks. In these tables, we also include the NC (in %) and SSIM value of the extracted watermark. It has been cleared from the output of the first and second columns of the tables that the extracted watermarks are visually identifiable and the quality of the extracted watermark is also acceptable even after attacks on watermarked images. Also, note that the method under the first column is not secure as the watermark image is not encrypted. So, the result provided in the second column shows that the BWBTC-PF method is robust and secure when the watermark images are encrypted by the FTTIE method and can be used to prove the copyright of the image. On the other hand, the output in the third column of the tables shows that the extracted watermark is like a noisy image under the attacks. So, this output shows that the BWBTC-PF method can be used as a fragile watermarking technique if the watermark is encrypted by the $FTTIE_{ext}$ method, and as a result, our proposed technique may be used to verify the authenticity of the images.

The performance of the proposed watermarking technique BWBTC-PF is also compared with some existing techniques under various attacks. The average NC (%) and SSIM of the extracted watermark under attacks are shown in Table 2.8 and Table 2.9. In these tables, we include the average performance of the robust BWBTC-PF method (when the watermark images are encrypted by the FTTIE method). From these two tables, we observe that the proposed method BWBTC-PF performs better compared to SoA methods under the attacks.

Table 2.3: Different attacks on watermarked image

Attack name	Description	Value used
No attack (NA)	The watermarked image without any attack	—
Salt & pepper noise (SP)	It adds salt & pepper noise (on and off pixels) to the image with given noise density ‘d’	d=0.05
Cropping (CP)	Cropping attack crops the image by setting pixel values to zeros (black) or 255 (white). The amount of cropping and position of cropping are determined by the user.	Setting pixels as 255, upper 50%
Blurring (BR)	Blurring is one important operation on the image. A kernel with disk size ‘s’ is used for blurring an image.	s=0.6
Gaussian filter (GF)	Gaussian filter is a low pass filter used for reducing noise. The filter size is 3×3 with standard deviation ‘v’	v=0.5
Sharpening (SRN)	This attack sharpens the image using name-value pairs to control aspects of unsharpened masking. Two parameters ‘Amount’ and ‘Radius’ are used to process the image.	Amount=1 and Radius=0.8.
Rotation (RT)	Rotation attacks rotate the image either clockwise or anti-clockwise. The angle of rotation is determined by the user.	clockwise, angle= 5°
Histogram equalization (HE)	Contrast enhancement of the image by transforming the values in an intensity image so that the histogram of the output image approximately matches a specified histogram	[0, 255]
JPEG compression (JPEG:X)	This attack comprises the image with varying quality factor ‘X’. This value is determined by the user	X=20, 40, 60, 80

2.5 Proposed Grayscale Image Watermarking Method

In the previous section, we have presented a binary watermarking technique that is semi-blind in nature. We also observe that the BWBTC-PF method can be used in copyright protection (as the method is robust) and authentication (since the method can be transformed as fragile).

In this section, we are going to present a grayscale watermarking technique. Usually, two approaches are used in grayscale watermarking:

1. When the watermark is embedded, a scaled value of the pixel intensity is considered and directly used to modify the host image.

Table 2.4: Robustness and Fragility of BWBTC-PF method: host image ‘Airplane’, watermark image ‘Logo’

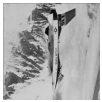





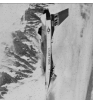
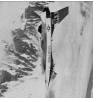
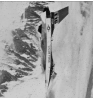
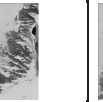
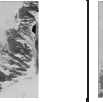
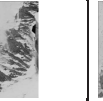
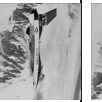
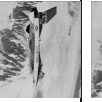
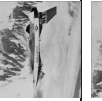
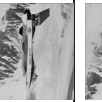
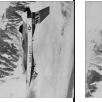
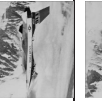
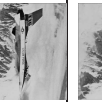
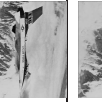
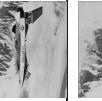
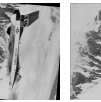
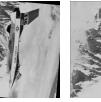
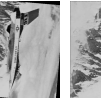
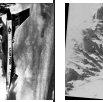
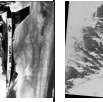
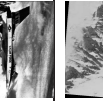
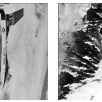
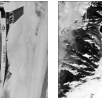
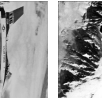
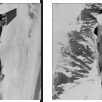
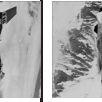
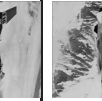
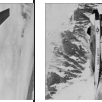
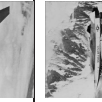
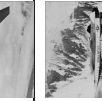
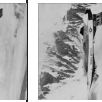
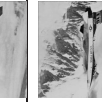
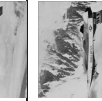
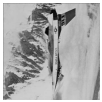



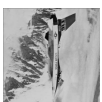


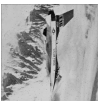

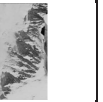
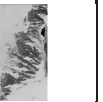

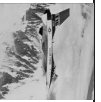
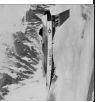

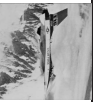
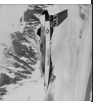

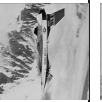
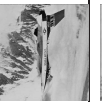
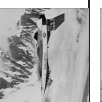
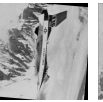
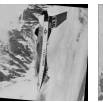
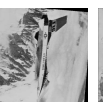
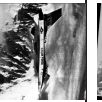
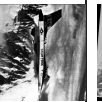
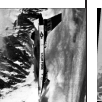
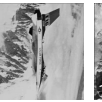
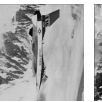
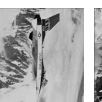
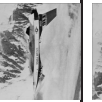
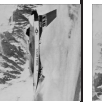
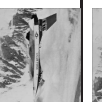
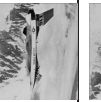
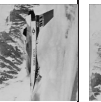
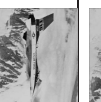
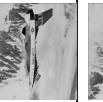
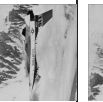
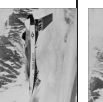
Attack	Plain watermark			Encryption by FTTE			Encryption by FTTE _{ext}		
NA	Marked 	NC ↓ Extract ↓ SSIM	100.0  1.000	Marked 	NC ↓ Extract ↓ SSIM	100.0  1.000	Marked 	NC ↓ Extract ↓ SSIM	100.0  1.000
SP		96.9665	0.7145		96.1130	0.7102		50.8606	0.0038
CP		73.2614	0.4185		72.1412	0.3949		51.0315	0.0595
BR		99.6704	0.9185		99.9448	0.9903		45.7627	0.0044
GF		96.8523	0.7669		96.1715	0.7631		51.6438	0.0021
SRN		96.1121	0.7176		97.3052	0.7016		51.8433	0.0044
RT		95.5181	0.8364		96.9852	0.8498		49.1882	0.0074
HE		98.3459	0.9154		98.6023	0.8129		50.1404	0.0046
JPEG(20%)		61.2061	0.0617		62.7915	0.0731		50.1812	0.0037
JPEG(40%)		64.9014	0.1478		65.6023	0.1585		51.8312	0.0038
JPEG(60%)		74.2193	0.2469		76.3451	0.2616		52.3451	0.0025
JPEG(80%)		87.2479	0.7828		88.1284	0.7921		51.0315	0.0034

Table 2.5: Robustness and Fragility of BWBTC-PF method: host image 'Lena', watermark image 'Logo'

Attack	Plain watermark				Encryption by FTTIE				Encryption by FTTIE _{ext}			
	Marked	NC	Extract	SSIM	Marked	NC	Extract	SSIM	Marked	NC	Extract	SSIM
NA		100.0		1.000		100.0		1.000		100.0		1.000
SP		96.8511		0.7123		96.9383		0.7169		50.9831		0.0039
CP		72.1675		0.4085		71.0784		0.3889		51.1012		0.0588
BR		99.3625		0.9459		99.9541		0.9979		46.0318		0.0047
GF		95.2458		0.7682		95.7019		0.7767		51.0218		0.0019
SRN		96.2315		0.6915		96.8842		0.6989		51.6401		0.0049
RT		96.6035		0.8201		96.4435		0.8158		50.1315		0.0043
HE		98.0613		0.9258		98.8098		0.9372		51.5139		0.0048
JPEG(20%)		56.8048		0.0416		59.8172		0.0409		49.6277		0.0029
JPEG(40%)		64.1701		0.1392		64.6927		0.1402		51.8433		0.0036
JPEG(60%)		72.1493		0.2355		73.9881		0.2492		52.8745		0.0020
JPEG(80%)		82.7965		0.6482		83.0679		0.6578		52.7232		0.0032

Table 2.6: Robustness and Fragility of BWBTC-PF method: host image ‘Airplane’, watermark image ‘Flower’

Attack	Plain watermark			Encryption by FTTIE			Encryption by FTTIE _{ext}		
NA	Marked 	NC ↓ Extract ↓ SSIM	100.0 	Marked 	NC ↓ Extract ↓ SSIM	100.0 	Marked 	NC ↓ Extract ↓ SSIM	100.0 
SP		97.4487	0.6475		97.2778	0.6387		53.1925	0.0077
CP		65.4114	0.4689		67.3157	0.3508		41.2450	0.0017
BR		99.4783	0.9765		99.9428	0.9834		52.5085	0.0081
GF		96.2341	0.5899		96.7238	0.5774		52.3936	0.0068
SRN		95.9473	0.6484		96.9238	0.6929		52.9541	0.0077
RT		93.7236	0.5964		95.0051	0.6022		50.9155	0.0074
HE		98.1445	0.7478		98.4375	0.7398		52.0264	0.0079
JPEG(20%)		61.6272	0.0618		60.3258	0.0542		52.2227	0.0080
JPEG(40%)		63.9382	0.1469		64.7926	0.1487		52.9722	0.0081
JPEG(60%)		72.3485	0.2431		73.8257	0.2557		51.5703	0.0076
JPEG(80%)		88.0310	0.7829		87.5483	0.7970		50.5745	0.0042

2. In other cases, the grayscale image is first converted into a bit-stream, and then the bit values are embedded into the host image.

Table 2.7: Robustness and Fragility of BWBTC-PF method: host image ‘Lena’, watermark image ‘Flower’

Attack	Plain watermark			Encryption by FTTE			Encryption by FTTE _{ext}		
	Marked	NC	Extract	Marked	NC	Extract	Marked	NC	Extract
NA		100.0	1.000		100.0	1.000		100.0	1.000
SP		96.8753	0.6861		97.5039	0.6917		52.2837	0.0078
CP		68.7256	0.5255		69.4239	0.4114		41.3452	0.0019
BR		99.5178	0.8931		99.6521	0.8807		52.5757	0.0078
GF		96.8750	0.5745		96.4111	0.5672		52.9917	0.0074
SRN		97.1628	0.7083		98.9126	0.7266		52.9358	0.0069
RT		93.8293	0.6126		95.8303	0.6313		51.1780	0.0070
HE		96.8994	0.6995		98.5769	0.7482		51.1414	0.0036
JPEG(20%)		61.2344	0.0566		61.4358	0.0530		51.9670	0.0069
JPEG(40%)		67.5704	0.1669		66.7404	0.1662		52.6343	0.0073
JPEG(60%)		76.4312	0.2862		75.3959	0.2773		51.7004	0.0038
JPEG(80%)		84.6514	0.6757		83.3960	0.6691		50.8168	0.0040

Table 2.8: Average NC (%) and SSIM of different techniques under attacks ‘Logo’ as watermark

Attack	Measures	BTC_L [86]	BTC_H [86]	BTC_B [86]	Makbol et. al. [126]	Dhani et. al. [127]	BWBTC-PF [128]
NA	NC (%)	100.00	100.00	100.00	100.00	100.00	100.00
	SSIM	1.000	1.000	1.000	11.000	1.000	1.000
SP	NC (%)	93.8713	94.8750	96.6112	76.9001	92.1013	96.8975
	SSIM	0.5202	0.5308	0.6153	0.2685	0.6125	0.7119
CP	NC (%)	70.7112	70.0413	71.1837	62.5263	70.0151	71.4563
	SSIM	0.2903	0.2750	0.3679	0.2127	0.3775	0.3899
BR	NC (%)	80.5450	82.4963	99.0702	75.6263	99.0800	99.8238
	SSIM	0.2410	0.2652	0.9022	0.3132	0.9449	0.9747
GF	NC (%)	63.3050	63.8775	89.2925	77.9488	93.7313	96.3125
	SSIM	0.0880	0.0975	0.3656	0.3578	0.7259	0.7771
SNR	NC (%)	50.4313	50.6250	71.4963	71.6862	96.9250	97.5763
	SSIM	0.0063	0.0064	0.1119	0.3246	0.6740	0.7167
RT	NC (%)	92.9850	93.9575	88.7525	71.1375	95.6713	96.0938
	SSIM	0.6824	0.6446	0.5272	0.2599	0.8109	0.8221
HE	NC (%)	50.7438	52.6775	97.4062	71.4163	97.3713	98.7562
	SSIM	0.0127	0.0197	0.7717	0.2797	0.8402	0.8628
JPEG(20%)	NC (%)	52.1787	49.1400	43.0925	54.6788	41.3913	60.1675
	SSIM	0.0114	0.0125	0.0055	0.0275	0.0044	0.0521
JPEG(40%)	NC (%)	55.5113	54.3388	45.7425	58.8613	43.0338	64.5700
	SSIM	0.0268	0.0269	0.0100	0.0793	0.0099	0.1468
JPEG(60%)	NC (%)	58.2538	57.3663	48.7925	65.6338	65.0788	73.0750
	SSIM	0.0409	0.0402	0.0104	0.1348	0.1507	0.2331
JPEG(80%)	NC (%)	71.0613	71.8125	62.4600	78.8313	85.3016	86.5737
	SSIM	0.2618	0.2658	0.2173	0.4621	0.7047	0.7533

In the present work, we follow the first approach. Our proposed method is blind. Here, we have applied the SVD method to decompose the host block at the time of embedding. Let us refer to the proposed method as Grayscale image Watermarking using SVD (GWSVD).

2.5.1 Embedding process of GWSVD

In this work, first, the host image is divided into four sub-images by sub-sampling method [129], [130]. Each sub-image is partitioned into blocks of size 4×4 . Each block of the sub-images is decomposed by the SVD method. Before embedding the watermark image (W) is normalized into $[0, 1]$, which is a lossy technique, and then encrypted as W' using a secret key ‘key’. The ‘key’ is the hash value of the watermark image. For encryption purposes, we can use either FTTIE or FTTIE_{ext} depending on the requirement (whether used for copyright protection or authentication). The encryption process ensures the security of the GWSVD method.

In the embedding process, to embed the i^{th} pixel (p_i), the largest singular value of the i^{th} block of the sub-images is modified using p_i . To insert the intensity value $W'(p_i)$, the largest singular value $\{S_{i_j}(1, 1) : 1 \leq j \leq 4\}$ of the four sub-images are

Table 2.9: Average NC(%) and SSIM of different techniques under attacks ‘Flower’ as watermark

Attack	Measures	BTC_L [86]	BTC_H [86]	BTC_B [86]	Makbol et. al. [126]	Dhani et. al. [127]	BWBTC-PF [128]
NA	NC (%)	100.00	100.00	100.00	100.00	100.00	100.00
	SSIM	1.000	1.000	1.000	11.000	1.000	1.000
SP	NC (%)	93.2953	95.2975	96.8325	76.6038	92.3001	97.4737
	SSIM	0.3205	0.3593	0.4777	0.1961	0.4011	0.6659
CP	NC (%)	68.1250	66.6014	67.4375	70.8800	68.4463	68.3910
	SSIM	0.2510	0.1964	0.3516	0.2705	0.3717	0.3798
BR	NC (%)	80.4917	82.5700	98.1338	76.8363	98.4312	99.8387
	SSIM	0.1181	0.1317	0.8095	0.1798	0.8889	0.9428
GF	NC (%)	63.3063	63.7062	89.2825	78.7462	93.6013	96.6401
	SSIM	0.0421	0.0506	0.1856	0.2068	0.4932	0.5700
SNR	NC (%)	50.2962	50.7838	72.1213	78.0575	97.3538	97.9675
	SSIM	0.0052	0.0062	0.0826	0.3376S	0.6765	0.7196
RT	NC (%)	92.7700	94.1187	86.7088	76.2038	95.4413	95.5612
	SSIM	0.3924	0.2781	0.1592	0.1712	0.5867	0.6085
HE	NC (%)	49.8825	52.1013	96.9825	77.3425	97.1575	98.4350
	SSIM	0.0043	0.0083	0.6261	0.2089	0.6310	0.7431
JPEG(20%)	NC (%)	52.3475	48.0375	39.9338	56.1250	38.4250	61.1163
	SSIM	0.0086	0.0077	0.0044	0.0414	0.0362	0.0575
JPEG(40%)	NC (%)	55.7212	53.7050	43.6775	59.3763	42.0800	65.5738
	SSIM	0.0177	0.0175	0.0072	0.0833	0.0686	0.1553
JPEG(60%)	NC (%)	58.5500	57.0425	47.5475	67.6325	68.8638	75.3050
	SSIM	0.0271	0.0271	0.0089	0.1989	0.1712	0.2775
JPEG(80%)	NC (%)	63.3325	63.5988	53.7975	77.2588	84.6637	86.2812
	SSIM	0.1226	0.1263	0.0223	0.4260	0.7075	0.7353

modified. In this process, we use an embedding strength ‘T’, which controls the imperceptibility/quality of the watermarked image. In the embedding process, a watermark pixel is embedded into four co-located blocks by making two pairs of blocks (the first pair contains first and second sub-images; the second pair has third and fourth sub-images). The same watermark pixel is embedded into both pairs of blocks. The embedding equations of the ‘GWSVD’ method are given in Eq (2.17) and Eq (2.18). Then, each block passes through the inverse SVD. Finally, the modified sub-images are merged to get the watermarked image (I'). The block diagram of the embedding process of ‘GWSVD’ is given in Fig. 2.6. The algorithmic steps of the proposed embedding process are presented in **Algorithm 3: GWSVD-EmbeddingProcess()**. In this method, if the size of the host image is $N \times N$ then the size of the watermark is $\frac{N}{8} \times \frac{N}{8}$.

$$\begin{aligned}
 x_i &= \frac{S_{i_1}(1, 1) + S_{i_2}(1, 1)}{2} \\
 S'_{i_1}(1, 1) &= x_i + T \times W'(p_i) \\
 S'_{i_2}(1, 1) &= x_i
 \end{aligned} \tag{2.17}$$

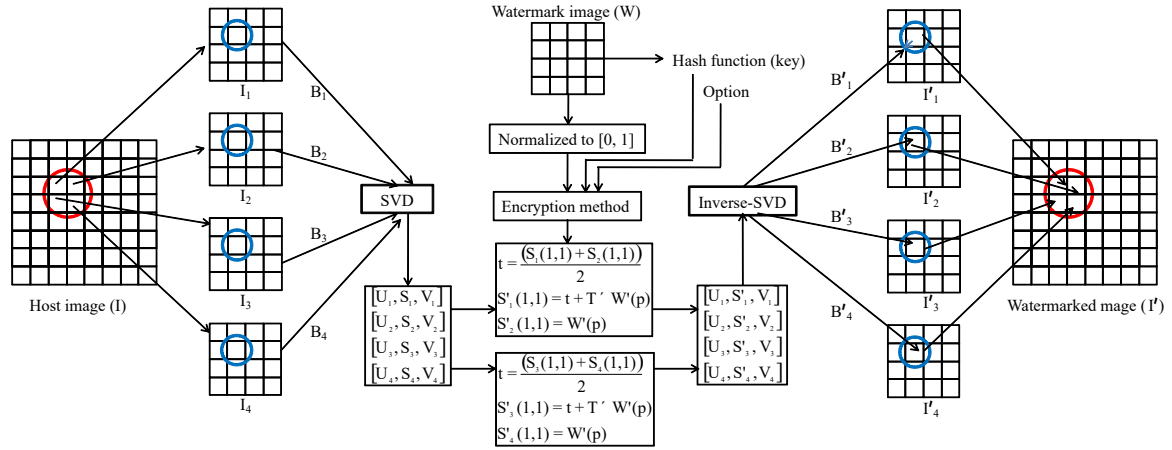


Figure 2.6: Block diagram of the embedding process of the proposed GWSVD method

$$y_i = \frac{S_{i_3}(1, 1) + S_{i_4}(1, 1)}{2}$$

$$S'_{i_3}(1, 1) = y_i + T \times W'(p_i)$$

$$S'_{i_4}(1, 1) = y_i$$
(2.18)

$$W'_1(p_i) = \frac{S'_{i_1}(1, 1) - S'_{i_2}(1, 1)}{T}$$

$$W'_2(p_i) = \frac{S'_{i_3}(1, 1) - S'_{i_4}(1, 1)}{T}$$

$$W'_x(p_i) = \frac{W'_1(p_i) + W'_2(p_i)}{2}$$
(2.19)

2.5.2 Extraction process of GWSVD

In the extraction process, the watermarked image is sub-sampled followed by the division into blocks and decomposition by SVD (like the embedding step). Then, the largest singular value of the blocks is taken and the expression given in Eq (2.19) is used to determine the value of the i^{th} pixel of the ciphered watermark ($W_c(p_i)$). Then the extracted watermark is deciphered using the secret key 'key' and the 'option' represents mode of encryption i.e., robust encryption (FTTIE) or fragile encryption (FTTIE_{ext}) and finally, de-normalized to obtain the desired watermark information $W'(p_i)$.

The proposed method is blind as no information is required other than the watermarked image. Figure 2.7 illustrates the block diagram of the proposed extraction

Algorithm 3 : GWSVD-EmbeddingProcess($I, W, key, 'opt', T, I'$)

Input: Host image (I), Grayscale watermark (W), Secret key (key), Option ('opt'), Embedding strength (T)

Output: Watermarked image (I')

- Step 1** I is divided into sub-images as $\{I_1, I_2, I_3, I_4\} \leftarrow \text{Sub-sampling}(I)$
- Step 2** W is normalized into $[0, 1]$ and denoted as W_n
- Step 3** $W' \leftarrow \text{Encrypt}(W_n, key, 'opt')$
- Step 4** $B_{ij} \leftarrow \text{PartitionIntoBlock}(I_j), 1 \leq j \leq 4$
- Step 5** For each co-located block B_{ij} of I_j
- $\{U_{ij}, S_{ij}, V_{ij}\} \leftarrow \text{SVD}(B_{ij})$
 - Largest singular values $\{S_{ij}(1, 1) : 1 \leq j \leq 4\}$ are modified as $\{S'_{ij}(1, 1)$ using Eq. (2.17) and Eq. (2.18)
 - $B'_{ij} \leftarrow \text{InverseSVD}(U_{ij}, S'_{ij}, V_{ij})$
- Step 6** $I'_j \leftarrow \text{Assembled}(B'_{ij})$
- Step 7** $I' \leftarrow \text{Up-sampling}(I'_1, I'_2, I'_3, I'_4)$
- Step 8** Return I'

method, and **Algorithm 4: GWSVD-ExtractionProcess**() describes the algorithmic flow of the extraction process.

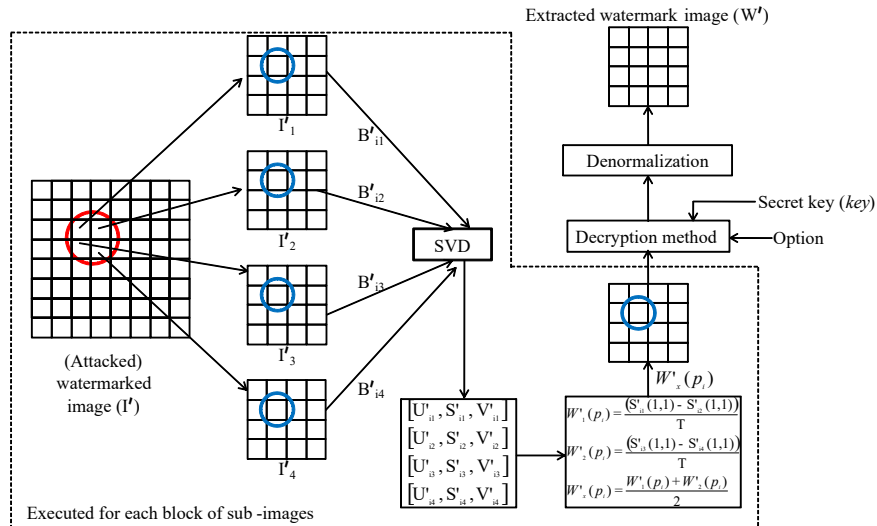


Figure 2.7: Block diagram of the extraction process of the proposed GWSVD method

Algorithm 4 : GWSVD-ExtractionProcess($I', key, 'opt', T, W'$)

Input: Watermarked image (I'), Secret key (key), Option ('opt'), Embedding strength (T)

Output: Extracted watermark (W')

- Step 1** I' is divided into sub-images as $\{I'_1, I'_2, I'_3, I'_4\} \leftarrow \text{Sub-sampling}(I')$
- Step 2** $B'_{i_j} \leftarrow \text{PartitionIntoBlock}(I'_j), 1 \leq j \leq 4$
- Step 3** For each co-located block B'_{i_j} of I'_j
1. $\{U'_{i_j}, S'_{i_j}, V'_{i_j}\} \leftarrow \text{SVD}(B'_{i_j})$
 2. Watermark pixel is extracted from the largest singular values $\{S'_{i_j}(1, 1) : 1 \leq j \leq 4\}$ using Eq. (2.19) and denote it as W'_c
- Step 8** $W'_d \leftarrow \text{Decrypt}(W'_c, key, 'opt')$
- Step 9** $W' \leftarrow \text{Denormalization}(W'_d)$
- Step 8** Return W'
-

2.5.3 Experimental Results of GWSVD

The proposed grayscale watermarking method 'GWSVD' is evaluated for various features to test the applicability of the method. The performance of 'GWSVD' is also compared with SVD-based existing SoA methods Ma et. al [117], and Ali et. al [131]. As per our understanding, we have implemented these two methods by ourselves.

Ma et. al [117] have proposed a grayscale image watermarking scheme based on SVD to embed the watermark. Here, one watermark pixel is embedded into an 8×8 block of the host image. During embedding, the scaled value of the watermark pixel is embedded into the block by modifying the largest singular values using some transformation function. The watermark is extracted using the inverse transformation function. The embedding strength is used to control the quality of the watermarked image and the robustness of the watermarking scheme.

Ali et. al [131] have proposed a grayscale watermarking scheme in a hybrid domain. They have used three transformations: Discrete Wavelet Transform (DWT), Discrete Cosine Transformation (DCT), and Singular Value Decomposition (SVD) to embed the watermark into the host image, by modifying the largest singular value of the blocks. To control the quality and robustness of the scheme the strength of the pixel intensity is changed as the applicant wants.

In this experiment, the same set of grayscale images of size 512×512 have been used as host images (shown in Fig. 2.3(a)). Two grayscale images 'Cameraman' and

‘Baboon’ each of size 64×64 are used as the watermark (shown in Fig. 2.3(d)-(e)) to judge the effectiveness of the proposed method. Empirically we set the value of ‘T’ to 50. Here, like previously, before embedding the watermark image will be encrypted by FTTIE or FTTIE_{ext} depending on the requirement, and accordingly we evaluate the performance of the GWSVD method. Each version of the encrypted watermark is shown in Fig. 2.3(d)-(e). Here, we evaluate the applicability of the proposed method using different features like imperceptibility, Capacity, and performance against attacks.




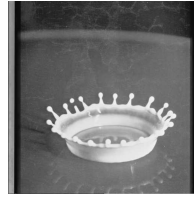




Imperceptibility and Capacity

Imperceptibility is computed to evaluate the perceptual quality of the watermarked image; a high imperceptible value means the quality of the watermarked image is good (see Fig. 2.8). Here, we report the average quality of the watermarked images when the watermark images ‘Cameraman’ and ‘Baboon’ are encrypted by FTTIE and FTTIE_{ext}. It is observed from the Fig. 2.8 that PSNR value is greater than 32 dB and SSIM is also greater than 0.98 (when T=50). Therefore, the proposed technique ‘GWSVD’ is good enough for general applications [75]. Table 2.10 presents a comparative study of the imperceptibility feature of the proposed technique against existing methods, demonstrating its superiority. In this table, we include the average performance with running over all the host images. The payload is also another feature, here, for all methods payload is the same. It is observed that the performance of the proposed technique is better than the existing schemes. The average PSNR and SSIM values of the watermarked images with varying embedding strength (T) are shown in Fig. 2.9 and Fig. 2.10, respectively. From Fig. 2.9, we note that at the lower value of strength, the performance of our method is poor; however, after a certain value of the strength the proposed method outperforms the existing method. Fig. 2.10 indicates that the performance of Ma et. al. [117] is inferior to the proposed method and the performance of the present is similar to the Ali et. al. [131].




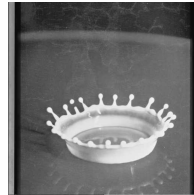




We can consider the quality vs. payload to study the overhead of the proposed GWSVD method, like the BWBTC-PF method. Here, all the methods have the same payload, so the technique giving a high PSNR value of the watermarked image will be considered better. Table 2.10 shows that the proposed GWSVD method gives high PSNR; therefore, the proposed method is better than the SoA methods.

Time complexity

The acceptability of the watermark system is also depends on the time complexity of the method. Like BWBTC-PF method, we report the execution time of the proposed

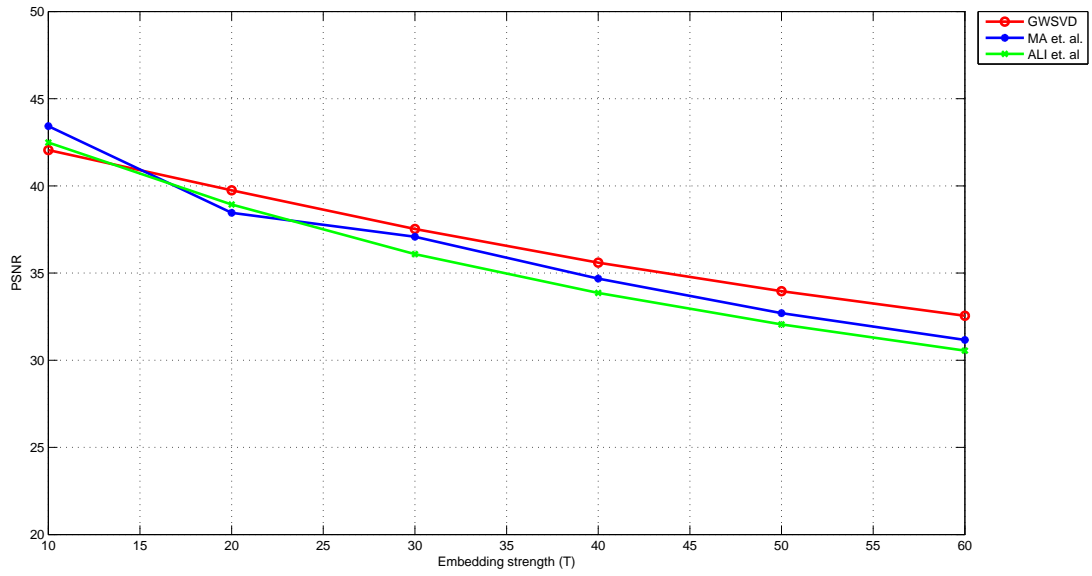
			
Lena PSNR=34.18 SSIM=0.9882	Tiffany PSNR=34.27 SSIM=0.9869	Zelda PSNR=34.23 SSIM=0.9876	Splash PSNR=33.77 SSIM=0.9833
			
Airplane PSNR=33.50 SSIM=0.9864	Boat PSNR=33.74 SSIM=0.9873	Couple PSNR=34.03 SSIM=0.9895	Man PSNR=33.94 SSIM=0.9796

(a) Watermarked images with ‘Cameraman’ as watermark

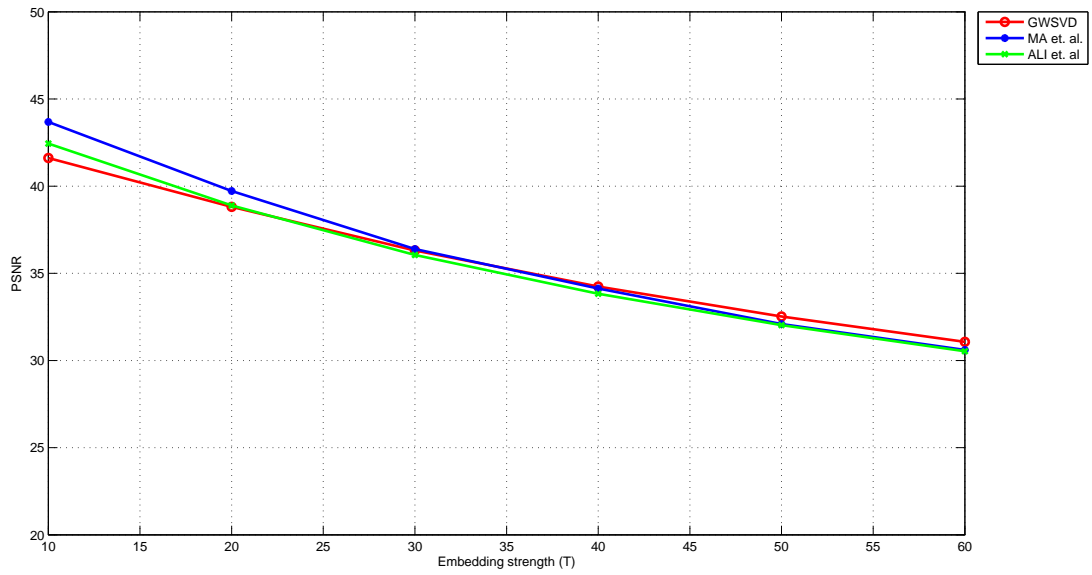
			
Lena PSNR=32.66 SSIM=0.9943	Tiffany PSNR=32.86 SSIM=0.9942	Zelda PSNR=32.71 SSIM=0.9923	Splash PSNR=32.37 SSIM=0.9912
			
Airplane PSNR=32.19 SSIM=0.9938	Boat PSNR=32.37 SSIM=0.9931	Couple PSNR=32.53 SSIM=0.9938	Man PSNR=32.50 SSIM=0.9785

(b) Watermarked images with ‘Baboon’ as watermark

Figure 2.8: Quality of the watermarked images of GWSVD method with embedding strength $T=50$

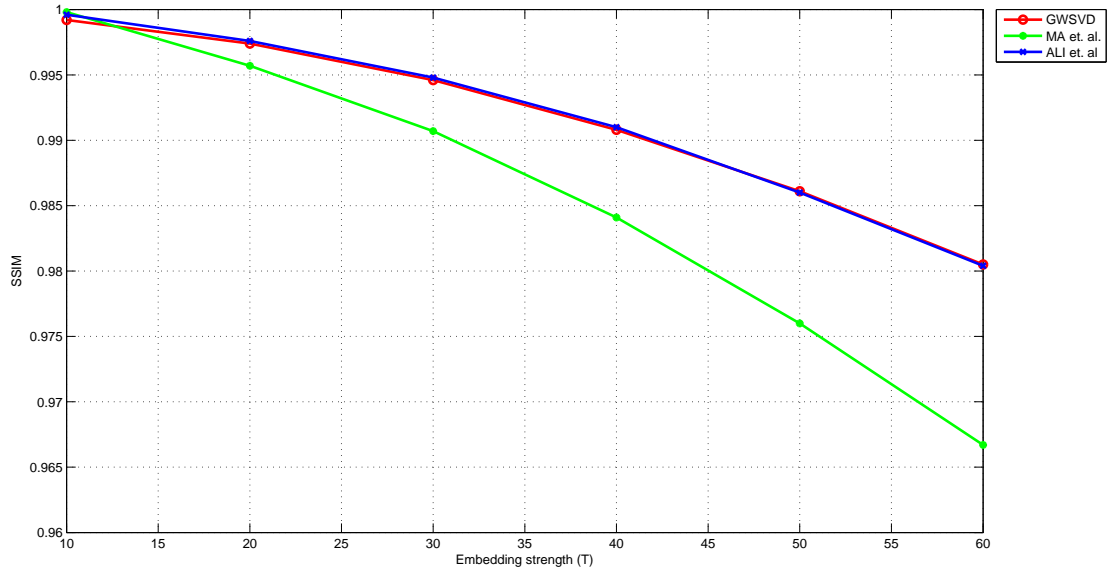


(a) Watermark image is 'Cameraman'

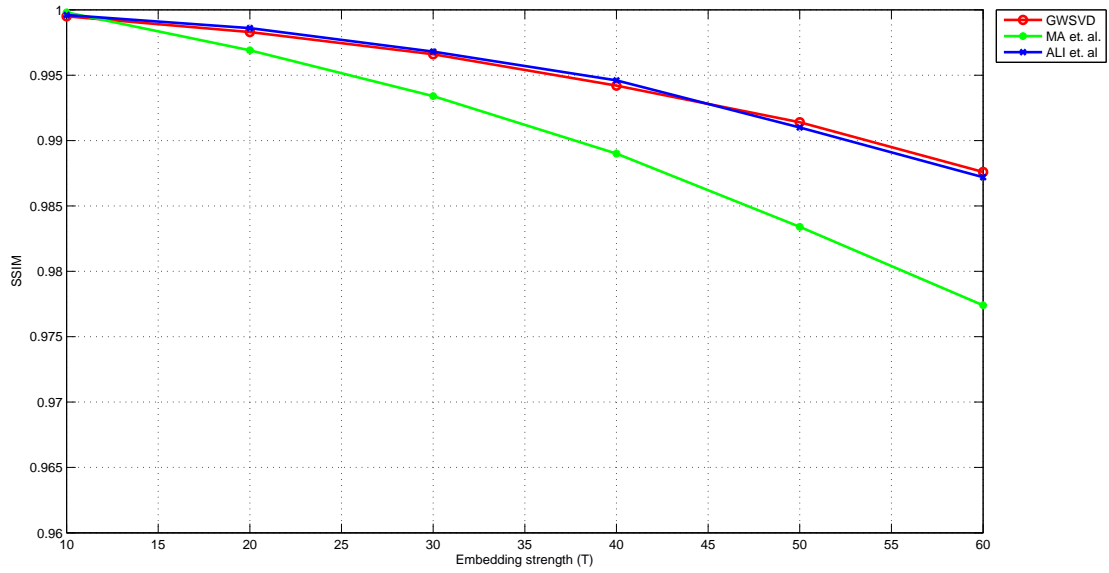


(b) Watermark image is 'Baboon'

Figure 2.9: Average PSNR of the watermarked images with varying embedding strength (T)



(a) Watermark image is 'Cameraman'



(b) Watermark image is 'Baboon'

Figure 2.10: Average SSIM of the watermarked images with varying embedding strength (T)

Table 2.10: Comparative study of the payload and the quality of the watermarked images given by different methods.

Watermarking Methods	Payload	Watermark ‘Cameraman’		Watermark ‘Baboon’	
		PSNR	SSIM	PSNR	SSIM
Ma et. al. [117]	$\frac{64 \times 64}{512 \times 512} = 0.015625$	32.70	0.9760	32.09	0.9834
Ali et. al. [131]	$\frac{64 \times 64}{512 \times 512} = 0.015625$	32.05	0.9860	32.03	0.9910
GWSVD [132]	$\frac{64 \times 64}{512 \times 512} = 0.015625$	33.96	0.9861	32.52	0.9914

Table 2.11: Average execution time (seconds) of the proposed GWSVD method and SoA methods

Methods	Execution time	
	Watermark ‘Cameraman’	Watermark ‘Baboon’
Ma et. al. [117]	0.326576	0.325894
Ali et. al. [131]	1.958417	1.957878
GWSVD [132]	0.623164	0.618442

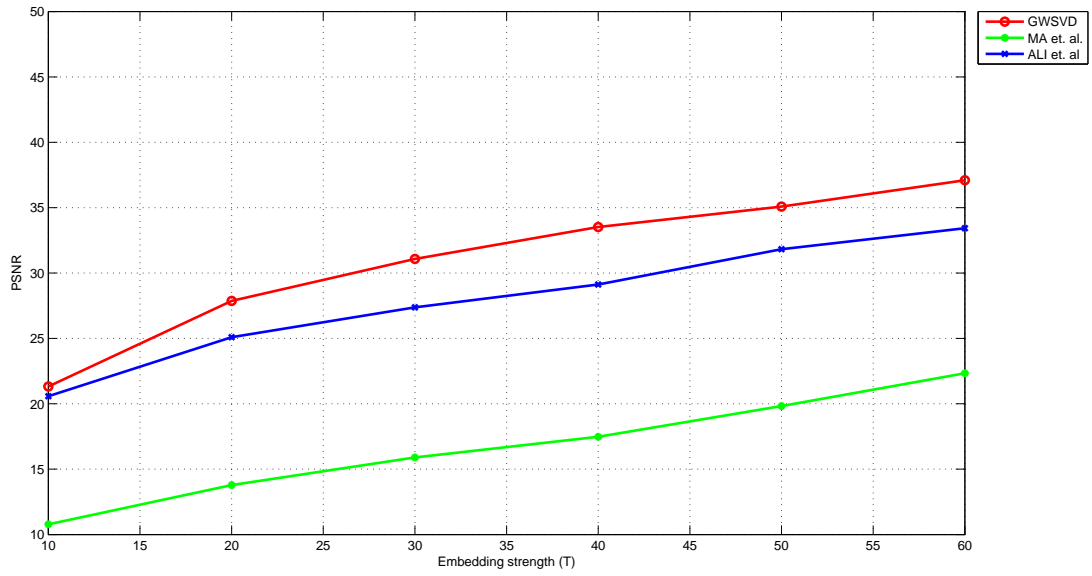
method in Table 2.11. The table reveals that the execution time of the proposed method twice of the Ma et. al. [117]. So, the proposed method is inferior to Ma et. al. [117]. Again, the proposed method is three times faster than Ali et. al. [131] method. Further, the quality of the proposed method is comparatively better, hence, the proposed method may be applied in some selective applications.

Performance against attacks

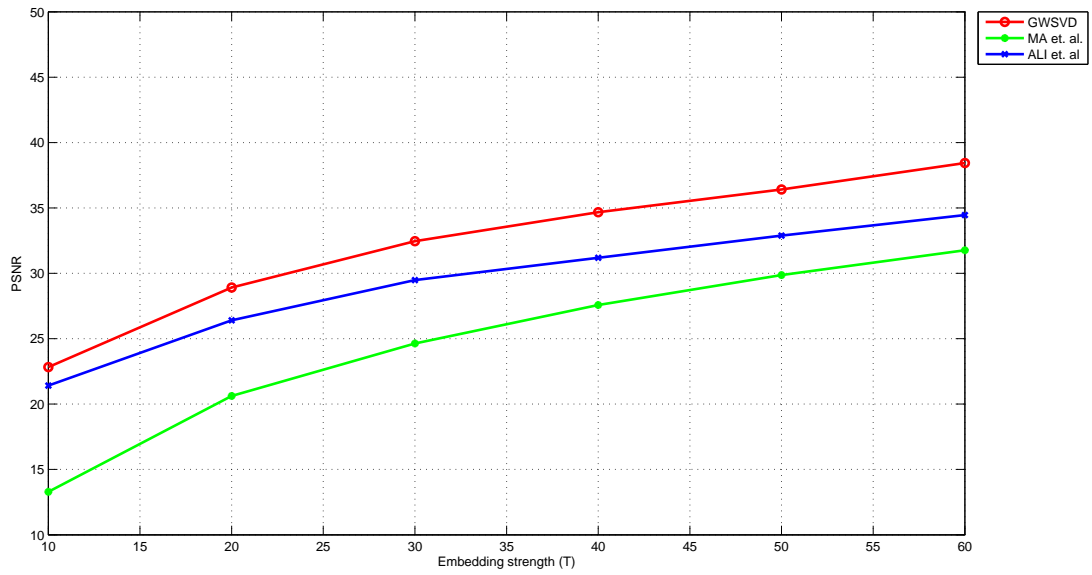
This section evaluates the performance of the proposed method against various attacks. Here, we consider three cases in terms of the watermark image: i) plain watermark image (not encrypted), ii) encrypted watermark image using FTTIE, and iii) encrypted watermark image using FTTIE_{ext} method. To measure the efficiency of the method, we consider some attacks on the watermarked image, given in Table 2.3. Due to space issues, we include the output on the two host images, say, ‘Zelda’ and ‘Splash’ with respect to both the watermark ‘Cameraman’ and ‘Baboon’. Table 2.12 - 2.15 provide the performance of the proposed method under different attacks. The first row of the tables shows the extracted watermark and it shows that the extracted watermark image is not the same as the original watermark. This is because, in the embedding process, we have normalized the watermark before embedding, which is a lossy process. In these tables, the first column shows the extracted watermark where the watermark is embedded without encryption. However, the second and third columns of the tables

show the extracted watermark where the watermark is embedded after encryption by FTTIE and FTTIE_{ext} methods, respectively. It has been understood from the output of the first and second columns of the tables that the extracted watermarks are visually identifiable and the quality of the extracted watermarks is also acceptable even after attacks on watermarked images. So, the proposed method can resist various attacks and able to extract the watermark that may be used for the copyright protection of images. It may be noted that the system under the first column is not secure; whereas the method in the second column is secure as the watermark image is encrypted by FTTIE and therefore, this scheme can be used to prove the copyright of an image. On the other hand, the output of the third column of the tables shows that the extracted watermark is like a noisy image under the attacks, where the watermark images are encrypted by FTTIE_{ext} method. So, this output shows that the GWSVD method can be used as a fragile watermarking technique if the watermark is encrypted by FTTIE_{ext}. As a result, our proposed technique may be used to verify the authenticity of an image. We have also reported the quality of the extracted watermark (PSNR and SSIM) for $T = 50$. The quality of the extracted watermark image increases as embedding strength 'T' increases, which is reflected in Fig. 2.11 and Fig. 2.12. The graphs demonstrate that the proposed method outperforms the SoA methods. Studying the Figs. 2.9 - 2.12, we set $T = 50$ considering the quality of the watermarked image, extracted watermark, and comparative performance.

The performance of the proposed grayscale watermarking technique 'GWSVD' is also compared with existing techniques under various attacks. The average PSNR and SSIM of the extracted watermark under the attacks are shown in Table 2.16. From the table, we notice that the proposed method 'GWSVD' outperforms the SoA methods under different attacks.

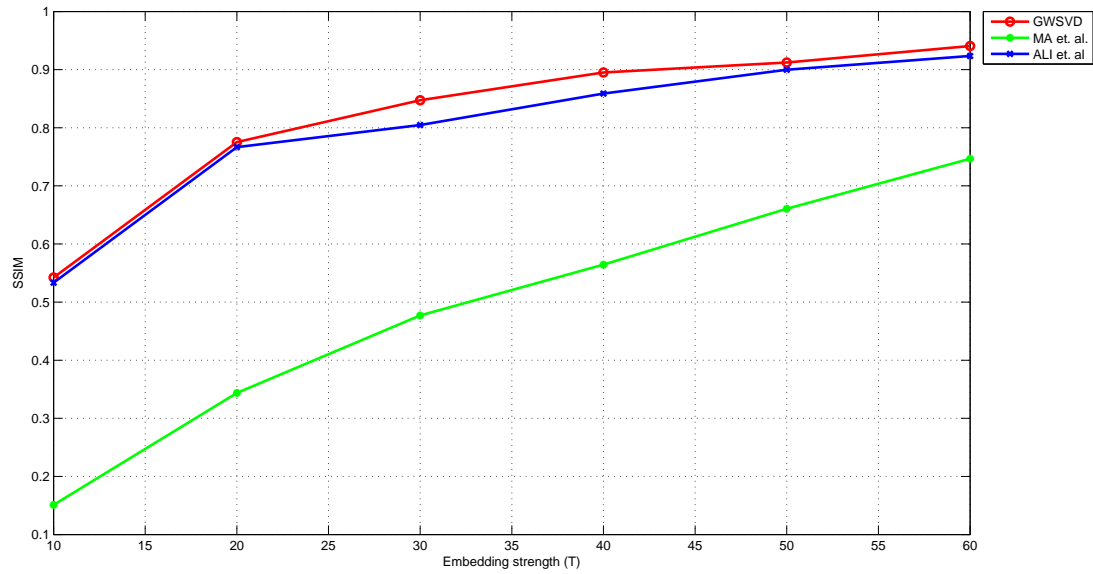


(a) Watermark image 'Cameraman'

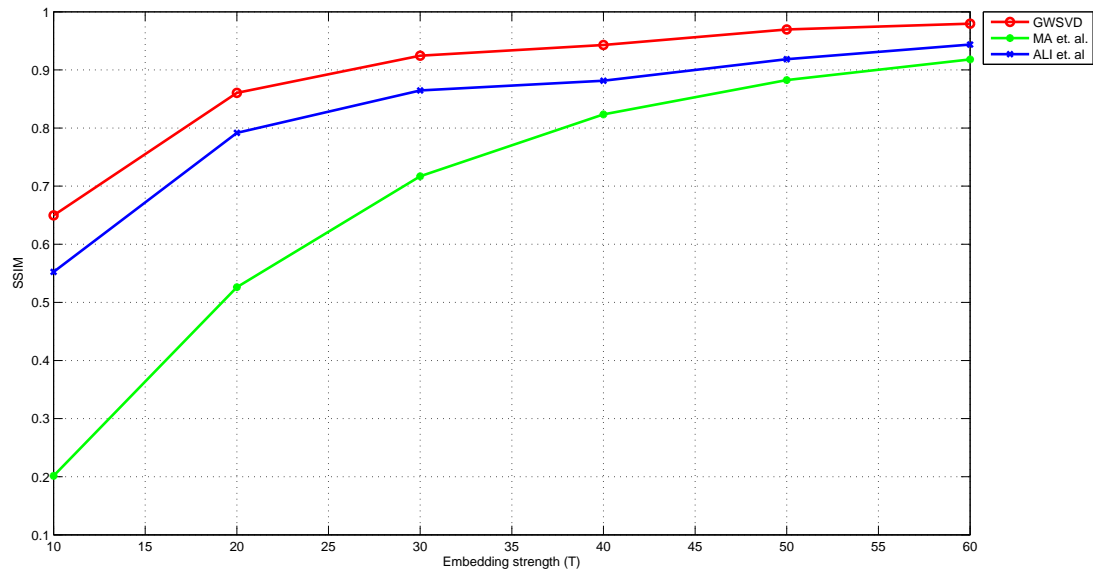


(b) Watermark image 'Baboon'

Figure 2.11: Average PSNR of the extracted watermark images with varying embedding strength (T)



(a) Watermark image 'Cameraman'



(b) Watermark image 'Baboon'

Figure 2.12: Average SSIM of the extracted watermark images with varying embedding strength (T)

Table 2.12: Robustness and Fragility of GWSVD method in terms of PSNR and SSIM (given in sideways): host image ‘Zelda’, watermark image ‘Cameraman’

Attack	Plain watermark				Encryption by FTTE				Encryption by FTTE _{ext}			
	Marked	NC	Extract	SSIM	Marked	NC	Extract	SSIM	Marked	NC	Extract	SSIM
NA		37.2965		0.9510		37.3571		0.9482		36.7281		0.9447
SP		14.5975		0.3299		14.7939		0.3381		8.8554		0.0065
CP		10.6721		0.2217		10.6740		0.2256		7.2531		0.0095
BR		21.0620		0.8278		22.1237		0.8346		10.1453		0.0207
GF		20.8642		0.8061		20.9810		0.7993		10.1355		0.0204
SRN		14.6357		0.5283		14.8924		0.5355		5.8498		0.0063
RT		18.5825		0.5422		19.1764		0.5668		7.2724		0.0123
HE		11.4240		0.3791		11.2671		0.3826		6.1927		0.0107
JPEG(20%)		15.9032		0.4381		15.5372		0.4292		6.7600		0.0055
JPEG(40%)		18.7603		0.6215		18.2256		0.6115		6.4718		0.0080
JPEG(60%)		23.1585		0.7573		23.0068		0.7516		9.3268		0.0169
JPEG(80%)		28.0712		0.8664		28.4293		0.8835		9.5736		0.0152

Table 2.13: Robustness and Fragility of GWSVD method in terms of PSNR and SSIM (given in sideways): host image ‘Splash’, watermark image ‘Cameraman’

Attack	Plain watermark				Encryption by FTTIE				Encryption by FTTIE _{ext}			
	Marked	NC	Extract	SSIM	Marked	NC	Extract	SSIM	Marked	NC	Extract	SSIM
NA		35.7933		0.9296		35.7252		0.9279		35.7540		0.9220
SP		15.1843		0.3672		15.0997		0.3578		8.9908		0.0189
CP		10.6713		0.2219		10.6725		0.2242		7.2581		0.0099
BR		20.5135		0.7734		21.5235		0.7613		10.1679		0.0189
GF		20.4613		0.7619		20.5189		0.7546		10.1654		0.0194
SRN		14.9687		0.4936		14.8605		0.4956		5.8310		0.0111
RT		17.9822		0.5343		18.7382		0.5499		7.1414		0.0055
HE		10.5774		0.2567		10.9899		0.2642		6.5012		0.0130
JPEG(20%)		15.5470		0.4359		15.8227		0.4446		6.7163		0.0062
JPEG(40%)		18.3531		0.6271		18.8583		0.6318		6.1858		0.0013
JPEG(60%)		22.1082		0.7383		22.9034		0.7495		9.3083		0.0166
JPEG(80%)		27.7976		0.8632		28.0541		0.8726		9.5908		0.0160

Table 2.14: Robustness and Fragility of GWSVD method in terms of PSNR and SSIM (given in sideways): host image ‘Zelda’, watermark image ‘Baboon’

Attack	Plain watermark			Encryption by FTTE			Encryption by FTTE _{ext}		
	Marked	NC	Extract	Marked	NC	Extract	Marked	NC	Extract
NA		39.0324	0.9854		39.1412	0.9837		38.7501	0.9821
SP		15.2341	0.2939		15.3647	0.2903		12.3106	0.0258
CP		13.7752	0.2391		13.6557	0.2369		8.1524	0.0108
BR		21.1893	0.7758		21.1887	0.7725		14.4959	0.0495
GF		21.0446	0.8347		21.0435	0.8285		14.4632	0.0486
SRN		13.4255	0.3875		13.7911	0.3991		7.4306	0.0329
RT		19.7012	0.5482		19.3772	0.5325		7.5388	0.0111
HE		11.0884	0.3680		11.1711	0.3684		8.0288	0.0147
JPEG(20%)		16.4525	0.4388		16.4910	0.4383		6.9763	0.0019
JPEG(40%)		17.7899	0.6178		18.5180	0.6231		8.9639	0.0174
JPEG(60%)		23.8247	0.8255		24.4585	0.8316		14.0987	0.0353
JPEG(80%)		29.2077	0.8714		30.0364	0.8887		14.2983	0.0366

Table 2.15: Robustness and Fragility of GWSVD method in terms of PSNR and SSIM (given in sideways): host image ‘Splash’, watermark image ‘Baboon’

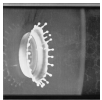

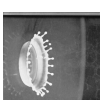
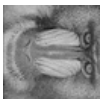


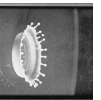

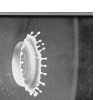



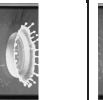
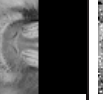
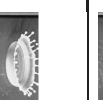
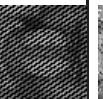


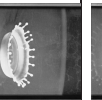
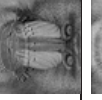
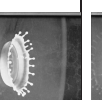
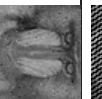
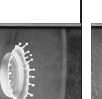
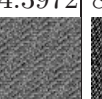
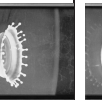
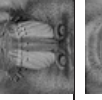
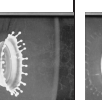
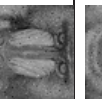
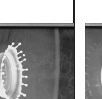

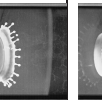
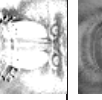
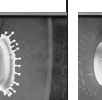
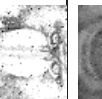
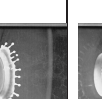
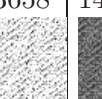
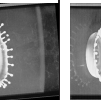
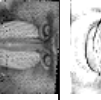
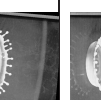
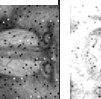
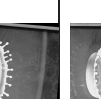
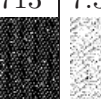
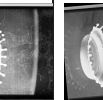
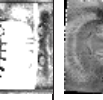
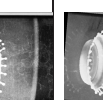
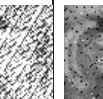
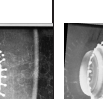
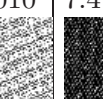
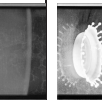
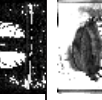
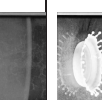
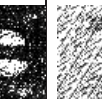
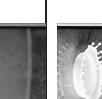
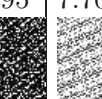
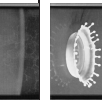
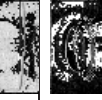
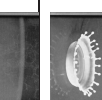
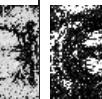
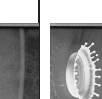

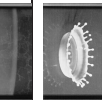
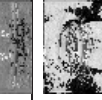
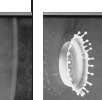
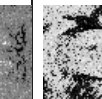
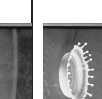
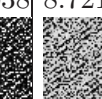
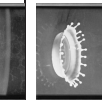
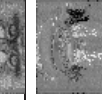
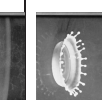
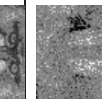
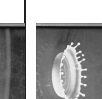

Attack	Plain watermark			Encryption by FTTIE			Encryption by FTTIE _{ext}		
	Marked	NC	Extract	Marked	NC	Extract	Marked	NC	Extract
NA		37.0072			36.9928			36.7377	
SP		15.3118			15.6195			12.2959	
CP		13.7739			13.6542			8.1539	
BR		20.5372			20.6027			14.3972	
GF		20.5010			20.5628			14.3896	
SRN		15.2679			14.2806			7.3658	
RT		19.2686			18.7735			7.4713	
HE		10.7504			10.8184			7.7010	
JPEG(20%)		15.3126			16.7094			6.8495	
JPEG(40%)		18.3388			18.8641			8.7212	
JPEG(60%)		24.1763			24.3240			14.6038	
JPEG(80%)		28.9050			29.5702			14.1824	

Table 2.16: Average PSNR and SSIM of different techniques under attacks ‘Camera-man’ and ‘Baboon’ as watermark with embedding strength T=50.

Attack	Measures	Watermark ‘Cameraman’			Watermark ‘Baboon’		
		Ma et. al. [117]	Ali et. al. [131]	GWSVD [132]	Ma et. al. [117]	Ali et. al. [131]	GWSVD [132]
NA	PSNR	19.8238	31.8256	35.0843	29.8663	32.8819	36.4112
	SSIM	0.6606	0.8998	0.9121	0.8826	0.9185	0.9696
SP	PSNR	12.8459	13.1885	15.1901	14.9629	13.5268	15.9126
	SSIM	0.2262	0.3020	0.3538	0.2240	0.2771	0.3164
CP	PSNR	10.6215	7.5591	10.7528	12.8215	7.3475	13.5583
	SSIM	0.2165	0.1437	0.2287	0.2065	0.1502	0.2350
BR	PSNR	18.3975	20.9355	21.9253	18.4046	19.8001	21.1663
	SSIM	0.5988	0.7464	0.7707	0.6171	0.6895	0.7944
GF	PSNR	17.1306	9.4476	20.2096	16.9774	9.4745	20.3167
	SSIM	0.5448	0.2359	0.7188	0.5504	0.2275	0.7907
SNR	PSNR	8.7056	14.0510	14.4999	11.2865	13.3150	13.8428
	SSIM	0.0355	0.4199	0.4515	0.2204	0.3869	0.4099
RT	PSNR	15.1409	17.1474	18.4967	18.0905	16.9261	18.7899
	SSIM	0.3665	0.4629	0.5288	0.4696	0.4736	0.5101
HE	PSNR	8.7441	10.8986	11.2166	9.9764	10.8912	11.1669
	SSIM	0.0154	0.2515	0.3482	0.2022	0.2553	0.3308
JPEG(20%)	PSNR	15.1068	8.8645	15.6005	14.9743	9.3773	16.4931
	SSIM	0.4164	0.0708	0.4348	0.4114	0.0909	0.4377
JPEG(40%)	PSNR	16.2194	10.2485	18.2738	16.5077	10.8149	18.5250
	SSIM	0.5148	0.1677	0.6169	0.5233	0.1741	0.6281
JPEG(60%)	PSNR	18.5484	11.0565	22.5358	19.4131	11.8699	23.7952
	SSIM	0.6091	0.2386	7411	0.6392	0.2723	0.7375
JPEG(80%)	PSNR	19.3720	14.1583	27.4408	23.7305	14.4820	28.8949
	SSIM	0.6370	0.3612	0.8596	0.7258	0.4180	0.8647

2.6 Conclusions

In this chapter, we have proposed binary image watermarking (BWBTC-PF) and grayscale image watermarking (GWSVD) methods for copyright protection/authentication of digital images. BWBTC-PF is a spatial domain-based technique and semi-blind; whereas, the GWSVD method is a blind and transform domain-based method. One serious concern of the proposed methods is the time complexity, which comparatively high; although, the overhead of the proposed methods are less than the SoA methods. The proposed methods are secure as the watermark image is encrypted and the person having the ‘*key*’ can only retrieve the watermark. The proposed method is robust, and the encryption of the watermark by FTTIE also gives the method as robust and secure and can be used for copyright protection. Again, the encryption of the watermark by FTTIE_{ext} method results in the watermarking method as a fragile method, which can be used to verify the authenticity of the watermarked image. Both methods perform well, and in most cases performance of the proposed methods is better than SoA methods.

The BWBTC-PF method has a disadvantage is that the recipient must have access to the same patternbook used for watermark embedding in order to extract the watermark, which may not always be feasible. On the other hand, the GWSVD method has a major limitation is that the original watermark cannot be recovered during the extraction process, even if no attack has occurred, as the watermark undergoes normalization, a lossy process, prior to embedding.

Audio signals are a crucial component of multimedia data, and ensuring their security is of equal importance. In the next chapter, we will introduce a secure watermarking method specifically designed for audio signals.

Chapter 3

Secure Watermarking Technique for Copyright Protection and Authentication of Audio Signals

3.1 Introduction

In the previous chapter, we have presented and discussed image watermarking techniques for copyright protection and authentication of digital images. In this chapter, we are going to present an audio watermarking technique for copyright protection and authentication of audio signals. There has been a massive increase in the utilization of digital audio data over the World Wide Web in the past two decades. Due to the enormous developments in the field of digital audio technology, anyone can download audio files from the Internet and then distribute or sell them after manipulation without the permission of the owner(s). This phenomenon is increasing progressively with the development of wireless technology [133, 134]. Therefore, it is mandatory to safeguard ownership as well as the integrity of the audio signal. In audio watermarking, some specific symbol is embedded into the host audio signal to detect the ownership or to check the authenticity of the signal.

Here, we have developed an audio watermarking algorithm that produces a good-quality audio signal, perceptually similar to the host audio. In this work, we consider binary watermarks only, and for the experimental purposes the ‘Logo’ and ‘Flower’ (previously used) are used as the watermark images. Different audio signals (.wav format) are taken as the host signal. Like the GWSVD method presented in the previous chapter, in this work, we have applied SVD on the audio signal and quantized

the largest singular value (LSV) of the blocks to embed the watermark bits. The proposed method is a blind watermarking technique. Here, also, the watermark images are encrypted by FTTIE and FTTIE_{ext} methods, according to establish the ownership or to check the integrity of the audio signal.

The organization of this chapter is as follows. The state-of-the-art of audio watermarking is studied in Section 3.2. In Section 3.3, the proposed audio watermarking algorithm is presented. In Section 3.4, experimental results are reported and also the performance of the proposed method is analyzed. Finally, the chapter is concluded in Section 3.5.

3.2 Literature Survey

Hua et al. [135] have discussed various audio watermarking methods. Their work serves as an overall tutorial for curious readers to acquire a historical and technical view of digital audio watermarking. In recent years, many audio watermarking methods have been developed. The audio watermarking methods are broadly classified into two groups: (i) time domain based methods [136–138] and (ii) transform domain based methods [139–142]. The time domain techniques are further subdivided into Least Significant Bit (LSB) and Phase-coding based techniques while transforming domain-based techniques are again classified into Spread Spectrum (SS), Quantization Index Modulation (QIM), and Patchwork. In general, the time domain based techniques are simple, easy, and have high data payload, but they are less robust against simple attacks. On the other hand, the transform domain based techniques are more complex, and have low data payload, but, they show improved robustness against various attacks.

In the literature, it has been seen that many image watermarking techniques [143, 144] can be applied for audio watermarking. However, developing an audio watermarking technique is difficult compared to image watermarking techniques for two reasons [141]. Firstly, the audio signals are represented by much fewer signals per time interval compared to images. As a result, the amount of information that can be added with audio signals is much lower than images. Secondly, the human auditory system (HAS) is much more sensitive than the human visual system (HVS). It is therefore difficult to satisfy the imperceptibility (inaudibility) property of audio watermarking techniques than the imperceptibility (invisibility) property of the watermarking techniques for images.

Khalidi et al. [145] have proposed an audio watermarking algorithm based on Empirical Mode Decomposition (EMD) where the host signal is converted into blocks and then EMD is applied on the blocks to generate Intrinsic Mode Functions (IMFs). The watermark bits are embedded into the extreme of the last IMFs. This method has a data payload of 50.3 bps. It also shows good robustness against a few attacks. In [146], a new audio watermarking technique has been proposed using gamma-tone dictionaries and a spike-gram for embedding and decoding watermarks. It has a data payload of about 56.5 bps which is slightly better than Khalidi et al.

Lei et al. [147] have presented an audio watermarking method using Singular Value Decomposition (SVD), Discrete Cosine Transform (DCT), and Synchronization Code(SC). SC is generated using a chaotic sequence and the watermark information is inserted blindly into the high-frequency band of the DCT-SVD frame. In this method, the data payload of about 43 bps is achieved and experimental analysis shows better robustness against various audio and Stirmark attacks as compared to other methods.

An audio watermarking algorithm has been proposed in [148] based on Quantization Index Modulation (QIM) and Discrete Wavelet Transform. The adaptive quantization steps using QIM are performed on the DWT coefficients of the audio signal. The watermark is embedded in the vector norm of the segmented approximate components. The experimental results demonstrate that it has shown good robustness on audio attacks and has a data payload of about 102.4 bps. In addition, the performance of the method against Stirmark attacks is not tested.

Li et al. [149] have designed a method for audio watermarking based on Dual-Tree Complex Wavelet Transform (DT-CWT) and Distortion Compensated Dither Modulation (DC-DM). The watermark is inserted in the low-pass coefficients by dither modulation process. The experimental results show superior performance against different audio attacks compared to other algorithms. However, the robustness of the method against Stirmark attacks is not tested and error analysis is not carried out. The data payload of their method is 128 bps.

Lei et al. [150] have designed a watermarking technique for audio signals with Lifting Wavelet Transform (LWT) and SVD with QIM. In this algorithm, the watermark is inserted into the low-frequency coefficients of LWT using SVD and QIM. The simulation results demonstrate the robustness against different audio attacks as well as Stirmark attacks. The data payload of their method is 170.6 bps.

The Spread Spectrum (SS) based watermarking method for the audio signal has been presented in [151]. The perceptual analysis technique is used during the watermark embedding and extraction process. This method achieves good performance

against various audio attacks as compared to other techniques. However, its performance is not tested against Stirmark attacks. The embedding payload of this method is 43.07 bps. Moreover, error analysis is not carried out.

In [152], norm ratio-based audio watermarking has been designed in the DWT domain to achieve a balance among imperceptibility, robustness, and data payload. The watermark is inserted into the audio signals by modifying the norm ratio of approximate coefficients and a chaotic sequence generated by the Tent map is used to encrypt the watermark before embedding. The performance of the method is tested on various attacks and the payload of the method is 172.4 bps.

Xiang et al. [153] have proposed a computationally efficient SS-based audio watermarking method. The DCT is performed on the segments of the audio signal. The watermark PN sequence is inserted into the audio segment using the watermark embedding method. The watermark bits are extracted by using the correlations between the watermarked audio segments and the PN sequences. The robustness of the method against common audio attacks is tested and the embedding rate is 84 bps. Moreover, the robustness of this method is not verified against the Stirmark audio benchmark, and error analysis is not also performed.

In [13], authors have proposed an audio watermarking algorithm based on LWT-DCT-SVD. For embedding the watermark, the algorithm decomposes the audio signals by LWT, and then the selected sub-band is transformed by DCT. The output of the DCT is used for the SVD process and the singular matrix is modified by quantization to embed the watermark. The robustness of this method against audio attacks is evaluated and the embedding payload is 172.3 bps.

An SVD-based audio watermarking algorithm using QIM is recently given in [154]. The audio signal is transformed into a frequency domain and SVD is applied. The watermark data is inserted using the QIM technique. The robustness of the method is evaluated against various audio attacks. The method performs better compared to other existing methods and the embedding rate is 187.5 bps. However, the robustness of the method is not verified against Stirmark attacks, and error analysis is also not computed.

In [141], An SVD-based audio watermarking has been developed where watermark information is inserted into the non-overlapping audio block by quantization of the Euclidean norms of singular values of the audio blocks. The robustness of the algorithm is tested on common audio attacks as well as against Stirmark attacks. The error analysis of the method has been evaluated and the embedding payload is about 196.

3.3 Proposed Audio Watermarking Method

From the literature survey of audio watermarking methods, it is noted that SVD-based audio watermarking algorithms have achieved a good trade-off between imperceptibility and robustness and are also acceptable for real-life applications [13, 155–157]. The reason behind the good performance of SVD-based methods is that they can adapt to variations in the local features of the signals. It is also seen that SVD-based algorithms have a high capacity. Still, there is a possibility to improve the performance of the watermarking algorithm by proper quantization of singular values. Inspired by the SoA methods, here, we have proposed an SVD-based audio watermarking method. The features of the proposed method are given below.

1. The proposed method can adapt the variations in local features of the given signal.
2. The Largest singular value is updated in a way such that the proposed method maintains good imperceptibility.
3. The watermark extraction process is simple and blind.
4. The data payload of the proposed method is quite high and is similar to the SoA methods.
5. The proposed scheme is secure and can be used as robust watermarking technique or fragile watermarking technique as per the requirement.

Like any watermarking technique, the proposed method has two phases: i) watermark embedding and ii) watermark extraction. We present the proposed Audio Watermarking technique using SVD and quantization of the largest singular value (AWSVD) in the following.

3.3.1 Embedding process of AWSVD method

In GWSVD method (see Section 2.5) to embed the watermark information, the largest singular value of the block is modified by the strength of the watermark image. In this work, we consider binary watermark images that are encrypted, with respect to the hash value of the watermark image as secret key (*key*), by FTTIE or FTTIE_{ext} method according to the requirement. The block diagram of the embedding process of AWSVD is depicted in Fig. 3.1. The host audio signal (1D) is transformed into a 2D signal and partitioned into blocks. The audio block is decomposed by the SVD

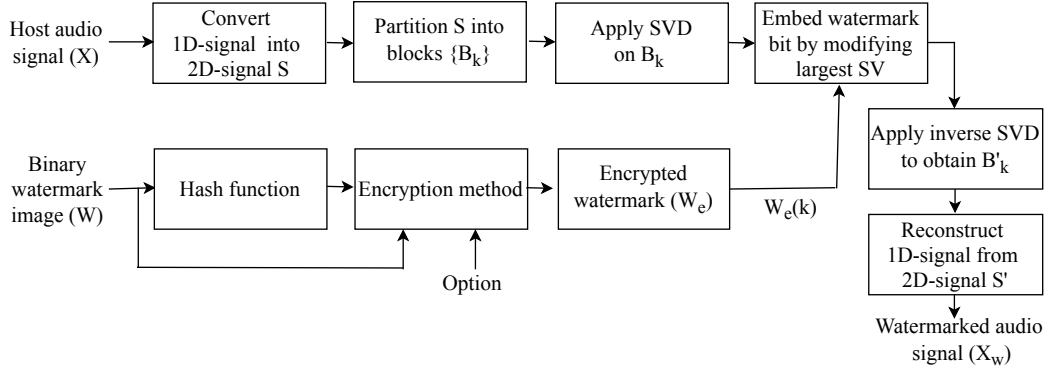
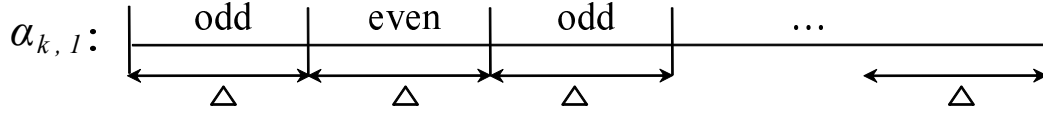


Figure 3.1: Block diagram of the embedding process of AWSVD

Figure 3.2: Partition of the range of $\alpha_{k,1}$ with interval length Δ

method. The watermark bit is embedded by modifying the Largest Singular Value (LSV) of each block. The possible range of the LSV is partitioned into intervals, where the length of each interval is, say, Δ . The LSV, $\alpha_{k,1}$, is quantized by the middle-most value of the interval (when required). One sample partition is shown in Fig. 3.2. The advantages of using quantization of LSV are: (i) quantization of the LSV has a negligible effect on the signal due to embedding, (ii) change of LSVs under attacks are very insignificant, (iii) quantization is simple, easy to implement, and less complex, (iv) quantization method helps to achieve a good trade-off among payload, imperceptibility, and robustness.

In the embedding process, the audio signal $X = \{x(i), i = 1, 2, \dots, L\}$ converted into 2D signal of size $M \times M$, where $M = \sqrt{L}$. The 2D audio is partitioned into non-overlapping blocks $\{B_k : k \in \{1, 2, \dots, T\}\}$ of size $q \times q$, where $T = \frac{M}{q} \times \frac{M}{q}$ is the size of the watermark image, i.e., in each block of the audio signal we embed one bit. From Fig. 3.2, we may note that the interval of the $\alpha_{k,1}$ can be categorized as ‘odd position intervals’ and ‘even position intervals’. Let us quantize $\alpha_{k,1}$ as

$$h_k = \left\lceil \frac{\alpha_{k,1}}{\Delta} \right\rceil \quad (3.1)$$

where $h_k \in \{0, 1, \dots, q\}$. The quantization level of an interval is represented by the middle value of the corresponding interval.

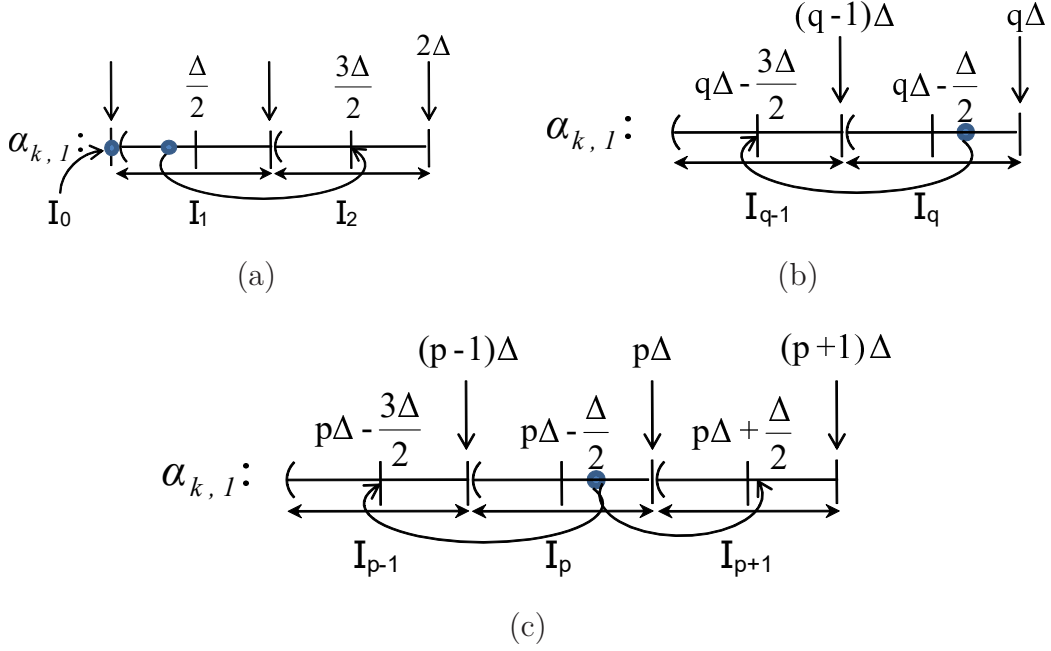


Figure 3.3: Watermark bit embedding when watermark bit is '0' and h_k is odd: (a) $h_k=1$, (b) $h_k = h_{max}$, (c) $1 < h_k < q$

In the embedding process, if the watermark bit is '0', then $\alpha_{k,1}$ is modified as $\alpha'_{k,1}$ with minimum error and $\alpha'_{k,1}$ belongs to an even interval. There are different cases, which are illustrated in Fig. 3.3. Let $h_k = 0$, which is denoted as ' I_0 ' in Fig. 3.3(a), then there is nothing to do (as h_k is even). If $h_k = 1$, corresponding interval is I_1 , then $\alpha_{k,1}$ is quantized as $\frac{3\Delta}{2}$ (shown in Fig. 3.3(a)). Suppose, h_k belongs to the last interval I_q . If h_k is even do nothing; otherwise, $\alpha_{k,1}$ will be quantized by $q\Delta - \frac{3\Delta}{2}$ (shown in Fig. 3.3(b)). The general case of the quantization process is shown in Fig. 3.3(c), where h_k is odd and the interval is I_p . According to the assumption, the h_k will be represented by either $p\Delta + \frac{\Delta}{2}$ or $p\Delta - \frac{3\Delta}{2}$, which one is closer to the $\alpha_{k,1}$. This embedding scheme is represented case-by-case in the Eq. (3.2).

$$\alpha'_{k,1} = \begin{cases} \alpha_{k,1} & \text{if } h_k \text{ is even} \\ \frac{3\Delta}{2} & \text{if } h_k = 1 \text{ (see Fig. 3.3.(a))} \\ h_k * \Delta - \frac{3\Delta}{2} & \text{if } h_k \text{ is odd and } h_k = q \text{ (see Fig. 3.3.(b))} \\ h_k * \Delta - \frac{3\Delta}{2} & \text{if } h_k \text{ is odd } 1 < h_k < q \text{ (see Fig. 3.3.(c)) and } \alpha_{k,1} \text{ is closer to } (k-1)^{th} \text{ interval} \\ h_k * \Delta + \frac{\Delta}{2} & \text{otherwise} \end{cases} \quad (3.2)$$

Similarly, when the watermark bit is ‘1’, we ensure that the $\alpha'_{k,1}$ belongs to an odd interval and incurs minimum error. Symmetric process will be followed, like Eq. (3.2), and it is given in Eq. (3.3).

$$\alpha'_{k,1} = \begin{cases} \alpha_{k,1} & \text{if } h_k \text{ is odd} \\ \frac{\Delta}{2} & \text{if } h_k = 0 \\ h_k * \Delta - \frac{3\Delta}{2} & \text{if } h_k \text{ is even and } h_k = q \\ & \text{if } h_k \text{ is even } 1 < h_k < q \text{ and} \\ h_k * \Delta - \frac{3\Delta}{2} & \text{if } \alpha_{k,1} \text{ is closer to } (k-1)^{th} \text{ interval} \\ h_k * \Delta + \frac{\Delta}{2} & \text{otherwise} \end{cases} \quad (3.3)$$

when the watermark bit is ‘1’. In this case, after modification $\alpha'_{k,1}$ belongs odd interval. We also describe the embedding process algorithmically, in **Algorithm 5: AWSVD-EmbeddingProcess()**.

Algorithm 5 : AWSVD-EmbeddingProcess($X, W, \Delta, Key, \text{'opt'}, X_w$)

Input: Host audio($X_{1...L}$), Binary watermark($W_{n \times n}$), Quantization step (Δ), Secret key (key), Option ('opt')

Output: Watermarked audio (X_w)

Step 1. 2D-Signal(S) \leftarrow 1D-Signal(X)

Step 2. Partition ‘ S ’ into blocks $B_k, k = 1, 2, \dots, T = (n^2)$.

Step 3. $W_e \leftarrow \text{Encrypt}(W, key, \text{'opt'})$

Step 4. For $k = 1$ to T

Step 4.a $[U_k, \Sigma_k, V_k] \leftarrow \text{SVD}(B_k), \alpha_{k,1} = \sigma_{1,1}$

Step 4.b Quantize $\alpha_{k,1}$ as $h_k = \lceil \frac{\alpha_{k,1}}{\Delta} \rceil$

Step 4.c If $W_e(k) == 0$ use Eq. (3.2)

Step 4.d Else use Eq. (3.3)

Step 4.e $B'_k \leftarrow \text{InvSVD}(U_k, \Sigma_k - \{\sigma_{1,1}\} \cup \{\alpha'_{k,1}\}, V_k)$

Step 5 1D-Signal(X_w) \leftarrow 2D-Signal(S')

Step 6 Return X_w

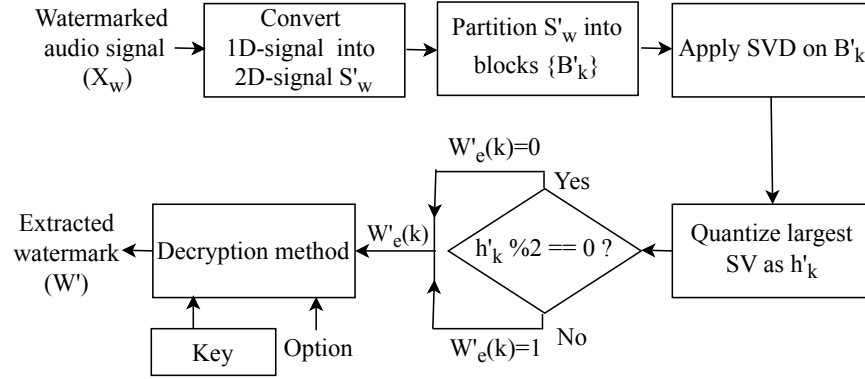


Figure 3.4: Block diagram of the extraction process of AWSVD

3.3.2 Extraction process of AWSVD method

The watermark extraction process is very simple. Here, the (attacked) watermarked audio signal, X'_w , is processed like the embedding method. The block diagram of the extraction process is given in Fig. 3.4. First, the watermarked signal is converted into a 2D signal and partitioned into blocks and each block is decomposed by the SVD method. The largest singular value, $\alpha'_{k,1}$, of each block is quantized by Δ as given in Eq. (3.4).

$$h'_k = \left\lceil \frac{\alpha'_{k,1}}{\Delta} \right\rceil \quad (3.4)$$

The parity (i.e. odd/even) of the h'_k determines the watermark bit (see Eq. (3.5)) and the collection of watermark bits constitute the encrypted watermark image W'_e .

$$W'_e(k) = \begin{cases} 0, & \text{if } h'_k \bmod 2 = 0 \\ 1, & \text{if } h'_k \bmod 2 = 1 \end{cases} \quad (3.5)$$

Finally, the watermark is obtained by deciphering the W'_e using *key*. The algorithmic sketch of the proposed extraction process is given in **Algorithm 6: AWSVD-ExtractionProcess()**.

3.4 Experimental Results and Performance Analysis of AWSVD

In this section, we demonstrate the experimental result and analyze the performance of the proposed AWSVD watermarking method. We have implemented this technique

Algorithm 6 : AWSVD-ExtractionProcess($X'_w, \Delta, key, \text{'opt'}, W'$)

Input: Watermarked audio (X'_w), Quantization step (Δ), Secret key (key), Option ('opt')

Output: Extracted watermark (W')

- Step 1.** 2D-Signal(S'_w) \leftarrow 1D-Signal(X'_w)
- Step 2.** Partition ' S'_w ' into blocks $B'_k, k = 1, 2, \dots, T$.
- Step 3.** For $k = 1$ to T
- Step 3.a** $[U'_k, \Sigma'_k, V'_k] \leftarrow \text{SVD}(B'_k), \alpha'_{k,1} = \sigma'_{1,1}$
- Step 3.b** Quantize $\alpha'_{k,1}$ as h'_k using Eq. (3.4).
- Step 3.c** $W'_e(k)$ is determined using Eq. (3.5).
- Step 4** $W' \leftarrow \text{Decrypt}(W'_e, key, \text{'opt'})$
- Step 5** Return W'
-

using MATLAB 2014 in the system with Intel core i5, 3.20 GHz CPU, 4GB RAM, and Windows 10.

Four different host audio signals namely, 'Classical', 'Jazz', 'Piano', and 'Tabla' are used as test signals [158] to evaluate the performance of the proposed method. The test signals are shown in Fig. 3.5. Audio signals are in 'wav' format with a sampling rate of 44.1 kHz and 16 bits/sample. The duration of the signals is 5.224 sec, i.e., each signal contains 230,400 samples. These audio signals are transformed into 2-D signals with size 480×480 , i.e., $M = 480$. Next, each audio 2D signal is partitioned into blocks of size 15×15 (i.e., $q = 15$). Number of blocks is $\frac{480}{15} \times \frac{480}{15} = 32 \times 32$. Since we assume that a bit will be embedded into each block then a binary image of size 32×32 is used as the watermark in our proposed method. Here, we use the same binary watermark images 'Logo' and 'Flower' as used in Chapter 2. In this experiment, we have down-sampled the original watermark images to 32×32 (originally, it was 128×128).

In the present method, before embedding the watermark, the watermark image is encrypted using its hash value (key) as the private key. Our method is originally robust (which means robustness can be proved without encrypting the watermark image) against the attacks. This encryption of the watermark image ensures the security of the proposed method. Since our targets are to protect the copyright information and

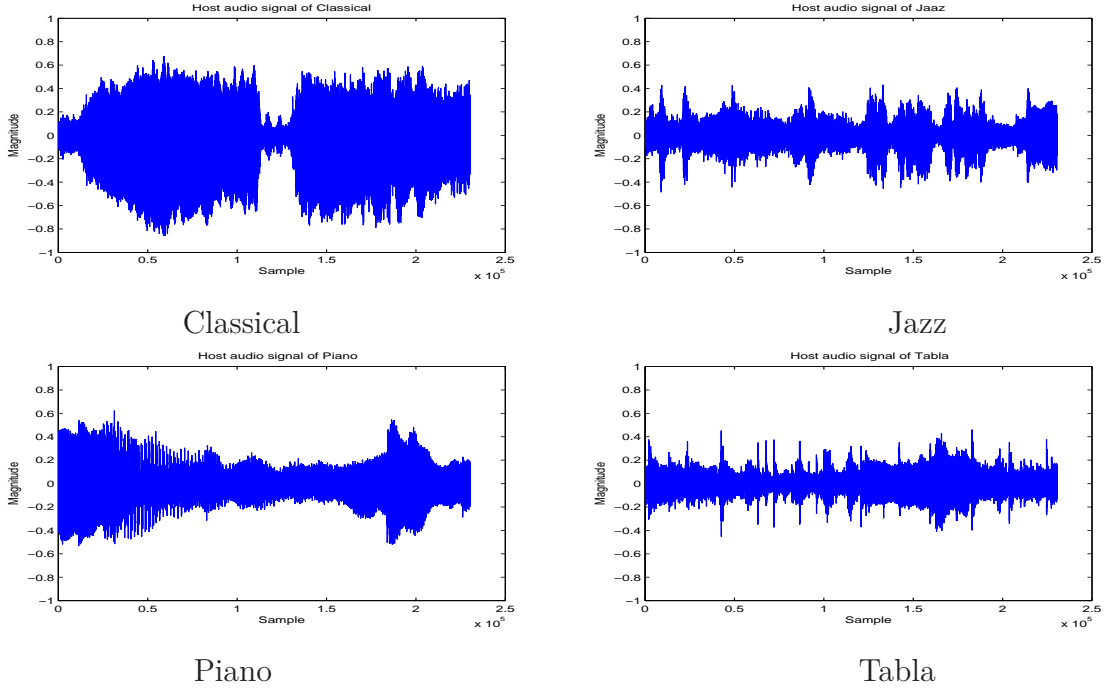


Figure 3.5: Host audio signals

Figure 3.6: Binary watermark images: original, encrypted by FTTIE, and encrypted by FTTIE_{ext}

test the authenticity of the audio data, so depending on the requirement we encrypt the watermark image either by FTTIE or FTTIE_{ext} method. The original and encrypted watermark images are shown in Fig. 3.6.

The quantization threshold Δ is set as 0.25 to achieve a good trade-off between imperceptibility and robustness and details of this are discussed in Section 3.4.4. From the above discussion, we may note that there are three different scenarios for embedding a watermark:

Case 1: Original watermark is embedded. We call this scenario as C_1 for future reference.

Case 2: The watermark image is encrypted by the FTTIE method using private key key and then it is embedded. We refer to this case as C_2 for future use.

Case 3: The watermark image is encrypted by the $\text{FTTIE}_{\text{ext}}$ method using private key key and then it is embedded. Hereafter, we name this scenario as C_3 .

Three watermarked signals for three respective cases are shown in Fig. 3.7(a), where ‘Logo’ is embedded into the host signal ‘Jazz’. The difference between the original signal and the watermarked signal is displayed in Fig. 3.7(b). The extracted watermarks with appropriate key(s) are shown in Fig. 3.7(c) and the extracted watermarks with wrong key(s) are shown in Fig. 3.7(d).

This figure shows that the watermark image can be extracted 100% provided there is no attack and correct decryption key is provided. On the other hand, when wrong decryption is used, then in case C_1 the watermark image can be extracted correctly as no key is used in this particular case. Whereas, for other two cases, noisy image is obtained since the watermark image is encrypted. Scheme C_2 and C_3 are secure and henceforth we exclude the result of C_1 . Therefore, C_2 and C_3 , these two versions of the proposed method may be applied in different applications.

We have compared the performance of the proposed with the SoA methods and we have implemented respective methods to the best of our understanding.

Bhat et. al. [141] have implemented a blind audio watermarking scheme to embed a binary watermark image into an audio signal. In their scheme, the host audio is first partitioned into different segments and then SVD is applied to each segment. The watermark bit is inserted into the segment by modifying the norm of the singular values of the segment.

Lei et. al. [150] have designed a blind audio watermarking algorithm based on Lifting Wavelet Transform (LWT) and SVD. LWT is applied to the audio signal and then the low-frequency subband is divided into blocks. The length of blocks depend on the watermark size and number of LWT decomposition levels. The watermark bit is embedded into the audio block by modifying the largest singular value of the block.

Li et. al. [152] have developed an audio watermarking algorithm where a binary image is embedded into audio the segments based on DWT. In the embedding process, a watermark bit is embedded by modifying the norm ratio of the approximate coefficients. Before embedding, the watermark symbol is encrypted using the Tent map function. The watermarking algorithm is tested under various attacks.

Novamizanti et. al. [13] have developed an audio watermarking scheme in the hybrid domain. They have used three transform domains LWT, DCT, and SVD. First, the audio signal is decomposed by LWT followed by DCT and SVD methods.

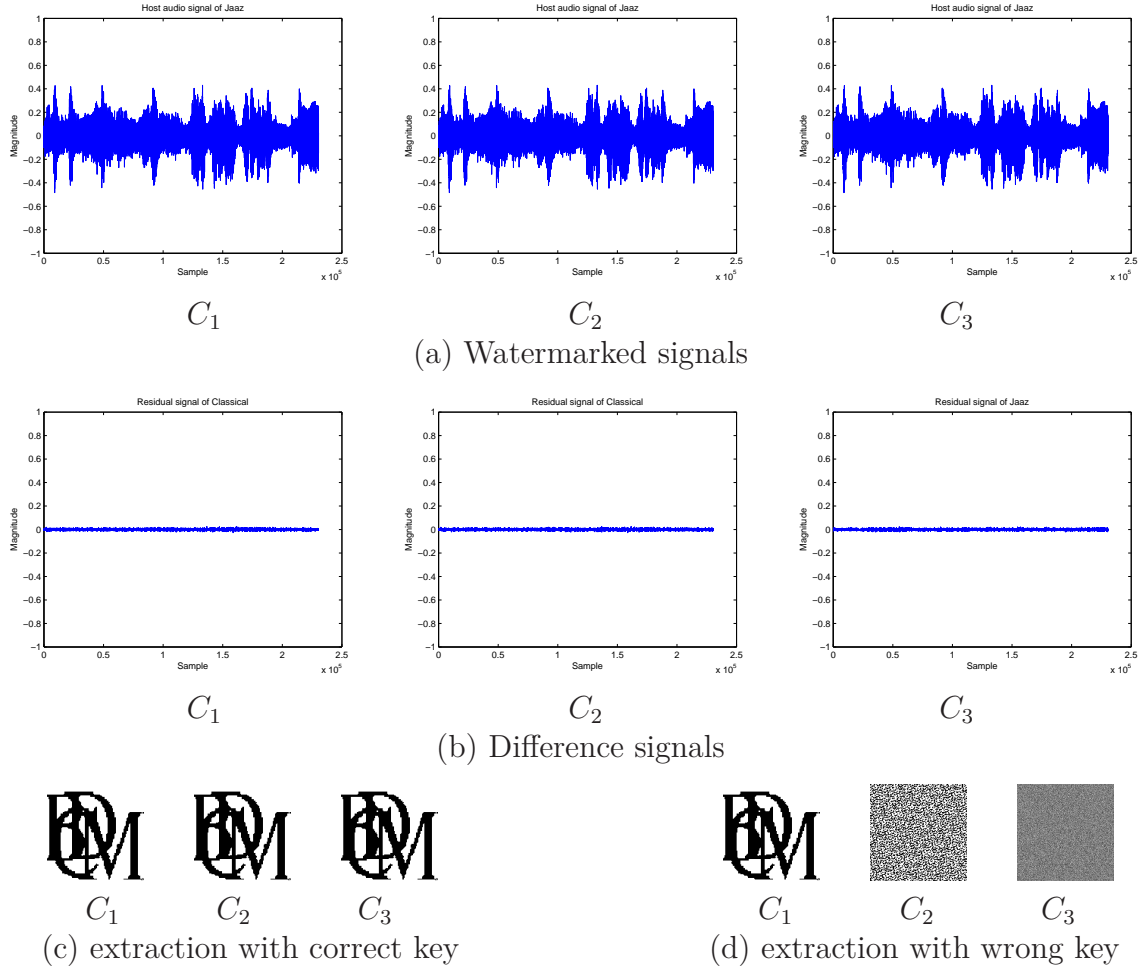


Figure 3.7: Performance of the proposed method AWSVD on the signal ‘Jazz’.

3.4.1 Imperceptibility and Payload

Imperceptibility is the difference between the original signal and the watermarked signal. In this experiment we ignore the subjective judgment since for the invisible watermarking method the watermarked signal is almost the same as the original signal (see the difference signals given in Fig. 3.7(b)) and it is difficult for a person to determine the difference. Here, we use Signal-to-Noise-Ratio (SNR) to measure the quality of the watermarked signal. The expression of SNR is given in Eq. (1.2). The SNR is more when the watermarked signal is close to the host signal, i.e., imperceptibility is high (which is the desired property of any watermarking technique).

Imperceptibility also depends on the payload (amount of data embedded). Here, payload is defined as bit-per-second (bps). In this experiment, we also vary the payload by changing the size of the blocks, and accordingly bits are embedded. The larger the

Table 3.1: SNR versus payload given by AWSVD method with ‘Logo’ as watermark

Audio signals	Payload				
	N=225 (196 bps)	N=256 (172 bps)	N=400 (110 bps)	N=576 (76 bps)	N=1024 (43 bps)
Classical	29.14	29.47	30.12	31.77	34.48
Jazz	22.16	22.88	25.00	26.24	28.69
Piano	26.91	27.38	28.12	28.66	31.94
Tabla	21.10	21.73	23.33	24.52	26.09
Average	24.83	25.37	26.64	27.80	30.30

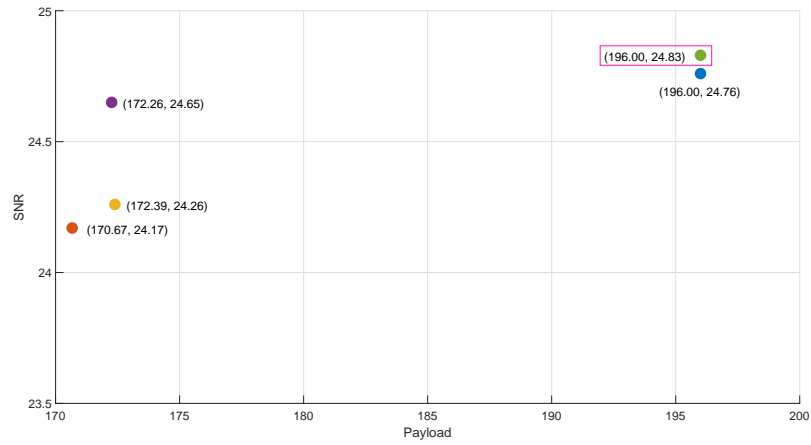
block size lesser the number of blocks and hence lesser number of bits are embedded. Therefore, SNR will be higher when we have a large block. The performance of the proposed method with varying payload is shown in Table 3.1 and Table 3.2.

Table 3.2: SNR versus payload given by AWSVD method with ‘Flower’ as watermark

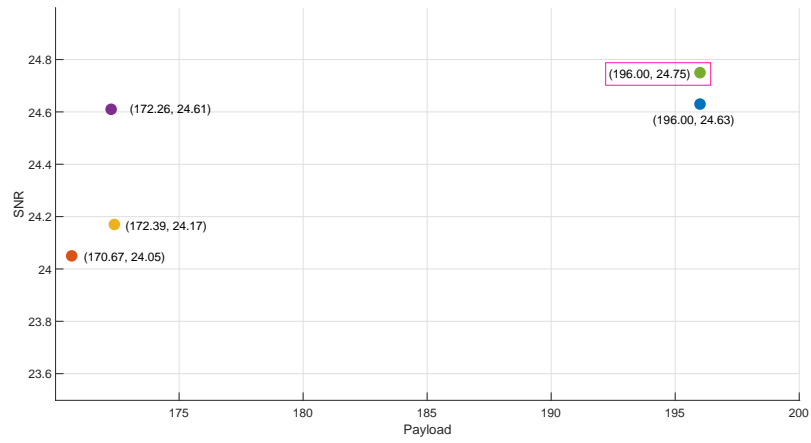
Audio signals	Payload				
	N=225 (196 bps)	N=256 (172 bps)	N=400 (110 bps)	N=576 (76 bps)	N=1024 (43 bps)
Classical	29.01	29.50	29.92	31.80	34.58
Jazz	22.12	22.80	25.11	26.26	28.73
Piano	26.97	27.23	28.18	28.73	32.02
Tabla	20.89	21.43	22.99	24.35	25.92
Average	24.75	25.24	26.55	27.79	30.31

According to the International Federation of the Phonographic Industry (IFPI) standard, the SNR value should be above 20 dB [159]. Our proposed technique satisfies the IFPI requirement and therefore the proposed method passes the imperceptibility test. The performance of the proposed method is also compared with the SoA method as given in Table 3.3. From this table, we note that the proposed method has the highest payload and maximum SNR value, i.e., the proposed method outperforms the existing methods.

To compare the overhead of the proposed AWSVD method with the SoA, like BWBTC-PF method, we take help of the distribution of the points defined by quality vs. payload. The comparative study is shown in Fig. 3.8, where the result for both the watermark images are shown. The figures established that overhead of the proposed



(a) Watermark as 'Logo'



(b) Watermark as 'Flower'

Figure 3.8: Payload vs. PSNR of different methods

Table 3.3: Comparative study of the performance of the proposed method with SoA methods

Methods	Techniques	Payload (bps)	‘Logo’ SNR (dB)	‘Flower’ SNR (dB)
AWSVD [160]	SVD-Qantization	196	24.83	24.75
Bhat et. al. [141]	SVD-QIM	196	24.76	24.63
Lei et. al. [150]	LWT-SVD	170.67	24.17	24.05
Li et. al. [152]	DWT-QIM	172.39	24.26	24.17
Novamizanti et. al. [13]	LWT-DCT-SVD	172.26	24.65	24.61

Table 3.4: Average execution time (seconds) of the proposed AWSVD method and SoA methods

Methods	Execution time	
	Watermark ‘Logo’	Watermark ‘Flower’
Bhat et. al. [141]	0.256545	0.253306
Lei et. al. [150]	0.316009	0.305788
Li et. al. [152]	2.896579	2.908312
Novamizanti et. al. [13]	2.592259	2.584234
AWSVD [160]	0.245708	0.249287

method is less than SoA methods, as the points resulted by the proposed method are located at the right-top position (marked by a box) in the respective figure.

3.4.2 Time complexity

To judge the efficacy of the AWSVD method with respect to the time complexity, we report the execution time of the different method in Table 3.4. The table reflects that our method is faster than all other methods. Therefore, the proposed AWSVD method can be applied in many applications (please note that overhead of the method is less).

3.4.3 Robustness test

Robustness means the ability to extract the watermark under the attacks, i.e., we need to measure the similarity/dissimilarity between the original watermark and the extracted watermark. In this experiment, we use two parameters: i) normalized coefficient (NC), see Eq. (1.4) and ii) bit error rate (BER), as given in Eq. (1.5). Here, it may be noted that the foreground is ‘1’ and the background is ‘0’, and display purposes we use the complement image. When NC is close to 1 then the foreground part of the

extracted watermark is similar to the original watermark. Again, the measure BER indicates the percentage of mismatch. If the value is close to zero then we may infer that both the watermarks (original and extracted) are close enough. So, a method is robust when NC is close to one as well as BER is close to zero, when the watermarked image is attacked.

To measure the robustness of the proposed method, we have considered different audio attacks. The audio editing and attacking tools used in this experiment are Adobe Audition 1.0 (for echo addition and inverse attacks), GoldWave 5.18 (for denoising, smoothing, MP3 compression, and re-sampling attacks), and MATLAB (for low-pass filtering, additive white Gaussian noise, cropping, and re-quantization attacks). The summary of the different attacks is given in Table 3.5. After the attacks, if the

Table 3.5: Different attacks on watermarked audio signal

Attack name	Description
Low-pass filtering (LP)	Low-pass filtering using a second-order Butterworth filter with cut-off frequency 'f=11.025 kHz' is performed on the watermarked audio signals
Additive noise (AN)	White Gaussian noise are added to the watermarked audio signals until the SNR of the resulting signal is below 20 dB
Cropping (CR)	Different segments of length 500 are set to '0'
Echo addition (EA)	An echo signal with a delay of 98ms and a decay of 41% is added to the watermarked audio signal
Denoising (DE)	The 'Hiss removal' function of GoldWave is used to denoise the watermarked audio signal
Reverse amplitude (RA)	Negate the sample of the watermarked signal
Smoothing (SM)	The smoothing operation of GoldWave is used to produce slow changes to the watermarked audio signal
Re-quantization (RQ)	The 16-bit/sample watermarked audio signals is quantized down to 8-bits/sample and then back to 16-bits/sample
MP3 compression (X kbps)	MPEG-1 layer-3 compression are applied using GoldWave on the watermarked audio signals. The watermarked audio signal is compressed at bit rates X=32
Re-sampling (RS)	As the audio signal is sampled at 44.1 kHz, the watermarked signal is down-sampled at different frequency 8.0 kHz and back to 44.1 kHz.

extracted watermark is easily recognizable then this establishes that the method is robust and can be used for copyright protection. On the other hand, due to attack,

if nothing can be guessed about the watermark image (i.e., the watermark looks like a noisy image) then the proposed method is fragile and the method can be applied to check the authenticity. The performance of the proposed method against different attacks is reported in Table 3.6 with the help of NC of the extracted watermark, and BER (written sideways). Here, we embed the watermark image on the ‘Jazz’ signal. From this table, the extracted watermark images under the scheme C_2 are fully recognizable; whereas, under C_3 nothing can be guessed about the watermark image. Theoretically, the value of NC should be close to one, and BER to be close to zero in case C_2 . Since, we consider binary image, so the scheme C_3 should provide both the NC and BER values close to 50%. This phenomenon is justified by the performance of the proposed method as given in Table 3.6. Hence, we may conclude that the C_2 scheme of the proposed method gives robust performance under different attacks with a higher level of security and therefore this scheme can be used to protect copyright information. Again, the C_3 scheme of the method is secure and behaves as fragile under the attacks. So, the C_3 scheme of the proposed method can be used for authentication purposes.

Similar results for the audio signals ‘Piano’, ‘Tabla’, and ‘Classical’ are shown in Table 3.7 and Table 3.8. The NC values are above 0.8879, so 90% or more accurate. On the other hand, the performance under the attack ‘echo addition’ is comparatively poor in terms of BER , which is around 12.5%. As a whole, the proposed method is robust against almost all attacks except ‘echo addition’.

Table 3.6: Performance of AWSVD against attacks: host signal ‘Jazz’ and watermark images ‘Logo’ and ‘Flower’













































Attack	Logo		Flower	
	C_2	C_3	C_2	C_3
NA	 ← NC ← BER 1.0000 0.0000	Marked  ← NC Extract ← BER 1.0000 0.0000	Marked  ← NC Extract ← SSIM 1.0000 0.0000	← NC  ← BER 1.0000 0.00000
LP	 0.9837 1.8555	 0.5431 51.36721	 0.9853 1.6602	 0.6657 44.4336
AN	 1.0000 0.0000	 0.5291 51.3672	 1.0000 0.0000	 0.5791 49.6094
CR	 1.0000 0.0000	 0.5397 52.1484	 1.0000 42.7734	 0.6798 100.0
EA	 0.9896 1.1719	 0.5618 47.3633	 0.9904 1.0742	 0.5874 49.8047
DE	 1.0000 0.0000	 0.5458 51.3672	 1.0000 0.0000	 0.6499 45.3125
RA	 1.0000 0.0000	 0.5759 48.0469	 1.0000 0.0000	 0.6711 43.9453
SM	 1.0000 0.0000	 0.5748 48.5690	 1.0000 0.0000	 0.6740 43.5547
RQ	 1.0000 0.0000	 0.5672 49.0234	 1.0000 0.0000	 0.6718 43.4951
MP3 (32 kbps)	 1.0000 0.0000	 0.5736 54.2442	 1.0000 0.0000	 0.6769 43.1641
RS (44.1-8.0-44.1)	 1.0000 0.0000	 0.5741 48.2422	 1.0000 0.0000	 0.6711 43.9453

Table 3.7: Quality of the extracted watermark from watermarked audio signals with ‘Logo’ as watermark under attacks

Audio attacks	Piano		Tabla		Classical	
	NC	BER (%)	NC	BER (%)	NC	BER (%)
NA	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
LP	0.9681	5.1758	0.9957	0.4883	0.9430	6.3477
AN	0.9971	0.3906	0.9983	0.1953	0.9974	0.2930
CR	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
EA	0.8879	12.5977	0.9957	0.4883	0.8917	12.1094
DE	0.9983	0.1953	0.9948	0.5859	0.9998	0.0977
RA	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
SM	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
RQ	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
MP3 (32 kbps)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
RS (44.1-8.0-44.1)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000

Table 3.8: Quality of the extracted watermark from watermarked audio signals with ‘Flower’ as watermark under attacks

Audio attacks	Piano		Tabla		Classical	
	NC	BER (%)	NC	BER (%)	NC	BER (%)
NA	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
LP	0.9737	4.7852	0.9971	0.3906	0.9492	5.9570
AN	0.9985	0.1953	0.9974	0.2930	0.9985	0.1953
CR	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
EA	0.8957	11.7188	0.9971	0.3906	0.8993	11.6211
DE	0.9983	0.1953	0.9957	0.4883	0.9998	0.0977
RA	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
SM	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
RQ	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
MP3 (32 kbps)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
RS (44.1-8.0-44.1)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000

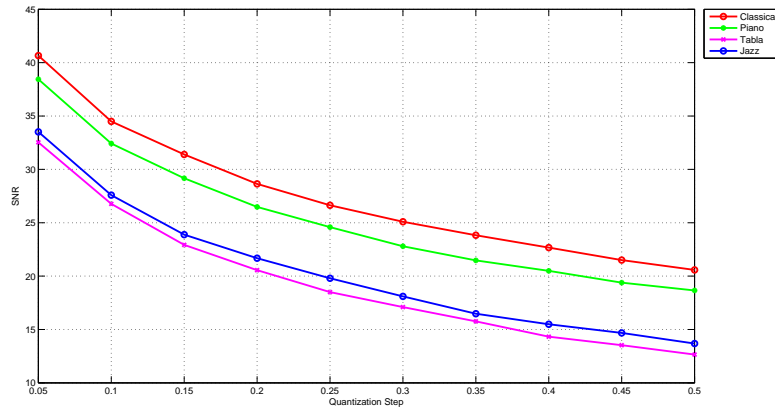
Table 3.9: Robustness performance of different audio watermarking techniques on the ‘Classical’ audio signal with watermark ‘Logo’.

Attacks	Bhat et. al.[141]		Lei et. al.[150]		Li et. al.[152]		Novamizanti et. al.[13]		AWSVD [160]	
	NC	BER(%)	NC	BER(%)	NC	BER(%)	NC	BER(%)	NC	BER(%)
NA	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
LP	0.9413	6.4453	0.9281	7.4219	0.9316	7.1289	0.9395	6.5430	0.9430	6.3477
AN	0.9957	0.4883	0.7892	22.9492	0.9947	0.5859	0.9651	5.3711	0.9974	0.2930
CR	1.0000	0.0000	0.9998	0.0977	1.0000	0.0000	0.9988	0.1953	1.0000	0.0000
EA	0.8688	14.6484	0.8843	12.7930	0.8648	15.0391	0.9195	9.2773	0.8917	12.1094
DE	0.9974	0.2930	0.9997	0.0977	0.9985	0.1953	0.9985	0.1953	0.9998	0.0977
RA	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
SM	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
RQ	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
MP3 (32 kbps)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
RS (44.1-8.0-44.1)	1.0000	0.0000	0.9803	2.1484	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000

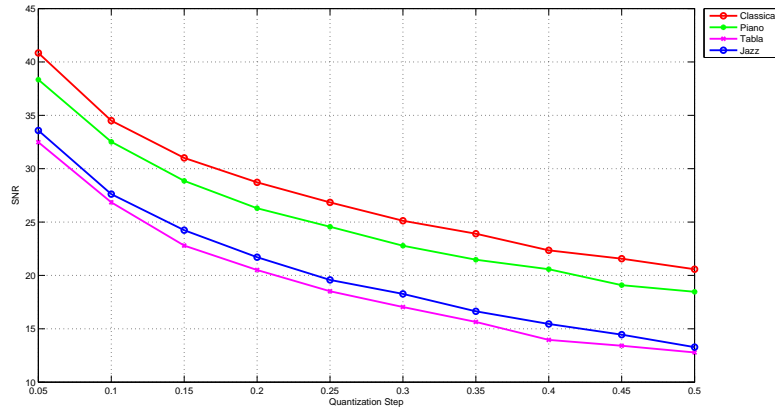
Table 3.10: Robustness performance of different audio watermarking techniques on the ‘Classical’ audio signal with watermark ‘Flower’.

Attacks	Bhat et. al.[141]		Lei et. al.[150]		Li et. al.[152]		Novamizanti et. al.[13]		AWSVD [160]	
	NC	BER(%)	NC	BER(%)	NC	BER(%)	NC	BER(%)	NC	BER(%)
NA	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
LP	0.9462	6.0547	0.9314	6.9336	0.9366	6.6406	0.9452	6.1523	0.9492	5.9570
AN	0.9957	0.4883	0.8417	19.3359	0.9957	0.4883	0.9664	5.2734	0.9985	0.1953
CR	1.0000	0.0000	0.9998	0.0977	1.0000	0.0000	0.9988	0.1953	1.0000	0.0000
EA	0.8879	12.5977	0.8993	11.6211	0.8774	13.0391	0.9195	9.2773	0.8993	11.6211
DE	0.9997	0.0977	0.9997	0.0977	0.9985	0.1953	0.9997	0.0977	0.9998	0.0977
RA	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
SM	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
RQ	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
MP3 (32 kbps)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
RS (44.1-8.0-44.1)	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000

In this section, the robustness of the proposed watermarking technique is compared with state-of-the-art techniques. Here, we have compared the proposed method with the methods described in [13, 141, 150, 152]. The comparative study is reported in Table 3.9 and Table 3.10. From these tables we observe that the performance of the proposed method better than SoA methods for LP, AN, and CR attacks; has similar performance with respect to DE, RA, SM, RQ, MP3 and RS attacks. Only, in case of EA attack, Novamizanti et. al.[13] method performs better than our method. Therefore, we may note that the performance of the proposed method is comparable with the SoA methods.

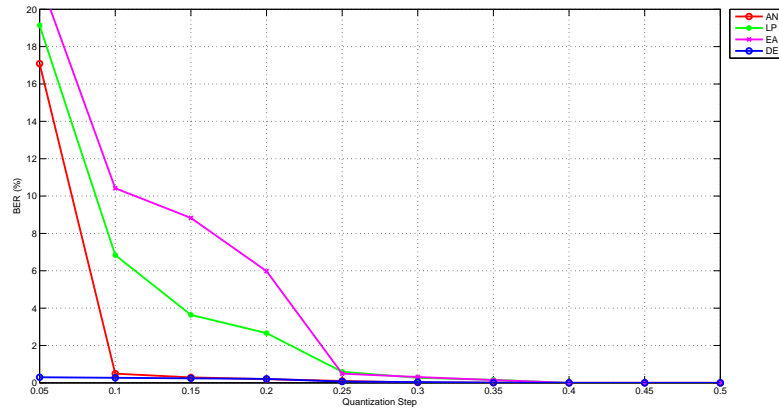


(a) Watermark image is 'Logo'

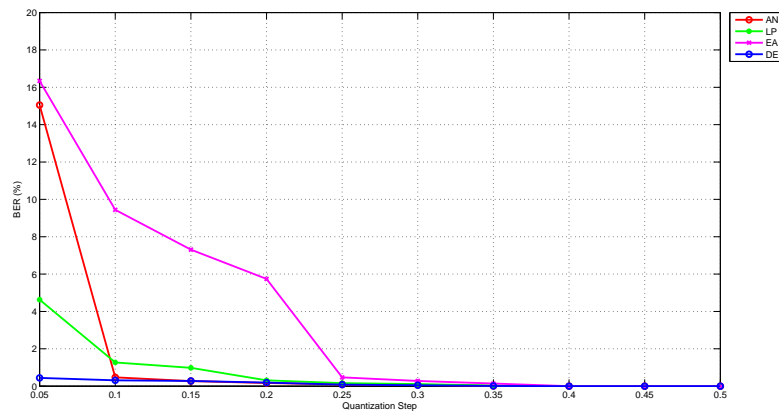


(b) Watermark image is 'Flower'

Figure 3.9: SNR (dB) value of the watermarked image of AWSVD method with varying quantization steps



(a) Watermark image is 'Logo'



(b) Watermark image is 'Flower'

Figure 3.10: BER (%) of extracted watermark of AWSVD method with varying quantization steps under AN, LP, EA, DE attacks

3.4.4 Quantization parameter

The quantization parameter Δ plays an important role in the performance of any quantization-based audio watermarking technique. The two conflicting requirements of watermarking, i.e., imperceptibility and robustness are very much dependent on the quantization parameter. As the value of Δ increases, the imperceptibility decreases. On the other hand, the robustness of the technique increases with the increased value of Δ . For an effective audio watermarking technique, a suitable Δ value is to be selected experimentally. Fig. 3.9 shows the imperceptibility of the watermarked image with changing the value of Δ . From this figure, we note that quality is quite good when $\Delta < 0.3$ as SNR is higher than 20 dB. In Fig. 3.10, the value of BER on different values of Δ is shown and $BER < 0.1$ for $\Delta > 0.25$. Compromising the SNR and BER values we set $\Delta = 0.25$ in our experiment.

3.5 Conclusions

In this work, we have proposed a secure and blind audio watermarking technique using SVD. In the proposed method, we have embedded a binary image by quantization of the largest singular value of the audio blocks. The proposed method provides a good trade-off among imperceptibility, robustness, and payload. The proposed method is faster than the SoA methods and overhead of the proposed method is also less than the existing methods. Depending on the encryption method applied on the watermark image, the proposed method can be used for copyright protection or authentication of the audio signals.

So far, we have proposed some secure watermarking techniques for copyright protection and authentication of images and audio signals. To achieve these, we need to encrypt the watermark symbols and as images are used as the watermark symbol; so image encryption plays an important role in the proposed methods.

Chapter 4

Image Encryption for Secure and Robust-Fragile Watermarking

Due to the fast growth of Internet technology, popularity of digital devices, and the style of multimedia information exchanges through the Internet, increasing the information confidentiality requirement. Since the image is an essential communication medium, image security is a significant concern.

The use of image encryption is quite prevalent in the corporate world, serving diverse purposes. For instance, in the medical industry, health records are often created and shared online, making it imperative to ensure their protection through encryption [161, 162]. Health records contain sensitive information such as patient details, medical history, symptoms, and more. Due to their confidentiality, it is crucial to protect them from unauthorized use. Similarly, in the military field, images such as maps, building positions, and enemy locations are of utmost importance. Images play a critical role in small target detection, tracking, and missile guidance, making it essential to prevent illegal access to them, which may jeopardize national security. The media industry operates on a 24×7 basis, with news being broadcast round the clock. In this industry, maintaining privacy and security of multimedia content, including images, audio, and video, is critical. Image encryption plays a vital role in safeguarding sensitive information from unauthorized access [163]. The cloud has become a popular storage option for individuals and organizations, with third-party providers storing client data remotely. Storing images in the cloud has proven to be a feasible solution for many users. However, ensuring privacy protection of these images in cloud applications is equally important [164–166].

Many traditional cryptosystem like IDEA [167], AES [168], DES [169] and RSA [5] are available to encrypt text data. These traditional cryptosystems cannot be used

to encrypt image data directly [170]. The use of a classical cryptosystem, to encrypt an image is not a good idea due to i) the huge volume of data, ii) the size of the decrypted image may not same as the original image, iii) high redundancy in data, iv) strong spatial correlation in data, v) complex mathematical computation and vi) computational overheads.

An image encryption algorithm is designed to transform an image into a cipher image using a unique encryption key. As a result, the cipher image appears to be a noisy image, making it impossible to predict anything about the original image from the cipher image. In a plain image, the correlation among neighboring pixels is often high. To reduce this correlation, the relative position of pixels can be changed by scrambling the image, thereby decreasing the correlation among adjacent pixels. After scrambling, the image will be shown as a noisy image so that it is difficult to recognize the original image. The scrambling is done using a random sequence. This is known as the confusion phase of encryption. In the confusion phase, pixels' positions are interchanged only. The scramble by random sequence applies to the large data set. It is also simple, computationally low overhead [171]. In confusion-based image encryption, the pixel intensity value does not change and as a result, the histogram of the original image and encrypted image remain the same. The diffusion phase is another step of the image encryption method in which the pixels' intensity value is modified, resulting in a uniform histogram, and the system becomes robust against different attacks. The general block diagram of the image encryption process is shown in Fig.1.2.

In previous chapters, we used the image encryption method to make the watermarking method more secure. We have studied the watermarking techniques where both binary images, as well as grayscale images, are used as the watermark. Here, we have proposed an image encryption method using Fibonacci Transformation, XOR operation, and Tribonacci Transformation, we refer to this method as *FTTIE* (Fibonacci and Tribonacci Transformations based Image Encryption). The proposed method is a plaintext sensitive method, robust against different attacks, and gives results similar to the SoA methods. To the best of our knowledge, this is the first time that the Tribonacci Transformation has been used in image encryption. The *FTTIE* method can be used in the watermarking method, to encrypt the watermark image, to establish the copyright information.

Since *FTTIE* is robust under the attacks, so this method cannot be used in fragile watermarking. In fragile watermarking, if the watermarked image is modified then the

watermark image cannot be extracted, and a noisy image will be obtained. We further extend the *FTTIE* method, using triple encryption scheme (i.e., first encrypt with a key key_1 , then decrypt using other key key_2 , and finally, encrypt again using key key_3 where key_1 and key_3 are normally same and key_1 is the hash value of the plain image and key_2 is computed from the hash value of the intermediate cipher image (obtained after the first encryption). The extended method is denoted as $FTTIE_{ext}$ and this method is designed to ensure that due to slight changes encrypted image results in a noisy decipher image and makes the method suitable for fragile watermarking. It may be noted that like any other image encryption method available in the literature, the proposed method also follows the line of Kerckhoff's principle of cryptography, i.e., our proposed encryption system is publicly available and the security of the ciphertext completely depends on the secret key.

The organization of this chapter is presented as follows. In Section 4.1, an analysis of the image encryption method is described. The review of the image encryption techniques is presented in Section 4.2. Mathematical background related to the proposed method is given in Section 4.3. In Section 4.4, the proposed image encryption method is presented. The security analysis of the proposed method is explained in Section 4.5. In Section 4.6, the extended version of the proposed scheme is presented. Finally, conclusions are given in Section 4.7.

4.1 Features to Analysis the Image Encryption Method

The performance and the applicability of an image encryption method are examined concerning different features, the features are described below.

1. **Larger key space:** An image encryption should have a large key space to withstand brute-force attack. If a method has keyspace $> 2^{100}$, with the use of current computing facilities it is impossible to break the system within a life span [172]. The key space of an encryption algorithm is determined by the total number of bits needed for the various components of the encryption scheme. The larger the key space, the more difficult it is for someone to guess the encryption key and gain unauthorized access to the encrypted data.
2. **Randomness of the cipher image:** It is known that there are strong correlations among the adjacent pixels of the original image. Due to these correlations, the value of the current pixel can be predicted from the neighboring pixels in the case of plain images. In image encryption, one of the primary objectives is

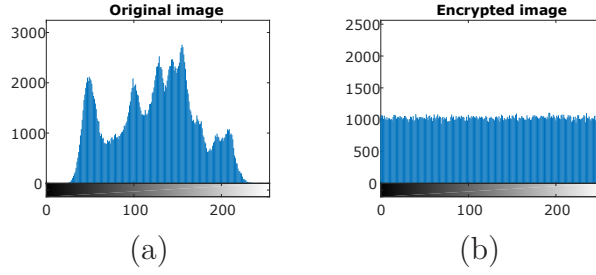


Figure 4.1: Histogram of the grayscale images: (a) Original image, (b) Encrypted image

to ensure that nothing can be inferred about the plain image from the encrypted image. The encrypted image should be wholly noisy or random. We need to measure the randomness of the encrypted image. Here, to study the randomness, we use different analyses like i) histogram analysis, ii) entropy analysis, iii) correlation analysis, and iv) χ^2 analysis.

- (a) **Histogram analysis:** The histogram of a plain (grayscale) image may be seen as in Fig. 4.1(a). From this histogram, we note that i) it has some specific peaks and ii) it does not cover the entire domain. These characteristics of the histogram of the plain images are due to the spatial correlations among the adjacent pixels. We can say that an image is random if the intensity of a pixel can be any value of the domain, i.e., $P(intensity = i) = P(intensity = j)$, for all $i, j \in \{0, 1, \dots, 2^k - 1\}$ (for k -bit image). All the intensity values are equally probable (i.e., the frequency of each intensity is almost the same). The expected phenomenon of the image encryption method in terms of the histogram of cipher image should be seen as given in Fig. 4.1(b). In the case of binary image, the frequency of occurrence of '0' and '1' should be very close, i.e., $frequency(0) \simeq frequency(1) \simeq 50\%$.
- (b) **Entropy analysis:** The entropy of a source measures the information contained in the source. A source has more information (less predictable or highly random) if the entropy of the source is large. The entropy of a source is measured by Eq. (4.1), where $p(s_i)$ is the probability of the i^{th} symbol, s_i , of the source S .

$$H(S) = - \sum_{i=0}^{N-1} p(s_i) \log_2 p(s_i) \quad (4.1)$$

When all symbols of S are equally likely, then the entropy of the source is maximum. So, when a source is genuinely random, its entropy is maximized, which is a necessary condition (i.e., maximum entropy does not indicate that the source is random). In the case of k -bit images, the maximum entropy is k .

If the entropy of an encrypted image is high, then also it may happen that for some blocks, entropy is low (i.e., some information about the image can be predicted). To test the robustness of an encryption method, local entropy (entropy at the block level) is also important. For this purpose, (p, T_B) -local Shannon entropy is computed as

$$H_{(p, T_B)} = \frac{1}{p} \sum_{i=1}^p H(B_i) \quad (4.2)$$

where p is the number of blocks and B_i 's (for $1 \leq i \leq p$) are randomly selected non-overlapping blocks with T_B pixels. The $H(30, 1936)$ -local Shannon entropy of the grayscale image lies in the range $[7.901901305, 7.903037329]$ and the local entropy of the binary image is in the $[0.46733934, 0.532666066]$ for $H(30, 2)$ [173].

- (c) **Correlation analysis:** The correlation coefficient (ρ_{xy}), between two random variables X and Y , determines the level of accuracy of a predicted value x' of $x \in X$ using $y \in Y$. The correlation coefficient (ρ_{xy}) between the random variables X and Y is computed as given in Eq. (4.3).

$$\rho_{xy} = \frac{cov(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$$

where

$$\begin{aligned} cov(X, Y) &= \frac{1}{N} \sum_{i=1}^N (x_i - E(X))(y_i - E(Y)) \\ E(X) &= \frac{1}{N} \sum_{i=1}^N x_i, \quad E(Y) = \frac{1}{N} \sum_{i=1}^N y_i \\ D(X) &= \frac{1}{N} \sum_{i=1}^N (x_i - E(X))^2, \quad D(Y) = \frac{1}{N} \sum_{i=1}^N (y_i - E(X))^2 \end{aligned} \quad (4.3)$$

In the case of an original image, the spatial correlations among the adjacent pixels are very high. For this analysis, we randomly selected pairs of

adjacent pixels. In the context of the selected pairs, the intensity of the first pixel is denoted by X , while the intensity of the second pixel is denoted by Y . In cryptographic attacks, the spatial correlation of the pixels is exploited, and one important objective of image encryption is to reduce the spatial correlation of the encrypted image. To compute the correlations, normally three adjacency relations, namely horizontal, vertical, and diagonal are taken into account. It is expected that correlations in the cipher image should be very low and if it is an attacker cannot obtain any information about the cipher image from some known pixels. So, nothing can be guessed about the plain image, and this is a very important feature of the image encryption method.

- (d) **χ^2 -Test:** The χ^2 -test calculates the difference between the observed and expected values of statistical samples. The χ^2 -test can be used as another important measure to analyze the uniformity of the cipher image's histogram (i.e., randomness) [174]. Here, we expect that the histogram is uniform and the expected frequency of any gray value $e \in [0, 2^k - 1]$ is $f_e = \frac{M \times N}{2^k}$, for an image of size $M \times N$. The χ^2 value of a cipher image is defined by Eq (4.4).

$$\chi^2 = \sum_{i=0}^{2^k-1} \frac{(f_i - f_e)^2}{f_e} \quad (4.4)$$

where f_i is the frequency of the grayvalue i in the cipher image. When the χ^2 value is high, the distribution is less analogous to the uniform distribution. If two distributions are identical, the χ^2 value is 0, indicating that the theoretical value is consistent. For an 8-bit grayscale image, when the significant level is 0.05, the ideal value is $\chi^2_{(255,0.5)} = 293.2478$ [174]. For binary image, the theoretical value of χ^2 at significant level 0.05 is 3.8415, i.e., $\chi^2_{(1,0.5)} = 3.8415$ [174].

3. **Pixel disparity analysis:** A straightforward method to establish the relationship between the plain and cipher images is to compute the pixel-wise difference between the two images. This can help identify any similarities between the two images, which can be useful for evaluating the effectiveness of an image encryption algorithm. Commonly used parameters are Mean Square Error (MSE), Peak-Signal-to-Noise-Ratio (PSNR), and Mean Absolute Error (MAE), which

are defined in Eq. (4.5).

$$\begin{aligned}
 MSE &= \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (P(i, j) - C(i, j))^2 \\
 PSNR &= 10 \times \log_{10} \frac{MAX^2}{MSE} \\
 MAE &= \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N |P(i, j) - C(i, j)|
 \end{aligned} \tag{4.5}$$

Here, P and C represent the corresponding plain and cipher images, respectively, and ‘ MAX ’ is the maximum possible intensity value of a pixel. MSE and PSNR have an inverse relation i.e., when two images are similar, MSE will be less, and in that case, the PSNR value will be high. In the case of image encryption, we expect that MSE will be as large as possible (i.e., PSNR is less).

4. **Key sensitivity:** An effective image encryption technique should be highly sensitive to the secret key both in encryption and decryption processes. An image encryption method is said to be key sensitive if the method produces two different cipher images from the same plain image when two different encryption keys (maybe a single-bit difference) are used. Similarly, key sensitivity in the decryption process reflects that if the actual key is used then the plain image can be recovered; otherwise, even due to minimum change in the decryption key the method will return a noisy image (from which no information about the plain image can be obtained). Two parameters - NPCR (Number of Pixel Change Rate) and UACI (Unified Average Changing Intensity) - are commonly used to measure the deviation between two cipher images. They are formulated given in Eq. (4.6).

$$\begin{aligned}
 NPCR &= \frac{\sum_{i=1, j=1}^{M, N} D(i, j)}{M \times N} \times 100\% \\
 UACI &= \frac{1}{M \times N} \left[\sum_{i, j}^{M, N} \frac{|C_1(i, j) - C_2(i, j)|}{MAX} \right] \times 100\%
 \end{aligned} \tag{4.6}$$

where $D(i, j)$ defined as

$$D(i, j) = \begin{cases} 1, & \text{if } C_1(i, j) \neq C_2(i, j) \\ 0, & \text{otherwise} \end{cases}$$

The formula is referring to two cipher images, C_1 and C_2 , which have the same size of $M \times N$. These cipher images are obtained from the same plain image using two different encryption keys and where ‘ MAX ’ is the largest possible intensity value of a pixel, which is either 255 or 1 for an 8-bit grayscale image or binary image, respectively. For grayscale images, the optimal values of NPCR and UACI are 100% and 33.33%, respectively [175]. For binary image, from the Eq. (4.5) we may conclude that $MAE = NPCR = UACI$. It is very simple to note that for binary images, the ideal value of NPCR is 50%.

5. **Plaintext sensitivity:** An encryption method is said to be plaintext sensitive, if two images, one obtained from the other by a minimum change in pixel intensity (may complement LSB of a pixel intensity), and give two different cipher images. NPCR and UACI are used to measure the key sensitivity of the method.
6. **Robustness against differential attacks:** Usually, an attacker would slightly modify a plain image and encrypt the images (plain and modified) with a given algorithm to obtain two cipher images. Then, an attacker tries to find the relationship between the plain image and the cipher image by comparing two encrypted images. This process is known as a ‘difference attack.’ If an image encryption system is plaintext sensitive, any change to a pixel in the plain image should result in a different encrypted image compared to the actual cipher image. This makes it difficult for an attacker to obtain any information about the plain image from the cipher image, thus increasing the security of the encryption system.

It is crucial for an image encryption method to resist differential attacks. This means that even a minor change in the plain image should result in a completely different cipher image while using the same cryptosystem. This is important to ensure the security and reliability of the encryption method. NPCR and UACI parameters are used to measure the difference between two cipher images. If a method is plaintext sensitive, then the method is also robust against differential attacks.

7. **Chosen-plaintext and Known-plaintext attacks:** In a chosen-plaintext attack, the attacker can gain temporary access to the encryption module and use common plaintext images such as ‘Black’, ‘White’, or other images to obtain information about the encryption key. This type of attack is particularly dangerous as it allows the attacker to gain information about the encryption key, which can then be used to compromise the security of the encrypted data. The

resistance against the chosen plaintext attack ensures robustness against the known-plaintext attack. In a chosen plaintext attack, the third party uses the information of some chosen plaintext-ciphertext pairs. In a chosen-plaintext attack, the attacker can use any plain image other than the actual plain image of the current cipher image they want to break. This makes the attack more versatile and dangerous, and it is important to implement measures to prevent and detect such attacks to ensure data security.

8. **Time complexity:** In the implementation of an image encryption system, the time complexity of the method is a crucial metric. For symmetric key cryptosystems, the encryption and decryption times are the same. Therefore, some research studies have only reported the encryption time and not the decryption time. It is important to consider both encryption and decryption times to accurately evaluate the performance of an image encryption system. Image encryption techniques with high time complexity are not applicable in real-time applications. Thus, a fast image encryption technique is desirable.
9. **Cropping attack:** A good image encryption method should have the capability to restore the features of the plain image after a certain degree of the cipher image is cropped out.
10. **Noise attack:** The anti-interference ability of a cryptosystem is the ability of the system to defend against a noise attack. During the transmission of the cipher image, the image may be distorted by the transmission noise, affecting the deciphered image. To test the anti-noise ability of a cryptosystem, the cipher image is distorted with some noise like salt-and-pepper noise, etc. and then decrypt the image. If the method can return a decipher image that has similar features (it may be less) to the original image then the cryptosystem is robust against the noise attack.

4.2 Literature Survey

At an earlier stage, the image encryption methods were developed on compressed data [176–178]. Among different classes of image encryption, chaotic-based techniques are quite popular due to their sensitivity to the initial parameters, pseudo-random behavior, and unpredictable motion trajectories. In the chaotic methods, a pseudo-random sequence is computed, and certain permutations are defined using this sequence. Moreover, these permutations are used to scramble the pixels, permute the bit

planes, or define the substitution matrices. One intriguing aspect of chaotic methods is that a slight change in the initial value of the parameter(s) generates quite different sequences. This chaotic behavior is matched with an encryption feature. Earlier researchers have worked using classical chaotic maps like Baker's map [179], Logistic map [180], Tent map [181], delayed coupled map [175, 182], 2D logistic-Sine-coupling map [183], 1D chaotic map [184], Lorenz chaotic system [185], etc. Recently, people have been designing different hybrid chaotic image encryption methods, like the hybrid chaotic map with an optimized substitution box [186]. In [187], a hybrid chaotic method using a 2D modified Henon map with a Sine map is proposed. Combining the Sine map, Logistic map, and Tent map, a new hybrid method is proposed in [188]. A hybrid method [189] using 1D and 2D chaotic maps to achieve image encryption is proposed, where a 2D sin-cosine cross-chaotic map is used in the confusion phase, and a 1D Logistic-Tent map is used to diffuse the image. Recently, high-dimensional chaotic (hyper-chaotic) systems have gained popularity [190, 191] because these systems increase the key space and also become robust against attacks. In [192], a 5D hyper-chaotic system is proposed to encrypt color images in which the plain image is decomposed into sub-bands using a complex wavelet transform. Then sub-bands are diffused using secret keys obtained from a 5D chaotic map. A novel adjustable visual encryption scheme using a 6D hyper-chaotic system is proposed in [193]. A new image cryptosystem using a hyper-chaotic system and a Fibonacci Q-matrix is proposed in [194].

Another class, based on DNA computing, of image encryption methods is now getting popular. In recent years, many DNA sequence-based encryption methods have been designed. In [195], a DNA sequence-based image encryption method is proposed. In this method, the image is converted into a DNA matrix, and then a 2D Logistic map is used to define circular row and column permutations. Finally, a row-by-row diffusion technique is adopted. The initial parameters are calculated from the SHA256 of the plain image. A 2D Henon-Sine map and DNA coding-based image encryption method are proposed in [196], where S-box is first employed to synthesize cryptographic effects of the complicated DNA encryption operations, and security evaluation is conducted using 2D Henon-Sine map and DNA coding. In [197], the research proposes a novel hybrid method combining the power of DNA and the randomness of a Binary Search Tree (BST), which creates a more accurate encryption method. In [198], a two-phase secure image encryption method is proposed where the concepts of DNA and RNA are used. In this method, the initial cipher image is created by applying DNA rules, the DNA XOR operator, and the chaotic function.

Then, the codon's truth table for RNA and secret key are exploited to obtain the final encrypted image. A new Chaotic Evolutionary Biomolecules Model (CEBM) based on the concepts of biological molecules (DNA and RNA) is presented in [199] for image encryption. In this work, the Chaotic Rate operator is used for permutation, and DNA XOR and RNA codon complement operators are used for diffusion. Some other recent DNA-based encryption methods are noticed in [200–205].

Nowadays, medical images are also vital data that are communicated through the Internet. So, the security of medical images like MRI, ultrasound sonography, X-ray, etc., is also essential because these images contain private information. Some recent proposals on medical image encryption are given in [206, 161, 207–210].

Chaos-based image encryption methods are popular in the research community. At the same time, researchers also proposed non-chaotic methods to scramble the images (i.e., to define the permutation). Non-chaotic techniques such as the Arnold Transform [211–213], Fibonacci Transform [214–216], Gray code [217], Rubik's cube principle [218], cyclic group [219], prime factorization [220], binary tree traversal [221], and iterative numerical methods for root finding [222–224] are used to define the permutation. The SCAN-based permutation is another group of techniques used in image encryption. Different encryption techniques based on different scan patterns are studied in [225, 226]. A sine curve-based pattern is utilized in [227] to encrypt an image. In [228], a circular scan pattern is employed to define the permutation. A double spiral scan-based method is proposed in [229]. Recently, a novel zig-zag scan-based feedback convolution algorithm has been designed for image encryption [230].

Binary (bi-level) images are used in many electronic applications like fingerprint analysis, motion detection, character recognition, etc. Moreover, binary images appear as the output of various tasks such as half-toning, edge detection, thresholding, segmentation, etc. Different devices like laser printers, fax machines, and biometric devices can only handle bi-level images. Due to the simplicity and various applications of binary images, encryption of binary images has been proposed by the researchers. In [231], a new technique to encrypt a binary image has been presented based on scan patterns. In this approach, a binary image is encrypted by considering different scan patterns at the same level in the scan tree structure and using a two-dimensional run-encoding method. The algorithm shows very good security and a satisfactory compression ratio. A simple image encryption technique has been designed by Jai et.al. [232] to encrypt a binary image. The idea of encryption is based on interference two-phase-only masks. In the encryption process, a random phase mask is used to modulate the binary image followed by separation into two phase-only masks. The ex-

perimental output is satisfactory in the field of cryptography and the time complexity is very low. In [233], an image encryption algorithm has been developed for binary images where the bit error issue has been solved during transmission of the image through an insecure channel. In their encryption scheme, an error correction code is applied to the encrypted image to protect it against channel errors. In [234], a binary image encryption has been developed. This approach encrypts a binary image based on the combination of confusion and diffusion process. The Logistic chaotic sequence is used to scramble (confuse) the binary image and then the diffusion process is applied to obtain the encrypted image. This algorithm uses a large key space and the results show that it is suitable for security of the binary images over the Internet. In [235], a novel image encryption has been presented to encrypt a binary image. In step one, they adopt a basis to represent the image for reducing the amount of data. In the second step, the test plain image is split into r blocks to obtain new images (the same size as the plain image) and these new bases are used to generate a key image and encrypted image. The key image is used in the decryption process to obtain decrypted images by subtracting the key image from the encrypted image. Radhakrishnan et. al. [236], have suggested a symmetric encryption algorithm for binary images based on diffuse representation. During the encryption process, a binary image is XORed with a random matrix and then it is divided into several sub-images (non-overlapping). The method can be used to encrypt grayscale images by considering each bit-plane as a binary image.

4.3 Mathematical Background

In the encryption process, we have used the Fibonacci Transform to scramble the pixels, XOR operation, and the Tribonacci Transform is applied to modify the pixels' intensity. First, we study the mathematics behind these transformations and then discuss the proposed image encryption method.

4.3.1 Fibonacci Numbers

In mathematics, the sequence $\{1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \dots\}$ is known as the Fibonacci sequence [237], [238]. The above sequence can recursively be defined as given in Eq. (4.7).

$$f_k = \begin{cases} 1, & \text{if } k = 1 \\ 1, & \text{if } k = 2 \\ f_{k-1} + f_{k-2}, & \text{otherwise} \end{cases} \quad (4.7)$$

The above relation is extended for any $k \in \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$, the numbers are given in Table 4.1. The Fibonacci sequence is powerful and possesses some amazing properties. Researchers have used elementary matrix operations, determinants, and their properties to generate a class of identities for generalized Fibonacci numbers [237]. In this work, we study three properties of generalized Fibonacci sequences such as Cassini's identity, d'Ocagne's identity, and Catalan's identity, which are given below.

$$\text{Cassini's identity:} \quad f_{q+1}f_{q-1} - f_q^2 = (-1)^q \quad (4.8)$$

$$\text{d'Ocagne's identity:} \quad f_{p+1}f_q - f_p f_{q+1} = (-1)^p f_{q-p} \quad (4.9)$$

$$\text{Catalan's identity:} \quad f_q^2 - f_{q+p}f_{q-p} = (-1)^{q-p} f_p^2 \quad (4.10)$$

The above identities can be expressed as the determinants as given below.

$$\text{Cassini's identity:} \quad T_{Cas} = \begin{vmatrix} f_{q+1} & f_q \\ f_q & f_{q-1} \end{vmatrix} \quad (4.11)$$

$$\text{d'Ocagne's identity:} \quad T_{dOc} = \begin{vmatrix} f_{p+1} & f_{q+1} \\ f_p & f_q \end{vmatrix} \quad (4.12)$$

$$\text{Catalan's identity:} \quad T_{Cat} = \begin{vmatrix} f_q & f_{q+p} \\ f_{q-p} & f_q \end{vmatrix} \quad (4.13)$$

The corresponding matrix of any of the above identities can be used to transform data in cryptography under modulo n if and only if the $\gcd(\det, n) \equiv 1 \pmod{n}$, where 'det' is the determinant of the matrix. From the definition of the Fibonacci series, we note that $f_1 = 1$ and $f_2 = 1$. So, for any n , we may note the following about the transformation matrices:

1. $T_{Cas} = \pm 1$, for any value of q .
2. $T_{dOc} = \pm 1$, when $q - p = 1$ or 2 .
3. $T_{Cat} = \pm 1$, if $p \in \{1, 2\}$.

From the above definitions (Eq. (4.8)-(4.10)), it is obvious that Cassini's identity is a special case of the other two identities as given below.

$$\begin{aligned} \text{d'Ocagne's identity} &\rightarrow \text{Cassini's identity, when } p+1 = q \\ \text{Catalan's identity} &\rightarrow \text{Cassini's identity, when } p = 1 \end{aligned}$$

Table 4.1: Fibonacci and Tribonacci numbers

k	\dots	-5	-4	-3	-2	-1	0	1	2	3	4	5	\dots
f_k	\dots	5	-3	2	-1	1	0	1	1	2	3	5	\dots
t_k	\dots	-3	2	0	-1	1	0	0	1	1	2	4	\dots

4.3.2 Tribonacci Numbers

Tribonacci numbers [239] are a generalization of Fibonacci numbers, denoted as t_k , for $k = 0, 1, 2, 3, \dots$, and defined by the recurrence relation as given below in Eq. (4.14).

$$t_{k+1} = t_k + t_{k-1} + t_{k-2}, \quad \text{where } t_0 = t_1 = 0, \quad t_2 = 1 \quad (4.14)$$

The Tribonacci negative numbers t_{-k} , for $k = 1, 2, 3, \dots$, satisfies the recurrence relation given in Eq (4.15).

$$t_{-k} = \begin{vmatrix} t_{k+1} & t_{k+2} \\ t_k & t_{k+1} \end{vmatrix} \quad (4.15)$$

Eq (4.14) and (4.15) provide the Tribonacci numbers t_k , for $k = 0, \pm 1, \pm 2, \pm 3, \dots$ shown in Table 4.1.

In [239], a Tribonacci coding technique is proposed, and this method is based on the Tribonacci numbers. For this purpose, a matrix $H_{3 \times 3}$ is introduced. The matrix H is defined as

$$H = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} t_3 & t_2 + t_1 & t_2 \\ t_2 & t_1 + t_0 & t_1 \\ t_1 & t_0 + t_{-1} & t_0 \end{pmatrix} \quad (4.16)$$

where $\det(H) = 1$ and the inverse of H is given as

$$H^{-1} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{pmatrix}$$

$$= \begin{pmatrix} t_0^2 - t_{-1}t_1 & t_{-1}t_2 - t_0t_1 & t_1^2 - t_0t_2 \\ t_1^2 - t_0t_2 & t_0t_3 - t_1t_2 & t_2^2 - t_1t_3 \\ t_0t_2 + t_{-1}t_2 - t_1^2 - t_0t_1 & t_1^2 + t_1t_2 - t_0t_3 - t_{-1}t_3 & t_1t_3 + t_0t_3 - t_2^2 - t_1t_2 \end{pmatrix} \quad (4.17)$$

The above article has also provided two important methods to compute H^k ($k \in \mathbb{Z}$).

The positive power of H (i.e., $H^k : k \in \mathbb{N}$) is computed as

$$H^k = \begin{pmatrix} t_{k+2} & t_{k+1} + t_k & t_{k+1} \\ t_{k+1} & t_k + t_{k-1} & t_k \\ t_k & t_{k-1} + t_{k-2} & t_{k-1} \end{pmatrix} \quad (4.18)$$

The negative power of H (i.e., $H^{-k} : k \in \mathbb{N}$) is computed as

$$H^{-k} = \begin{pmatrix} t_{k-1}^2 - t_{k-2}t_k & t_{k-2}t_{k+1} - t_{k-1}t_k & t_k^2 - t_{k-1}t_{k+1} \\ t_k^2 - t_{k-1}t_{k+1} & t_{k-1}t_{k+2} - t_k t_{k+1} & t_{k+1}^2 - t_k t_{k+2} \\ (t_{k-1} + t_{k-2})t_{k+1} - & t_k(t_k + t_{k+1}) - & (t_k + t_{k-1})t_{k+2} - \\ (t_k + t_{k-1})t_k & (t_{k-1} + t_{k-2})t_{k+2} & (t_{k+1} + t_k)t_{k+1} \end{pmatrix} \quad (4.19)$$

The Eqs. (4.18) and (4.19) can be easily established by using mathematical induction. From the above definitions of H^p , $p \in \mathbb{Z}$, the following properties can be easily proved.

P1: $H^p = H^{p-1} + H^{p-2} + H^{p-3}$

P2: $H^p H^q = H^q H^p = H^{p+q}$ ($p, q = \pm 0, \pm 1, \pm 2, \dots$)

P3: $\det H^p = 1$

So, from the above discussion, we may note that a Tribonacci matrix H^i for any i can be used to transform data into another domain, and the original data can be retrieved since H^i is invertible. Therefore, H^i can be used in the encryption method.

4.4 Proposed Image Encryption Method

This section introduces the FTTIE method and evaluates its performance. The extended method, FTTIE_{ext}, is presented and evaluated in the following section. The proposed FTTIE encryption method consists of three phases:

1. Key generation phase, in this phase, the key for the encryption is generated, and this is derived from the plain image.
2. The confusion phase scrambles the pixel positions using the Fibonacci Transformation.

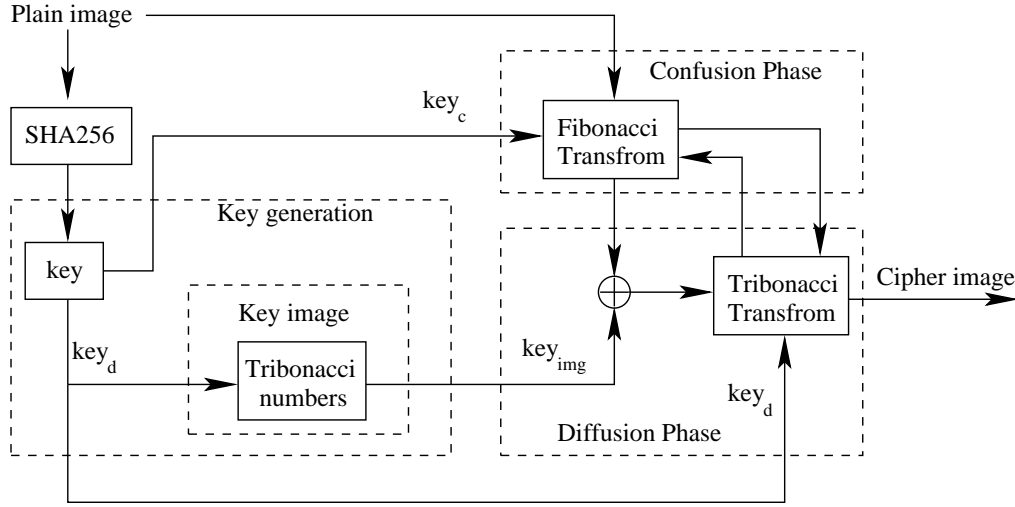


Figure 4.2: Block diagram of the proposed FTTIE method.

3. The diffusion phase is applied to modify the intensity values of the pixels, where XOR operation and Tribonacci Transformation are used.

The last two phases of the proposed method are performed iteratively in a loop. Fig. 4.2 shows the block diagram of the proposed method. The following subsections present the phases and a summary of the proposed image encryption method.

4.4.1 Key generation

In the present method, the key to encrypt the image is derived from the given plain image. The hash value of the image is computed using the SHA256 method, and the hash value of the image is key . The size of the key is 256 bits. The hash value is sensitive and depends on the input image. The key is partitioned into two halves (key_1 and key_2), each with 128 bits. To get key_i ($i = 1, 2$), we may consider $key_1 = key(1 : 128)$ and $key_2 = key(129 : 256)$ or $key_1 = key(1 : 2 : 256)$ and $key_2 = key(2 : 2 : 256)$ or some other techniques. In this work, we have considered the first one (i.e., $key_1 = key(1 : 128)$ and $key_2 = key(129 : 256)$). From these two keys, a single key is derived as $key_1 \oplus key_2$, which is considered as the confusion key (key_c) (i.e., $key_c = key_1 \oplus key_2$). The key key_2 is considered the diffusion key key_d .

In this phase, the key key_d is used to define a key image (key_{img}) of the same size as the plain image. This key image is used in the diffusion phase to modify the pixel intensity. A sequence of Tribonacci numbers $Tri = \{t_{T+1}, \dots, t_{T+S}\}$ and the threshold

Algorithm 7 : KeyGeneration ($Img, key_c, key_d, key_{img}$)

Input: Img the plain image

Output: Keys $\{key_c, key_d\}$ and key image key_{img}

Step 1. $key_c(1 : 128) = 0, key_d(1 : 128) = 0$ Step 2. $[M \ N] = \text{Size}(Img), S = M \times N$ Step 3. $key(1 : 256) = \text{SHA256}(Img)$ Step 4. For $i = 1$ to 128 a. $key_c(i) = key(i)$ b. $key_d(i) = key(128 + i)$ Step 5. $key_c = key_c \oplus key_d$

Step 6. //Compute the key image

 a. Determine threshold T using key_d b. Compute the Tribonacci sequence $Tri = \{t_{T+1}, \dots, t_{T+S}\}$ c. $key_{img_{M \times N}} \leftarrow \text{mod}(Tri, 256)$ Step 7. Return $\{key_c, key_d, key_{img}\}$

value T is computed using key_d , and t_j is the j^{th} Tribonacci number (computed using Eq. (4.14)) and, S is the number of pixels in the plain image. The sequence $\text{mod}(Tri, 256)$ defines an image of the same size as the plain image.

The algorithmic sketch of the key generation phase is given in **Algorithm 7: KeyGeneration()**. From the symmetries of the above steps, it may be noted that if the same key ‘key’ is used then key_c , key_d , and the key image key_{img} can be obtained.

4.4.2 The Confusion phase

In the confusion phase, we applied a Fibonacci Transformation to permute the positions of the pixels. We note that any matrix corresponding to T_{Cas} , T_{dOc} , or T_{Cat} can be used as a transformation matrix to reposition the pixels. It is also worth noting that a single variable can parameterize these transformations. Again, Cassini’s identity is a particular case of the other two identities. In this experiment, we employed Cassin’s identity in the confusion phase. The confusion key key_c is used to determine the value q of the T_{Cas} , and then pixels are scrambled. To compute q , we consider

Algorithm 8 : ConfusionPhase ($Img_{N \times N}$, key_c , pr , Img_{conf})

Input: Plain image Img , confusion key key_c , prime pr

Output: Confused image Img_{conf}

Step 1. $q = \text{mod}(key_c, pr) + 3$

Step 2. For $r = 1$ to N

a. For $c = 1$ to N

i. Compute r' and c' using Eq. (4.20)

ii. $Img_{conf}(r', c') = Img(r, c)$

Step 3. Return Img_{conf}

$q = \text{mod}(key_c, pr) + 3$ where pr is a prime number with moderate value and make it public. For an image of size $N \times N$, let (r, c) be the coordinate of a pixel, and after transformation, the new coordinate is (r', c') then

$$\begin{pmatrix} r' \\ c' \end{pmatrix} = \begin{pmatrix} f_{q+1} & f_q \\ f_q & f_{q-1} \end{pmatrix} * \begin{pmatrix} r \\ c \end{pmatrix} \text{ mod } N. \quad (4.20)$$

Algorithm 8: ConfusionPhase(), illustrates the confusion process of the proposed method. Concerning the same key, the same transformation can be derived; therefore, the confusion phase is invertible.

4.4.3 The Diffusion phase

The confusion phase returns a scrambled image (Img_{conf}), where the intensity of the pixels remains the same. As a result, the intensity profile of both the plain and confused images remains unchanged. Therefore, an attacker may guess the original image from the histogram profile of the confused image. To resolve this problem, we need to modify the intensity of the pixels so that nothing can be predicted about the plain image. In this phase, two operations have been executed to achieve the goal. The operations are: i) the scrambled image is XOR-ed with the image key_{img} ; ii) the pixels' values are changed by the Tribonacci Transformation (TT).

The intensity value of the pixels can be changed by applying the TT, except in the case of a pure black image. If the pure black image is transformed by TT, then it results in the same black image, i.e., the method fails to encrypt the pure black image. To

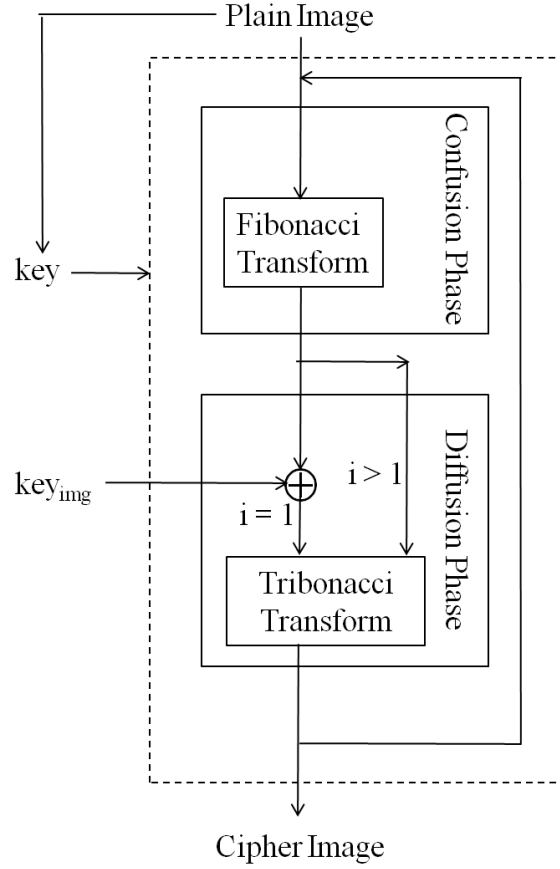


Figure 4.3: Simplified diagram of the FTTIE method which shows the looping between the confusion and diffusion phases.

change the black image into some random image, the Img_{conf} is XOR-ed with key_{img} which is defined with the diffusion key key_d . This XOR operation is applied at the very fast iteration, and for the remaining iterations of the diffusion phase, the XOR operation is ignored. A simplified diagram of the proposed *FTTIE* method is shown in Fig. 4.3, which shows the looping between the confusion and diffusion phases.

The TT is defined by a 3×3 matrix, the intermediate image is partitioned into groups of three pixels and the method is as follows. The TT is applied on a group of three pixels, as given in Eq. (4.21).

$$\begin{pmatrix} p'_1 \\ p'_2 \\ p'_3 \end{pmatrix} = H^k * \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \bmod 2^r \quad (4.21)$$

where $\{p_1, p_2, p_3\}$ are pixels' intensity, H^k is the transformation matrix, and the value of k is determined from key_d . The inverse transformation (called ITT) is given in Eq. (4.22).

$$\begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = H^{-k} * \begin{pmatrix} p'_1 \\ p'_2 \\ p'_3 \end{pmatrix} \bmod 2^r \quad (4.22)$$

In the above transformation, the pixels of the image are partitioned into groups and each group has three pixels, except the last group may have fewer pixels. Assume we have r groups, G_1, G_2, \dots, G_r and pixels in group G_i are denoted as $\{p_{i,1}, p_{i,2}, p_{i,3}\}$ (i.e., $G_i = \{p_{i,1}, p_{i,2}, p_{i,3}\}$). There are three cases:

1. The Number of pixels in the image is $3r$,
 - (a) At the time of encryption, each group G_i ($1 \leq i \leq r$) is transformed by Eq. (4.21) and obtained G'_i , i.e., $G'_i = TT(G_i)$.
 - (b) At the time of decryption, G'_i (for $1 \leq i \leq r$) is transformed back into G_i with the help of Eq. (4.22) (i.e., $G_i = ITT(G'_i)$).
2. The Number of pixels is $3(r-1) + 1$, i.e., $G_r = \{p_{r,1}\}$

In the encryption process,

- (a) First $(r-1)$ groups $\{G_1, G_2, \dots, G_{r-1}\}$ are transformed by Eq. (4.21).
- (b) Then, G_r is redefined as $G_r = \{p'_{r-1,2}, p'_{r-1,3}, p_{r,1}\}$ and transformed as $G'_r = \{p''_{r-1,2}, p''_{r-1,3}, p'_{r,1}\} = TT(G_r)$ using Eq. (4.21).
- (c) The diffused image is defined as $G'_1 || G'_2 || \dots || G'_{r-2} || G'_r || \{p'_{r-1,1}\}$.

In the decryption process,

- (a) First $r-2$ groups of the diffuse image are inversely transformed as $\{G_1, G_2, \dots, G_{r-2}\}$ (i.e., $G_i = ITT(G'_i)$, for $1 \leq i \leq r-2$).
- (b) Then, compute $\{p'_{r-1,2}, p'_{r-1,3}, p_{r,1}\} = ITT(G'_r)$.
- (c) Next, compute $\{p_{r-1,1}, p_{r-1,2}, p_{r-1,3}\} = ITT(p'_{r-1,1}, p'_{r-1,2}, p'_{r-1,3})$.

The previous version of the diffused image can be obtained as

$$G_1 || G_2 || \dots || G_{r-2} || \{p_{r-1,1}, p_{r-1,2}, p_{r-1,3}\} || \{p_{r,1}\}$$

3. Consider the case of $3(r-1) + 2$ pixels, i.e., $G_r = \{p_{r,1}, p_{r,2}\}$

In the encryption process,

Algorithm 9 : DiffusionPhase ($Img_{conf}, key_{img}, key_d, i, Img_{enc}$)

Input: Confused image Img_{conf} , key image key_{img} , diffusion key key_d , iteration number i

Output: Cipher image Img_{enc}

-
- Step 1. If ($i == 1$) $Img_{temp} = Img_{conf} \oplus key_{img}$ //iteration# 1
 Else $Img_{temp} = Img_{conf}$ //iteration# >1
- Step 2. $k = \text{mod}(key_d, 5) + 1$ //Determine H^k
- Step 3. $Img_{enc} \leftarrow \phi$; $temp \leftarrow \phi$
- Step 4. Select 3 pixels $\{p_1, p_2, p_3\}$ from Img_{temp}
 a. $Img_{enc} = Img_{enc} || temp$
 b. $\{p'_1, p'_2, p'_3\} = TT(\{p_1, p_2, p_3\})$
 c. $temp = \{p'_1, p'_2, p'_3\}$
 d. If three or more pixels are available in Img_{temp} GOTO Step 4
- Step 5. Is there is no pixel in Img_{temp}
 a. $Img_{enc} = Img_{enc} || temp$
 b. GOTO Step 8
- Step 6. If there is one pixel, p_1 , in Img_{temp}
 a. $\{p''_2, p''_3, p'_1\} = TT(\{temp(2), temp(3), p_1\})^\xi$
 b. $Img_{enc} = Img_{enc} || \{p''_2, p''_3, p'_1\} || \{temp(1)\}$
 c. GOTO Step 8
- Step 7. If there are two pixels, $\{p_1, p_2\}$, in Img_{temp}
 a. $\{p''_3, p'_1, p'_2\} = TT(\{temp(3), p_1, p_2\})$
 b. $Img_{enc} = Img_{enc} || \{p''_3, p'_1, p'_2\} || \{temp(1), temp(2)\}$
- Step 8. Return Img_{enc}
-

ξ ' $temp(i)$ ' represents the i^{th} element of $temp$

- (a) Transform first $(r - 1)$ groups and gives $G'_1, G'_2, \dots, G'_{r-1}$.
- (b) $p'_{r-1,3}$ of G'_{r-1} is combined with G_r and gives $G_r = \{p'_{r-1,3}, p_{r,1}, p_{r,2}\}$
- (c) G_r is transformed by Eq. (4.21), obtained $Gr' = \{p''_{r-1,3}, p'_{r,1}, p'_{r,2}\}$.
- (d) The diffuse image is $G'_1 || G'_2 || \dots || G'_{r-2} || Gr' || \{p'_{r-1,1}, p'_{r-1,2}\}$.

In the decryption process,

- (a) First $r-2$ groups of the diffuse image are inversely transformed as $\{G_1, G_2, \dots, G_{r-2}\}$ (i.e., $G_i = ITT(G'_i)$, for $1 \leq i \leq r-2$).
- (b) Then, compute $\{p'_{r-1,3}, p_{r,1}, p_{r,2}\} = ITT(G'_r)$.
- (c) Next, compute $\{p_{r-1,1}, p_{r-1,2}, p_{r-1,3}\} = ITT(p'_{r-1,1}, p'_{r-1,2}, p'_{r-1,3})$.

The previous version of the diffused image can be obtained as

$$G_1 || G_2 || \dots || G_{r-2} || \{p_{r-1,1}, p_{r-1,2}, p_{r-1,3}\} || \{p_{r,1}, p_{r,2}\}$$

In the above process, to handle the pixels, when we have either one or two pixels in the last group, we merge these pixels with some transformed pixels from the immediate previous group. This technique ensures that the size of the input and output images will remain the same. This technique is known as ‘cipher stealing’ in cryptography.

The outline of the proposed diffusion phase is presented in **Algorithm 9: DiffusionPhase()**. Here, the two operations, namely, TT and XOR, are used, and these operations are invertible. Therefore, the diffusion process of the proposed method is also invertible.

Algorithm 10 : ImageEncryption ($Img_{N \times N}$, itr , Img_{enc})

Input: Plain image Img , no of iterations itr

Output: Cipher image Img_{enc}

Step 1. $[key_c, key_d, key_{img}] = \text{KeyGeneration}(Img)$

Step 2. For $i = 1$ to itr

- a. $Img_{conf} = \text{ConfusionPhase}(Img, key_c)$
- b. $Img_{enc} = \text{DiffusionPhase}(Img_{conf}, key_{img}, key_d, i)$
- c. $Img = Img_{enc}$

Step 4. Return Img_{enc}

4.4.4 Image encryption

This section provides a summary of the proposed image encryption technique. From the plain image, the hash value of the image is computed using the SHA256 algorithm,

and then keys (key_c , key_d) and key image (key_{img}) are computed. Then, the confusion and diffusion phases are executed within the loop. **Algorithm 10: ImageEncryption()**, demonstrates the algorithmic structure of the proposed image encryption method. Since each step of the proposed image encryption method is invertible, we also have an image decryption method to reconstruct the plain image. Let us refer to the image decryption method as **Algorithm 11: ImageDecryption()** (the code of this algorithm is not included). The decryption algorithm executes as an encryption method with the corresponding inverse operation and in reverse order. Here, we have ignored the pseudo-code of the decryption algorithm.

4.4.5 Experimental Results

This section presents the performance evaluation of the proposed image encryption method and compares it with other SoA methods. In this experiment, we have used both grayscale and binary images to establish the applicability of the proposed image encryption method in secure watermarking.

The test dataset consists of ten images, including five standard test images commonly used in the image processing community (USC-SIPI database) [240]. The remaining five images are synthetic and were generated using a computer program. These images are shown in Fig. 4.4. This figure contains ten images, where (a)-(e) are standard test images commonly used in the image processing community (USC-SIPI database) and (f)-(j) are synthetic images. The synthetic images include an entirely black image with an intensity value of 0 (binary representation is '00000000' for an 8-bit image), a purely white image with pixel intensity of 255 (binary representation '11111111'), a checkerboard image with 50% black and 50% white pixels, a constant image with an intensity value of 170 (binary representation '10101010'), and another constant image with an intensity value of 85 (binary representation '01010101'). Corresponding binary images are shown on the right side. From the figure, we note that (f') and (j') are black images and (g') and (i') are white images. So, there are eight binary images for testing purposes.

In the proposed method, first, we compute the hash value of the plain image using SHA256, and this hash value is considered the key of the proposed method. The size of the key is 256 bits. From this key, the keys of the confusion phase (key_c) and diffusion phase (key_d) are computed, and a key image (key_{img}) is generated. In the confusion phase, to define the Fibonacci transform, we set $pr = 5$, which is public information. In the diffusion phase, the transformation matrix is H^k where $k = \text{mod}(key_d, 5) + 1$. The number of iterations is 3 (i.e., $itr = 3$). The output of the proposed schemes

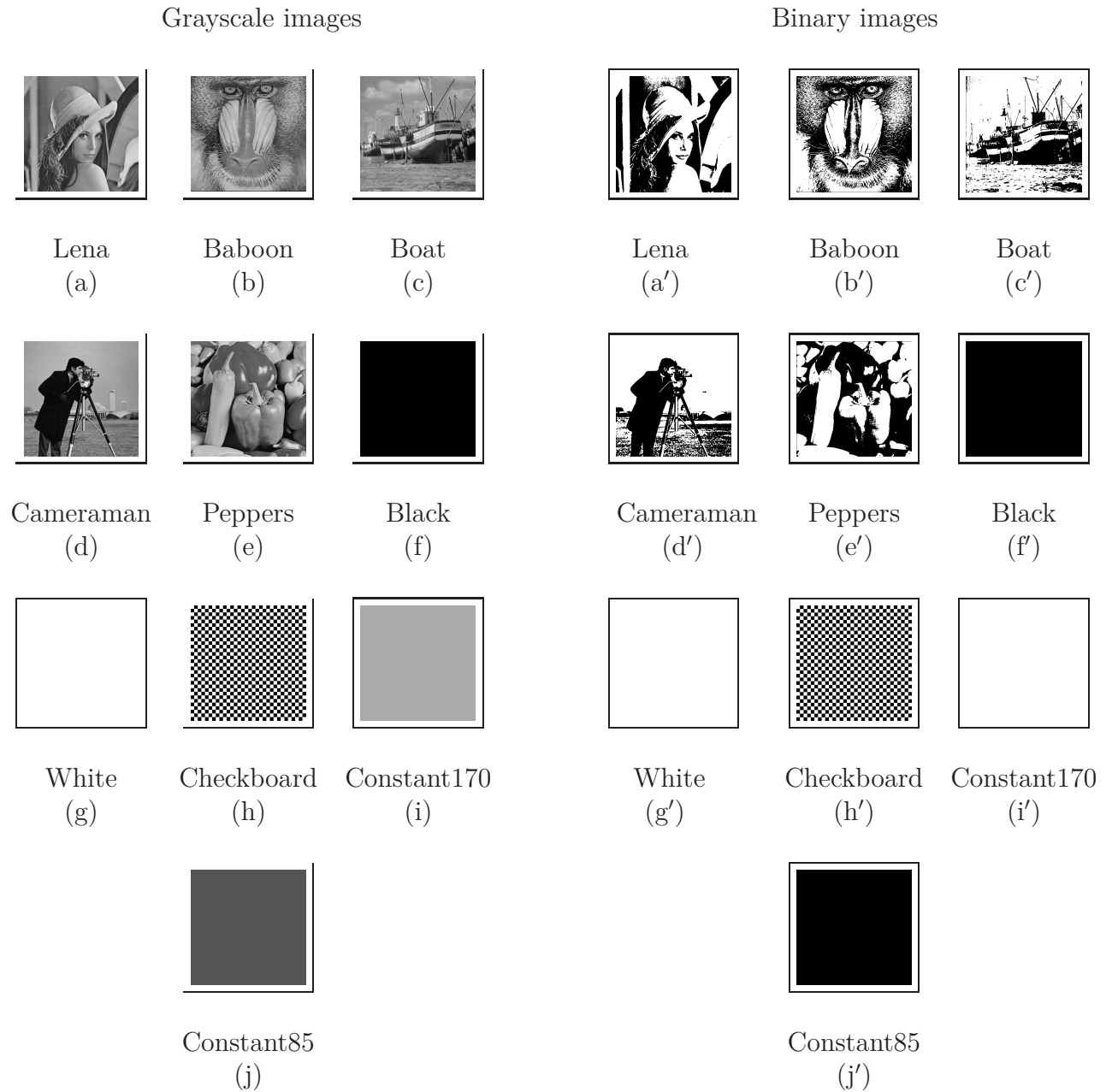


Figure 4.4: Test images used in this experiment: grayscale and corresponding binary images

on the test images is shown in Fig. 4.5. From this result, we may note that all the output images are noisy irrespective of the input image, and so the plain image cannot be predicted from the encryption image. The reconstructed images with the correct and incorrect decryption keys are shown in Fig. 4.6 and Fig. 4.7, respectively. These

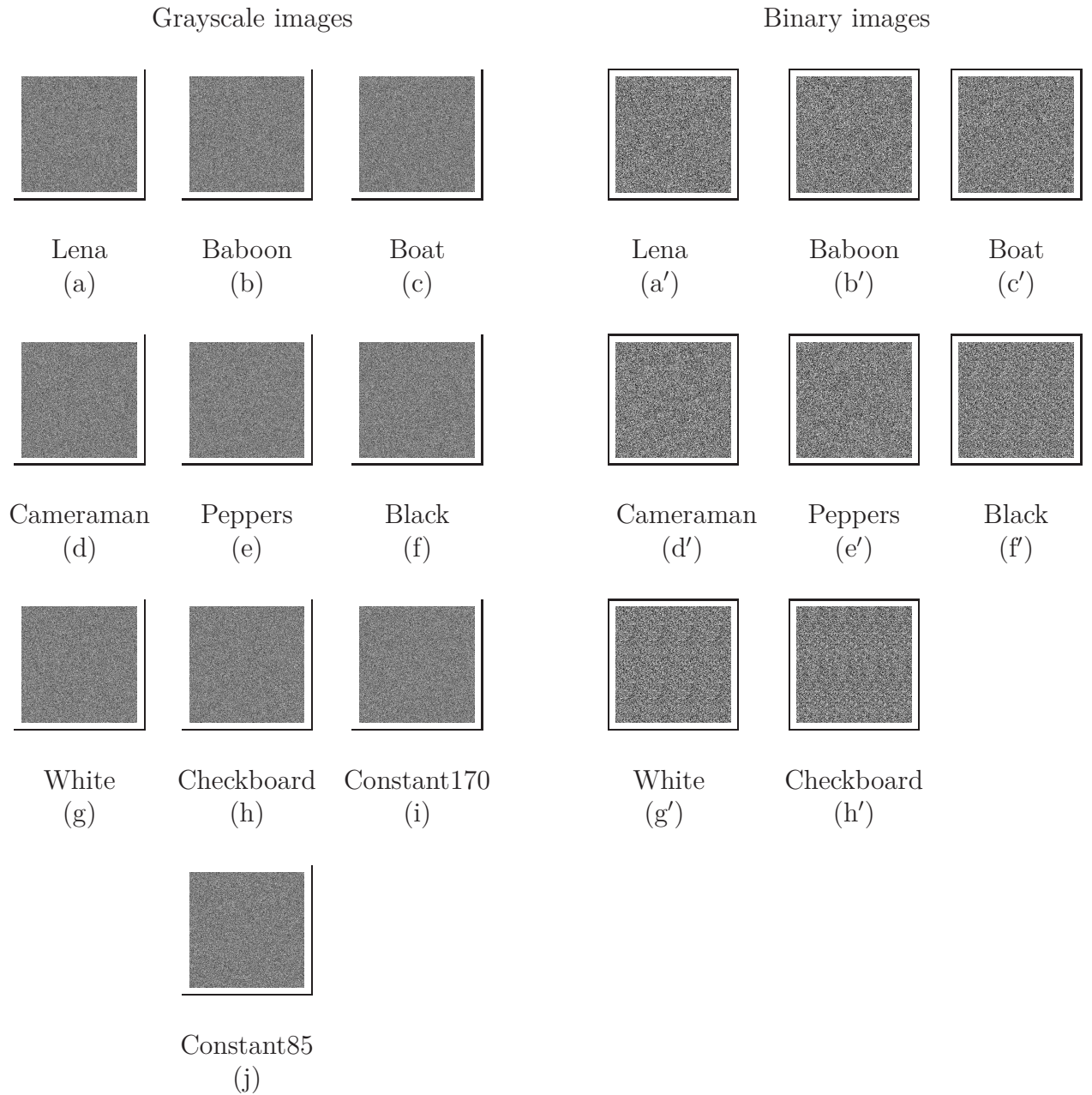


Figure 4.5: Encrypted images using FTTIE method

results demonstrate that the encrypted images can be perfectly reconstructed using the correct decryption key. However, using an incorrect decryption key, it is impossible to guess anything about the original image from the reconstructed image. The proposed method's performance is convincing and can be applied to various applications.

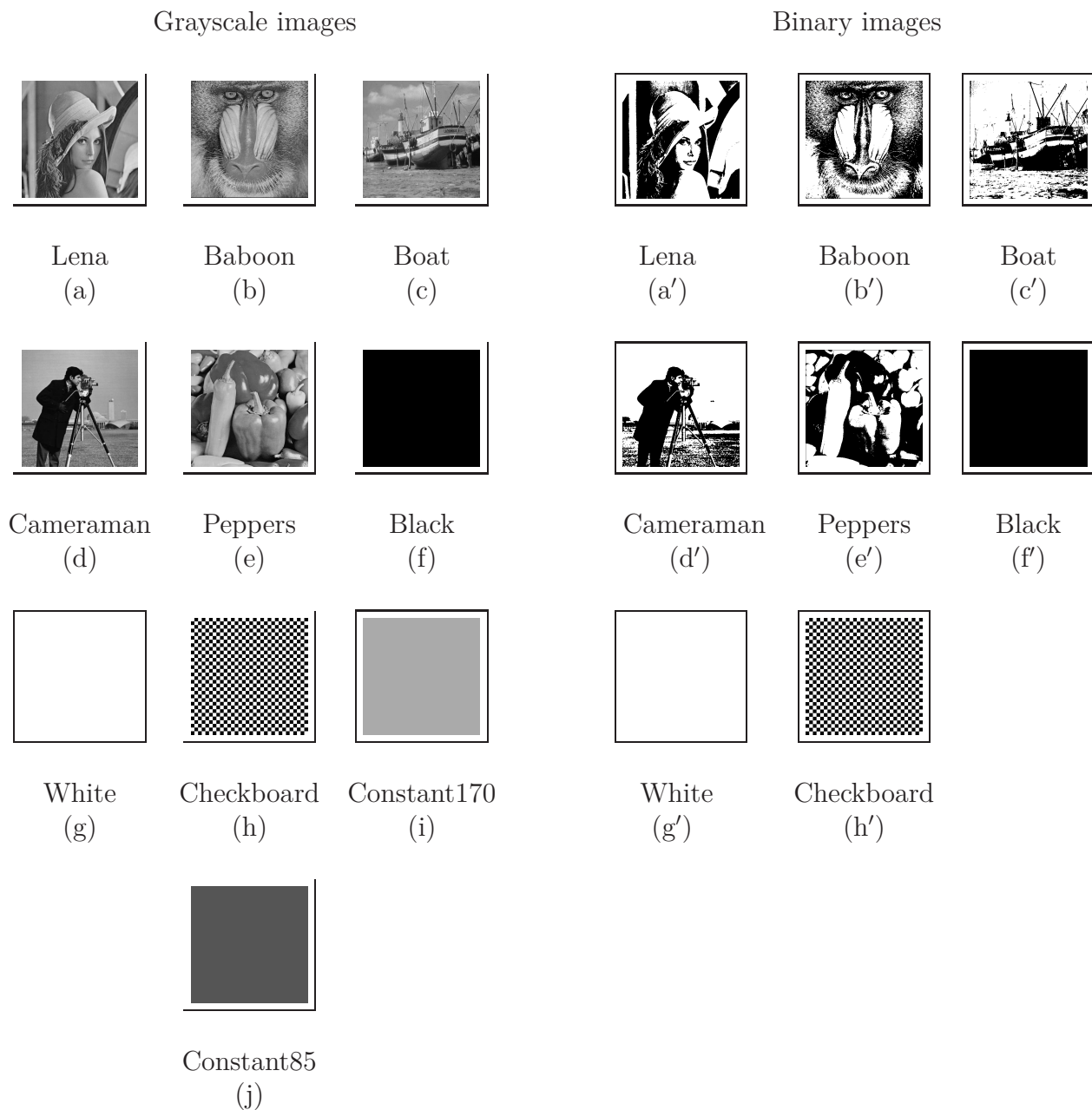


Figure 4.6: Decipher images using FTTIE method with correct decryption key

4.5 Security analysis

This section establishes the usability of the proposed image encryption method by analyzing its performance in terms of the parameters presented in Section 4.1. Moreover, we compare the proposed method's performance with SoA methods, including M1 [230], M2 [198], M3 [199], M4 [197], M5 [222], M6 [224], and M7 [174]. We have

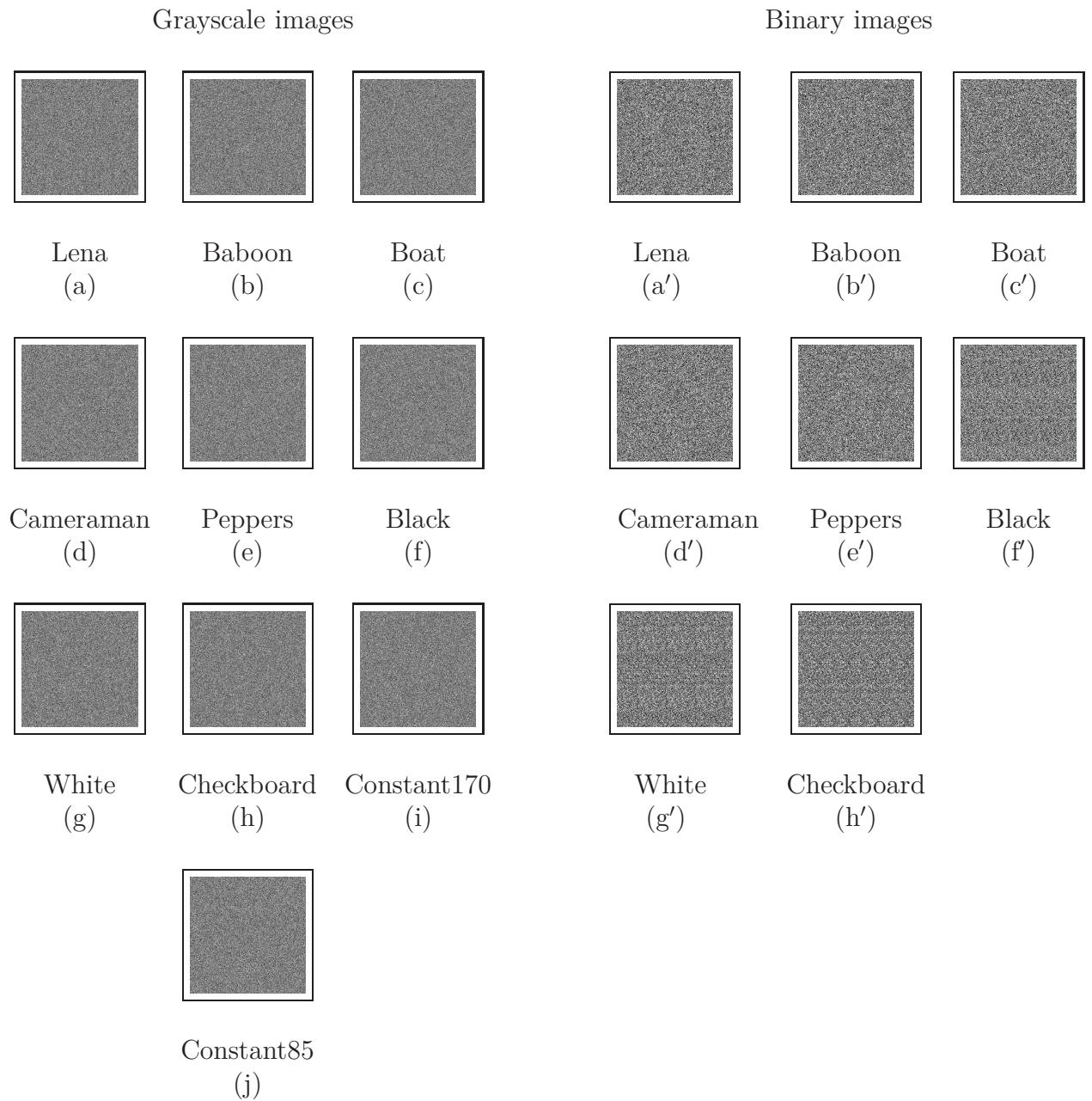


Figure 4.7: Decipher images using FTTIE method with wrong decryption key

summarized the encryption process of the SoA methods and the proposed method in Table 4.2. This presentation will help to visualize all these methods and make them easy to understand for comparison purposes. Since the SoA methods reported the result on grayscale images, so, for comparison purposes, we have considered the performance of the proposed method on grayscale images.

4.5.1 Key space analysis

As mentioned earlier, the keys for the confusion phase (key_c) and diffusion phase (key_d) are derived from the SHA value of the plain image. Since SHA256 is used to find the hash value of the plain image, the size of the key space of the proposed method is 256 bits, which is significantly high ($> 2^{100}$) and sufficient to resist the brute-force attack [172].

4.5.2 Complexity and Execution time

From the above discussion about the proposed method, we note the following.

1. First, we compute the key, using SHA256, from the given plain image. The complexity of this computation is $O(M \times N)$, where $M \times N$ is the size of the image.
2. Next, we define a random image (key_{img}) of the same size as the input image. Its complexity is $O(M \times N)$.
3. In the confusion phase, pixels are scrambled by using Eq. (4.20), which is nothing but a multiplication between 2×2 and 2×1 matrices. So, for each iteration of the confusion phase, the complexity is $O(M \times N)$.
4. Case I. In the diffusion phase, at the very first iteration, the random image (key_{img}) is XOR-ed with the confused image, and its complexity is $O(M \times N)$.
Case II. In all iterations the pixels' values are modified using Eq. (4.21), a multiplication of two matrices of size 3×3 and 3×1 . So, $O(\lceil \frac{M \times N}{3} \rceil)$ is the complexity of each iteration of the diffusion phase.

Therefore, if the confusion-diffusion phase is iterated itr times, then the complexity of the proposed image encryption is $O(M \times N) + O(M \times N) + O(M \times N) + itr \times (O(M \times N) + O(\lceil \frac{M \times N}{3} \rceil))$. Hence, the complexity of the proposed image encryption is $O((itr + 3) \times M \times N)$. In this experiment, we set $itr = 3$ (see Section 4.4.5). The average execution

Table 4.2: Techniques and encryption process used in FTTIE and SoA methods.

Methods	Encryption process	Techniques applied
M1 [230]	1) Incorporates a zig-zag scan-based pattern to generate the permutation 2) Generates the initial seed of the chaotic map from the plain image 3) Diffusion is achieved using an XOR operation	zig-zag scan-based chaotic feedback convolution model
M2 [198]	1) Incorporates a DNA module to create an intermediate cipher image 2) Using an RNA module to produce the final cipher image	DNA rules, DNA-XOR operator, and chaotic map
M3 [199]	1) Hash value of the plain image is used for the initial value of the chaos and to permute the pixels of the image 2) DNA XOR and RNA codons complement operators are used in diffusion.	A new Chaotic Evolutionary Biomolecules Model, DNA, and RNA computing
M4 [197]	1) The plain image is converted to the DNA image using binary coding rules 2) A BST is created from the DNA image using a Logistic map 3) DNA-BST is superimposed on DNA Image using XOR operation to get encrypted image	Binary search tree, DNA coding, Logistic map
M5 [222]	1) Given a polynomial, the value of the polynomial is computed at some selected points where the points are selected by interval bisection method 2) This sequence is used to define i) circular shift in the confusion phase and ii) substitution matrix and mask image same size as the given image of the diffusion phase.	Interval bisection method, Circular shift, Substitution, and XOR method
M6 [224]	1) The permutation is defined using points selected by the Regula-Falsi method 2) image encryption is achieved by pixel value substitution and iterative addition with the cyclic shift.	Regula-Falsi method, Substitution, Iterative addition, Circular shift
M7 [174]	1) A unified key is used to select the key pixels 2) DNA encryption is performed on the key pixels in their original locations 3) Other pixels of the image are encrypted by using the key stream generated by the hyper-chaotic Lorenz system to perform operations such as scrambling, DNA encoding, cyclic shift, etc.	Key pixels, hyper-chaotic Lorenz system Scrambling, DNA encoding, Cyclic shift
FTTIE [241]	1) The hash value of the image is used as key 2) The pixels are permuted using Fibonacci Transformation 3) Scrambled image is XOR-ed with a key image generated using the key 4) The pixel values are changed using Tribonacci Transformation	Key image, Fibonacci Transformation, and Tribonacci Transformation

Table 4.3: Average execution time (seconds)

Image	Execution time	
	Gray image	Binary image
Lena	0.284235	0.087974
Baboon	0.281999	0.078064
Boat	0.300635	0.076609
Cameraman	0.284769	0.078739
Peppers	0.308050	0.077587
Black	0.269685	0.081219
White	0.286664	0.082344
Checkerboard	0.286427	0.075928
Constant170	0.290878	-
Constant85	0.290954	-
Average	0.288429	0.079808

time of the proposed FTTIE method is reported in Table 4.3 and for each test image the program is runs ten times. The overall average, considering all images, is 0.288 sec and 0.079 sec, respectively, for grayscale and binary images. For binary images, the proposed method is 3.6 times faster than the grayscale images. The comparison of the execution time with the SoA method is given in Table 4.4. Since the SoA methods consider grayscale images, so we report the result on grayscale images for comparison with SoA methods. For better understanding, the system configuration of the individual method is included. We note that the proposed method is faster than the SoA methods.

4.5.3 Randomness analysis

It is known that there are high correlations among the adjacent pixels of the original image, i.e., a plain image is not a random one. In image encryption, one of the primary goals is to produce random noisy images so that nothing can be guessed from the cipher images. The proposed method returns noisy-like images, as shown in Fig. 4.5. To test the randomness, the cipher image is analyzed with respect to different parameters which are described in Section 4.1.

Table 4.4: Comparative study with respect to the execution time (seconds)

Method (image size)	System architecture	Execution Time
M1 [230] (256 × 256)	Matlab R2018a, Windows 10, 8 GB RAM, Intel Pentium N3540, 2.16 GHz CPU	10.440
M2 [198] (512 × 512)	Python 3, Windows 8, 8 GB RAM, Intel Core i7, 2.3 GHz CPU	1.494
M4 [197] (512 × 512)	MATLAB 2014, Windows 8, 8 GB RAM Intel Core i7, 2.3 GHz CPU	3.009
M6 [224] (512 × 512)	MATLAB R2018b, Intel core i5, 2.3 GHz CPU	0.620
M7 [174] (512 × 512)	Matlab R2020b, Windows 10, 8 GB RAM, Intel Core i7-8700, 3.20 GHz CPU	0.798
FTTIE [241] (512 × 512)	MATLAB R2022a, Windows 10, 4 GB RAM, Intel Core i5-6500 CPU, 3.20GHz	0.288

Table 4.5: Distribution of 0's and 1's in binary images (in %).

Image Name	Plain image		Cipher image	
	# 0s	# 1s	# 0s	# 1s
Lena	48.90	51.10	50.01	49.99
Baboon	47.37	52.63	49.80	50.20
Boat	31.49	68.51	50.01	49.99
Cameraman	40.98	59.02	50.08	49.92
Peppers	52.51	47.49	50.01	49.99
Black	100.0	00.00	49.80	50.20
White	00.00	100.00	50.20	49.80
Checkerboard	50.00	50.00	49.66	50.34

Histogram analysis

For histogram analysis, we display the histogram of the grayscale images, and for binary images, we report the percentage of '0' pixels and '1' pixels in the images. The distribution of '0's and '1's in the binary images (plain and cipher) is given in Table 4.5. This table shows that whatever the distribution of 0s and 1s in the original images, the proposed method gives almost 50-50 0s and 1s. So, for binary images, the proposed method performs well. The histogram of some of the test images (grayscale) is given in Fig. 4.8. (a). From these histograms, we note that i) they have specific peaks and ii) they do not cover the entire domain. These characteristics of the histogram of the plain images are due to the spatial correlations among the adjacent pixels. The histogram of the cipher images is shown in Fig. 4.8. (b). The figure shows that the

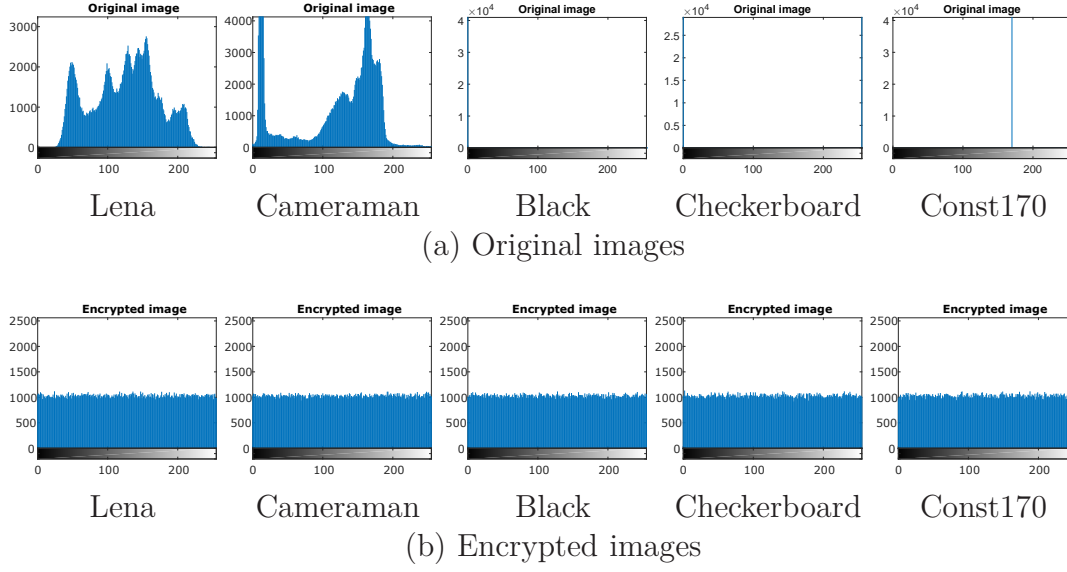


Figure 4.8: Histogram of the images: (a) Original images, (b) Encrypted images.

histograms are uniformly distributed and cover the entire intensity domain. Since our proposed method returns a noisy image and a uniform histogram, we may assume that encrypted images are random.

Entropy analysis

The entropy of an image is a good indicator of randomness. If an image is truly random, its entropy will be maximum; when the entropy of an image is close to the maximum possible value we may assume that the image is random. In Table 4.6, the entropy of the plain and encrypted images is reported. The entropy value of the encrypted grayscale images is almost 8 (here, images are 8-bit grayscale images), and for binary images it is nearly 1. The noisy structure of the cipher images guarantees that the cipher images are random. Table 4.7 study of the FTTIE method with existing methods. The table shows that the accomplishment of the FTTIE method is comparable with the SoA method.

Entropy is a global parameter, it is significantly high and does not imply that at the local level (block level) the cipher image is also random. For this, we compute the block level entropy (detail is given in Section 4.1). According to [173], if we compute local entropy on 30 non-overlapping blocks and each block has 1936 pixels, the average local entropy line is in the range $[7.901901305, 7.903037329]$. In a binary image, the value lies in the range $[0.467333934, 0.532666066]$ for 30 non-overlapping random blocks with blocks having 2 pixels only. The performance of the proposed

Table 4.6: Analysis of randomness with respect to entropy

Image	Grayscale		Binary	
	Original	Encrypted	Original	Encrypted
Lena	7.445061	7.999287	0.999651	1.000000
Baboon	7.358337	7.999289	0.997999	0.999988
Boat	7.191370	7.999302	0.898992	1.000000
Cameraman	7.047955	7.999367	0.976406	0.999998
Peppers	7.593654	7.999270	0.998185	1.000000
Black	-0.000000	7.999353	0.000000	0.999928
White	-0.000000	7.999410	0.000000	0.999988
Checkerboard	1.000000	7.999244	1.000000	0.999967
Constant170	-0.000000	7.999199	-	-
Constant85	-0.000000	7.999191	-	-

Table 4.7: The achievement of entropy of the proposed method in compare with SoA methods

Method→	M1	M2	M3	M4	M5	M6	M7	FTTIE
Image ↓	[230]	[198]	[199]	[197]	[222]	[224]	[174]	[241]
Lena	7.9994	7.9994	7.9995	7.9992	7.9993	7.9993	7.9994	7.9993
Baboon	7.9991	7.9994	7.9994	7.9990	7.9993	7.9993	7.9992	7.9993
Boat	7.9990	7.9992	7.9994	7.9991	-	-	-	7.9993
Cameraman	7.9992	7.9993	7.9994	7.9991	7.9993	7.9993	-	7.9994
Peppers	7.9993	7.9993	7.9994	7.9983	7.9994	7.9993	7.9993	7.9993
Black	-	-	-	-	7.9994	7.9993	-	7.9994
Checkerboard	-	-	-	-	7.9993	7.9993	-	7.9992

Table 4.8: Local Shannon entropy of the cipher images.

Image	Grayscale images (30, 1936)		Binary images (30, 2)	
	local entropy	Remark	local entropy	Remark
Lena	7.902567	Pass	0.472385	Pass
Baboon	7.902946	Pass	0.514365	Pass
Boat	7.903012	Pass	0.483158	Pass
Cameraman	7.902496	Pass	0.503672	Pass
Peppers	7.902021	Pass	0.469782	Pass
Black	7.902365	Pass	0.522462	Pass
White	7.901960	Pass	0.520703	Pass
Checkerboard	7.902347	Pass	0.489765	Pass
Constant170	7.903025	Pass	-	-
Constant85	7.901948	Pass	-	-

Table 4.9: Correlation coefficients of the images

Image		Grayscale			Binary		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
lena	Org img:	0.969557	0.985108	0.961273	0.903397	0.920394	0.888190
	Enc img:	-0.014825	-0.000664	0.007183	-0.014010	0.017374	0.010607
Baboon	Org img:	0.869555	0.759337	0.731440	0.706542	0.614299	0.581511
	Enc img:	0.000300	-0.003300	0.002763	0.015758	0.001940	0.009703
Boat	Org img:	0.931975	0.970481	0.918183	0.834579	0.858604	0.784536
	Enc img:	-0.007857	0.005314	-0.003888	0.006222	-0.000201	-0.011401
Cameraman	Org img:	0.982116	0.989307	0.971991	0.905331	0.906797	0.863369
	Enc img:	-0.003485	-0.012058	-0.002873	0.019643	0.025248	0.007040
Peppers	Org img:	0.975656	0.980522	0.959344	0.925471	0.929893	0.909070
	Enc img:	-0.015383	-0.009301	0.004001	-0.007694	-0.004968	-0.007512
Black	Org img:	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	Enc img:	-0.007286	-0.002917	-0.000073	-0.016802	0.023612	-0.015799
White	Org img:	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	Enc img:	-0.015854	0.009324	-0.005013	-0.016802	0.023612	-0.015799
Checherboard	Org img:	0.880613	0.875203	0.776411	0.880613	0.875203	0.776411
	Enc img:	-0.004700	0.014261	0.011992	-0.006026	-0.021435	-0.014751
Constant170	Org img:	1.000000	1.000000	1.000000	-	-	-
	Enc img:	-0.001335	-0.024532	-0.019457	-	-	-
Constant85	Org img:	1.000000	1.000000	1.000000	-	-	-
	Enc img:	-0.020115	-0.004376	-0.003340	-	-	-

method on both the grayscale and binary images is shown in Table 4.8. The table proves that the present method passed the test and therefore the cipher images are highly random.

Correlation analysis

A plain image has a high spatial correlation among the adjacent pixels. One image encryption method reduces the correlation. A detail of the correlation among the pixels of an image is given in Section 4.1. In this implementation, we consider random pairs of adjacent pixels (the number of pairs is around 20% of the pixels of the image) and then compute the correlations and repeat the process ten times with different collections of random pairs, and the average value is reported in Table 4.9. The table shows that the correlation in plain images is very high, a prominent characteristic of an original image. For the encrypted images, the correlation is close to zero; therefore, an attacker cannot obtain any information about the cipher image from some known pixels. So, nothing can be guessed about the plain image, which is an essential feature of an image encryption method. Therefore, the proposed method performs good and

Table 4.10: Comparative study of randomness analysis with respect to the correlation coefficients

Method→ Image ↓	Direction	M1 [230]	M2 [198]	M4 [197]	M5 [222]	M6 [224]	M7 [174]	FTTIE [241]
Lena	Horizontal	-0.002062	0.0054	0.0007	0.002097	0.001853	-0.0026	-0.014825
	Vertical	0.003685	0.0192	0.0017	0.002767	0.001984	-0.0033	-0.000664
	Diagonal	0.000249	0.0055	0.0008	-0.002901	0.000743	0.0004	0.007183
Baboon	Horizontal	-0.005890	0.0059	0.0009	-0.002659	0.000433	0.0019	0.000300
	Vertical	-0.001884	0.0047	0.0026	-0.001401	0.001475	-0.0041	-0.003300
	Diagonal	-0.009670	0.0058	0.0006	-0.002419	0.000648	-0.0099	0.002763
Boat	Horizontal	-0.002261	0.0052	0.0007	-	-	-	-0.007857
	Vertical	-0.000162	0.0007	0.0031	-	-	-	0.005314
	Diagonal	0.000207	0.0151	0.0007	-	-	-	-0.003888
Cameraman	Horizontal	-0.000280	0.0068	0.0035	0.004615	-	-	-0.003485
	Vertical	-0.000445	0.0023	0.0041	-0.000353	-	-	-0.012058
	Diagonal	0.000259	0.0092	0.0014	0.003272	-	-	-0.002873
Peppers	Horizontal	-0.000434	0.0149	0.0029	-0.001076	0.001180	-0.0063	-0.015383
	Vertical	-0.000446	0.0077	0.0059	0.001394	0.000721	-0.0006	-0.009301
	Diagonal	0.000195	0.0002	0.0018	-0.002952	0.001784	-0.0046	0.004001
Black	Horizontal	-	-	-	0.001646	-	-	-0.007286
	Vertical	-	-	-	-0.001508	-	-	-0.002917
	Diagonal	-	-	-	0.005440	-	-	-0.000073
Checkerboard	Horizontal	-	-	-	-0.002520	-	-	-0.004700
	Vertical	-	-	-	-0.001171	-	-	0.014261
	Diagonal	-	-	-	0.004867	-	-	0.011992

provide acceptable result. To prove the efficacy, the correlation coefficients of the test images are compared with SoA methods. The comparative result is shown in Table 4.10. In some cases, the proposed method performs a little better and, a little inferior. Hence, the performance of the proposed method is similar to that of the SoA methods.

Also, to visualize the relationships among the adjacency pixels of the images, we consider the selected pairs of pixels and plot the intensity value as the scatter diagram, which is a joint distribution. We consider only grayscale images since binary images have only four pairs $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$, it is not possible to judge the distribution of the intensity values. Here, we consider all three neighbors: vertical, horizontal, and diagonal. The scatter diagrams are shown in Fig. 4.9. This figure has three parts (a) - (c). In each part, the first row shows the scatter diagram of the original images, and the second row represents the scatter diagram of the encrypted images. In the case of the original images, (i) For the original images, the scatter plot is concentrated along the straight line $y = x$, and this shows that there is a high correlation among the adjacent pixels of these images; (ii) The ‘Black’ image has only one dot at $(0, 0)$ location, and this implies that pixels are 100% correlated; (iii) All the scatter diagrams of the ‘Checkerboard’ image has four dots at the locations $\{(0, 0), (0, 255), (255, 0), (255, 255)\}$ and the existence of the off-diagonal points $(255, 0)$ and $(0, 255)$ reduces the correlation among the pixels; (iv) The scatter diagrams of the

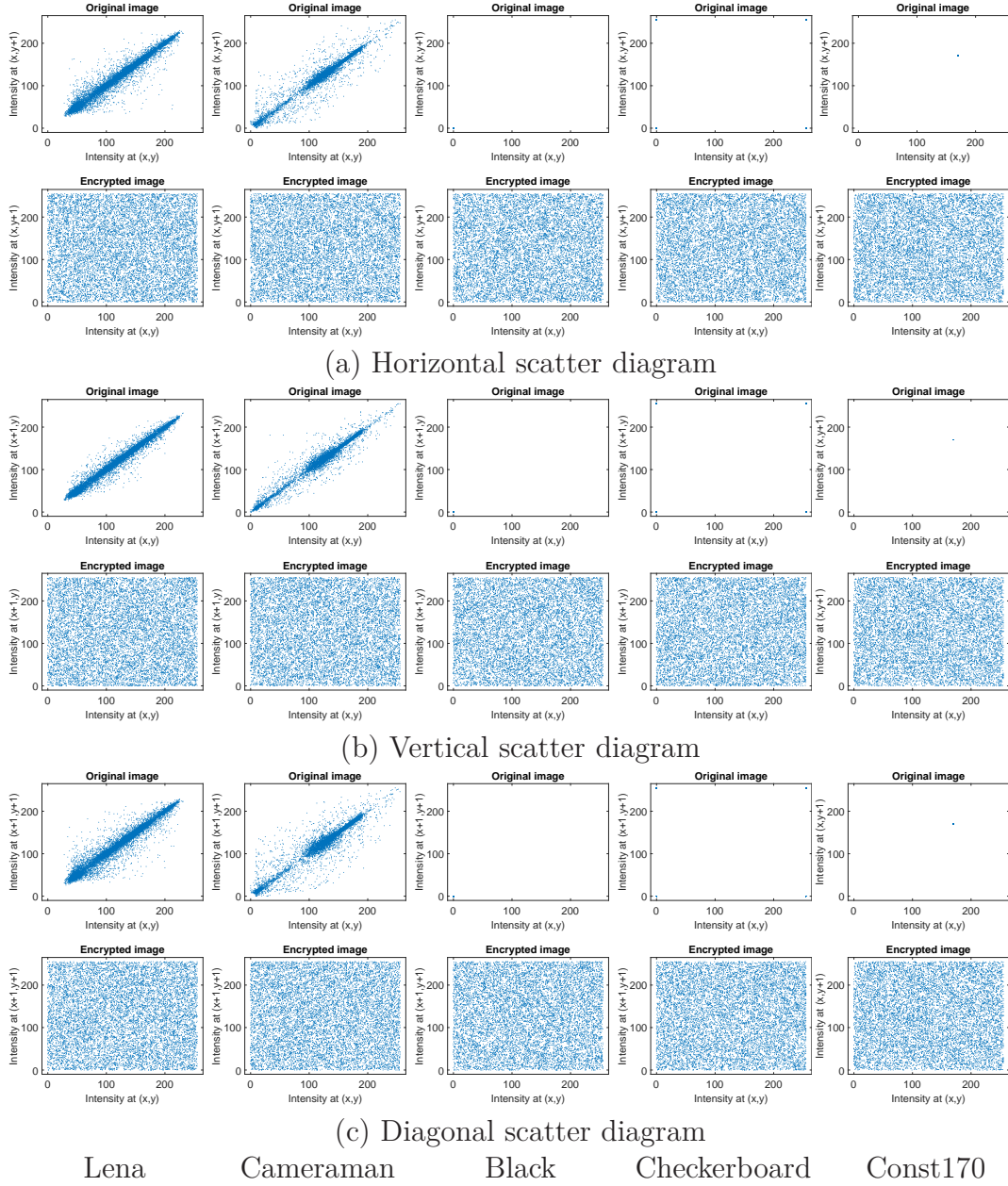


Figure 4.9: Scatter diagrams of some original and corresponding encrypted images.

‘Constant170’ has only one point at location (170, 170), so pixels are 100% correlated. We also have similar observations for the other original images. The result of Table 4.9 supports our above observations. For the cipher images, we note that the points of the scatter diagram are distributed over the whole domain, this shows that the correlation among the neighbor pixels of the cipher images is almost zero (this is also supported by Table 4.9). So, the performance of the proposed method is pretty good.

Table 4.11: Comparative study of randomness analysis with respect to the χ^2 value

Image	Grayscale		Binary	
	Original	Cipher	Original	Cipher
Lena	158349.355469	259.451172	126.738342	0.003433
Baboon	187356.572266	259.167969	726.681534	3.416565
Boat	383969.687500	253.525391	35842.180725	0.005508
Cameraman	418530.146484	229.529297	8527.368164	0.718521
Peppers	120165.796875	265.492188	659.245777	0.002197
Black	66846720.000000	234.914063	262144.000000	3.367447
White	66846720.000000	214.857422	262144.000000	3.367447
Checkerboard	33292288.000000	275.041016	*0.000000	3.751065
Constant170	66846720.000000	290.718750	-	-
Constant85	66846720.000000	289.310547	-	-

* This value is zero because frequency(0)=frequency(1)= expected frequency=50%.

χ^2 -Test

To compute the difference between the observed frequency distribution of the cipher image and the uniform distribution (expected for cipher image) of the intensity values χ^2 -test done. To compute the value we use the Eq (4.4). For an 8-bit image, when the significant level is 0.05, the ideal value of $\chi^2_{255,0.5}$ is 293.2478 and for a random binary image the value of χ^2 value is 3.8415 when the significant level is 0.05 [174]. The χ^2 value of the cipher images is given in Table 4.11. Table 4.11 supports our conclusions: i) original images do not have a uniform distribution, and the χ^2 values are very high; ii) for cipher images, the value is less than the threshold value for both types of images, indicating that the intensity distribution is almost uniform. Therefore, the proposed image encryption method can successfully counter statistical attacks.

4.5.4 Pixel disparity analysis

This test finds the affinity between the plain and cipher images. We use the parameters MSE, PSNR, and MAE (defined in Eq. (4.5)). For binary images we consider only MAE as it has a direct relation with other metrics; this parameter also gives the number of pixels mismatch. For binary image, the highest possible value of MAE is 50% and the experimental data support this. Table 4.12 reflects our observations, and we note that the PSNR value is very low (or MSE/MAE is high). From the above, we may conclude that predicting the plain image from the cipher image is not possible.

Table 4.12: Pixel difference between the plain and cipher images

Image	Grayscale image			Binary image
	MAE	MSE	PSNR	MAE
Lena	73.017334	7775.909210	9.223292	50.1499
Baboon	70.936752	7236.867638	9.535297	49.8981
Boat	72.429760	7628.511555	9.306406	49.8745
Cameraman	79.425556	9396.519512	8.401133	49.9424
Peppers	75.414028	8396.533710	8.889803	49.9126
Black	127.426994	21708.028267	4.764600	50.2041
White	127.302147	21667.287865	4.772758	50.2041
Checkerboard	127.459908	21698.784279	4.766450	51.3933
Constant170	70.995754	7256.970516	9.523250	-
Constant85	71.022579	7265.353962	9.518236	-

4.5.5 Key sensitivity analysis of encryption process

An image encryption method is said to be key sensitive if the method produces two different cipher images from the same plain image when two different encryption keys (maybe a single bit difference) are used. In this testing part, we have derived the second key (for the encryption) from the first by complementing a particular bit. Two parameters, NPCR and UACI (see Eq. (4.6)) are used to measure the deviation between two cipher images. The optimal value of UACI and NCPR are 33.3% and 100%, respectively [175], for 8-bit grayscale image. From Eq. (4.6), it is obvious that in the case of binary image $MAE = NPCR = UACI$ and in ideal its value is 50%. In this experiment, the key size is 256, and we consider five cases, as given in Table 4.13, to analyze the key sensitivity of the proposed method. The values of NPCR and UACI under the above test cases, along with their average values for the grayscale test images, are given in Tables 4.14 and 4.15.

Table 4.13: Test cases of key sensitivity analysis.

Test cases →	Case 1	Case 2	Case 3	Case 4	Case 5
bit complemented	4 th	56 th	132 nd	208 th	256 th

From these two tables, we note that the computed values are very close to the ideal value, and we may conclude that the proposed image encryption method is highly key-sensitive. The comparison with SoA methods is given in Table 4.16, where the NPCR value is reported. This table shows that the efficiency of the current method is similar

Table 4.14: NPCR measure for key sensitivity analysis in % (for grayscale images)

Test condition →	Case 1	Case 2	Case 3	Case 4	Case 5	Average
Image ↓	NPCR	NPCR	NPCR	NPCR	NPCR	NPCR
Lena	99.582291	99.598694	99.608612	99.200058	99.582291	99.514389
Baboon	99.606323	99.614716	99.605942	99.605942	99.594879	99.605560
Boat	99.611664	99.606323	99.584198	99.595642	99.615479	99.602661
Cameraman	99.596024	99.605560	99.232101	99.594498	99.628067	99.531250
Peppers	99.603271	99.604797	99.598694	99.597931	99.619293	99.604797
Black	99.621582	99.621582	99.612808	99.621582	99.593353	99.614181
White	99.626923	99.602509	99.618149	99.623871	99.613190	99.616928
Checkerboard	99.615479	99.579620	99.610901	99.615479	99.214935	99.527283
Constant170	99.548340	99.615097	99.580383	99.600601	99.644470	99.597778
Constant85	99.603653	99.595642	99.622345	99.622345	99.612808	99.611359

Table 4.15: UACI measure for key sensitivity analysis in % (for grayscale images)

Test condition →	Case 1	Case 2	Case 3	Case 4	Case 5	Average
Image ↓	UACI	UACI	UACI	UACI	UACI	UACI
Lena	33.348443	33.442854	33.459736	33.409161	33.348443	33.401727
Baboon	33.309637	33.410598	33.424123	33.358507	33.420190	33.384611
Boat	33.414022	33.483049	33.402626	33.425547	33.522388	33.449526
Cameraman	33.539064	33.485827	33.489577	33.494475	33.530526	33.507894
Peppers	33.408230	33.468824	33.460194	33.413883	33.503502	33.450927
Black	33.450742	33.453988	33.443636	33.453988	33.546138	33.469698
White	33.515939	33.451167	33.454111	33.518425	33.511207	33.490170
Checkerboard	33.489221	33.394068	33.493416	33.489221	33.396577	33.452501
Constant170	33.550738	33.464792	33.509507	33.360810	33.497044	33.476578
Constant85	33.421457	33.477122	33.406869	33.406869	33.402857	33.423035

to the first two existing methods [198], [199]. However, it is slightly less than the third method [197].

The NPCR value for the binary test images is reported in Table 4.17. This table shows that the value is very close to 50%. So, the proposed method performs efficiently in the case of binary image also.

4.5.6 Key sensitivity of the decryption process

In this section, we study the key sensitivity of the proposed decryption process. It is already mentioned that the proposed decryption method can only produce the original

Table 4.16: The key sensitivity of the proposed method and SoA methods in terms of NPCR %.

Method→ Image ↓	M2 [198]	M3 [199]	M4 [197]	FTTIE [241]
Lena	99.60	99.5998	99.93	99.5144
Baboon	99.61	99.6101	99.93	99.6056
Boat	99.60	99.5977	99.95	99.6027
Cameraman	99.60	99.6071	99.88	99.5313
Peppers	99.62	99.5983	99.92	99.6048

Table 4.17: NPCR/UACI/MAE measure for key sensitivity analysis in % (for binary images)

Test condition →	Case 1	Case 2	Case 3	Case 4	Case 5	Average
Image ↓	NPCR	NPCR	NPCR	NPCR	NPCR	NPCR
Lena	50.053406	50.022982	49.983528	49.989239	49.991517	50.008134
Baboon	49.972534	49.890137	50.052643	49.995045	50.131226	50.008317
Boat	49.973679	49.957657	49.942386	49.952914	49.982999	49.961927
Cameraman	49.948883	50.057220	49.948120	49.983925	49.978321	49.983294
Peppers	50.230026	50.085068	50.021744	50.117111	50.042835	50.099357
Black	49.800873	50.051498	49.558640	49.923427	49.727749	49.812437
White	49.773026	50.021744	49.953019	49.962527	49.952937	49.932651
Checkerboard	49.915695	49.932827	49.942932	49.932527	49.952982	49.935393

image if the correct key is given (see Fig. 4.6). To test the key sensitivity of the decryption process, we consider the grayscale images and there are two cases:

1. The cipher image is decrypted by keys key_a and key'_a , where key_a is the actual key and key'_a is derived from key_a by flipping only one bit. So, the decrypted image with key_a is the original image, and the deciphered image using key'_a is a noise-like image, as shown in Fig. 4.7. Then the NPCR and UACI values between the original and noisy decipher images are computed.
2. In this scenario, the same cipher image is decrypted using key'_a and key''_a , respectively, where key''_a is obtained from key'_a by complementing a bit other than the bit which was complemented during the computation of key'_a . So, key'_a and key''_a differ by a single bit, and none of these is the actual key. Both the decipher images, obtained in this case, are noisy, and then we compute the NPCR and UACI values between these two decipher images.

The test results, on grayscale images, under the above two cases are reported in Table 4.18. Since in computation of NPCR considers whether the pixels are the same or not so, in both cases, the NPCR value is acceptable. In UACI computation, the actual difference between two pixels is taken. In the first case, one deciphered image is the original one, i.e., not a random image, and the other image is a random one. So, in the first case, we do not achieve the expected value of UACI. In the second case, both the decipher images are noisy (i.e., random), so we achieve the expected result. From these test cases and the result given in Table 4.18, we may conclude that the proposed decryption process is also highly key-sensitive.

Table 4.18: Key sensitivity measure of the decryption process in % (for grayscale images)

Test condition \rightarrow	$key_a \rightarrow key'_a$		$key'_a \rightarrow key''_a (\neq key_a)$	
Image \downarrow	NPCR	UACI	NPCR	UACI
Lena	99.6033	28.6088	99.5903	33.4537
Baboon	99.5424	27.8631	99.6250	33.5700
Boat	99.5636	28.4531	99.5861	33.4177
Cameraman	99.6132	31.1033	99.6216	33.5664
Peppers	99.5773	29.5813	99.5937	33.5376
Black	99.6033	49.9736	99.5987	33.4442
White	99.6296	50.1252	99.6162	33.4315
Checkerboard	99.5987	49.9795	99.5983	33.3967
Constant170	99.5926	27.8675	99.6067	33.4897
Constant85	99.5964	27.8515	99.5995	33.4336

Table 4.19: Key sensitivity measure of the decryption process in % (for binary images)

Test condition \rightarrow	$key_a \rightarrow key'_a$	$key'_a \rightarrow key''_a (\neq key_a)$
Image \downarrow	NPCR/UACI	NPCR/UACI
Lena	50.2041	50.1245
Baboon	50.2338	50.0875
Boat	50.0542	50.1689
Cameraman	50.4086	50.3248
Peppers	50.6027	50.2762
Black	50.2026	50.1237
White	50.4711	50.1252
Checkerboard	50.0675	50.0342

Similar, performance is also achieved for binary images and the test performance is reported in Table 4.19, where the number of mismatched pixels is almost 50%.

4.5.7 Plaintext sensitivity analysis

An image encryption method is said to be plaintext sensitive if a minor change in plaintext gives a different cipher image. In the proposed method, the secret key is the hash value of the plain image. As different versions of the plain images are used, then for each image different hash value (here, key) will be generated and therefore, transformations (Fibonacci and Tribonacci) will be applied and therefore we obtain different cipher images. To test the plaintext sensitivity, we complement the LSB of different pixels and then generate the corresponding cipher images. The pixels that are modified as the test cases are given in Table 4.20. We report the NPCR and UACI values of the underlying test cases in Tables 4.21 and 4.22. These tables show that the test values are close to the ideal values. To achieve a more robust result, we also report the average value. Under the same test environment, we also test plaintext sensitivity for binary images. The result on binary images is given in Table 4.23. The table shows that the average performance is almost 50%, the ideal value. Hence, we can conclude that the proposed FTTIE method is highly key-sensitive for both grayscale and binary images.

4.5.8 Differential analysis

In a differential attack, an attacker tries to gather the affinity between the plain image and the cipher image by considering different versions of the plain image (with slide modification in the plain image) and the corresponding cipher image. If different cipher images are obtained subsequently, then the attacker will not get any information. NPCR and UACI parameters are used to measure the difference between two cipher images (see Eq. (4.6)). Since the proposed method is plaintext sensitive, obviously the method is robust against differential attacks. Table 4.23 shows that our method can resist the differential attack in the case of binary image. For grayscale images, to compare with SoA methods we report the average result in Table 4.21 and 4.22. The comparative study is given in Table 4.24. This study shows that the proposed method performs similarly to the SoA methods.

Table 4.20: Test cases of differential attacks.

Test cases →	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Pixel modified	(1,1)	(1,512)	(512,1)	(512,512)	(256,256)	(150,210)

Table 4.21: NPCR value (%) due to modified plain image (grayscale)

Position→	(1,1)	(1,512)	(512,1)	(512,512)	(256,256)	(150,210)	Average
Image ↓	NPCR	NPCR	NPCR	NPCR	NPCR	NPCR	NPCR
Lena	99.624252	99.600983	99.612427	99.592209	99.624252	99.626923	99.613508
Baboon	99.600220	99.589157	99.616241	99.625397	99.601364	99.596405	99.604797
Boat	99.589539	99.613190	99.600601	99.605942	99.626541	99.612808	99.608103
Cameraman	99.619293	99.593735	99.608994	99.607086	99.597168	99.601364	99.604607
Peppers	99.631882	99.616241	99.633026	99.592590	99.624252	99.610901	99.618149
Black	99.618530	99.587631	99.607468	99.617004	99.596024	99.612045	99.606450
White	99.635315	99.591827	99.607849	99.616623	99.611282	99.582672	99.607595
Checkerboard	99.607086	99.595261	99.607468	99.605942	99.614334	99.607468	99.606260
Constant170	99.607086	99.617004	99.629974	99.632645	99.605179	99.597168	99.614843
Constant85	99.603653	99.610519	99.610519	99.609756	99.606705	99.606705	99.607976

Table 4.22: UACI value (%) due to modified plain image (grayscale)

Position→	(1,1)	(1,512)	(512,1)	(512,512)	(256,256)	(150,210)	Average
Image ↓	UACI	UACI	UACI	UACI	UACI	UACI	UACI
Lena	33.413755	33.454439	33.379728	33.454401	33.505837	33.458171	33.444389
Baboon	33.451004	33.440667	33.520867	33.439270	33.499757	33.462248	33.468969
Boat	33.520524	33.427105	33.468743	33.427565	33.468623	33.474045	33.464434
Cameraman	33.515205	33.484876	33.475239	33.480551	33.521425	33.505224	33.497087
Peppers	33.396327	33.423345	33.432240	33.446555	33.414833	33.464578	33.429646
Black	33.475123	33.467127	33.490922	33.446733	33.477324	33.429009	33.464373
White	33.514914	33.518723	33.495575	33.459799	33.434952	33.522321	33.491047
Checkerboard	33.462349	33.494892	33.477976	33.448968	33.473994	33.413604	33.461964
Constant170	33.465041	33.510117	33.463981	33.460575	33.432919	33.407864	33.456749
Cconstant85	33.419471	33.470540	33.420730	33.388661	33.458832	33.454155	33.435398

Table 4.23: NPCR/UACI value (%) due to modified plain image (binary)

Position→	(1,1)	(1,512)	(512,1)	(512,512)	(256,256)	(150,210)	Average
Image ↓	NPCR	NPCR	NPCR	NPCR	NPCR	NPCR	NPCR
Lena	50.166321	50.080490	49.978638	50.191498	50.255966	50.115585	50.131416
Baboon	50.114441	50.007629	50.074768	49.851227	50.169375	50.372915	50.098392
Boat	50.000381	50.030518	50.839020	50.117111	50.043106	50.265884	50.216003
Cameraman	49.910736	49.917984	49.891663	49.872971	49.987030	50.046921	49.937884
Peppers	50.230026	49.825287	49.782181	50.082016	50.079346	49.923706	49.987094
Black	50.197601	50.215149	50.125885	50.052261	50.217819	50.230132	50.173141
White	50.127411	49.837875	50.049591	49.792099	50.185013	49.792179	49.964028
Checkerboard	49.931335	50.069046	49.887086	49.985504	50.297928	49.870682	50.006930

Table 4.24: The performance of the proposed method and SoA methods against differential attack in % (for grayscale image).

Method→ Image ↓		M1 [230]	M2 [198]	M3 [199]	M4 [197]	M5 [222]	M6 [224]	M7 [174]	FTTIE [241]
Lena	NPCR	99.6379	99.61	99.5998	99.6289	99.6231	99.7979	99.6167	99.6135
	UACI	33.4617	33.27	33.4715	33.5420	33.4556	33.4338	33.4589	33.4444
Baboon	NPCR	99.6264	99.62	99.6101	99.3665	99.6162	99.8088	99.6063	99.6048
	UACI	33.4948	33.19	33.4238	33.5084	33.5117	33.3604	33.4565	33.4690
Boat	NPCR	99.6322	99.57	99.5977	99.6742	-	-	-	99.6081
	UACI	33.4423	33.14	33.4447	33.6392	-	-	-	33.4644
Cameraman	NPCR	99.6462	99.53	99.6071	99.5283	99.6250	99.8052	-	99.6046
	UACI	33.5464	33.08	33.4647	33.4292	33.4656	33.4173	-	33.4971
Peppers	NPCR	99.6338	99.62	99.5983	99.6047	99.5918	99.7873	99.6112	99.6181
	UACI	33.4852	33.20	33.4823	33.3447	33.3994	33.4353	33.4776	33.4296
Black	NPCR	-	-	-	-	99.6761	99.8051	-	99.6065
	UACI	-	-	-	-	33.9281	33.4059	-	33.4644
Checkerboard	NPCR	-	-	-	-	99.6112	99.7983	-	99.6063
	UACI	-	-	-	-	33.5467	33.4340	-	33.4620

4.5.9 Chosen-plaintext and Known-plaintext attacks

In chosen plaintext attack, the attacker can use typical plaintext images like pure ‘White’ or ‘Black’ or some other special images to get the knowledge about the encryption key. If system can resist the chosen-plaintext attack, then the system is also robust against known-plaintext attack. As the key of the proposed encryption process depends on the plain image and the present method is plain image sensitive (i.e., strong against the differential attack), it is impossible to establish any connection between the actual cipher image and the cipher image achieved from the used plain image. Further, it may be noted that in our experiment, we have considered both ‘Black’ and ‘White’ images in gray and binary format, and their performance also establishes that our method is robust. Hence, the proposed method is robust against the known-plaintext and chosen-plaintext attacks.

4.5.10 Cropping attack

A good image encryption method should have the capability to restore the features of the plain image after a certain degree of the cipher image is cropped out. Here, we show the result on image ‘Lena’ in Fig. 4.10. The cipher image of ‘Lena’ is cropped with different levels like 50%, 25%, 12.5%, and 6.25%, and then the decipher im-

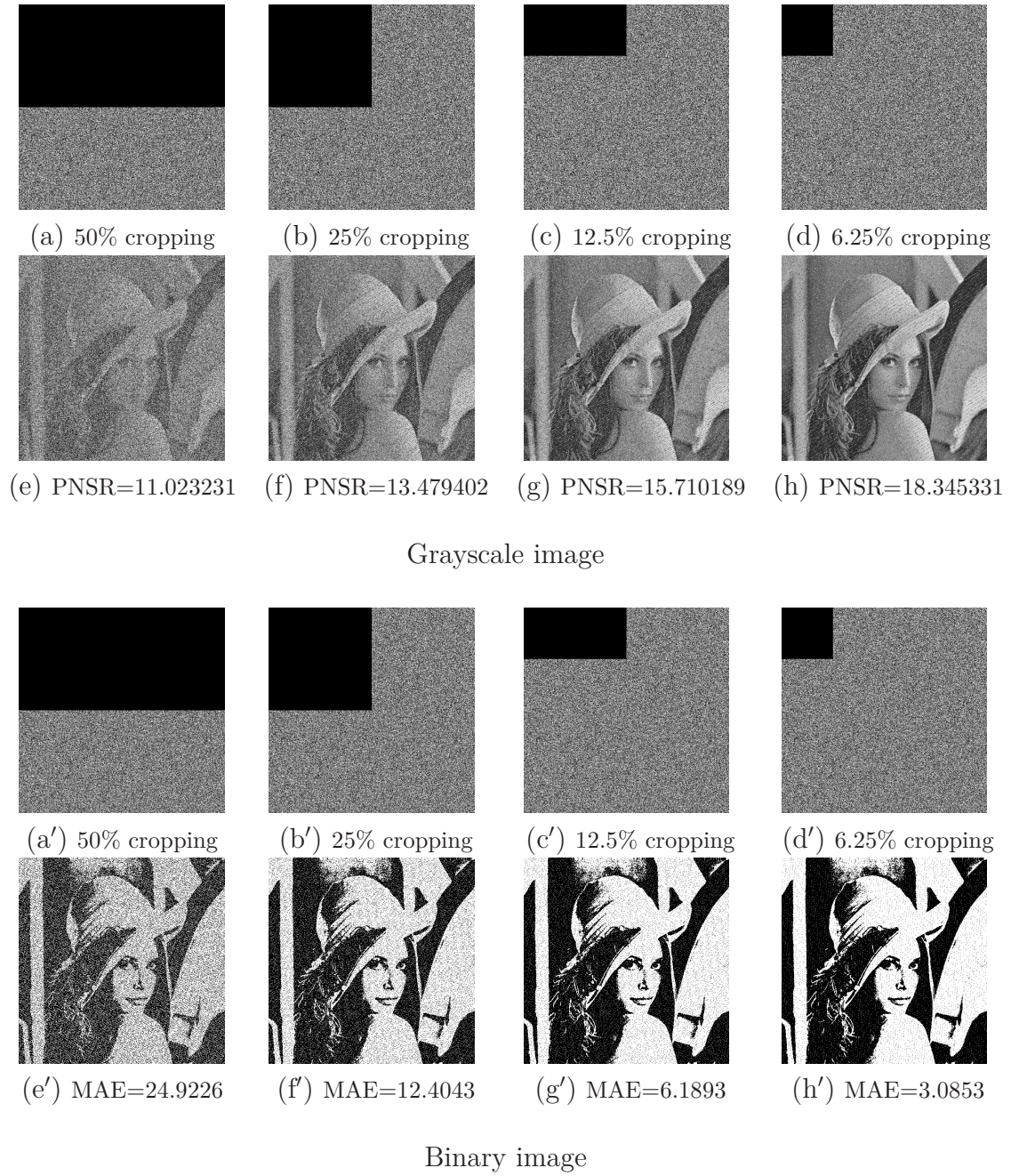


Figure 4.10: Cropped cipher images with different degrees of cropping (a - d, a'-d') and corresponding decipher images (e - f, e'-f').

ages are obtained. Figs. 4.10(a)-(d) show the cropped grayscale cipher images, while Figs. 4.10(e)-(h) show the corresponding decipher images. The PSNR between the plain and deciphered images is also calculated. From the figures, it is easy to understand that the deciphered images are 'Lena', and the quality of the deciphered images

is increasing when better when the level of cropping is decreasing. In Figs. 4.10(a')-(d'), we show the cropped version of the binary encrypted images, and their corresponding decipher images are displayed in Figs. 4.10(e')-(h'). This result proves that the proposed FTTIE method is robust against cropping attacks.

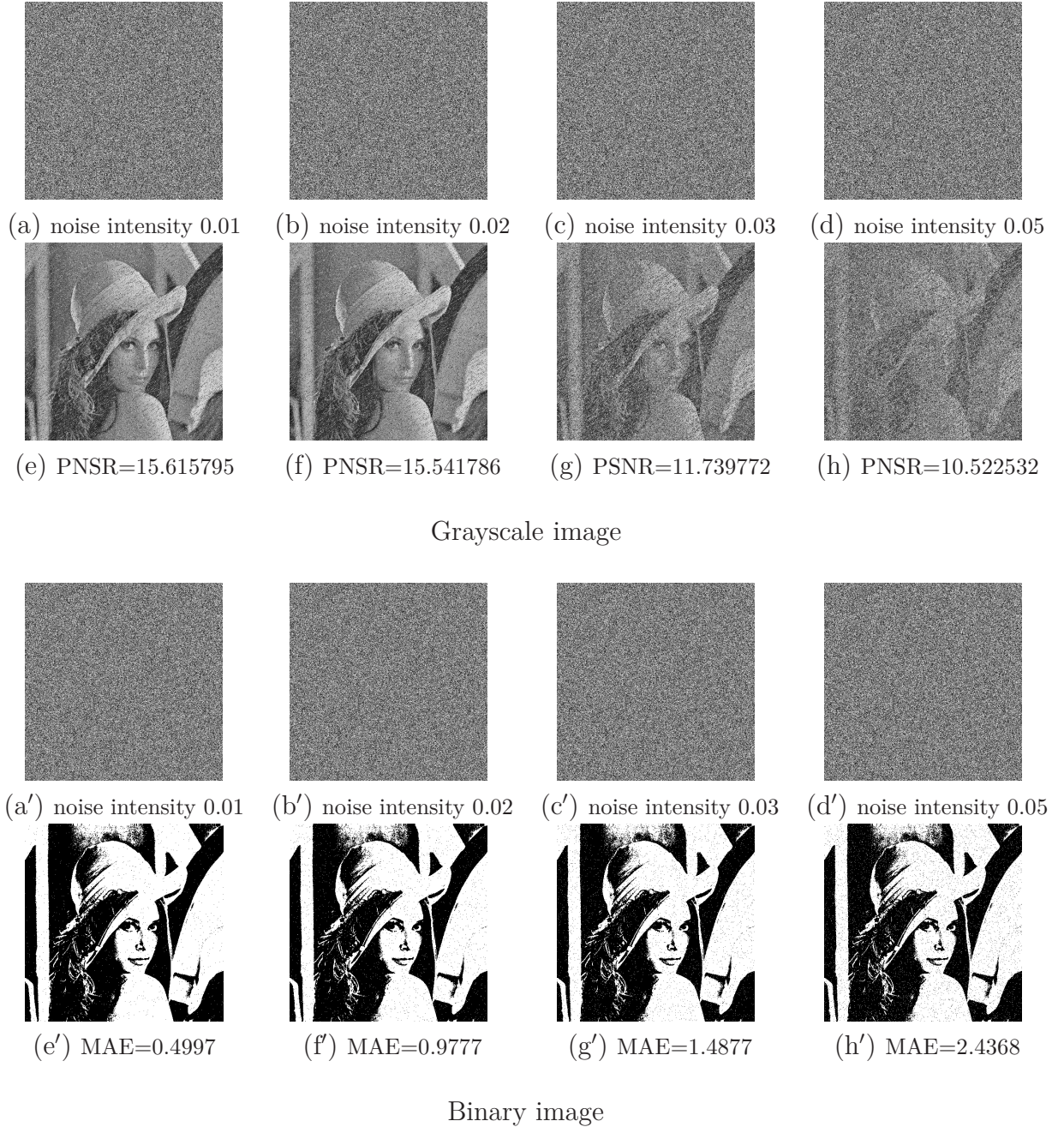


Figure 4.11: Cipher images with different degree of Salt & Pepper noise (a - d, a'-d') and corresponding decipher images (e - f, e'-f').

4.5.11 Noise attack

The ant-interference ability of a cryptosystem is the ability of the system to defend against a noise attack. During the transmission of the cipher image, the image may be distorted by the transmission noise, affecting the deciphered image. To test the anti-noise ability of the proposed method, we distort the cipher image with salt-and-pepper noise and then decrypt the image. Here, we test on the ‘Lena’ image. The grayscale noisy cipher images and their corresponding decipher images are shown in Fig. 4.11(a)-(h). The result shows that the quality of the deciphered image is gradually decreasing when the intensity of the noise increases. The result on the binary ‘Lena’ image is shown in Fig. 4.11(a’)-(h’). The result shows that when the intensity of noise is increasing the quality of the deciphered image decreases. Also note that in all the test cases, it easily follows that the deciphered image is the ‘Lena’ image. Therefore, the proposed FTTIE method is capable of withstanding the ‘Salt & Pepper’ noise.

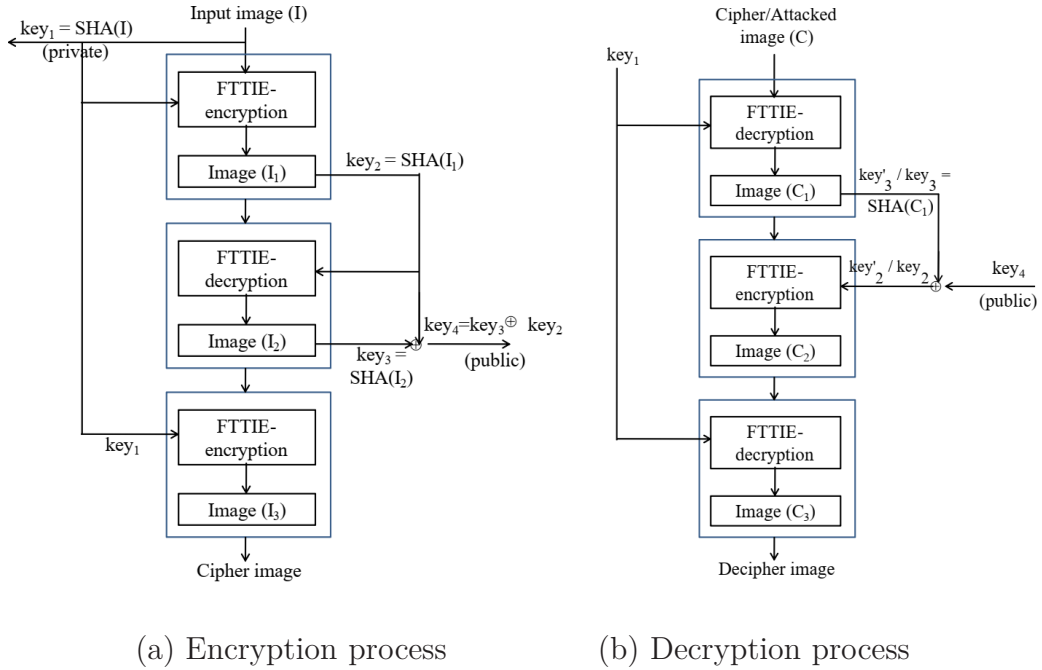


Figure 4.12: Block diagram of the $FTTIE_{ext}$ method.

4.6 Extended FTTIE method

In the last section we observe that the proposed FTTIE provides good result, faster, and robust against different attacks and the present method is applicable to encrypt the watermark image in case of copyright protection of the multimedia data. The

performance shows that the method is not suitable for fragile watermarking, because under the attack the FTTIE method is robust, i.e., from the reconstructed image we can easily identify the original image. The requirement of the fragile watermarking is the opposite, i.e., if there is a slightest modification in the cipher image then the decipher image should look like a noisy image. Here, we extend the method FTTIE as $\text{FTTIE}_{\text{ext}}$ to make it fit for fragile watermarking. Fig. 4.12 shows a simplified flow diagram of the $\text{FTTIE}_{\text{ext}}$ method. This extended $\text{FTTIE}_{\text{ext}}$ technique is achieved by adopting triple encryption technique using FTTIE, where the plain image (I) is first encrypted by FTTIE with the hash value (key_1) of the image as the key. Then, the encrypted image (I_1) is decrypted using the hash value (key_2) of I_1 . Let the decrypted image is I_2 and its hash value is key_3 . Finally, image I_2 is encrypted as I_3 using key_1 . Here, the key key_1 is the secret information and $\text{key}_2 \oplus \text{key}_3$ is public. Now, we have to prove that if the encrypted image I_3 is not modified, then the original image I can be obtained. We also need to prove that if there is a minimum change in I_3 , the decipher image, obtained using the decryption of $\text{FTTIE}_{\text{ext}}$, becomes noisy and there is no concurrence exists between the decipher image and the original image I . There are two cases that I_3 , the cipher image (produced by $\text{FTTIE}_{\text{ext}}$) is modified (attacked) or not.

1. No modification in I_3 : In the decryption process:

- i. The image I_3 is decrypted using key_1 and this gives the image I_2 .
- ii. Then compute the hash value of I_2 , which is key_3 and it is XOR-ed with the public information and this gives key_2 ($\text{key}_2 = (\text{key}_2 \oplus \text{key}_3) \oplus \text{key}_3$), the hash value of I_1 .
- iii. Note that I_2 is derived from I_1 , using decryption process of FTTIE with the help of key_2
 So, I_2 is encrypted by FTTIE using key_2 as key and it gives the image I_1
- iv. Finally, the image I_1 is decrypted as I using the decryption process of FTTIE with key_1 as key.

So, the original image I can be obtained.

2. I_3 is modified: In this scenario, if we follow the decryption process, then we have

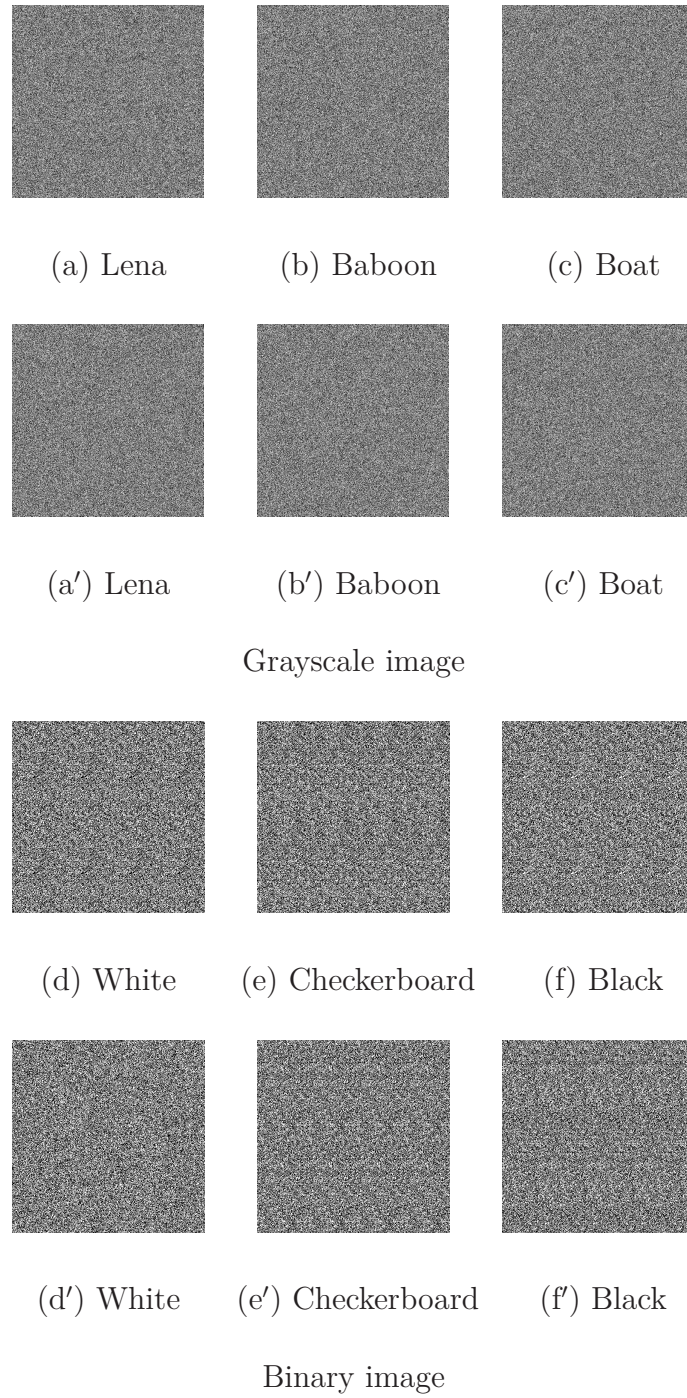


Figure 4.13: Fragility output of $\text{FTTIE}_{\text{ext}}$ when cipher images (a - f) are modified at (100, 100) location and corresponding decipher images are in (a' - f').

- i. The image I'_3 (the modified version) is decrypted using key_1 and this gives the image I'_2 ($\neq I_2$).

- ii. Then compute the hash value of I'_2 , which is key'_3 ($\neq key_3$) and it is XOR-ed with the public information and this gives key'_2 (different from the hash value of I_1).
- iii. Note that I_2 is derived from I_1 , using decryption process of FTTIE with the help of key_2

So, I'_2 is encrypted by FTTIE using key'_2 as key and it gives the image I'_1 , which is completely different from I_1 as the encryption method of FTTIE is key sensitive as well as plaintext sensitive.

- iv. Finally, the image I'_1 is decrypted as I' with key_1 as key and obviously I' is completely different from I as I'_1 is heavily different from I_1 .

Therefore, I' is completely different from I and hence can be used in fragile watermarking.

In support of the method $FTTIE_{ext}$, we have demonstrated some test results which are shown in Fig. 4.13, and in Table 4.25 we include the result for all test images. Here, only a single pixel at location (100,100) of each cipher image is modified. The results in Fig. 4.13 as well as in Table 4.25 are as expectation. For grayscale images, the

Table 4.25: Fragility of the $FTTIE_{ext}$ method, only modified the pixel at (100, 100) of the cipher image.

Image ↓	Grayscale images		Binary images
	NPCR	UACI	NPCR/MAE
Lena	99.5930	28.5020	50.3261
Baboon	99.6262	27.8712	50.0562
Boat	99.5979	28.4090	49.9346
Cameraman	99.6078	31.0989	50.2678
Peppers	99.5987	29.6537	50.1562
Black	99.5975	50.0449	49.9821
White	99.6120	49.9271	50.0326
Checkerboard	99.6052	50.0259	50.1034
Constant170	99.5965	27.8245	-
Constant85	99.6204	27.8812	-

NPCR value between the original image and the deciphered images obtained after modification is as expected; however, the UCAI value does not fulfill the expectation. Because, NPCR counts the number of pixels that differ and UACI considers the deviation of the values between two pixels. For binary images, we obtain the desired result. In Fig. 4.14, we demonstrate the output of the FTTIE and $FTTIE_{ext}$ methods, against

cropping attack, side-by-side to realize the effect of the $\text{FTTIE}_{\text{ext}}$ method. Similarly, in Fig. 4.15 we have included the performance of FTTIE and $\text{FTTIE}_{\text{ext}}$ methods against the Slat & Pepper noise. These results prove that the $\text{FTTIE}_{\text{ext}}$ method is a fragile image encryption method.

4.7 Conclusions

This article introduces a novel plaintext-based image encryption technique that utilizes the Fibonacci and Tribonacci transformations (FTTIE). The encryption key is obtained from the plain image using SHA256, and a 2×2 Fibonacci matrix is employed to scramble the image. In the diffusion phase, a 3×3 Tribonacci matrix is utilized to modify the pixel values, combined with an XOR operation with a random image defined by the key. The result is a highly secure encryption method that can be used to protect sensitive image data. This work marks the first known instance of using the Tribonacci Transformation in image encryption. The proposed method is faster than the SoA methods. The proposed method offers a vast key space along with comparable performance to state-of-the-art methods. Moreover, the proposed FTTIE method is highly suitable for creating robust and secure watermarking systems, while the extended encryption method $\text{FTTIE}_{\text{ext}}$ can be used for designing fragile and secure watermarking. Additionally, both FTTIE and $\text{FTTIE}_{\text{ext}}$ can be applied to grayscale and binary images, making them highly versatile methods for image encryption and watermarking. The proposed method demonstrates comparable performance for various types of special images. This article introduces a novel approach to pixel diffusion through the use of the Tribonacci Transformation. The phases of the FTTIE method can be applied to encrypt a color image. The Fibonacci transformation can be used to scramble the pixels of a color image, and the 3×3 tribonacci transformation can be employed to diffuse the color triple of a pixel. So, the method can be used for color image encryption.

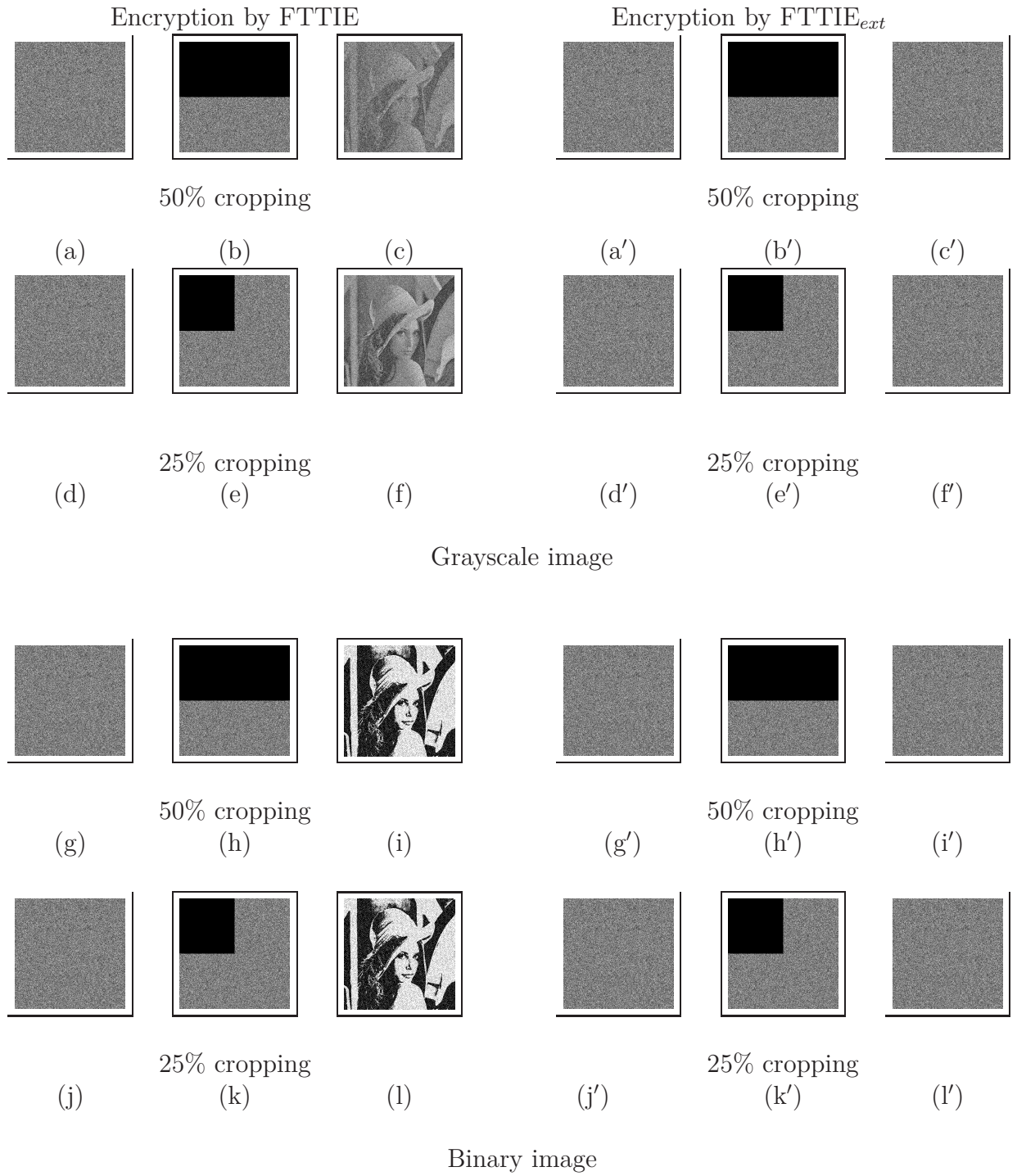


Figure 4.14: Output of FTTIE and FTTIE_{ext} methods under cropping attack with 50% and 25% cropping of the cipher image 'Lena'

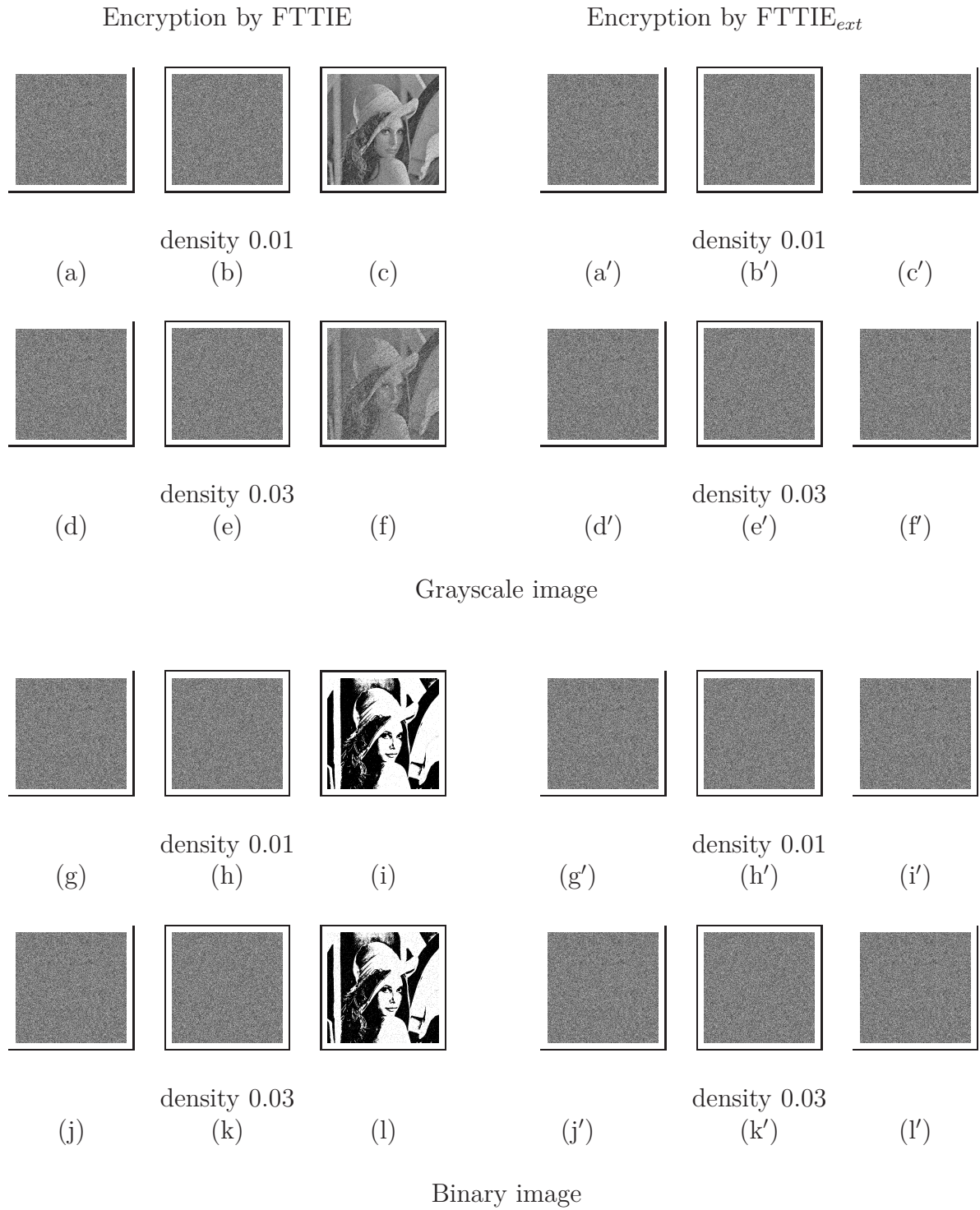


Figure 4.15: Output of FTTIE and FTTIE_{ext} methods under salt & pepper noise with noise density 0.01 and 0.03 on the cipher image 'Lena'

Chapter 5

Conclusions and Future Scopes

In this thesis, we have discussed watermarking techniques for information security of digital data such as image, and audio. Our primary goal is to develop secure watermarking methods for copyright protection and authentication of images and audio signals. The proposed watermarking methods have been evaluated concerning different watermarking features and the performance of the proposed methods has been compared with SoA methods. Finally, the thesis is concluded and future scopes are highlighted for further improvements.

5.1 Conclusions

In this work, we have focused on the security of digital images and audio signals using watermarking techniques as these digital data contain various sensitive information like a person's medical reports, military data, music data, business secrets, etc. Due to the advancement of modern multimedia devices and Internet technology, digital data have been shared on open platforms easily and there is a possibility of misuse of those data. As a result, it may harm personal reputation, national security, and huge financial loss in business, etc. In this thesis, we have proposed two watermarking schemes for images (Chapter 2) and one watermarking scheme for audio signals (Chapter 3).

The first watermarking technique is based on the BTC-PF method to embed a binary image (BWBTC-PF). A 2-level patternbook (PB) with 128 patterns (each of size 4×4) is used by the proposed method 'BWBTC-PF' to embed the watermark image into the host image. The PB is partitioned into two sub-patternbooks PB_0 and PB_1 with 64 patterns each, and the same partition is used for entire experiment. The watermark bit '0' (or '1') is embedded by encoding the host block using BTC-PF method using PB_0 (or PB_1) sub-patternbook. At the time of watermark extraction,

the (attacked) watermarked image is again encoded by the BTC-PF method using PB . The best pattern of a block belongs to either PB_0 or PB_1 and accordingly, the watermark bit is determined. This method is semi-blind as the same patternbook PB is used in the extraction process. The watermark image is encrypted before embedding using the secret key 'key' (the hash value of the watermark) to ensure the security of the BWBTC-PF method. The proposed method is a semi-blind, a limitation of the method. Also, the embedding time of the proposed method is high and this does not affect our purpose as the embedding step is considered as off-line process.

The second watermarking method is presented to embed a grayscale image into the host image based on SVD (GWSVD). Our proposed grayscale watermarking method is blind. Here, the host image is divided into four sub-images by the sub-sampling method. Each sub-image is then partitioned into blocks of size 4×4 and each block of the sub-images is decomposed by the SVD method. In the embedding process, to embed a watermark pixel, the largest singular value of the four co-located blocks of the sub-images is modified using embedding strength 'T'. Experimentally, the value of T is set to 50. Before embedding, the watermark image is normalized into $[0, 1]$ followed by encryption using a secret key 'key' (the hash value of the watermark image). During extraction, a watermark pixel is computed from the largest singular value of the four co-located blocks.

The experimental output of both the watermarking methods BWBTC-PF and GWSVD is satisfactory with respect to different features like imperceptibility, robustness, capacity, etc. Both the methods are secure and robust as the watermark image is encrypted by the FTTIE method and they can be used for copyright protection which is our goal of the thesis. Again, the same watermarking scheme can be used as a fragile watermarking if the watermark image is encrypted by the FTTIE_{ext} method and the methods can be used for authentication of the data which is another goal of the thesis. The proposed methods have been compared with SoA methods and output shows that our methods perform similar or better than some existing SoA methods.

The BWBTC-PF method has a disadvantage in that the recipient must have access to the same patternbook used for watermark embedding in order to extract the watermark, which may not always be feasible. On the other hand, the GWSVD method has a major limitation in that the original watermark cannot be recovered during the extraction process, even if no attack has occurred, as the watermark undergoes normalization, a lossy process, prior to embedding.

The utilization of digital audio over the World Wide Web is increasing day-by-day and it is mandatory to protect the copyright information as well as to verify the in-

tegrity of audio data. In this thesis, we have proposed a new audio watermarking technique for copyright protection and authentication of audio signals. In the proposed audio watermarking technique, a binary image is embedded into the host audio signals based on SVD (AWSVD). Here, the watermark image is encrypted using the its hash value either by FTTIE or FTTIE_{ext} method. In the embedding process of AWSVD, the watermark bit is embedded into the host audio block by modifying the Largest Singular Value (LSV) of the block. The amount of modification is determined using the quantization threshold Δ and the value of Δ is experimentally fixed as 0.25. Similar to image watermarking techniques, audio watermarking techniques can be applicable for copyright protection or authentication depending upon the encryption scheme FTTIE or FTTIE_{ext} is selected to encrypt the watermark image respectively. The proposed method has been compared with SoA methods and the result shows that our method outperforms compared to SoA methods. The main challenge of AWSVD is to make proper trade-offs among imperceptibility, robustness, and payload concerning the embedding strength Δ .

From the proposed watermarking methods, the GWSVD method is lossy (i.e., the grayscale watermark image can not be retrieved), and the other two methods (i.e., BWBTC-PF and AWSVD) are lossless (i.e., the original watermark can be extracted from the watermarked signal). The quality of the extracted watermark is not so crucial for the current problem; the only point is whether we can detect the ownership/authenticate the data. From the thesis's point of view, reversibility of the extracted watermark is not mandatory. Therefore, our proposed methods can be applied as modules in different applications.

As the watermarking technique is public, the watermark embedding and extraction process is known to everyone. As a result, one can extract the watermark and claim ownership. This is one major drawback of any watermark technique. To address this problem, we have proposed a new plaintext-based image encryption technique using the Fibonacci and Tribonacci transformations (FTTIE) in this thesis (Chapter 4). The encryption key is the hash value of the plain image itself, and it is computed using SHA256. A 2×2 Fibonacci matrix is defined to scramble the image, and a 3×3 Tribonacci matrix is used in the diffusion phase to modify the value of the pixels. We have XOR the scramble image with a random image to ensure that any image (specially, the pure black image) can be encrypted by our method. To the best of our knowledge, this is the first time that the Tribonacci Transformation has been used in image encryption. The proposed method has a high key space, and the proposed method has similar performance to SoA methods. The proposed FTTIE method can

be used to design a robust and secure watermarking system. Again, the extended encryption method $\text{FTTIE}_{\text{ext}}$ can be used to design fragile and secure watermarking. Both FTTIE and $\text{FTTIE}_{\text{ext}}$ can be applicable for grayscale images as well as binary images. The performance of the proposed method is equally suitable for different kinds of images (the test set includes some special images also). This paper highlights a new direction to diffuse the pixels using the Tribonacci Transformation. Since the Tribonacci matrix is 3×3 , the proposed method can be extended for color image encryption.

5.2 Future scope of improvement

In this thesis, we have discussed different issues related to watermarking techniques. In this section, the future directions of the work related to the thesis are discussed.

1. In BWBTC-PF method, to embed the watermark bit we encode the image block by BTC-PF method (which is a lossy encoding method). This results the degradation of the quality of the watermark image. So, we need to devise some modification over the BTC-PF method or plan for new methods on the skeleton of the BTC-PF to improve the quality of the watermarked image as well as reduce the embedding time.
2. The GWSVD method has inherent problem that the original watermark image cannot be retrieved, as we adopt the normalization step which is lossy. The method may be further extended for lossless recovery of the watermark image.
3. For copyright protection or authentication of images, we use binary/grayscale images as watermark. People can propose some methods to handle the color images as watermark image.
4. There is a scope to improve the AWSVD method by combining the SVD with DCT/DWT transformations.
5. The FTTIE method efficiently handles the binary and grayscale images. However, in case of color images there are three plains and straight we extend the FTTIE method for color image then its execution time will be three times than the time taken by grayscale image, and this may not be suitable for real-time application. So, there is an option to design a method for color images, utilizing the concept used in FTTIE .

6. In our work, we consider the issues for images and audio signals. Video is an important multimedia signal and it is ingored in this thesis. So, as a future scope people can consider the copyright protection and authentication of video as a research problem, though the method will not straight forward like image or audio.

Bibliography

- [1] K. Sreenivas and V. K. Prasad, "Fragile watermarking schemes for image authentication: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 9, pp. 1193–1218, 2018.
- [2] W. Sun, J. Zhou, Y. Li, M. Cheung, and J. She, "Robust high-capacity watermarking over online social network shared images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 1208–1221, 2021.
- [3] Z. Yun-Peng, L. Wei, C. Shui-ping, Z. Zheng-jun, N. Xuan, and D. Wei-di, "Digital image encryption algorithm based on chaos and improved des," in *International Conference on Systems, Man and Cybernetics*, pp. 474–479, IEEE, 2009.
- [4] M. Zeghid, M. Machhout, L. Khriji, A. Baganne, and R. Tourki, "A modified aes based algorithm for image encryption," *Journal of Computer Science and Engineering*, vol. 1, pp. 745–750, 2007.
- [5] G. Zhao, X. Yang, B. Zhou, and W. Wei, "Rsa-based digital image encryption algorithm in wireless sensor networks," in *Second International Conference on Signal Processing Systems*, vol. 2, pp. 640–643, IEEE, 2010.
- [6] M. Kumar, A. Iqbal, and P. Kumar, "A new rgb image encryption algorithm based on dna encoding and elliptic curve diffie-hellman cryptography," *Signal Processing*, vol. 125, pp. 187–202, 2016.
- [7] C. Gupta and N. V. S. Reddy, "Enhancement of security of diffie-hellman key exchange protocol using ras cryptography," in *Journal of Physics: Conference Series*, vol. 2161, 2022.
- [8] Y. Alkady, M. I. Habib, and R. Y. Rizk, "A new security protocol using hybrid cryptography algorithms," in *9th International Conference on Computer Engineering Conference*, pp. 109–115, IEEE, 2013.
- [9] A. A. Gutub and F. A. Khan, "Hybrid crypto hardware utilizing symmetric-key and public-key cryptosystems," in *International Conference on Advanced Computer Science Applications and Technologies*, pp. 116–121, IEEE, 2012.
- [10] S. M. Hardi, J. T. Tarigan, and N. Safrina, "Hybrid cryptosystem for image file using elgamal and double playfair cipher algorithm," *Journal of Physics: Conference Series*, vol. 978, pp. 012–068, 2018.

- [11] H. T. Hu, L. Y. Hsu, and H. H. Chou, "An improved svd-based blind color image watermarking algorithm with mixed modulation incorporated," *Information Sciences*, vol. 519, pp. 161–182, 2020.
- [12] I. Jacaman and M. Farajallah, "A lightweight spatial domain image encryption algorithms: A review paper," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 3, pp. 1275–1290, 2023.
- [13] L. Novamizanti, G. Budiman, and E. N. F. Astuti, "Robust audio watermarking based on transform domain and svd with compressive sampling framework," *Telecommunication Computing Electronics and Control*, vol. 18, no. 2, pp. 1079–1088, 2020.
- [14] A. Kumar and M. Dua, "Audio encryption using two chaotic map based dynamic diffusion and double dna encoding," *Applied Acoustics*, vol. 203, p. 109196, 2023.
- [15] W. M. Salama and M. H. Aly, "Chaotic maps based video encryption: A new approach," in *31-st International Conference on Computer Theory and Applications*, pp. 18–25, IEEE, 2021.
- [16] C. Chen, X. Wang, G. Huang, and G. Liu, "An efficient randomly selective video encryption algorithm," in *8-th International Conference on Computer and Communications*, pp. 1287–1293, IEEE, 2022.
- [17] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, "Steganography," *Digital Watermarking and Steganography*, pp. 425–467, 2008.
- [18] K. H. Jung and K. Y. Yoo, "Steganographic method based on interpolation and lsb substitution of digital images," *Multimedia Tools and Applications*, vol. 74, no. 6, pp. 2143–2155, 2015.
- [19] P. Li and A. Lu, "Lsb-based steganography using reflected gray code for color quantum images," *International Journal of Theoretical Physics*, vol. 57, no. 5, pp. 1516–1548, 2018.
- [20] S. K. Das and B. C. Dhara, "An lsb based novel data hiding method using extended lbp," *Multimedia Tools and Applications*, vol. 77, no. 12, pp. 15321–15351, 2018.
- [21] V. M. Potdar and E. Chang, "Grey level modification steganography for secret communications," in *IEEE International Conference on Industrial Informatics*, pp. 223–228, IEEE, 2004.
- [22] M. Hussain, A. W. A. Wahab, A. T. Ho, N. Javed, and K. H. Jung, "A data hiding scheme using parity-bit pixel value differencing and improved rightmost digit replacement," *Signal Processing: Image Communications*, vol. 50, pp. 44–57, 2017.
- [23] M. A. Hameed, S. Aly, and M. Hassaballah, "An efficient data hiding method based on adaptive directional pixel value differencing (adpvd)," *Multimedia Tools and Applications*, vol. 77, no. 12, pp. 14705–14723, 2018.

- [24] C. Y. Chen, N. Kand Su, C. Y. Shih, and Y. T. Chen, "Reversible watermarking for medical images using histogram shifting with location map reduction," in *IEEE International Conference on Industrial Technology*, pp. 792–797, IEEE, 2016.
- [25] S. Zhang, L. Yang, X. Xu, and T. Gao, "Secure steganography in jpeg images based on histogram modification and hyper chaotic system," *International Journal of Digital Crime and Forensics*, vol. 10, no. 1, pp. 40–53, 2018.
- [26] W. Hong and T. S. Chen, "A novel data embedding method using adaptive pixel pair matching," *IEEE transactions on information forensics and security*, vol. 7, no. 1, pp. 176–184, 2012.
- [27] W. Hong, "Efficient data hiding based on block truncation coding using pixel pair matching technique," *Symmetry*, vol. 10(2):36, 2018.
- [28] B. Kaur, A. Kaur, and J. Singh, "Steganographic approach for hiding image in dct domain," *International Journal of Advances in Engineering & Technology*, vol. 1(3):72, 2011.
- [29] M. Saidi, H. Hermassi, R. Rhouma, and S. Belghith, "A new adaptive image steganography scheme based on dct and chaotic map," *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 13493–13510, 2017.
- [30] W. Y. Chen, "Color image steganography scheme using dft, spiht codec, and modified differential phase-shift keying techniques," *Applied Mathematics and computation*, vol. 196, no. 1, pp. 40–54, 2008.
- [31] M. Douglas, K. Bailey, M. Leeney, and K. Curra, "Using svd and dwt based steganography to enhance the security of watermarked fingerprint images," *Telkomnika*, vol. 15, no. 3, 2017.
- [32] V. Kumar and D. Kumar, "A modified dwt-based image steganography technique," *Multimedia Tools and Applications*, vol. 77, no. 11, pp. 13279–13308, 2018.
- [33] C. Maiti and B. C. Dhara, "A new audio watermarking techniques based on empirical mode decomposition and quantization," in *Fifth International Conference on Computing, Communication and Sensor Network*, pp. 177–181, 2016.
- [34] B. Yang, Z. Li, and T. Zhang, "A real-time image forensics scheme based on multi-domain learning," *Journal of Real-Time Image Proc.*, vol. 17, pp. 29–40, 2020.
- [35] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [36] G. R. Blakley, "Safeguarding cryptographic keys," in *IEEE International Workshop on Managing Requirements*, pp. 313–318, IEEE, 1979.
- [37] B. B, "An efficient verifiable secret redistribution scheme," *Journal of Information Security and Applications*, vol. 69, p. 103295, 2022.

- [38] S. Kandar and B. C. Dhara, "A verifiable secret sharing scheme with combiner verification and cheater identification," *Journal of Information Security and Applications*, vol. 51, p. 102430, 2020.
- [39] K. J. Tan, H. W. Zhu, and S. J. Gu, "Cheater identification in (t, n) threshold scheme," *Computer Communications*, vol. 22, no. 8, pp. 762–765, 1999.
- [40] Y. Liu, "Linear (k, n) secret sharing scheme with cheating detection," *Security and Communication Networks*, vol. 9, no. 13, pp. 2115–2121, 2016.
- [41] Y. X. Liu, Q. D. Sun, and C. N. Yang, " (k, n) secret image sharing scheme capable of cheating detection," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–6, 2018.
- [42] A. Cheddad, J. Condell, K. Curran, and P. M. Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, vol. 90, pp. 727–752, 2010.
- [43] M. Asikuzzaman and M. R. Pickering, "An overview of digital video watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2131–2153, 2018.
- [44] N. S. Kamaruddin, A. Kamsin, Y. L. Por, and H. Rahman, "A review of text watermarking: theory, methods, and applications," *IEEE Access*, vol. 6, pp. 8011–8028, 2018.
- [45] N. Agarwal, A. K. Singh, and P. K. Singh, "Survey of robust and imperceptible watermarking," *Multimedia Tools and Applications*, vol. 78, pp. 8603–8633, 2019.
- [46] M. Begum and M. Uddin, "Digital image watermarking techniques: A review," *Information*, vol. 11, pp. 110–151, 2020.
- [47] I. Cox, M. L. Miller, and J. A. Bloom, *Digital watermarking*. Morgan Kaufmann Publishers, CA, USA, 2001.
- [48] D. Jiang, X. Weixin, and Y. Jianping, "Study on capacity of information hiding for still images," *Fifth International Conference on Signal Processing proceedings*, vol. 2, pp. 1010–1013, 2000.
- [49] M. G. Almutiri and O. M. T. B, "Digital image watermarking based on lsb techniques: A comparative study," *International Journal of Computer ApplicationsPattern Recognition Letters*, vol. 181, pp. 30–36, 2018.
- [50] N. M. Abdulwahab and N. M. Basheer, "Sptial domain block based blind image watermarking for hardware applications," in *International Conference on Advanced Computer Applications*, IEEE, 2021.
- [51] S. N. Bal, M. R. Nayak, and S. K. Sarkar, "On the implementation of a secured watermarking mechanism based on cryptography and bit pairs matching," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, pp. 552–561, 2021.

- [52] Z. Faheem, M. Ali, M. Raza, F. Arslan, J. Ali, M. Masud, and S. Mohammad, "Image watermarking scheme using lsb and image gradient," *Applied Sciences*, vol. 12, pp. 4202–4213, 2022.
- [53] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, pp. 1673–1687, 1997.
- [54] M. Begum and M. S. Uddin, "Analysis of digital image watermarking techniques through hybrid methods," *Journal of Advances in Multimedia*, vol. 2020, pp. 1–12, 2020.
- [55] S. Mellimi, V. Rajput, I. A. Ansari, and C. W. Ahn, "A fast and efficient image watermarking scheme based on deep neural network," *Pattern Recognition Letters*, vol. 151, pp. 222–228, 2021.
- [56] P. Y. Lin, Y. H. Chen, and J. S. Chang, C C Lee, "Secure spread spectrum watermarking for multimedia," *Image and Vision Computing*, vol. 31, pp. 311–321, 2013.
- [57] Z. Yuan, D. Liu, X. Zhang, and Q. Su, "New image blind watermarking method based on two-dimensional discrete cosine transform," *Optik*, vol. 204, p. 164152, 2020.
- [58] Q. Su, D. Liu, and Y. Sun, "A robust adaptive blind color image watermarking for resisting geometric attacks," *Information Sciences*, vol. 606, pp. 194–212, 2022.
- [59] O. Jane, E. Elbasi, and H. G. Ilk, "Hybrid non-blind watermarking based on dwt and svd," *Journal of Applied Research and Technology*, vol. 12, pp. 750–761, 2014.
- [60] C. Pradhan, S. Rath, and A. K. Bisoi, "Non blind digital watermarking technique using dwt and cross chaos," *Procedia Technology*, vol. 6, pp. 897–904, 2012.
- [61] J. M. Shieh, D. C. Lou, and C. M. C, "A semi-blind digital watermarking scheme based on singular value decomposition," *Computer Standards and Interfaces*, vol. 28, pp. 428–440, 2006.
- [62] S. Prasanth Vaidya and P. V. S. S. R. Chandra Mouli, "A robust semi-blind watermarking for color images based on multiple decompositions," *Multimedia Tools and Applications*, vol. 76, pp. 25623–25656, 2017.
- [63] L. Y. Hsu and H. T. Hu, "Robust blind image watermarking using crisscross inter-block prediction in the dct," *Journal of Visual Communication and Image Representation*, vol. 46, pp. 33–47, 2017.
- [64] Q. Su, Y. Niu, G. Wang, S. Jia, and J. Yue, "Color image blind watermarking scheme based on qr decomposition," *Signal Processing*, vol. 94, pp. 219–235, 2014.

- [65] F. N. Thakkar and V. K. Srivastava, "A blind medical image watermarking: Dwt-svd based robust and secure approach for telemedicine applications," *Multimedias Tools and Applications*, vol. 76, pp. 3669–3697, 2017.
- [66] D. Singh and S. K. Singh, "Dwt-svd and dct based robust and blind watermarking scheme for copyright protection," *Multimedias Tools and Applications*, vol. 76, pp. 13001–13024, 2017.
- [67] J. J. Shen, C. F. Lee, F. W. Hsu, and S. Agrawal, "A self-embedding fragile image authentication based on singular value decomposition," *Multimedia Tools and Applications*, vol. 79, pp. 25969–25988, 2020.
- [68] E. Gul and S. Ozturk, "A novel pixel-wise authentication-based self-embedding fragile watermarking method," *Multimedia Systems*, vol. 27, pp. 531–545, 2021.
- [69] X. Qi and X. Xin, "A singular-value-based semi-fragile watermarking scheme for image content authentication with temper localization," *Journal of Visual Communication and Image Representation*, vol. 30, pp. 312–327, 2015.
- [70] B. B. Haghighi, A. H. Taherinia, and R. Monsefi, "An effective semi-fragile watermarking method for image authentication based on lifting wavelet transform and feed-forward neural network," *Cognitive Computation*, vol. 12, pp. 863–890, 2020.
- [71] F. Ernawan and M. N. Kabir, "A block-based rdwt-svd image watermarking method using human visual system characteristics," *Visual computer*, vol. 36, no. 1, pp. 19–37, 2020.
- [72] A. Vacavant, "A novel definition of robustness for image processing algorithms," in *International Workshop on Reproducible Research in Pattern Recognition*, pp. 75–87, Springer, 2017.
- [73] C. Kumar, A. K. Singh, and P. Kumar, "Sa recent survey on image watermarking techniques and its applications in e-governance," *Multimedia Tools and Applications*, vol. 77, pp. 3597–3622, 2018.
- [74] B. C. Dhara and B. Chanda, "Block truncation coding using pattern fitting," *Pattern Recognition*, vol. 37, no. 11, pp. 2131–2139, 2004.
- [75] B. C. Dhara and B. Chanda, "Color image compression based on block truncation coding using pattern fitting principle," *Pattern Recognition*, vol. 40, pp. 2408–2417, 2007.
- [76] Z. Chen, L. Li, H. Peng, Y. Liu, and Y. Yang, "A novel digital watermarking based on general non-negative matrix factorization," *IEEE Trans. Multimedia*, vol. 20, pp. 1973–1986, 2018.
- [77] T. Huynh-The, C. H. Hua, N. A. Tu, T. Hur, J. Bang, D. Kim, M. B. Amin, B. H. Kang, H. Seung, and S. Lee, "Selective bit embedding scheme for robust blind color image watermarking," *IEEE Trans. Multimedia*, vol. 426, pp. 1–18, 2018.

- [78] Y. Liu, S. Tang, R. Liu, L. Zhang, and Z. Ma, "Secure and robust digital image watermarking scheme using logistic and rsa encryption," *Expert Syst. Appl.*, vol. 97, pp. 95–105, 2018.
- [79] R. G. V. Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *1st International conference on image processing*, pp. 86–90, 1994.
- [80] R. Z. Wang, C. F. Lin, and J. C. Lin, "Image hiding by optimal lsb substitution and genetic algorithm," *Pattern Recognition*, vol. 34, pp. 671–683, 2001.
- [81] G. J. Lee, E. J. Yoon, and K. Y. Yoo, "A new lsb based digital watermarking scheme with random mapping function," in *International Symposium on Ubiquitous Multimedia Computing*, pp. 130–134, 2008.
- [82] S. Fazli and G. Khodaverdi, "Trade-off between imperceptibility and robustness of lsb watermarking using ssim quality metrics," in *Second International Conference on Machine Vision*, pp. 101–104, 2009.
- [83] A. B. Dehkordi, S. N. Esfahani, and A. N. Avanaki, "Robust lsb watermarking optimized for local structural similarity," in *19th Iranian Conference on Electrical Engineering*, pp. 1–6, 2011.
- [84] S. Heidari and M. Naseri, "A novel lsb based quantum watermarking," *International Journal of Theoretical Physics*, vol. 55, pp. 4205–4218, 2016.
- [85] X. Kuang, W. A. Ling, L. S. Ke, G. Lei, P. J. Ping, L. Z. Yue, and L. F. Ping, "Watermark embedding and extraction based on lsb and four-step phase shift method," in *7th International Conference on Information Technology: IoT and Smart City*, pp. 243–247, 2019.
- [86] M.-H. Lin and C.-C. Chang, "A novel information hiding scheme based on btc," in *The Fourth International Conference on Computer and Information Technology, 2004. CIT '04.*, pp. 66–71, IEEE, 2004.
- [87] S. F. Tu and C. S. Hsu, "A btc-based watermarking scheme for digital images," *International Journal on Information and Security*, vol. 15, no. 2, pp. 216–228, 2004.
- [88] J. M. Guo, M. F. Wu, and Y. C. Kang, "Watermarking in conjugate order dither block truncation coding images," *Signal Processing*, vol. 89, pp. 1864–1882, 2009.
- [89] C. N. Yang and Z. M. Lu, "A blind image watermarking scheme utilizing btc bitplanes," *Journal of Digital Crime and Forensics*, vol. 3, no. 4, pp. 42–53, 2011.
- [90] W. Wan, J. Wu, X. Xie, and G. Shi, "A novel just noticeable difference model via orientation regularity in dct domain," *IEEE Access*, vol. 5, pp. 22953–22964, 2017.

- [91] J. Wu, L. Li, W. Dong, G. Shi, W. Lin, and C. C. J. Kuo, "Enhanced just noticeable difference model for images with pattern complexity," *IEEE Trans. Image Process.*, vol. 26, pp. 2682–2693, 2017.
- [92] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A dct-domain system for robust image watermarking," *Signal Processing*, vol. 66, pp. 357–372, 1998.
- [93] B. Chen and G. W. Wornell, "Digital watermarking and information embedding using dither modulation," in *IEEE Second Workshop on Multimedia Signal Processing*, pp. 273–278, 1998.
- [94] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inf. Theory*, vol. 47, pp. 1423–1443, 2001.
- [95] Y. Zhang, Z. Wang, Y. Zhan, L. Meng, J. Sun, and W. Wan, "Jnd-aware robust image watermarking with tri-directional inter-block correlation," *Int. J. Intell. Syst.*, 2021.
- [96] J. Wu, G. Shi, and W. Lin, "Survey of visual just noticeable difference estimation," *Frontiers of Computer Science*, vol. 13, pp. 4–15, 2019.
- [97] Q. Li and I. J. Cox, "Improved spread transform dither modulation using a perceptual model: robustness to amplitude scaling and jpeg compression," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. II–185–II–188, 2007.
- [98] H. J. Ko, C. T. Huang, G. Horng, and W. Shiuh-Jeng, "Robust and blind image watermarking in dct domain using inter-block coefficient correlation," *Information Science*, vol. 517, pp. 128–147, 2020.
- [99] A. B. Watson, "Dct quantization matrices visually optimized for individual images," in *Proceedings of SPIE-The International Society for Optical Engineering*, pp. 1913–1914, 1993.
- [100] F. Ernawan and M. N. Kabir, "A robust image watermarking technique with an optimal dct-psychovisual threshold," *IEEE Access*, vol. 6, pp. 20464–20480, 2018.
- [101] M. Barni, F. Bartolini, and A. Piva, "Improved wavelet-based watermarking through pixel-wise masking," *IEEE Trans. Image Process.*, vol. 10, pp. 783–791, 2001.
- [102] S. Giri, K. J. and Quadri, R. Bashir, and J. I. Bhat, "Dwt based color image watermarking: a review," *Multimedia Tools Appl.*, vol. 66, pp. 1–15, 2020.
- [103] Y. Gao, J. Wang, and Y. Q. Shi, "Dynamic multi-watermarking and detecting in dwt domain," *J. Real-Time Image Proc.*, vol. 16, pp. 565–576, 2019.
- [104] N. Kashyap and G. R. Sinha, "Image watermarking using 3-level discrete wavelet transform dwt," *Int. J. Modern Educ. Comput. Sci.*, vol. 3, pp. 50–56, 2012.

- [105] R. Choudhary and G. Parmar, "A robust image watermarking technique using 2-level discrete wavelet transform dwt," in *2nd International Conference on Communication Control and Intelligent Systems*, pp. 120–124, 2016.
- [106] C. Li, Z. Zhang, Y. Wang, B. Ma, and D. Huang, "Dither modulation of significant amplitude difference for wavelet based robust watermarking," *Int. J. Modern Educ. Comput. Sci.*, vol. 166, pp. 404–415, 2015.
- [107] M. Sudha and T. Thanuja, "A robust image watermarking technique using dtcwt and pca," *Int. J. Appl. Eng. Res.*, vol. 12, pp. 8252–8256, 2017.
- [108] K. Zebbiche, F. Khelifi, and K. Loukhaoukha, "Robust additive watermarking in the dtcwt domain based on perceptual masking," *Multimedia Tools Appl.*, vol. 77, pp. 21281–21304, 2018.
- [109] N. Jimson and K. Hemachandran, "Dft based coefficient exchange digital image watermarking," in *Second International Conference on Intelligent Computing and Control Systems*, pp. 567–571, 2018.
- [110] S. Prajwalasimha, S. C. Suputhra, and C. Mohan, "Performance analysis of combined discrete fourier transformation dft and successive division based image watermarking scheme," *Int. J. Recent Technol. Eng.*, vol. 8, pp. 34–39, 2019.
- [111] C. C. Lai and C. C. Tsai, "Digital image watermarking using discrete wavelet transform and singular value decomposition," *IEEE Trans. on Instrumentation and Measurement*, vol. 59, no. 11, pp. 3060–3063, 2010.
- [112] S. Agrawal and A. Bhalchandra, "Robust implementation gray image watermarking with lwt, svd and qr decomposition," *International Journal of Computer Theory and Engineering*, vol. 14, no. 3, pp. 89–96, 2022.
- [113] S. E. Naffouti, K. A, and S. A, "A sophisticated and provably grayscale image watermarking system using dwt-svd domain," *The Visual Computer*, <https://doi.org/10.1007/s00371-022-02587-y>, 2022.
- [114] W. J. Yang, W. W. Wang, and G. X. Song, "A novel blind gray image watermarking scheme," *Computer Engineering and Applications*, vol. 41, pp. 124–126, 2005.
- [115] R. Yang and J. X. Pu, "A blind grayscale watermarking algorithm based on block dct," *Journal of Research of Computers*, vol. 22, pp. 165–166, 2005.
- [116] Z. Pao, J. W. Zhang, and D. S. Xia, "A gray-scale blind watermarking algorithm in dct domain based on chaotic encryption," *Journal of Computer Engineering*, vol. 32, pp. 157–160, 2006.
- [117] X. Ma and X. Shen, "A novel blind grayscale watermark algorithm based on svd," in *IEE International Conference on ALIP*, pp. 1063–1068, IEEE, 2008.
- [118] B. C. Dhara and B. Chanda, "A fast progressive image transmission scheme using block truncation coding by pattern fitting," *Journal of Visual Communication and Image Representation*, vol. 23, no. 2, pp. 313–322, 2012.

- [119] E. Delp and O. Mitchell, "Image compression using block truncation coding," *IEEE Transactions on Communications and Processing*, vol. 27, pp. 1335–1342, 1979.
- [120] M. N. Nasrabadi and R. B. King, "Image coding using vector quantization: a review," *IEEE Trans. on Communication*, vol. 36, pp. 957–971, 1988.
- [121] V. Aslantas, "An optimal robust digital image watermarking based on svd using differential evolution algorithm," *Optics Communications*, vol. 282, pp. 769–777, 2009.
- [122] C. C. Lai, "An improved svd-based watermarking scheme using human visual characteristics," *Optics Communications*, vol. 284, pp. 938–944, 2011.
- [123] B. Zhou and J. Chen, "A geometric distortion resilient image watermarking algorithm based on svd," *International Journal on Information and Security*, vol. 9, no. 4, pp. 506–512, 2004.
- [124] S. P. Vaidya and V. Kishore, "Adaptive medical image watermarking system for e-health care applications," *SN Computer Science*, vol. 3, 03 2022.
- [125] M. Mehrabi, V. Zarei, and M. Ghanbari, "A highly robust medical image watermarking method for medical real-time applications," *Journal of Medical Signals and Sensors*, vol. 13, pp. 199–207, 2023.
- [126] N. M. Makbol, B. E. Khoo, and T. H. Rassem, "Block-based discrete wavelet transform-singular value decomposition image watermarking scheme using human visual system characteristics," *IET Image Processing*, vol. 10, pp. 34–52, 2016.
- [127] A. Dhani and F. Ernawan, "Adaptive scaling factors based on the impact of selected dct coefficients for image watermarking," *Journal of King Saud University Computer and Information Sciences*, vol. 34, pp. 605–614, 2022.
- [128] C. Maiti and B. C. Dhara, "A robust binary watermarking scheme using btc-pf technique," in *International Conference on Eco-friendly Computing and Communication Systems*, pp. 178–185, Springer, 2012.
- [129] W. C. Chu, "Dct-based image watermarking using sub-sampling," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 34–38, 2003.
- [130] Z. Wei, H. Lu, and F.-L. Chung, "Robust digital image watermarking based on subsampling," *Applied Mathematics and Computation*, vol. 181, no. 2, pp. 886–893, 2006.
- [131] A. Ali and N. A. Memon, "Blind and robust watermarking scheme in hybrid domain for copyright protection of medical images," *IEEE Access*, vol. 9, pp. 113714–113734, 2021.
- [132] C. Maiti and B. C. Dhara, "A grayscale watermark technique using sub-sampling and singular value decomposition," in *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 1511–1516, IEEE, 2016.

- [133] C. K. Huynh, D. W. Yun, J. P. Choi, and W. C. Lee, "Performance analysis for coexistence of lte-laa and wi-fi systems in the spatial, time, spectrum domain," *ICT Express*, vol. 5, no. 1, pp. 72–76, 2019.
- [134] S. Chowdhury, "Matching theory for cognitive radio networks: An overview," *ICT Express*, vol. 5, no. 1, pp. 12–15, 2019.
- [135] G. Hua, J. Huang, Y. Q. Shi, J. Goh, and V. L. L. Thing, "Twenty years of digital audio watermarking—a comprehensive review," *Signal Processing*, vol. 128, pp. 222–242, 2016.
- [136] A. Binny and M. Koilakuntla, "Hiding secret information using lsb based audio steganography," in *Proc. of IEEE International Conferences on Soft Computing and Machine Intelligence*, pp. 56–59, 2014.
- [137] W.-N. Lie and L.-C. Chang, "Robust and high-quality time-domain audio watermarking on low-frequency amplitude modification," *IEEE Transaction on Multimedia*, vol. 8, no. 1, pp. 46–59, 2006.
- [138] S. Subbarayan and S. K. Ramanatha, "Effective watermarking of digital audio and image using matlab technique," in *Proc. of IEEE International Conferences on Machine Vision*, pp. 317–319, 2009.
- [139] P. K. Dhar and T. Shimamura, "Blind svd-based audio watermarking using entropy and log-polar transformation," *Information Security and Application*, vol. 20, pp. 74–83, 2015.
- [140] A. R. Elshazly, M. E. Nasr, M. M. Fouad, and F. S. Abdel-Samie, "High payload multi-channel dual audio watermarking algorithm based on discrete wavelet transform and singular value decomposition," *International Journal of Speech Technology*, vol. 20, pp. 951–958, 2017.
- [141] K. V. Bhat, I. Sengupta, and A. Das, "A new audio watermarking scheme based on singular value decomposition and quantization," *Circuits, Systems and Signal Processing*, vol. 30, pp. 915–927, 2011.
- [142] J. Zhang, "Audio dual watermarking scheme for copyright protection and content authentication," *International Journal of Speech Technology*, vol. 18, pp. 443–448, 2015.
- [143] N. Muhammad and N. Bibi, "Digital image watermarking using partial pivoting lower and upper triangular decomposition into the wavelet domain," *IET Image Processing*, vol. 9, no. 9, pp. 795–803, 2015.
- [144] P. Rasti, S. Samiei, M. Agoyi, S. Escalera, and G. Anbarjafari, "Robust non-blind color video watermarking using qr decomposition and entropy analysis," *Visual Communication and Image Representation*, vol. 38, pp. 838–847, 2016.
- [145] K. K and A. O. Boudraa, "Audio watermarking via emd," *IEEE Trans. Audio, Speech, Language Processing*, vol. 21, no. 3, pp. 675–680, 2013.

- [146] Y. Erfani, R. Pichevar, and J. Rouat, "Audio watermarking using spikegram and a two-dictionary approach," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 840–852, 2017.
- [147] B. Y. Lei, I. Y. Soon, and Z. Li, "Blind and robust audio watermarking scheme based on svd-dct," *Signal Processing*, vol. 91, no. 8, pp. 1973–1984, 2011.
- [148] X. Wang, P. Wang, P. Zhang, S. Xu, and H. Yang, "A norm-space, adaptive, and blind audio watermarking algorithm by discrete wavelet transform," *Signal Processing*, vol. 93, pp. 913–922, 2013.
- [149] M. Li, X. Yuon, and J. Li, "Dual-tree complex wavelet transform based audio watermarking using distortion-compensated dither modulation," *IEEE Access*, vol. 6, pp. 60834–60842, 2018.
- [150] B. Y. Lei, I. Y. Soon, F. Zhou, Z. Li, and H. Lei, "A robust audio watermarking scheme based on lifting wavelet transform and singular value decomposition," *Signal Processing*, vol. 92, no. 9, pp. 1985–2001, 2012.
- [151] R. Li, S. Yu, and H. Yang, "Spread spectrum audio watermarking based on perceptual characteristic aware extraction," *IET Signal Processing*, vol. 10, no. 3, pp. 266–273, 2016.
- [152] J.-f. Li, H.-X. Wang, T. Wu, X.-m. Sun, and Q. Qian, "Norm ratio-based audio watermarking scheme in dwt domain," *Multimedia Tools and Applications*, vol. 77, no. 12, pp. 14481–14497, 2018.
- [153] Y. Xiang, I. Natgunanathan, D. Peng, G. Hua, and B. Liu, "Spread spectrum audio watermarking using multiple orthogonal pn sequences and variable embedding strengths and polarities," *IEEE/ACM Trans. Audio, Speech, Language Process*, vol. 26, no. 3, pp. 529–539, 2018.
- [154] J. Hwang, M J Lee, M. Lee, and H. G. Kang, "Svd-based adaptive qim watermarking on stereo audio signals," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 45–54, 2018.
- [155] P. K. Dhar and T. Shimamura, "Audio watermarking in transform domain based on singular value decomposition and cartesian-polar transformation," *International Journal of Speech Technology*, vol. 17, pp. 133–144, 2014.
- [156] M. Patil and J. Chitode, "Svd based audio watermarking algorithm using dual watermark for copyright protection," *New Visions in Science and Technology Vol. 5*, pp. 107–120, 2021.
- [157] A. R. Elshazly, M. E. Nasr, M. M. Fouad, and F. E. Abdel-Samie, "Intelligent high payload audio watermarking algorithm using colour image in dwt-svd domain," in *Journal of Physics: Conference Series*, vol. 2128, pp. 012–019, IOP Publishing, 2021.
- [158] A. Ghosal, "A hierarchical approach for content based audio classification," *PhD Thesis, Jadavpur University*, 2014.

- [159] V. K. Bhat, I. Sengupta, and A. Das, "An adaptive audio watermarking based on the singular value decomposition in the wavelet domain," *Digital SignalProcessing*, vol. 20, pp. 1547–1558, 2010.
- [160] C. Maiti and B. C. Dhara, "A blind audio watermarking based on singular value decomposition and quantization," *International Journal of Speech Technology*, vol. 25, pp. 759–771, 2022.
- [161] J. C. Dagadu, J.-P. Li, and E. O. Aboagye, "Medical image encryption based on hybrid chaotic dna diffusion," *Wireless Personal Communications*, vol. 108, no. 1, pp. 591–612, 2019.
- [162] M. Sokouti, A. Zakerolhosseini, and B. Sokouti, "Medical image encryption: an application for improved padding based ggh encryption algorithm," *The open medical informatics journal*, vol. 10, p. 11, 2016.
- [163] A. S. Dongare, A. Alvi, and N. Tarbani, "An efficient technique for image encryption and decryption for secured multimedia application," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 4, pp. 3186–3190, 2017.
- [164] Y. Li, H. Yu, B. Song, and J. Chen, "Image encryption based on a single-round dictionary and chaotic sequences in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 7, pp. 1–1, 2021.
- [165] R. K. Dwivedi, R. Kumar, and R. Buyya, "Secure healthcare monitoring sensor cloud with attribute-based elliptical curve cryptography," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 11, no. 3, pp. 1–18, 2021.
- [166] B. Joshi, B. Joshi, A. Mishra, V. Arya, A. K. Gupta, and D. Peraković, "A comparative study of privacy-preserving homomorphic encryption techniques in cloud computing," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 12, no. 1, pp. 1–11, 2022.
- [167] P. P. Dang and P. M. Chau, "Implementation idea algorithm for image encryption," in *Proceedings of SPIE*, pp. 1–9, 2000.
- [168] M. Zeghid, M. Machhout, L. Khriji, A. Baganne, and R. Tourki, "A modified aes based algorithm for image encryption," *International Journal of Computer Science and Engineering*, vol. 1, no. 1, pp. 70–75, 2007.
- [169] Z. Yun-Peng, L. Wei, C. Shui-ping, Z. Zheng-jun, N. Xuan, and D. Wei-di, "Digital image encryption algorithm based on chaos and improved des," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pp. 474–479, IEEE, 2009.
- [170] Z. Chen, Jand Zhu, C. Fu, L. Zhang, and Z. Y, "An image encryption scheme using nonlinear inter-pixel computing and swapping based permutation approach," *Communications in Nonlinear Science and Numerical Simulation*, vol. 23, no. 1-3, pp. 294–310, 2015.

- [171] N. K. Pareek, V. Patidar, and k. k. Sud, "Image encryption using chaotic logistic map," *Image and Vision Computing*, vol. 24, no. 9, pp. 926–934, 2006.
- [172] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International journal of bifurcation and chaos*, vol. 16, no. 08, pp. 2129–2151, 2006.
- [173] R. E. Boriga, A. C. Dăscălescu, and A. V. Diaconu, "A new fast image encryption scheme based on 2d chaotic maps," *IAENG International Journal of Computer Science*, vol. 41, no. 4, pp. 249–258, 2014.
- [174] M. Li, M. Wang, H. Fan, K. An, and G. Liu, "A novel plaintext-related chaotic image encryption scheme with no additional plaintext information," *Chaos, Solitons & Fractals*, vol. 158, p. 111989, 2022.
- [175] I. Hussain and M. A. Gondal, "An extended image encryption using chaotic coupled map and s-box transformation," *Nonlinear Dynamics*, vol. 76, no. 2, pp. 1355–1363, 2014.
- [176] X. Li, J. Knipe, and H. Cheng, "Image compression and encryption using tree structures," *Pattern Recognition Letters*, vol. 18, pp. 1253–1259, 1997.
- [177] H. K.-C. Chang and J.-L. Liu, "A linear quadtree compression scheme for image encryption," *Signal Processing: Image Communication*, vol. 10, no. 4, pp. 279–290, 1997.
- [178] J. Wang, X. Song, and A. A. A. El-Latif, "Efficient entropic security with joint compression and encryption approach based on compressed sensing with multiple chaotic systems," *Entropy*, vol. 24, no. 7, p. 885, 2022.
- [179] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *International Journal of Bifurcation and chaos*, vol. 8, no. 06, pp. 1259–1284, 1998.
- [180] J.-C. Yen and J.-I. Guo, "Efficient hierarchical chaotic image encryption algorithm and its vlsi realisation," *IEE Proceedings-vision, image and signal processing*, vol. 147, no. 2, pp. 167–175, 2000.
- [181] C. Li, G. Luo, K. Qin, and C. Li, "An image encryption scheme based on chaotic tent map," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 127–133, 2017.
- [182] B. Wang, B. Zhang, and X. Liu, "An image encryption approach on the basis of a time delay chaotic system," *Optik*, vol. 225, p. 165737, 2021.
- [183] Z. Hua, F. Jin, B. Xu, and H. Huang, "2d logistic-sine-coupling map for image encryption," *Signal Processing*, vol. 149, pp. 148–161, 2018.
- [184] P. Biswas, S. Kandar, and B. C. Dhara, "A novel image encryption technique using one dimensional chaotic map and circular shift technique," in *International Conference on Software and Computer Applications*, pp. 112–116, IEEE, 2017.
- [185] X.-Y. Wang, L. Yang, R. Liu, and A. Kadir, "A chaotic image encryption algorithm based on perceptron model," *Nonlinear Dynamics*, vol. 62, no. 3, pp. 615–621, 2010.

- [186] M. Farah, A. Farah, and T. Farah, "An image encryption scheme based on a new hybrid chaotic map and optimized substitution box," *Nonlinear Dynamics*, vol. 99, no. 4, pp. 3041–3064, 2020.
- [187] S. J. Sheela, K. V. Suresh, and D. Tandur, "Image encryption based on modified henon map using hybrid chaotic shift transform," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 25223–25251, 2018.
- [188] A. Pourjabbar Kari, A. Habibizad Navin, A. M. Bidgoli, and M. Mirnia, "A new image encryption scheme based on hybrid chaotic maps," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 2753–2772, 2021.
- [189] N. Khalil, A. Sarhan, and M. A. Alshewimy, "An efficient color/grayscale image encryption scheme based on hybrid chaotic maps," *Optics & Laser Technology*, vol. 143, p. 107326, 2021.
- [190] Z. Li, C. Peng, L. Li, and X. Zhu, "A novel plaintext-related image encryption scheme using hyper-chaotic system," *Nonlinear Dynamics*, vol. 94, no. 2, pp. 1319–1333, 2018.
- [191] E. Hasanzadeh and M. Yaghoobi, "A novel color image encryption algorithm based on substitution box and hyper-chaotic system with fractal keys," *Multimedia Tools and Applications*, vol. 79, no. 11, pp. 7279–7297, 2020.
- [192] M. Kaur, D. Singh, K. Sun, and U. Rawat, "Color image encryption using non-dominated sorting genetic algorithm with local chaotic search based 5d chaotic map," *Future Generation Computer Systems*, vol. 107, pp. 333–350, 2020.
- [193] X. Wang, Q. Ren, and D. Jiang, "An adjustable visual image cryptosystem based on 6d hyperchaotic system and compressive sensing," *Nonlinear Dynamics*, vol. 104, no. 4, pp. 4543–4567, 2021.
- [194] K. M. Hosny, S. T. Kamal, M. M. Darwish, and G. A. Papakostas, "New image encryption algorithm using hyperchaotic system and fibonacci q-matrix," *Electronics*, vol. 10, no. 9, p. 1066, 2021.
- [195] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using dna sequence operations," *Optics and Lasers in engineering*, vol. 88, pp. 197–213, 2017.
- [196] J. Chen, L. Chen, and Y. Zhou, "Cryptanalysis of a dna-based image encryption scheme," *Information Sciences*, vol. 520, pp. 130–141, 2020.
- [197] H. Nematzadeh, R. Enayatifar, M. Yadollahi, M. Lee, and G. Jeong, "Binary search tree image encryption with dna," *Optik*, vol. 202, p. 163505, 2020.
- [198] M. Yadollahi, R. Enayatifar, H. Nematzadeh, M. Lee, and J.-Y. Choi, "A novel image security technique based on nucleic acid concepts," *Journal of Information Security and Applications*, vol. 53, p. 102505, 2020.

- [199] A. A. Abbasi, M. Mazinani, and R. Hosseini, "Evolutionary-based image encryption using biomolecules and non-coupled map lattice," *Optics & Laser Technology*, vol. 140, p. 106974, 2021.
- [200] S. Zhu and C. Zhu, "Secure image encryption algorithm based on hyperchaos and dynamic dna coding," *Entropy*, vol. 22, no. 7, p. 772, 2020.
- [201] X. Xue, D. Zhou, and C. Zhou, "New insights into the existing image encryption algorithms based on dna coding," *PLoS One*, vol. 15, no. 10, p. e0241184, 2020.
- [202] K. Jithin and S. Sankar, "Colour image encryption algorithm combining arnold map, dna sequence operation, and a mandelbrot set," *Journal of Information Security and Applications*, vol. 50, p. 102428, 2020.
- [203] X. Wang and Y. Su, "Image encryption based on compressed sensing and dna encoding," *Signal Processing: Image Communication*, vol. 95, p. 116246, 2021.
- [204] S. Zhang and L. Liu, "A novel image encryption algorithm based on spwlcmm and dna coding," *Mathematics and Computers in Simulation*, vol. 190, pp. 723–744, 2021.
- [205] D. Wei and M. Jiang, "A fast image encryption algorithm based on parallel compressive sensing and dna sequence," *Optik*, vol. 238, p. 166748, 2021.
- [206] S. Kumar, B. Panna, and R. K. Jha, "Medical image encryption using fractional discrete cosine transform with chaotic function," *Medical & biological engineering & computing*, vol. 57, no. 11, pp. 2517–2533, 2019.
- [207] A. Banu S, R. Amirtharajan, *et al.*, "A robust medical image encryption in dual domain: chaos-dna-iwt combined approach," *Medical & biological engineering & computing*, vol. 58, no. 7, pp. 1445–1458, 2020.
- [208] P. T. Akkasaligar and S. Biradar, "Selective medical image encryption using dna cryptography," *Information Security Journal: A Global Perspective*, vol. 29, no. 2, pp. 91–101, 2020.
- [209] D. Ravichandran, A. Banu S, B. Murthy, V. Balasubramanian, S. Fathima, R. Amirtharajan, *et al.*, "An efficient medical image encryption using hybrid dna computing and chaos in transform domain," *Medical & Biological Engineering & Computing*, vol. 59, no. 3, pp. 589–605, 2021.
- [210] Y. Ding, F. Tan, Z. Qin, M. Cao, K.-K. R. Choo, and Z. Qin, "Deepkeygen: a deep learning-based stream cipher generator for medical image encryption and decryption," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [211] M. Li, T. Liang, and Y.-j. He, "Arnold transform based image scrambling method," in *3rd International Conference on Multimedia Technology*, pp. 1309–1316, 2013.

- [212] H. Zhu, C. Zhao, X. Zhang, and L. Yang, "An image encryption scheme using generalized arnold map and affine," *Journal of Optik*, vol. 125, pp. 6672–6677, 2014.
- [213] G. S. Chandel and V. Sharma, "X-or and arnold cipher based double phase image encryption technique," *International Journal of Emerging Research in Management and Technology*, vol. 4, no. 12, pp. 175–181, 2015.
- [214] M. Mishra, P. Mishra, M. Adhikary, and S. Kumar, "Image encryption using fibonacci-lucas transformation," *International Journal on Cryptography and Information Security*, vol. 2, no. 3, pp. 131–141, 2012.
- [215] C. Maiti and B. C. Dhara, "Image encryption with a new fibonacci transform," in *5th International Conference on Emerging Applications of Information Technology*, pp. 1–4, IEEE, 2018.
- [216] H. Tora, E. Gokcay, M. Turan, and M. Buker, "A generalized arnold's cat map transformation for image scrambling," *Multimedia Tools and Applications*, pp. 1–14, 2022.
- [217] J. Chen, Z. Zhu, C. Fu, H. Yu, and L. Zhang, "An efficient image encryption scheme using gray code based permutation approach," *Optics and Lasers in Engineering*, vol. 67, pp. 191–204, 2015.
- [218] R. K. Sinha, I. Agrawal, K. Jain, A. Gupta, and S. Sahu, "Image encryption using modified rubik's cube algorithm," in *Advances in computational intelligence*, pp. 69–78, Springer, 2020.
- [219] S. Kandar, D. Chaudhuri, A. Bhattacharjee, and B. C. Dhara, "Image encryption using sequence generated by cyclic group," *Journal of information security and applications*, vol. 44, pp. 117–129, 2019.
- [220] R. Vidhya and M. Brindha, "A chaos based image encryption algorithm using rubik's cube and prime factorization process (cierpf)," *Journal of King Saud University-Computer and Information Sciences*, 2020.
- [221] A. Priya, K. Sinha, M. P. Darshani, and S. K. Sahana, "A novel multimedia encryption and decryption technique using binary tree traversal," in *Proceeding of the Second International Conference on Microelectronics, Computing & Communication Systems (MCCS 2017)*, pp. 163–178, Springer, 2019.
- [222] P. Biswas, S. Kandar, and B. C. Dhara, "An image encryption scheme using sequence generated by interval bisection of polynomial function," *Multimedia Tools and Applications*, vol. 79, no. 43, pp. 31715–31738, 2020.
- [223] A. Paul and S. Kandar, "Simultaneous encryption of multiple images using pseudo-random sequences generated by modified newton-raphson technique," *Multimedia Tools and Applications*, vol. 81, no. 10, pp. 14355–14378, 2022.
- [224] A. Paul, S. Kandar, and B. C. Dhara, "Image encryption using permutation generated by modified regula-falsi method," *Applied Intelligence*, pp. 1–20, 2022.

- [225] T. Sivakumar and P. Li, "A secure image encryption method using scan pattern and random key stream derived from laser chaos," *Optics & Laser Technology*, vol. 111, pp. 196–204, 2019.
- [226] T. Sivakumar, M. Pandi, N. S. Madasamy, and R. Bharathi, "An image encryption algorithm with hermite chaotic polynomials and scan pattern," in *Journal of Physics: Conference Series*, vol. 1767, p. 012044, IOP Publishing, 2021.
- [227] S. K. Das and B. C. Dhara, "An image encryption technique using sine curve," in *International Conference on Advances in Pattern Recognition (ICAPR-2017)*., p. accepted, IEEE, 2017.
- [228] S. K. Das and B. C. Dhara, "A new image encryption method using circle," in *Computing Communication and Networking Technologies 2017. IEEE International Conference on*, pp. 1–6, IEEE, 2017.
- [229] Z. Tang, Y. Yang, S. Xu, C. Yu, and X. Zhang, "Image encryption with double spiral scans and chaotic maps," *Security and Communication Networks*, vol. 2019, 2019.
- [230] R. Vidhya, M. Brindha, and N. A. Gounden, "Analysis of zig-zag scan based modified feedback convolution algorithm against differential attacks and its application to image encryption," *Applied Intelligence*, vol. 50, no. 10, pp. 3101–3124, 2020.
- [231] K. L. Chung and L. C. Chang, "Large encrypting binary images with higher security," *Pattern Recognition Letters*, vol. 19, no. 5, pp. 461–468, 1998.
- [232] W. Jia, F. J. Wen, Y. T. Chow, and C. Zhou, "Binary image encryption based on interference of two phase-only masks," *Journal of Appl. Opt.*, vol. 51, no. 21, pp. 5253–5258, 2012.
- [233] M. F. Mursi, H. E. H. Ahmed, F. E. Abd El-samie, and A. H. Abd El-aziem, "Image security with different techniques of cryptography and coding: A survey," *IOSR Journal of Computer Engineering*, vol. 16, no. 3, pp. 39–45, 2014.
- [234] R. Liu, "New binary image encryption algorithm based on combination of confusion and diffusion," *Journal of Chemical and Pharmaceutical Research*, vol. 6, no. 7, pp. 621–629, 2014.
- [235] A. Houas, Z. Mokhtari, K. E. Melkemi, and A. Boussaad, "A novel binary image encryption algorithm based on diffuse representation," *Engineering Science and Technology, an International Journal*, vol. 19, no. 4, pp. 1887–1894, 2016.
- [236] S. Radhakrishnan, R. Arthy, M. Sivasankari, B. Jegajothi, and M. Mohamed, "A modified binary encryption algorithm based on diffuse representation," *International Journal of Advanced Engineering, Management and Science*, vol. 3, no. 8, pp. 825–828, 2017.
- [237] M. Spivey, "Fibonacci identities via the determinant sum property," *The College Mathematics Journal*, vol. 37, no. 4, pp. 286–289, 2006.

- [238] Y. K. Panwar, B. Singh, and V. K. Gupta, "Generalized fibonacci sequences and its properties," *Palestine Journal of Mathematics*, vol. 3, no. 1, pp. 141–147, 2014.
- [239] M. Basu and M. Das, "Tribonacci matrices and a new coding theory," *Discrete Mathematics, Algorithm and Applications*, World Scientific, vol. 6, no. 1, pp. 1450008(1)–1450008(17), 2014.
- [240] "Usc-sipi image database website."
- [241] C. Maiti, B. C. Dhara, S. Umer, and V. Asari, "An efficient and secure method of plaintext-based image encryption using fibonacci and tribonacci transformations," *IEEE Access*, vol. 11, pp. 48421–48440, 2023.

Chinmay Maiti
20/02/2024

