

A GENERALIZATION OF ARNOLD'S CAT MAP AND FRACTION BASED  
EMBEDDING IN IMAGE STEGANOGRAPHY



MOHAMED BUKER

July, 2019

A GENERALIZATION OF ARNOLD'S CAT MAP AND FRACTION BASED  
EMBEDDING IN IMAGE STEGANOGRAPHY

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF  
ATILIM UNIVERSITY

BY

MOHAMED BUKER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE DEGREE OF DOCTOR OF PHILOSOPHY

IN

MODELING AND DESIGN OF ENGINEERING SYSTEMS

(MAIN FIELD OF STUDY: COMPUTER ENGINEERING)

JULY, 2019

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

---

Prof. Dr. Ali Kara  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Doctor of Philosophy in Modeling and Design of Engineering Systems (MODES)** **Atilim University**.

---

Assoc. Prof. Dr. Ender Keskinliç  
Head of Department

This is to certify that we have read the thesis A GENERALIZATION OF ARNOLD'S CAT MAP AND FRACTION BASED EMBEDDING IN IMAGE STEGANOGRAPHY submitted by MOHAMED BUKER and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

---

Asst. Prof. Dr. Erhan Gökçay  
Co-Supervisor

---

Asst. Prof. Dr. Hakan Tora  
Supervisor

**Examining Committee Members:**

Asst. Prof. Dr. Emre Sumer  
Computer Engineering Department, Baskent University \_\_\_\_\_

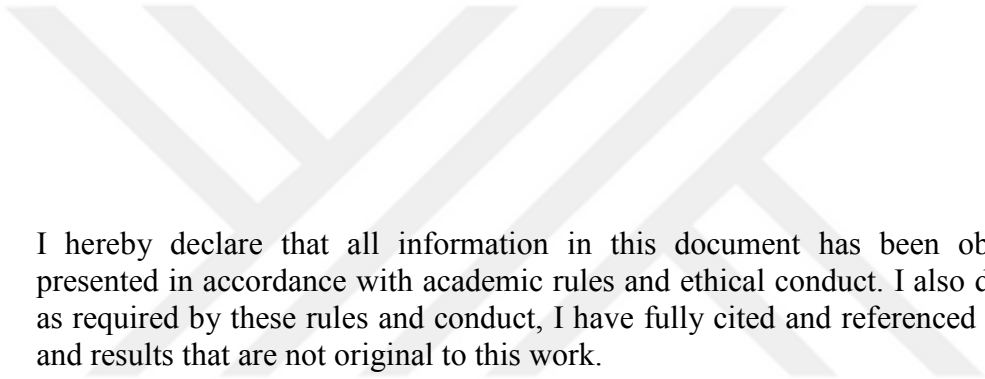
Asst. Prof. Dr. Hakan Tora  
Electric & Electronic Department , Atilim University \_\_\_\_\_

Asst. Prof. Dr. Baran Uslu  
Electric & Electronic Department , Atilim University \_\_\_\_\_

Assoc. Prof. Dr. Mehmet Turan  
Mathematics Department, Atilim University \_\_\_\_\_

Assoc. Prof. Dr. Fikret Ari  
Electrical & Electronics Eng. Department, Ankara University \_\_\_\_\_

**Date:** June 27, 2019



I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, LastName : Mohamed Buker

Signature :

## ABSTRACT

### A GENERALIZATION OF ARNOLD'S CAT MAP AND FRACTION BASED EMBEDDING IN IMAGE STEGANOGRAPHY

Buker, Mohamed

PH.D, Department of Modeling and Design of Engineering Systems

Supervisor : Asst. Prof. Dr. Hakan Tora

Co-Supervisor : Asst. Prof. Dr. Erhan Gökçay

July 2019, 90 pages

The rapid development of data communication, and the increased amount of information that are communicated via networks, make it very important to find new ways to protect exchanged information. Encryption is one of the most widely used methods nowadays in this area. Steganography is a recent field of research in which the communicated information is being invisible to anyone rather than being only encrypted. The idea behind steganography is to hide the existence of information itself.

As long as a third party knew there were information, whether encrypted or not encrypted, the information will be at risk. In this thesis, we present a steganographic model with two levels of security. First, the secret image is scrambled using our Generalized Arnold Cat Map (ACM). Then, the scrambled image is embedded into another image using our Fraction Based Embedding Technique (FBE) in the transform domain using both Discrete Wavelet Transform (DWT) and Lifted Wavelet Transform (LWT). The efficiency of our model was tested on benchmark color images. Peak Signal-to-Noise Ratio (PSNR), Mean Square Error (MSE), Structural Similarity (SSIM) and correlation values are calculated. Results show that our Generalized ACM is more robust compared to standard and modified versions of ACM. At the same time, results of our new FBE technique performs better than those of other techniques regarding to PSNR and MSE values.

**Keywords:** Steganography, Arnold Transform, Least Significant Bit, Discrete Wavelet Transform, Lifted Wavelet Transform.

## ÖZ

### ARNOLD CAT DÖNÜŞÜMÜNÜN GENELLEŞTİRİLMESİ VE GÖRÜNTÜ STEGANOĞRAFİSİNDE KESİR TABANLI GÖMME

Buker, Mohamed

Doktora, Mühendislik Sistemlerinin Modellenmesi ve Tasarımı

Tez Yöneticisi : Dr. Öğr. Üyesi Hakan Toru

Ortak Tez Yöneticisi : Dr. Öğr. Üyesi Erhan Gökçay

Temmuz 2019, 90 sayfa

Veri iletişiminin hızlı gelişimi ve ağlar aracılığıyla iletilen bilgilerin artması, değişik tokuş edilen bilgileri korumanın yeni yollarını bulmayı çok önemli kılmaktadır. Şifreleme günümüzde bu alanda en yaygın kullanılan yöntemlerden biridir. Steganografi, iletilen bilgilerin yalnızca şifrelenmekten ziyade herkes tarafından görünmez olduğu araştırma alanıdır. Steganografinin arkasındaki fikir bilginin varlığını gizlemektir.

Bir üçüncü taraf bilgi olduğunu bildiği sürece, şifreli olsun ya da olmasın, bilgi risk altında olacaktır. Bu tezde, iki güvenlik seviyeli bir steganografik model sunuyoruz. İlk olarak, gizli görüntü Genelleştirilmiş Arnold CAT Haritamız (ACM) kullanılarak karıştırılmıştır. Daha sonra, karıştırılmış görüntü, dönüşüm bölgesinde hem Ayrık Dalgacık Dönüşümü (DWT) hem de Kaldırılmış Dalgacık Dönüşümü (LWT) ile Kesir Tabanlı Gömme Tekniğimizi (FBE) kullanarak başka bir görüntünün içine gömülür. Modelimizin verimliliği, referans renkli görüntüler üzerinde test edildi. Tepe Sinyal Gürültü Oranı (PSNR), Ortalama Kare Hatası (MSE), Yapısal Benzerlik (SSIM) ve Korelasyon değerleri hesaplandı. Sonuçlar, Genelleştirilmiş ACM'mizin, ACM'nin standart ve değiştirilmiş versiyonlarına kıyasla daha sağlam olduğunu göstermektedir. Aynı zamanda, yeni FBE tekniğimizin sonuçları, PSNR ve MSE değerleri ile ilgili diğer tekniklerden daha iyi performans göstermektedir.

Anahtar Kelimeler: Steganografi, Arnold Transform, En Az Önemli Bit, Kesikli Dalgacık Dönüşümü, Kaldırma Dalgacık Dönüşümü.

To my family; my Mother, my Wife and my Sons for their love and support.

## ACKNOWLEDGMENTS

I would like to thank my supervisor, Assistant Professor Dr. Hakan Tora, for his assistance and guidance during my research. He spent about three years of constant support, interesting discussions, and valuable advises. I consider it my good fortune that I had the opportunity to work with and learn from him.

A great thankfulness goes to the assistance supervisor, Assistant Professor Dr. Erhan Gökçay for his valuable recommendations of which helped me a lot in finalizing my research in this form.

Also I must thank Associate Professor Dr. Mehmet Turan for his support and his assistance especially in the mathematical issues of my research.

Finally, my gratitude goes to both Dr. Emre Sümer for his useful additions to this thesis, and Dr. Fikret Arı who provided me valuable comments.

They all, read and evaluated my thesis in order to help me finalize the study in a timely manner. I'm sure that my thesis is stronger because of all of them.



## TABLE OF CONTENTS

ABSTRACT .....	iv
ÖZ .....	v
DEDICATION .....	vi
ACKNOWLEDGMENTS .....	vii
TABLE OF CONTENTS .....	viii
LIST OF TABLES .....	x
LIST OF FIGURES .....	xi
CHAPTER	
1. INTRODUCTION .....	1
1.1 BACKGROUND ON STEGANOGRAPHY .....	1
1.2 RELATED TOPICS .....	2
1.3 RESEARCH OBJECTIVES .....	3
1.4 MAIN CONTRIBUTIONS .....	4
1.5 OUTLINE OF THE THESIS .....	5
2. DATA SECURITY .....	6
2.1 ENCRYPTION .....	6
2.1.1 HISTORY .....	7
2.1.2 MODERN ENCRYPTION .....	7
2.1.2.1 SYMMETRIC KEY ENCRYPTION (Private-Key) .....	8
2.1.2.2 ASYMMETRIC KEY ENCRYPTION (Public-Key).....	9
2.1.3 ARNOLD CAT MAP (ACM).....	9
2.2 STEGANOGRAPHY .....	11
2.2.1 HISTORY .....	11
2.2.2 TYPES OF STEGANOGRAPHY .....	12
2.2.3 EVALUATION CRITERIA FOR STEGANOGRAPHY .....	14
2.2.4 STEGANOGRAPHY TECHNIQUES .....	15
2.2.4.1 STEGANOGRAPHY IN SPATIAL DOMAIN.....	15
2.2.4.1.1 LEAST SIGNIFICANT BIT (LSB) SUBSTITUTION.....	15
2.2.4.1.2 LSB MATCHING.....	16

2.2.4.1.3	LSB MATCHING REVISITED (LSBMR).....	17
2.2.4.1.4	BIT PLANE COMPLEXITY SEGMENT ( BPCS).....	17
2.2.4.1.5	PÍXEL VALUE DIFFERENCING TECHNIQUE (PVD)...	18
2.2.4.2	STEGANOGRAPHY IN TRANSFORM DOMAIN .....	18
2.2.4.2.1	DISCRETE COSINE TRANSFORM (DCT) .....	18
2.2.4.2.2	DISCRETE WAVELET TRANSFORM (DWT) .....	19
2.3	STEGANALYSIS.....	21
3.	RELATED STUDIES .....	22
3.1	ARNOLD CAT MAP .....	22
3.2	STEGANOGRAPHY.....	25
3.2.1	SPATIAL DOMAIN METHODS.....	25
3.2.2	TRANSFORM DOMAIN METHODS .....	28
4.	METHODOLOGY AND PROPOSED WORK .....	30
4.1	DEFINING THE STEGANOGRAPHY MODEL.....	30
4.2	EVALUATION METRICS .....	31
4.3	ENCRYPTION PART .....	33
4.3.1	GENERALIZED ARNOLD CAT MAP.....	33
4.3.2	EXPERIMENTAL RESULTS.....	39
4.3.2.1	ADJACENT PIXEL CORRELATION ANALYSIS.....	39
4.3.3	ROBUSTNESS OF THE PROPOSED METHOD.....	42
4.3.4	ENCRYPTING NON-SQUARE IMAGES .....	43
4.3.5	ADVANTAGES OF THE PROPOSED METHOD.....	44
4.4	STEGANOGRAPHY PART .....	45
4.4.1	LSB INSERTION METHOD .....	45
4.4.1.1	LSB INSERTION WITH ENCRYPTED SECRET IMAGE .	47
4.4.1.2	LSB LOSSLESS METHOD .....	47
4.4.2	EMBEDDING IN WAVELET TRANSFORMATION .....	48
4.4.2.1	DISCRETE WAVELET TRANSFORM (DWT) .....	49
4.4.2.2	STATIONARY WAVELET TRANSFORM (SWT) .....	50
4.4.2.3	LIFTED WAVELET TRANSFORM (LWT).....	52
4.4.2.4	COMBINATION OF DWT,SWT,LWT .....	54
4.4.3	LWT WITH LSB TECHNIQUE .....	56

4.4.4	PROPOSED METHOD (FBE)	58
4.4.4.1	EXPERIMENTAL RESULTS	61
4.4.4.2	ROBUSTNESS AGAINST JPEG COMPRESSION	70
4.4.4.3	ROBUSTNESS AGAINST GAUSSIAN NOISE	72
4.5	THE INTEGRATED MODEL	74
5.	DISCUSSION ON EXPERIMENTAL RESULTS	76
6.	CONCLUSIONS AND FUTURE WORK	79
6.1	CONCLUSIONS	79
6.2	FUTURE WORK	80
	REFERENCES	81

## LIST OF TABLES

### TABLES

Table 4.1 Algorithm for generating transformation matrix .....	35
Table 4.2 Correlation values of original and encrypted image .....	40
Table 4.3 Similarity analysis between original and encrypted images .....	41
Table 4.4 Correlation of neighborhood in matrix sample .....	43
Table 4.5 MSE and PSNR values for LSB method .....	46
Table 4.6 PSNR values of stego images in lossless embedding .....	48
Table 4.7 PSNR values of stego images at different scaling factor $\alpha$ .....	50
Table 4.8 PSNR values for extracted images at different value of $\alpha$ .....	50
Table 4.9 PSNR values for stego images at different values of $\alpha$ .....	52
Table 4.10 PSNR values for extracted images at different values of $\alpha$ .....	52
Table 4.11 PSNR and MSE values for stego images with different values of $\alpha$ .....	53
Table 4.12 PSNR and MSE values for Extract images with different values of $\alpha$ ..	54
Table 4.13 PSNR values for stego and extracted images (peppers) .....	56
Table 4.14 PSNR values of stego images in LWT-LSB method .....	56
Table 4.15 PSNR values of extracted images in LWT-LSB method .....	56
Table 4.16 PSNR of stego image using LWT-LSB method (5 bits) .....	57
Table 4.17 PSNR of extracted image using LWT-LSB method (5 bits) .....	57
Table 4.18 PSNR of stego images with scaling $\alpha=0.5$ .....	57
Table 4.19 Results for stego and extracted images in FBE method in SWT .....	60
Table 4.20 PSNR values for stego and extracted images in FBE method .....	62
Table 4.21 PSNR resulting values for extracted images .....	63
Table 4.22 Resulting values for extracted images from <i>Peppers</i> Image .....	64
Table 4.23 Resulting values for extracted images from <i>Lenna</i> Image .....	65
Table 4.24 Resulting values for extracted images from <i>Barbara</i> Image .....	66
Table 4.25 Resulting values for extracted images from <i>Sailboat</i> Image .....	67
Table 4.26 Resulting values for extracted images from <i>Peng</i> Image .....	68
Table 4.27 PSNR values for stego and extracted using JPEG format .....	69

Table 4.28 PSNR values of FBE method in the gray scale level.....	70
Table 4.29 PSNR values for extracted images with compression .....	72
Table 4.30 PSNR values for the extracted images with Gaussian noise (m=0).....	73
Table 4.31 PSNR values for the extracted images with Gaussian noise (m=0.5)....	74
Table 5.1 PSNR values for stego and extracted images in FBE method .....	76
Table 5.2 PSNR values for stego images in different methods .....	77



## LIST OF FIGURES

### FIGURES

Figure 2.1 Encryption/Decryption Process .....	6
Figure 2.2 Encryption process at different iteration numbers.....	10
Figure 2.3 Types of secret data carriers .....	12
Figure 2.4 LSB process .....	15
Figure 2.5 The 4 LSB embedding.....	16
Figure 2.6 DCT transformation of Lenna Image .....	19
Figure 2.7 Block diagram of DWT process .....	20
Figure 2.8 First and second level decomposition of DWT .....	21
Figure 4.1 Mathematical Model of Steganography.....	30
Figure 4.2 Matrix Operations Sequence Order .....	36
Figure 4.3 Encryption process .....	38
Figure 4.4 Decryption process .....	38
Figure 4.5 Decrypted image at iterations 48 and 98 .....	38
Figure 4.6 Correlation Analysis of original image .....	39
Figure 4.7 Peppers image at different iterations during scrambling process .....	39
Figure 4.8 Correlation Analysis of Encrypted Peppers image at iteration 157.....	40
Figure 4.9 Correlation scores for different iterations.....	41
Figure 4.10 Best Correlation scores for different matrices .....	42
Figure 4.11 Best retrieved image .....	42
Figure 4.12 Encryption process of non-square image.....	44
Figure 4.13 Decryption process of non-square image .....	44
Figure 4.14 Bit-planes of a grey-scale image.....	45
Figure 4.15 Embedding one bit in LSB .....	46
Figure 4.16 Embedding four bits 4LSB .....	46
Figure 4.17 4LSB Embedding with XOR encryption.....	47
Figure 4.18 Divided Cover Image.....	47
Figure 4.19 Lossless Embedding Procedure .....	47
Figure 4.20 Lossless Embedding .....	48

Figure 4.21 Embedding process using DWT .....	49
Figure 4.22 Stego images of DWT Embedding .....	49
Figure 4.23 Stego and extracted images at different value of $\alpha$ .....	50
Figure 4.24 SWT first level decomposition .....	51
Figure 4.25 Stego and extracted images using SWT .....	51
Figure 4.26 comparison between DWT,SWT and LWT Decompositions .....	53
Figure 4.27 Stego and extracted images using LWT .....	53
Figure 4.28 Stego images in different wavelet combinations.....	54
Figure 4.29 Extracted images by inverse in different wavelet combinations .....	55
Figure 4.30 Directly Extracted images in different wavelet combinations.....	55
Figure 4.31 Stego images of LWT-LSB method (4 bits).....	56
Figure 4.32 Stego images of LWT-LSB method (5 bits).....	57
Figure 4.33 SWT and DWT coefficient blocks during embedding .....	59
Figure 4.34 FBE in SWT transform.....	59
Figure 4.35 Stego and Extracted images using FBE.....	60
Figure 4.36 Block Diagramm of FBE - Embedding process .....	61
Figure 4.37 Block Diagramm of FBE - Extracting process .....	61
Figure 4.38 A graphical representation for SSIM values.....	69
Figure 4.39 Stego and Extracted Images using JPEG format .....	69
Figure 4.40 Stego and extracted images in the gray scale level.....	70
Figure 4.41 The integrated Model for embedding encrypted images .....	71
Figure 4.42 Extracted images with different compression ratios.....	72
Figure 4.43 Extracted images with Gaussian noise (m=0) .....	73
Figure 4.44 Extracted images with Gaussian noise (m=0.5) .....	74

## LIST OF ABBREVIATIONS

ACM	Arnold Cat Map
AES	Advanced Encryption Standard
ARP	Address Request Protocol
BMP	Bit Map image
BPCS	Bit Plane Complexity Segmentation
dB	Decibels
DCT	Discrete Cosine Transform
DES	Data Encryption Standard
DOS	Disk Operating System
DWT	Discrete Wavelet Transform
EOF	End of File
FBE	Fraction Based Embedding
HTML	Hyper Text Mark-up Language
HVS	Human Visual System
IP	Internet Address protocol
JPEG	Joint Photographic Experts Group
LCG	Linear Congruential number Generator
LSB	Least Significant Bit
LSBMR	LSB Matching Revisited
LWT	Lifted Wavelet Transform
MSB	Most Significant Bit
MSE	Mean Square Error
PFM	Prime First Mapping
PM1	Plus Minus one method
POS	Point of Sale
PRNG	Pseudo-Random Number Generator
PSNR	Peak Signal-to-Noise Ratio
PVD	Pixel Value Differencing Technique
RGB	Red Green Blue color image



SSIM	Structural Similarity
SSL	Secure Socket Layer
SWT	Stationary Wavelet Transform
TCP	Transfer Control Protocol
UDP	User Data Protocol
UED	Uniform Embedding Distortion
VQ	Vector quantization method
WEP	Wired Equivalent Privacy
XML	Extensible Markup Language



# CHAPTER 1

## INTRODUCTION

### 1.1 Background

In spite of the development of encryption techniques, there is a parallel effort in the development of intrusion hackers that became able to interrupt and access the communicated information. This showed the need to adopt more sophisticated technology in order to protect the information and preserve confidentiality. The best way to achieve this is by using steganography in which the communicated information is being invisible to anyone rather than being encrypted.

It could be said that there are very strong encryption algorithms available and it may take several years to be decrypted. However, there could be some information worth trying over these years. Furthermore, hackers continue to evolve and what it may take years to break, it may not need a few hours tomorrow.

Relying only on encryption, any attacker can see the data in an encrypted form, and this is sufficient for him to believe that sensitive data exists and he will start using anti-encryption techniques to try to obtain the original content, even if he could not recover the original content, he may destroy it to prevent it from reaching its goal.

In steganography, secret communication is masked by an unsuspecting looking communication. The way to achieve this task is to embed the secret data in a familiar looking cover media.

By using steganography, the hacker must first look for the place where the information is hidden, then after that think about decrypting or extracting it.

The objective of the concealment process is not to raise any point of doubt about the presence of hidden data. There are many techniques used to conceal secret data inside different media types such as images.

A very simple way to conceal text data inside an image can be achieved by applying a single DOS command window. It is known that many file types contain an end tag at the end. Anything written after the EOF tag will be discarded and will not be considered a part of the image. So we can write any secret data there and this information will not appear when the file is opened using any photo editor normally. Of course this method is easily detected.

This can be done simply by issuing the following DOS command in windows platform

```
C:\> Copy carrier.bmp /b + Secret.txt /b secret.bmp
```

This command will append the secret data contained in the file "Secret.txt" at the end of the image file "Carrier.bmp". A new image file will be created with the name "secret.bmp".

When any one tries to view the *secret.bmp* file using any photo editor, only the image will be displayed and anything embedded after the EOF tag will be ignored.

However, we can see the hidden message at the end of the stego file if we opened it using the notepad program or any other text editor.

Different steganography techniques will be discussed in chapter 2.

## 1.2 Related Topics

There is a confusion between three techniques in data security. These techniques are interlinked with each other. These are cryptography, watermarking and steganography. Here we will give a brief discussion about them.

**Cryptography** is the procedure of transforming data into a non-understandable code and only the involved people can decode the message and read it. Encryption is used in everyday life as a common way to protect transactions over insecure channels of communication including the internet. Encryption algorithms are responsible to protect data being communicated among several devices such as personal computers, point of sale POS, mobile telephones, and more. The algorithm that is used to encrypt and decrypt information is called a cipher. The oldest known cipher is called the Caesar cipher.

**Watermarking** is a special case of data hiding like steganography in which both techniques conceal some data inside a digital media. The difference is in the purpose of data concealing. The most common application of watermarking is to protect copyright, by intending to prevent any illegal copying of digital media.

Watermarking is done by embedding the information which verifies the owner into a digital media. This media could be image or video. Embedded information is used to verify ownership, ensure fidelity, and protect copyrights by tracking a particular copy of the content.

Depending on the purpose of Watermarking, it is classified in two types; visible watermarking and invisible watermarking. Visible watermarks are usually logos or text. As an example, TV channels are putting their logo at one of the corners of the screen. A visible watermark is used in paper currency to protect it from forgery. Invisible watermarking is the process that adds some information inside a video, audio or picture such that it will not be visible or noticeable except to its creator which uses this watermark to verify originality.

**Steganography** differs from cryptography. While the main goal of cryptography is to make the meaning of a message secret, steganography is intended to hide the existence of the secret message [1]. It is also different from watermarking. While watermarking embeds data into a digital image to protect copyrights, steganography embeds secret data inside a digital media to protect its confidentiality.

### **1.3 Research Objectives**

Many researchers presented new techniques for image encryption and image steganography. One of the most widely used methods in scrambling digital images is the Arnold Cat Map (ACM). This method is not secure and scrambled images could be descrambled easily. Although many modified versions of the original ACM was produced by some researchers, these versions still have their weak points. In this thesis we try to propose a Generalized ACM that overcomes the weak points of those methods.

Meanwhile, we know that relying only on image scrambling is not enough for secret communication, we try to make some contribution in the steganography part. Embedding secret messages into cover images usually affects the image resolution

with some kind of distortion and hence reducing the image quality. This distortion may be not visible to the human eye, but detectable by statistical analysis.

The question is that, can there be any way of embedding secret messages into cover images such that we can get some or all of the following benefits?

- Keeping the quality of the stego image as high as possible, and making it robust enough against steganalysis tools.
- Embedding as much secret data as possible,
- Extracting the secret image in a reasonable quality.

The objectives of the research can be summarized in:

- Investigation and evaluation of existing methods of image steganography
- producing an effective and robust steganography method.

Our goal is to implement a hybrid model that can be used in covert communication by combining encryption and steganography techniques in one system.

#### **1.4 Contribution**

The contribution of this work lies in two areas: image encryption and image steganography.

1- We have improved the well known image scrambling method *Arnold Cat Map*. Our method is equipped with some features which are not found in the standard and previously modified versions of ACM.

We used a new method for generating generalized transform matrices that can be used instead of the traditional or the modified Arnold variations. In our approach, the four coefficients of the transform matrix are unknown and there is no fixed structure for the transform matrix. On the other hand, we calculate the correlation value between each two adjacent pixels in the candidate encrypted image for each iteration during the period and selecting the iteration number in which the encrypted image has the lowest correlation value, this ensures best encryption degree.

2- In the steganography part, we developed a new transform domain method to conceal color image into another color image without degrading the quality of the

stego image. Our method is called Fraction Based Embedding (FBE) which exploits both Discrete Wavelet Transform and Lifted Wavelet Transform. Experimental results showed that our method outperforms most of the methods in literature.

### **1.5 Outline of the thesis**

This thesis deals with two aspects of data security; steganography and image encryption.

The thesis is organized as follows, in Chapter 1, we provide the reader with general concepts of data hiding using steganography.

In chapter 2 we present a detailed information about encryption and steganography including the techniques used to achieve them. We also give some background about steganalysis. In chapter 3 we review the literature of both Arnold Cat Map scrambling and different steganographic techniques.

In chapter 4 we introduce our methodology for implementing different applications of Arnold Cat Map and many steganographical techniques. Results are obtained and analyzed. At the end of this chapter, we introduce our new proposed method for Generalized ACM and our Fraction Based Embedding Technique FBE.

In Chapter 5 a detailed discussion on experimental results of our proposed method is presented and compared to results found in the literature. Finally, Chapter 6 summed up with conclusions and suggested future work.

## CHAPTER 2

### DATA SECURITY

In this modern and fast paced world, security became more important than ever. Huge amount of data is being transferred around the world through the internet.

In everyday life, we are communicating data. Using our mobile phones we can make calls, connect to the internet, send and receive e-mails, search for information, connect to social media, share our personal photos, do banking, online reservations, shopping and more.

Companies and organizations have sensitive information that are saved in their computers. Many business information need to be protected like company's financial records, future confidential business plans, research projects, trade secrets, and other information. These information have to be kept secret as a competitive edge.

#### 2.1 Encryption

Encryption is the process of protecting the content of messages or making their meaning not understandable.

The raw data that can be read and understood directly is called plaintext or clear text. Encrypting plaintext results in unreadable codes of what is called cipher text.

The cipher text can be converted back to its original readable plain text by a process called decryption. Figure 2.1 illustrates the encryption and decryption process.

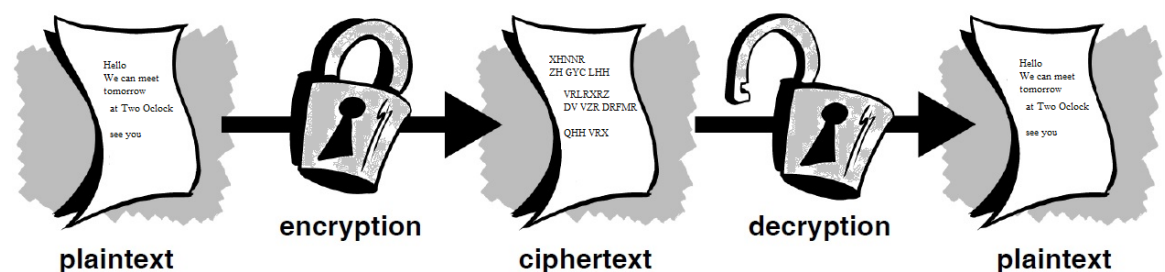


Figure 2.1 Encryption/Decryption Process

Cryptography or encryption enables us to save sensitive and confidential information during the transmission across insecure networks like the Internet and we guarantee that even the information is captured by a third party it cannot be read. The only one that can read and understand the message is the intended recipient.

The encrypted message is protected by a key of which can be used to decrypt the message.

The security of the encrypted message is depends on the strength of the used algorithm and the strength of the secret key. The same plaintext will be encrypted to different cipher text if different key is used and of course the original image cannot be decrypted using different key.

According to the *Kerchoff's* principle, he assumes that the used algorithm for encryption is not secret, the only thing that is considered to be secret is the key used to encrypt and decrypt the message.

### **2.1.1 History**

The origins of encryption began in the early ages. The oldest known application of encryption was done by Julius Caesar before 2,000 years. Caesar, the Roman commander, found a way to send and receive secure messages. He developed a substitution cipher method that substitutes letters for different letters. The secret messages could be deciphered only by those who knew the substitution order.

The substitution method he followed was very simple. He just shifted the letters of the message by a predefined number as a cipher key used to decrypt the encrypted message. For example if the Key=3, then each letter will be replaced by the letter occurring after three positions of that letter. For example, the letter A will be replaced by D, the letter B with E and so on.

### **2.1.2 Modern Encryption**

People tried to find better ways to send secret messages, they used different techniques for this purpose. In 1917, The "one-time pad" encryption algorithm was invented by Major J. Mauborgne and G. Vernam and became the most common used method for data encryption for decades [2]. This encryption method relies on a modular addition by XOR operation between a plain text and a predefined key. The method proved to be unbreakable if the secret key is kept secret. The same key that used for encryption is also used for the Decryption process. Suppose we have the



following binary representation of the plain text to be encrypted, 11010011, and we have the secret key as 10000101. Encryption and Decryption will take place as follows.

**Encryption:**

Plain text	11010011	
Secret key	10000101	<b>XOR</b>
Cipher text	01010110	

**Decryption :**

Cipher text	01010110	<b>XOR</b>
Secret key	10000101	
Plain text	11010011	

Applying the same key to the cipher text results back to the plaintext.

The key point in any encryption algorithm is the secret key. The key should be random and difficult to expect. In case of non-randomness occurrence in the key of the algorithm, the security is decreased and the algorithm will be no more unbreakable. There are numerous algorithms for generating random numbers which are called Pseudo-Random Number Generators (PRNGs).

For recent encryption algorithms, two types of encryption are available according to the used key; Symmetric key encryption and asymmetric key encryption.

**2.1.2.1 Symmetric Key Encryption (Private-Key)**

A symmetric key or private-key, sometimes called the conventional cryptography, in which one key is used for both encryption and decryption. This key should be transmitted securely to the authorized members. If any unauthorized person obtain that key, he will have the ability to decrypt all messages that are encrypted by this key.

Two types of symmetric key encryption are used, stream cipher and block cipher. stream cipher uses a stream of random numbers combined with the original message. RC4 is an example of this type. This cipher is commonly used in Secure Socket Layer (SSL) and Wired Equivalent Privacy (WEP).

Block cipher works on a fixed number of bits, commonly used block sizes are 64 and 128 bits. It converts a block of raw data to a block of cipher data of the same size.

Data Encryption Standard DES, and Advanced Encryption Standard AES are common block ciphers.

### **2.1.2.2 Asymmetric Key Encryption (Public-key)**

The main problem with private key encryption is how to distribute the key. It is necessary to submit the key to the targeted receiver securely. This problem is solved when Whitfield Diffie and Martin Hellman introduced the public key cryptography in 1975 [3].

The idea behind Public key cryptography is to use a pair of keys; a public key to encrypt data, and a corresponding private key to decrypt data. In this type of encryption, one can publish a public key to others and keeps his private key secret. This way others who received the public key can encrypt data but only the owner of the private key can decrypt and read these data.

RSA encryption is one of the most widely used asymmetric key encryption systems.

### **2.1.3. Arnold Cat Map (ACM)**

Cryptography deals with the encryption of data such as text, images, video, etc. When the encrypted data is an image, encryption techniques can be classified into two categories. They are called substitution and transposition methods. The former is based on changing the pixel's values [5]. The algorithms used for this is known as diffusion algorithms. In other words, the image pixels are completely changed. The latter is based on changing the pixel's positions. This technique utilizes algorithms called confusion. They shuffle the locations of image pixels. Arnold Cat Map (ACM) is one of the confusion techniques that scramble the image pixels. ACM is simply built on matrix transformation [4]. ACM is an iterative procedure that starts with a given image at the beginning and ends when the given image is reconstructed again. This points out that the ACM is periodic.

Russian mathematician Vladimir I discovered a method to scramble the positions of matrix elements according to a 2 by 2 transformation matrix[6]. He invented this method when he discussed his ergodic theory. Defining the linear toral endomorphism, If  $\det A = \pm 1$  then  $A^{-1}$  is an integer matrix and  $T$  is invertible with  $T^{-1} = TA^{-1}$ . In this case,  $T$  is called a linear toral automorphism.

Given a space  $X$  (the 'phase space'), a dynamical system is a rule governing how points in  $X$  evolve in time. If we consider Time to be discrete, in this case a dynamical system is given by iterating a single map.

Let  $T : X \rightarrow X$  be a dynamical system. Let  $x \in X$ . The orbit of  $x$  is the set  $\{T^n X | n \geq 0\}$

This chaotic algorithm is known as Arnold's cat map (ACM). CAT is the abbreviation of Continuous Automorphism of a Torus. The ACM is a two-dimensional invertible chaotic map[7]. For an image of size  $N \times N$ , Arnold transform is defined by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \text{ mod } N \quad (1)$$

where  $x, y \in \{0, 1, 2, \dots, N-1\}$ ,  $(x, y)$  is the pixel of original image and  $(x', y')$  is the mapped pixels. Note that the determinant of the transformation matrix is unity which guarantees that the mapping is one to one. Furthermore, area preserving is ensured by taking the module. Therefore, iterative application of the transformation will recover the original image in a finite number of steps.

Figure 2.2 illustrates an example of this transformation. The numbers above the images indicate the iteration numbers. As can be seen, original image is exactly recovered back at iteration of 100. Hence, the iteration number at which the given image is obtained is called the period of the process. Note that there exist many other images between the initial and the final states. These images are the scrambled versions of the original image. Any of them can be utilized as an encrypted image.

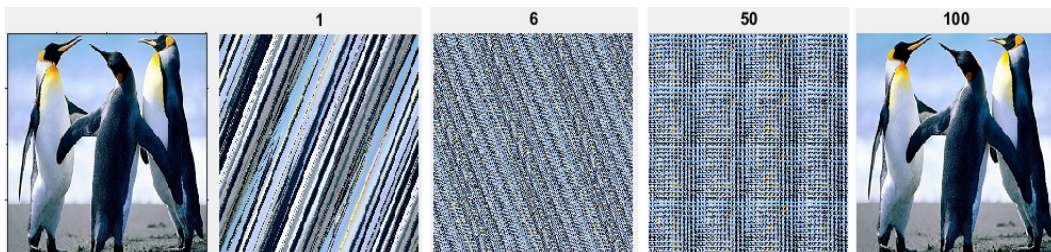


Figure 2.2 Encryption process at different iteration numbers

Beside the traditional Arnold cat map, several modifications are also introduced by researchers [8,9]. These modified ACMs differ from the traditional method in terms of the matrix elements. One of them is defined as follows.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ i & i + 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \text{mod} N \quad (2)$$

Eq.(2) allows one to select any matrix which is specified by the variable  $i$ .

Additionally, Eq. (3) is another modified version of ACM [7]

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab + 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \text{mod} N \quad (3)$$

This version gives more flexibility to the user for selecting the transformation matrix. Compared to the other two ACMs in Eqs. (1) and (2) , it is more difficult for attackers to estimate the matrix since two independent variables  $a$  and  $b$  are used such that  $a$  and  $b$  could be any integer values. That makes a scrambled image much safer.

## 2.2 Steganography

Steganography is the science that used to hide secret data in digital media. Using steganography is safer than using encryption because encryption only codes the content of the information into unreadable form; whereas steganography keeps the existence of the message hidden. In steganography, secret communication is masked by an unsuspecting looking communication. The way to achieve this task is done by embedding the secret data in a familiar looking cover media.

### 2.2.1 History

The word steganography is derived from the Greek two words “stegos” which means “cover” and “grafia” which means “writing”, it gives the meaning of “covered writing” [10].

Although the idea of data hiding is not a novelty that it has been used centuries ago, it is still unknown to most people till to date.

Ancient Romans used to write their secret messages during wars using invisible ink which is made of various natural substances such as fruit juices, lemon and milk. Messages can be read only when heated.

Ancient Greeks also used steganography in their secret communications. Greek historian Herodotus in his histories, stated that around 440 B.C., the famous Greek tyrant *Histiaeus* used a secret method to send message to *Aristagoras* to provoke them to revolt against the Persian rulers. He shaved the head of one of his slaves and tattooed a message on his scalp, and waited for the hair to grow back and send him. When the slave reached his destination, they shaved his head again to read the secret message[11].

In another story, a Greek spy in the Persian empire Demeratus wanted to send a secret message to warn Spartans of the attack that the Persian king Xerxes planning to do. He removed the wax from the wooden tablet, and scratched the secret message on the tablet and finally he reapplied the wax on the tablet and send it.

### 2.2.2 Types of Steganography

Steganography techniques can be classified according to the type of the carrier digital media or the cover. Figure 2.3 shows the different cover types.

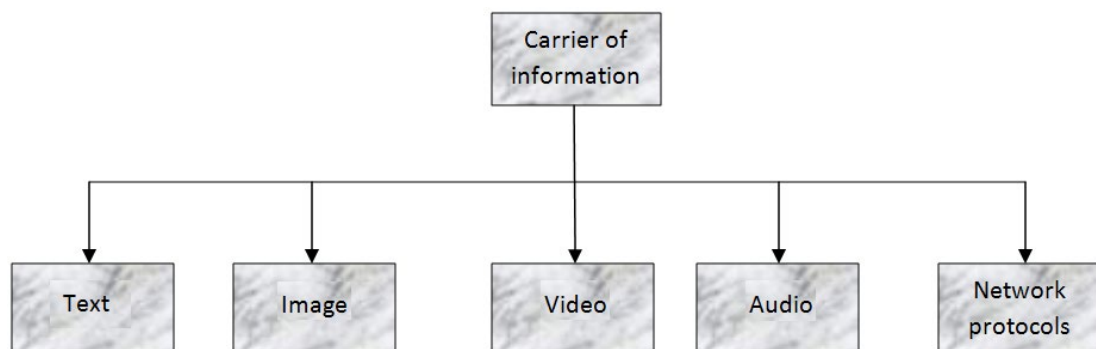


Figure 2.3 Types of secret data carriers

- **Text Steganography**

In this type, secret messages are embedded into text files. This type of steganography is somewhat difficult to achieve because of limited places to hide text in text carrier because unlike other types of carriers, text files have less redundant information. Actually the amount of payload that can be hidden in this type of carriers is small

and only text messages can be embedded. Many techniques are used to do so, one of them is to embed the secret data in extra white spaces and extra lines. Hyper Text Mark-up Language (HTML) pages are a common way to implement text steganography [12].

Secret data can be hidden in other types of files such as executable files (.EXE) and Extensible Markup Language (XML) [13].

- **Image Steganography:**

Here, secret data is embedded into a digital image. The cover image could be binary, gray scale or color image of any format. This type is preferred among other types because of many reasons. First, because of the high degree of redundancy in image data that allow us to embed much more secret data without visibly affecting the original image. Second, images are widely used and communicated throughout the internet and without arousing suspicion. Images are preferred on videos because of their relative small sizes to be communicated faster. The focus of this thesis is on steganography in digital images.

- **Audio Steganography:**

Hiding data inside different audio files such as WAV or MP3 is another alternate for steganography. It is possible to embed secret data in audio files with a very little affect on the quality of the sound as the human ear cannot notice any difference. For example, if we take a WAV file, one audio sample can be represented with a 16-bit number, and this number will take value in the range 0 to 65535. We can embed secret bits inside each of these samples, causing only a little change of these sample by 1. Thus the human ear cannot notice the affect of this little change.

- **Video Steganography:**

Video files are very big in size and can be used as a carrier for huge amount of data. Video files can be considered of multiple still images since video consists of multiple consecutive frames to represent the motion. Data can be embedded in video frames and also in its corresponding audio part. Although video files can hold huge amount of secret data, they are not preferred because of the size of video files which is usually very big to be communicated through the network.

- **Network or Protocol Steganography:**

protocol steganography is relatively a new approach for data hiding. Many network protocols can be used for data hiding, unfortunately they can be used also in stealing data. Most protocols used for data hiding are IPv4, IPv6, TCP, UDP and ARP protocols. Any reserved or unused bits of the protocol fields and header fields can be used to achieve steganography.

### 2.2.3 Evaluation Criteria for Image Steganography

There are some criteria that determine the robustness or the strength of any steganographic algorithm [14]. No one algorithm can fulfill all of these criteria, it might be good in one criteria but not in an other criteria. These criteria can be summarized as follows.

**Imperceptibility:** The algorithm is said to be imperceptible if the stego image do not contain visual artifacts after concealing secret data in it. That means, the presence of hidden data in the stego image cannot be visually detected.

**Robustness:** A reliable steganographic method must be robust enough against different image manipulations. An algorithms is said to be of robust if the hidden data is not destroyed when some image manipulations like cropping, filtering, reshaping or resizing is performed on the stego image.

Also, robustness against steganalysis is very important, that means the hidden message should remain statistically and visually undetected by statistical analysis.

**Independent of file format:** A good steganographic algorithm should have the ability to hide secret messages in any image file regardless of its format.

**Payload capacity:** Good algorithms can embed much data in cover images such that the quality of the stego image will not be degraded. This is measured by the embedding rate which refers to the ration between the size of a payload and the size of the cover media. For example if a cover file is of 10 MB size and a 1 MB of secret data is hidden in it, the embedding rate would be 10%.

**Speed:** The computational effort and time required to embed secret data and extract it should be as small as possible.

## 2.2.4 Steganography Techniques

There are a lot of steganography embedding techniques presented in the literature. These techniques are classified into two domains, spatial domain and transform domain. In the former, pixel values of the cover image are used directly to hide message bits in a way such that they will not be visible by human vision system. In the latter, a special transformation is performed on the cover image and then the message can be embedded in the transformed form of the image [15]

### 2.2.4.1 Steganography in Spatial Domain

As mentioned above, in this type of techniques, pixels of cover image are altered directly during the embedding process.

The main advantages of spatial domain techniques is that, more data can be embedded in images, while it has some disadvantages including:

1. Lack of robustness, any image manipulation causes losing the hidden data.
2. Easy detection of Hidden data.

#### 2.2.4.1.1 Least Significant Bit LSB substitution

LSB substitution or LSB replacement is the most used technique in the spatial domain because of its simplicity, its reasonable capacity with the advantage of visual imperceptibility. There are many steganography tools using this method such as Steghide, S-tools, Steganos and others.

Although LSB stego images are of visually imperceptible, presence of hidden data can be easily detected by current steganalysis methods.

The mechanism of LSB insertion method is shown in Figure 2.4. As can be seen, the LSB of the cover image is replaced with the MSB of the secret image. We can choose any number of bits to be replaced considering that more embedded bits causes more quality degradation of stego images [16].

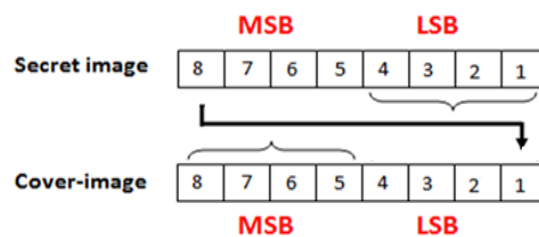


Figure 2.4 LSB process



Again, inserting bits into the LSB layer has a 50% probability of changing the value of each original pixel by at most, 1. There is a chance of one out of two that the inserted bit will be the same as the original bit and will not be changed.

In the original LSB insertion method, only one bit out of the eight bits is used to embed secret bits that is the size of secret message should not exceed 12.5 % of the cover image size. That means to embed all bits of a secret byte, we need eight bytes of cover image using the LSB method.

Perfect retrieval of a secret image requires hiding all the bits of the secret image, that means we can only hide a secret image of  $\frac{1^{th}}{8}$  the cover image size.

To increase the capacity of LSB replacement, more than one bit is replaced by secret data including the use of lowest two, three, or even four LSBs to insert a message.

Figure 2.5 illustrates four LSB embedding method.

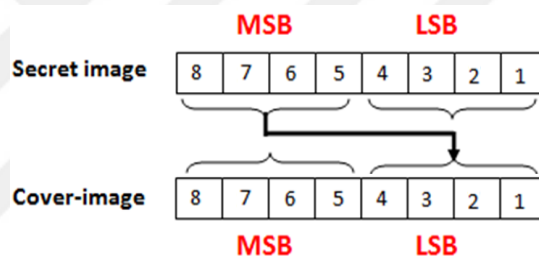


Figure 2.5 The 4 LSB embedding

Using more LSBs to hide the secret message increases the capacity of cover image, but the distortion in the stego image will be high. So there is always trade-off between payload and cover image distortion.

As the LSB replacement method is easy to detect, modified versions have been proposed to increase its capacity and to reduce the detection probabilities [17].

#### 2.2.4.1.2 LSB Matching

In LSB insertion method, embedding one bit causes even values of cover pixels to either increased by one or stay unmodified, and odd values either to be decreased by one or stay unmodified. For example, the pixel value 4, if changed, will only be changed to 5 and never to 3. This makes the LSB method easy to be detected by simple statistical analysis like Chi-Squared attack. This problem was solved by the LSB Matching or Plus Minus 1 (PM1) method.

LSB matching method was proposed in [18] and is considered to be harder to detect than LSB replacement [19]. In PM1 method the probability of increasing and decreasing for each modified pixel is the same and thus the asymmetry artifacts introduced by the traditional LSB insertion is avoided.

Of course during embedding we should not subtract 1 from pixel values of 0 and not to add 1 to the pixel values of 255.

#### **2.2.4.1.3 LSB Matching Revisited (LSBMR)**

There is a disadvantage of the LSB matching method that it changes both the histogram of the image as well as the correlation between adjacent pixels and this makes it easy for this method to be attacked by steganalysis methods [20].

In LSBMR, the embedding is performed using a pair of pixels as a unit. The LSB of the first pixel carries one bit of information, and a function of the two pixel values carries another bit of information. This process allows embedding the same payload as LSB matching with the advantage of making fewer changes to the cover image. This can avoid the problem of asymmetry and make it more difficult for detection compared to the LSBM method.

The experimental results performed on the LSBMR method show that it has a better performance than both LSB and LSB matching. It has less distortion and higher resistance against existing steganalysis [19].

#### **2.2.4.1.4 Bit Plane Complexity Segmentation (BPCS) Technique**

This technique makes use of the important characteristic of human vision. Here, the cover image is divided into two regions; informative region and high complexity region (noise-like region) according to defined threshold, then the secret data is embedded in noise region of the cover image hoping not to degrade the image quality. It differs from LSB technique, while the LSB technique hides data only in any of the 4 LSBs, BPCS technique can hide data also in MSB planes along with the LSB planes[21].

#### **2.2.4.1.5 Pixel Value Differencing Technique (PVD)**

The PVD method was proposed by Wu and Tsai to overcome the problem of asymmetry found in LSB methods [22]. This method can provide both high embedding capacity and notable imperceptibility for the stego images.

In this method, the number of embedded bits in each pixel is variable. It depends on the difference value between the pixel and its neighbor in a block. The larger the difference value, the more secret bits are embedded.

This method provides high imperceptibility to the stego image with high payloads, while keeping the consistency of the stego image characteristic.

#### **2.2.4.2 Steganography in Transform Domain**

In these techniques, the cover image is transformed into its frequency domain using a frequency oriented mechanism, then secret data is embedded in the frequency coefficients. Hence, transform domain techniques have an advantage over spatial domain techniques because we hide secret messages in areas that are less exposed to compression, cropping, and other image manipulations.

Advantages include higher level of robustness against statistical analysis. Unfortunately, it lacks high capacity of embedding.

##### **2.2.4.2.1 Discrete Cosine Transform (DCT)**

The Discrete Cosine Transform is a common process used in data compression. The JPEG and MPEG formats are examples of using DCT in data compression.

DCT separates the image into different parts of importance namely; high, middle and low frequency sub-bands. Much of the signal energy lies in the low frequency sub-band which contains the important visual parts of the image and appears in the upper left corner of the DCT. The lower right values represent the higher frequencies which contains the less important components of the image and can be removed during compression and noise removal.

In steganography, the secret messages can be embedded by modifying the coefficients of the middle or high frequency sub-band, this embedding does not affect the visibility of the image. An inverse DCT is applied to get the stego-image.

The general equation for a 2D DCT is defined by Equation 4:

$$C(u, v) = \alpha(u)\alpha(v)\alpha \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[ \frac{(2x + 1)u\pi}{2M} \right] \cos \left[ \frac{(2y + 1)v\pi}{2N} \right] \quad (4)$$

for  $u, v = 0, 1, 2, \dots, N-1$

Here, N and M are the dimensions of the image,  $c(i, j)$  is the intensity of the pixel in row  $i$  and column  $j$ ;  $C(u, v)$  is the DCT coefficient in row  $u$  and column  $v$  of the DCT matrix.

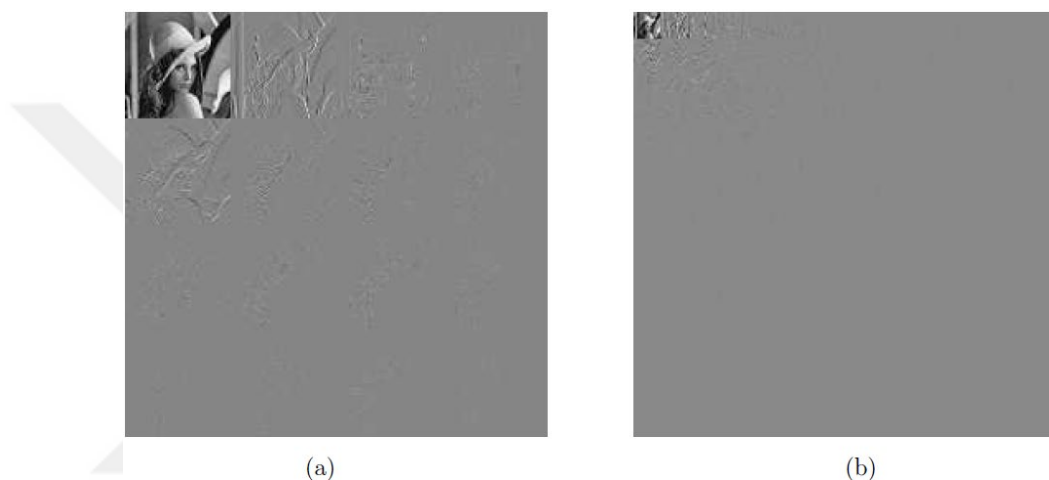


Figure 2.6 DCT transformation of Lena, (a) by 4x4 DCT, (b) by 16x16 DCT.

Example of tools that are based on DCT is the *Outguess* which was proposed by Provos [23].

#### 2.2.4.2.2 Discrete Wavelet Transforms (DWT)

A powerful multi resolution representation using Discrete Wavelet Transforms has emerged as a better alternative to the DCT. Efficient implementations of wavelet decompositions were established and the new JPEG-2000 standard replaced the old JPEG that implemented with DCT.

Compared to DCT, DWT is more powerful. It provides high compression ratios and also avoids interferences due to artifacts.

For each level of decomposition in 2D applications, first DWT is performed in the vertical direction followed by horizontal direction. resulting in dividing the image

into four frequency sub bands by performing low and high pass filters. These sub-bands are known as:

*LL*- Represents approximation details of the image resulting from applying low pass filter in both horizontal and vertical directions.

*LH* - - Represents vertical details resulting from applying vertical low pass filter and horizontal high pass filter.

*HL* - Horizontal details resulting from applying vertical high pass filter and horizontal low pass filter.

*HH* - Represents diagonal image details resulting from applying high pass filter in both horizontal and vertical directions.

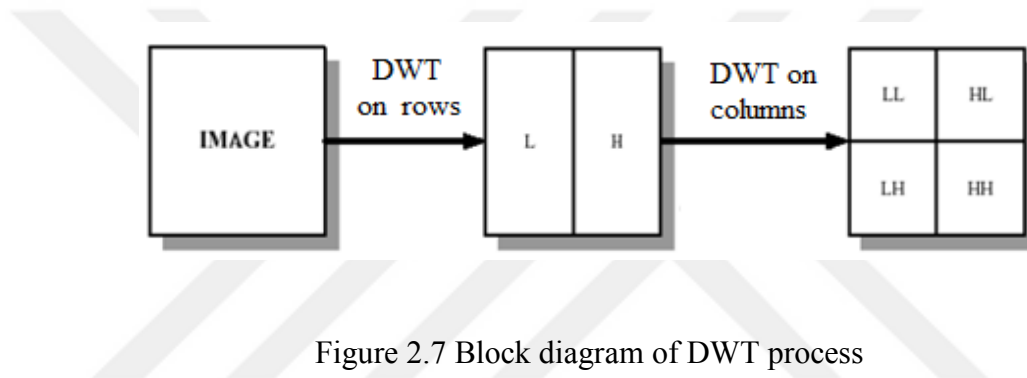


Figure 2.7 Block diagram of DWT process

A second level decomposition can be made by applying the DWT on the LL sub-band.

Usually, human eyes are more sensitive to the low frequency (LL sub band), while the other three sub-bands ( HL, LH and HH) contain unimportant information like the edge and texture details which are not sensitive to small changes. Because of that, we can hide secret data in these sub-bands without reducing the image quality. By embedding secret data inside high frequency regions, we maintain the image quality and protect the wavelet coefficients in the low frequency region which contains the most important features of the image. As a result, change in statistical properties is reduced and the probability that the secret message will be detected is also reduced. Figure 2.8 illustrates the first and second level decomposition of DWT.

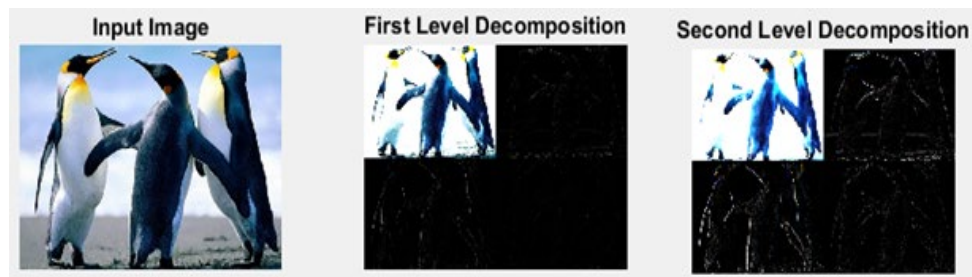


Figure 2.8 First and second level decomposition of DWT

### 2.3 Steganalysis

Steganalysis is the study of detecting secret data hidden in cover media by steganography algorithms. Once the presence of hidden data is revealed or even suspected, the purpose of steganography is partly defeated [1] and then this data could be retrieved or at least destroyed by some image manipulation.

The goal of the steganalyst is to suspect all messages sent, and check them for hidden data. Any steganalysis algorithm is called to be successful if it can determine that a file contains secret hidden information in it, and the message itself does not have to be decoded.

Embedding data bits in an image often introduces many visual artifacts and changes the image statistics. The image statistics can be analyzed by steganalysis methods, whereas visual artifacts may be detected by the human eye.

There have been many steganalysis techniques presented in the literature, each technique operates with its own unique approach. There are two approaches of steganalysis techniques.

Techniques proposed for a specific embedding algorithm, and techniques independent of the embedding algorithm which are theoretically able to operate on any embedding technique.

## CHAPTER 3

### RELATED STUDIES

In recent years, many researchers have done a lot of research on different encryption and steganography techniques. In encryption part we are interested in Arnold Cat Map for image scrambling. In the following section a review of some research papers about image scrambling and image steganography is presented.

#### 3.1 Arnold CAT Map

Arnold Cat Map ACM was first presented by the Russian mathematician Vladimir I. This method is used to scramble the positions of matrix elements according to a 2 by 2 transformation matrix[6]. Although digital images can be represented as a matrix of pixels containing integer numbers that determine the intensity of the image, ACM can be applied to scramble these images.

Standard ACM was criticized because it uses a fixed transform matrix to scramble and de-scramble images. This makes it easy to descramble any scrambled image to get the original image.

A new modified ACM that uses more flexible transform matrix was introduced. the new transform matrix takes the form  $\begin{bmatrix} 1 & 1 \\ i & i + 1 \end{bmatrix}$  instead of the standard form  $\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$ .

The new ACM make it harder to retrieve the original image because the matrix elements are no more fixed. However, only one variable  $i$  has to be known to construct the matrix. Since there are only N (size of the image) possibilities of this variable  $i$ , the construction of the transform matrix is very easy, and anyone can construct the matrix and retrieve the scrambled image in a seconds.

Another modified version of ACM is presented in [7] in which the transform matrix takes the form  $\begin{bmatrix} 1 & a \\ b & ab + 1 \end{bmatrix}$ . This version gives more flexibility to the user for selecting the transformation matrix. Logistic map was used for parameter sequences generation. The image is divided into several blocks. Finally, for each image block

Arnold transformation is applied with different parameters. Compared with the other two ACMs mentioned previously, it is more difficult for attackers to estimate the matrix since two independent variables are used. That makes a scrambled image much safer.

Kekre et al. [24] proposed an image scrambling algorithm that uses the concept of relative prime numbers. The main goal the algorithm is to minimize the correlation between any two rows and columns. Based on this aspect, correlation is calculated between the first row and every subsequent prime row, the row with the minimum correlation is moved next to the first row, this process is repeated with all rows. Once all rows are finished, a similar process is applied to columns.

In [25], the same researchers have extended their R-Prime shuffle technique over the blocks of image. The new method is applied to different block sizes and compared to the earlier R-Prime shuffle technique of image scrambling. They divided the image into non overlapping blocks. For every block, the lowest correlation obtained between the first row/column and different Relative Primes numbers (Row/Column positions) in every block is used as a key for carrying out the shuffling in each block. Different block sizes of image is applied (8x8, 16x16 and 32x32).

Daong Xu et al make use of Fibonacci sequence and show that Arnold transformation is related with Fibonacci sequence[26]. They introduced a method for image scrambling based on Fibonacci numbers called Fibonacci Q-transformation. The encoding and decoding process of this method is very simple that can be applied in real-time systems.

[27] presents a method combining the scrambling in locational space with scrambling in color space. This work offers a digital image scrambling method based on Arnold and Fermat number transformation to improving the security. Experimental results show that the gray histogram of the image is well-distributed and differs from the histogram of the original image.



In [28], a couple of new image scrambling methods are presented. The first is based on the 3-dimensional Arnold transformation matrix, while the other is based on the generalized gray code transformation.

In [29], the performance of both bakers and Arnold chaotic maps are evaluated. The performed statistical tests showed that the Arnold cat map creates more chaos than the bakers map.

In [30], a new technique for image encryption was proposed. It is based on a combination of Arnold map pixel shuffling and s-box non-linear byte substitution.

Kuang Tsan Lin[5], proposed an image encryption method by combining the Arnold transform method in the spatial domain and the Hartley transform method in the transform domain. Experimental results show that the proposed method is secure and robust against noise attacks.

Farajallah et al. proposed an algorithm that consists of two layers; substitution and permutation to encrypt image values. The algorithm uses two components: a hash function and a chaotic generator. The hash function generates dynamic keys for the substitution-permutation layers, while the secret hash key is used to authenticate the decrypted image [31].

In [32], a developed permutation and substitution image encryption method was introduced, based on Arnold 3D cat map and Turing machine in the form of dynamic random growth technique.

This hybrid technique increased the performance and enlarge the key space required to resist the brute force attacks.

A new color image encryption method is presented by [33] combining compressive sensing with Arnold transform. The compressive sensing is used for encrypting and compressing the three color components of the color image, then these components are grouped into a gray image. Finally, the gray image is scrambled by Arnold Cat Map.

A new image scrambling algorithm was proposed in [34]. In this method, the algorithm changes pixel positions and pixel values as well. Experiments show that the new algorithm with a large key space.

### **3.2 Steganography**

During last decade, a lot of research was done for different steganography techniques in both Spatial Domain and Transform Domain. One of the earliest studies concerning modern steganography was present by Simmons [35] by the famous story of Prisoners' problem (Alice and Bob) explaining what capabilities and merits steganography has to offer in public communication channel. In the following sections we will review the literature on steganography in both domains.

#### **3.2.1 Spatial Domain Methods**

In [36],[37] Bit Plane Complexity Segmentation steganography (BPCS) method is used to embed secret data in noise regions of the cover image without degrading image quality.

Deepesh, et al. presented two methods to embed the MSB of secret image into the LSB of cover image [38]. In first method, last 2 LSBs of each plane (red, green and blue) of cover image, is replaced by 2 MSBs of secret image. In the second method, last LSB of each red plane is replaced by first MSB of secret image, last 2 LSBs of each green plane by next 2 MSBs of secret image and then last 3 LSBs of blue plane is replaced by next 3 MSBs of secret image. This means that a total of 6 bits of secret image can be hidden in RGB color image.

In [39], a new LSB technique is presented by using more than one bit in a pixel. Instead of storing the data in only the least significant bit of the pixels such that this change will not affect the visual appearance of the host image. side information of neighboring pixels is used to estimate the number of bit that can be used to hide the secret data.

Kamau, et al. propose an enhanced LSB method that follows a selective and randomized approach in picking specific number of image bits to replaced with the secret data bits [40]. To facilitate the selective picking of the cover image bits, they used the standard minimal linear congruential number generator (LCG). The numbers

generated by this generator determines what bits of the cover image will be selected for embedding.

Juneja, et al. proposed an embedding algorithm to hide encrypted data in nonadjacent and random pixel locations in edges and smooth areas of the cover image [41]. They use the 1-3-4 LSBs of red, green and blue components respectively across randomly selected pixels in smooth areas of the cover image.

A novel algorithm was proposed in [42] where a combination of two methods were used. First, BPCS analysis was utilized. Next, QR Decomposition was performed to determine the region having high security in which the secret information will be embedded.

In [43], The negative of secret image is hidden in cover image using Even Odd algorithm with LSB method.

In [44], a proposed system has been designed. This method combines the use of K-Means Clustering Algorithm and LSB substitution technique (2 LSBs). Two approaches were followed.

In the first approach, secret data is embedded in one cluster, when pixels in that cluster are completely embedded proceed to the next cluster. In second approach the embedding process is done in even position pixels of even clusters and odd position pixels of odd clusters.

The research work presented in [45] employs the concept of logistic maps and genetic algorithm to select the optimal pixel positions for Least Significant Bit method.

The proposed algorithm shown in [46], hides the secret message in the cover image by manipulating either the difference or the sum of the non-overlapping blocks of two consecutive pixels. Experimental results showed that the proposed algorithm provides higher capacity than original PVD with visually high quality stego images.

In [47], a vector quantization (VQ) method is applied to a secret color image to get compressed data to be concealed into a color cover image using LSB replacement technique.

Many researchers worked on embedding the secret message in the edges of the cover image. Edge detection is based on identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities.

The alteration in edges cannot be noticed, that's why edges can hide secret data without losing the quality of the stego image. Unfortunately, edge regions are not big enough to embed secret images especially if the secret image is a color image, that's why most of the researchers are using this type of technique to embed text messages.

Some researchers like [48], [49], [50], [51] use the Canny edge detection technique to detect the edges of the image and least significant bit insertion method is used to embed the data in to the image.

In [52], the researchers use the Pixel Indicator Technique on RGB colored cover image. The technique uses two LSBs in one of the channels Red, Green or Blue as an indicator of data existence or data size in the other two channels.

In papers [53], [54] the hybrid edge detector was used to detect the edges of the cover image, then the secret data is embedded by the LSB method.

Arora, et al. [55] uses a different technique for edge detection. Here, Edges of an image are detected by scanning using 3x3 window and then text message is concealed in edges using first component alteration technique. After that Sorting method is used to randomize the edge pixels. Finally, Text data is concealed in the blue channel of sorted edge pixels.

In [56], LSB Matching Revisited LSBMR method and an edge detection method is used such that the embedding regions can be selected according to the size of the secret message and the difference between two consecutive pixels in the cover image. For lower embedding rates, only sharper edge regions are used while keeping the other smoother regions untouched.

The proposed approach in [57] uses both LSB matching and LSB substitution to hide the secret data in images. Using hybrid LSB techniques increase the level of security. The proposed approach stores two bits in a pixel to increase the maximum embedding capacity to double compared to the LSB and LSBM methods. One bit is concealed in the seventh bit using LSB-M technique; and the other bit is embedded in the eighth bit using LSB replacement technique.

### 3.2.2 Transform Domain Methods

KUMAR, et al. followed a Technique in which The cover image is converted from RGB to Grey scale and DWT is applied to the gray scale image [58]. The secret message is being hidden at the edges of the cover image.

Al-Ataby, et al. [59] followed a different technique, they made many corrections to pixels of the cover image. These corrections include level correction, contrast correction and color balance correction. A 2D Wavelet Transform is performed, and finally, a threshold calculation is done to define the redundant pixels that will be used to conceal secret data.

A proposed method combining LSB and DCT with threshold is presented in [60], random locations in the cover image based on a threshold is used to hide data. The confidential information was randomly spread throughout the cover image to make it difficult to attack by steganalysts.

Elsayed, et al. suggests using the low frequency component of Curvelet transform in embedding process to reduce the computation time [61] and to avoid affecting edges coefficients which leads to better stego image quality.

A refinement of the Uniform Embedding Distortion UED method is presented In [62]. That was done by considering the relative changes of statistical model for digital images, hoping to make the embedding modifications to be proportional to the coefficient of variation. The new method was called Uniform Embedding Revisited Distortion (UERD).

In [63], A 2D Haar Discrete Wavelet Transform is applied on the cover image, followed by applying a proposed Prime First Mapping (PFM) approach to embedded secret data, where unique concepts based on the prime and nonprime location values of the existing pixels have been used.

Kamila, et al. presented a method to hide secret bits in the three higher frequency sub-bands of the DWT to minimize the embedding impact on the cover image and not to centralize them in sensitivity domain [64].

In [65], a 2D Haar DWT has been applied to the cover, then, the secret message is encrypted by Arnold's Cat Map. After that, the secret message is concealed in the wavelet coefficients using Pixel Position Modulus Method (PPMM).

Ahani, et al. address the use of sparse representation to hide messages within non-overlapping blocks of a given color image in the wavelet domain [66].

One of color bands was used for dictionary learning and the other two color bands are used for the data embedding without causing visible artifacts or detectable statistical alteration.

Nag et al. proposed a steganographic method based on DWT and applied Huffman coding on Secret message before embedding it in the high frequency sub-bands [67]. The low frequency sub-band is kept untouched, to retain the visual quality of the image.

Aayushi Verma et al. proposed an algorithm for embedding secret image inside cover gray scale image [68]. A 2-level DWT is applied to the cover image and then the targeted band is selected to be modified. Then the size of secret image is calculated, and the five MSBs of the secret image is embedded into the high frequency bands.

## CHAPTER 4

### METHODOLOGY AND PROPOSED WORK

In this chapter we discuss the methodology used in our research work on analyzing different image scrambling and image steganography techniques and implementing new improved algorithms for both image steganography and image scrambling. We developed a highly secure model for image steganography by combining our improved algorithms in a double layered secure model.

#### 4.1 Defining the Steganography Model

The mathematical Model of the proposed method can be formulated as follows:

Let :  $M$  be the set of possible secret images to be embedded,

$C$  be the set of covers, and  $S$  be the set of stego images.

The steganographic Model consists of two processes, an embedding process which is denoted by  $F(C, M) \rightarrow S$ , and an extracting process with reverse function which is denoted by  $F^{-1}(S) \rightarrow M$  such that  $F^{-1}(F(c, m)) = m$  for arbitrary  $c \in C$ , and  $m \in M$ .

The embedding process generates a stego image  $s$  by embedding a message  $m$  into a cover image  $c$ ,  $s = F(c, m)$ . Figure 4.1 depicts the mathematical model of steganography.

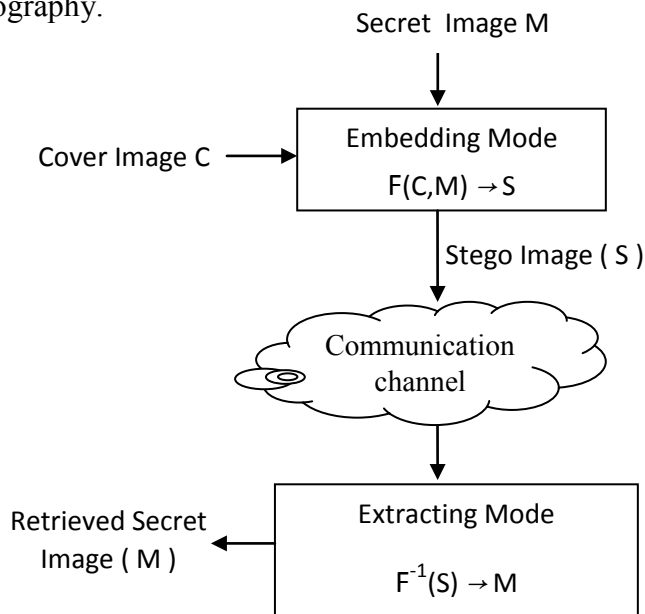


Figure 4.1 Mathematical Model of Steganography

Here we have implemented all the programs for all presented techniques using Matlab 2016a framework.

To simulate our programs we used a carefully selected images. They are selected with different features and characteristics to test the affect of the image characteristics on the steganography method. We used two image formats, JPEG format for compressed images and BMP format for uncompressed images.

For the secret image, we have used the penguin image. the reason for this selection because it contains solid patterns that could be difficult to embed in any cover image without causing quality degradation for the stego image. Also we used the secret image in the same size of the cover image to face the challenge of embedding as much as possible of secret data. For the cover images, we have used six color images with different textures, patterns, dominant colors and smooth areas to test the effect of embedding process on images with different characteristics.

The image "bird" has a huge smooth region especially with the view of the clear blue sky as a dominant background. This image and the "sea" image are considered to be the smoothest images in the set. "Peppers" is a multi-colored image and has sharp edges at almost constant intervals. "Barbara" is representative of heavy texture images. Finally, Lena was chosen to act as a moderator, because it is widely used in the field of image processing and steganogaphy.

#### 4.2 Evaluation Metrics:

To evaluate the effectiveness of our method, we have applied security analysis tests using many evaluation metrics. As a common used measures of image quality in the field of image processing, we have used the Mean Square Error MSE and Peak Signal-to-Noise Ratio PSNR as a main quality measurement for stego and extracted images. Lower MSE values correspond to better stego images while higher PSNR values correspond to better stego images. MSE is defined as :

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [C(i, j) - S(i, j)]^2 \quad (5)$$

where  $C$  is the cover image,  $S$  is the stego image, and  $m$  and  $n$  represents the dimensions of the image matrix.



PSNR is often expressed on logarithmic scale in decibels (dB) given in (6).

$$PSNR = 10 \log_{10} \left( \frac{C_{max}^2}{MSE} \right) \quad (6)$$

where,  $C_{max}$  is the maximum pixel value in the cover image.

In addition to these two metrics, we used the Structural Similarity (SSIM) Index. SSIM is commonly used to measure the quality of noisy or blurred images. In steganography, a secret image is embedded inside a cover image resulting in stego image. This stego image can be considered as a type of noisy image, so we believe that SSIM can be used as a metric to measure the quality of both stego and extracted images.

SSIM is based on the computation of three terms, namely the luminance term (brightness values), the contrast term and the structural term of the image being tested taking into consideration the assumption that the Human Visual System HVS is highly adapted to extract structural information from the viewing field.

These terms are defined as :

$$L(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (7)$$

$$C(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (8)$$

$$S(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (9)$$

Where, X and Y are the two images to be compared,

The luminance comparison function  $L(x,y)$  measures the closeness of the two images by calculating the mean luminance  $\mu_x$  and  $\mu_y$ ,

the contrast comparison function  $C(x,y)$  measures the closeness of the contrast of the two images. by computing the standard deviation  $\sigma_x$  and  $\sigma_y$ .

the structure comparison function  $S(x,y)$  measures the correlation coefficient between the two images by computing the covariance  $\sigma_{xy}$  between x and y

$C_1$ ,  $C_2$ , and  $C_3$  are small constants that handle the division by zero exception and take the values:

$C1 = (0.01 L)^2$ , where L is the specified Dynamic Range of the pixel values.

$C2 = (0.03 L)^2$ , where L is the specified Dynamic Range of the pixel values.

$C3 = C2/2$ .

And finally, SSIM is calculated with following formula:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (10)$$

### 4.3 Encryption Part

In this part, we analyzed the standard and the modified versions of Arnold Cat Map ACM for image encryption. We developed a new Generalized transform matrix that can be used in ACM with better security.

#### 4.3.1 Generalized Arnold Cat Map

As discussed in chapter 2, Standard Arnold Cat Map and improved versions of it have some disadvantages of which we try to overcome. The proposed method is based on the Arnold Cat Map and aims to provide an easy and secure scheme for encryption and decryption of digital images. We started from the fact that the transform matrix of traditional Arnold Cat Map and its variations have a determinant value of 1. Arnold cat map works too for matrices with determinants of minus one. This leads us to think of using transform matrix M in any other form such that its elements satisfy the criteria  $\det(M) = \pm 1$ .

It is well known that for a square matrix, its determinant remains unchanged when it undergoes to some matrix operations such as addition/subtraction of rows and columns, taking the transpose and inverse of the matrix, exchanging rows and columns, multiplication of two matrices with determinants of unity, moving the elements of the matrix clockwise and counter clockwise direction. In this study, we exploited such properties of determinant of a matrix to generate a generalized transformation matrix which shuffles a given image. In addition, a set of operation sequence is introduced to calculate the matrix. This sequence is required to regenerate the matrix in decryption phase and it will be called operSeq. It consists of several operations of any length and order with the condition that the resulted matrix

has a determinant of one like  $operSeq = \{\text{inverse, rotation, inverse, transpose, row operations, column operations, etc}\}$ .

The proposed algorithm starts with an initial matrix  $M1 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$  and  $M2 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$  with unity determinant and an  $operSeq$  which is a list of operations and it determines the sequence of operations applied to the initial matrix. By performing multiple operations mentioned above, the algorithm is able to generate a large number of various matrices. The execution order, type and number of operations can be modified for each encryption process which will make the estimation much harder for an attacker. Only one of these matrices will be chosen according to our  $operSeq$ .

The algorithm leads to the following matrix,  $Mg = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ .

The elements of this generalized matrix can take any integer values. There is no predefined dependency among the elements except that the determinant is  $\pm 1$ . This differs from the modified matrix in equations (2) and (3) in Chapter 2. Notice that upper left elements is always set to 1 in traditional and modified matrices. However, there is no such restriction in our proposed matrix (the set of matrices generated by our algorithm is simply a super set of the traditional and modified versions). Therefore, the proposed algorithm may result in a matrix that is exactly the same as the conventional version of ACM matrix.

Since the generalized matrix has four independent elements, estimation of this matrix is much harder than the conventional matrices. This is a desired feature for a secure encryption. Table 1 introduces a pseudo code for the proposed method. As can be seen from Table, inputs to the algorithm generate-matrix are  $M1$ ,  $M2$ ,  $B$  and  $operSeq$ . The  $operSeq$  list determines the type and number of operations needed to obtain the final transformation matrix. There is no limit to the number of operations as long as there is no integer overflow in the calculation.

To increase the possible number of combinations, the algorithm uses 2 different matrices (i.e.  $M1$  and  $M2$ ) to start with. Several operations are applied to  $M1$  matrix resulting in  $M11$ ,  $M12$  and  $M13$  where  $M1$  is reinitialized with  $M13$  for the next repetition. The same set of operations are applied to  $M2$  as well resulting a different  $M2$  at the end. The scaling constant  $c$  is used in row multiplication and it is incremented at each repetition. The largest value is limited by the parameter  $B$ .

The algorithm returns the generated matrix once the number of operations reaches the limit indicated by the *operSeq* list. Basic matrix operations are listed as, row addition with a scaling constant, transpose, inverse, interchanging rows and columns, multiplication and rotation.

Selecting the transform matrix depending on a predefined set of operations means that it will be very difficult for a potential attacker to recover the original image. This is because even in the case if the attacker knows the algorithm, he has to know the parameters of the matrix with a huge number of combinations limited only by integer overflow, otherwise he has no way to recover the image.

The pseudo code of the algorithm is given in Table 4.1.

Table 4.1 Algorithm for generating transformation matrix

<p><b>Generate-matrix</b>  <b>Input:</b> <math>M1, M2, key</math>  <b>Output:</b> <math>Mg</math> (2x2 matrix with <math>det=1</math>)  <math>counter = 0</math>  <b>for</b> <math>m = 1</math> <b>to</b> <math>A</math>  <b>for</b> <math>c = 1</math> <b>to</b> <math>B</math>            <b>step1:</b> Set <math>M = M1</math>            <b>step2:</b> add <math>c</math> times row2 to row1 and set to <math>M11</math>            <b>step3:</b> call <b>operation-proc</b>(<math>M11, key</math>)            <b>step4:</b> add <math>c</math> times row1 to row2 and set to <math>M12</math>            <b>step5:</b> call <b>operation-proc</b>(<math>M12, key</math>)            <b>step6:</b> Set <math>M13 = M11 * M12</math>            <b>step7:</b> call <b>operation-proc</b>(<math>M13, key</math>)            <b>step8:</b> Set <math>M = M2</math>            <b>step9:</b> repeat step2 to step 7            <b>step10:</b> Set <math>M23 = M13</math>  <b>end</b>  <b>step10:</b> Set <math>M1 = M13</math>  <b>step10:</b> Set <math>M2 = M23</math>  <b>end</b></p>	<p><b>Operation-proc</b>(<math>M, key</math>)          call <b>check-key</b>(<math>M, key, counter</math>)          Take Transpose of <math>M</math>          call <b>check-key</b>(<math>M, key, counter</math>)          Take Inverse of <math>M</math>          call <b>check-key</b>(<math>M, key, counter</math>)          Swap rows of <math>M</math>          call <b>check-key</b>(<math>M, key, counter</math>)          Swap columns of <math>M</math>          call <b>check-key</b>(<math>M, key, counter</math>)          Rotate <math>M</math> counter-clock wise by <math>90^\circ</math>          call <b>check-key</b>(<math>M, key, counter</math>)          Rotate <math>M</math> counter-clock wise by <math>180^\circ</math>          call <b>check-key</b>(<math>M, key, counter</math>)</p> <p><b>check-key</b>(<math>M, key, counter</math>)          if <math>counter = key</math>            obtain <math>Mg = M</math>            return(<math>Mg</math>)          end</p>
--	---

The key variable determines the operation sequence number at which the algorithm will stop running and returning the matrix  $Mg$ . The order of operations can be different for  $M11$ ,  $M12$  and  $M13$ . Figure 4.2 shows an example of the sequence order for  $M11$ .

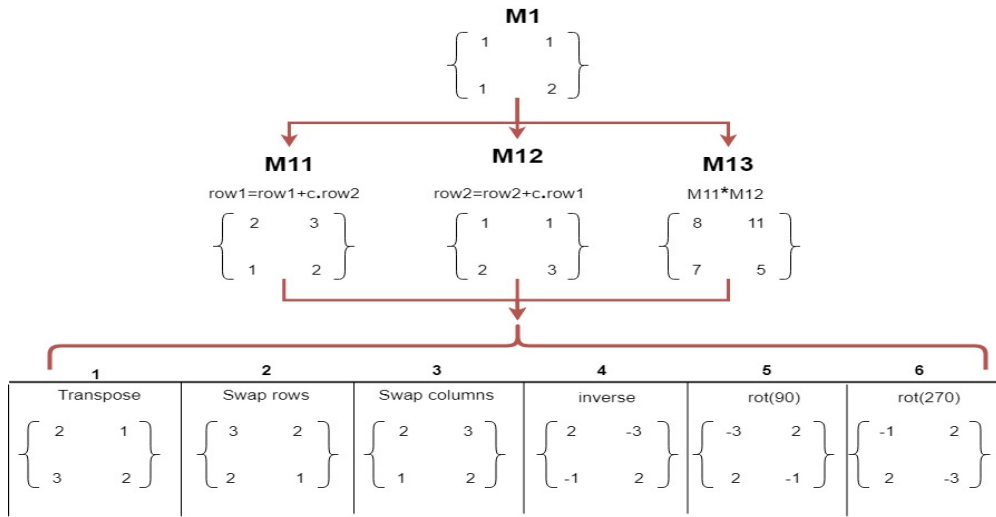


Figure 4.2 Matrix Operations Sequence Order

By changing the order of the six operations, there will be possibilities of 720 (6!) different sequence executions and the generated matrix  $M_g$  will be different for each sequence order. To get the exact  $M_g$  at the decryption side, the same value for *key* must be applied in the decryption phase.

We define a generalized ACM transformation as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \text{mod } N \quad (11)$$

where  $a$ ,  $b$ ,  $c$  and  $d$  are the coefficients of the transform matrix  $M_g$  which is already discussed in the previous section in detail. Algorithm-1 is employed for encryption using the generalized ACM. It is well known that ACM generates  $P - 1$  scrambled images each of which is a candidate for being an encrypted image. However, in our approach, encrypted image is selected based on the correlation score of each candidate image. Thus, it is ensured that the appearance of selected image shows no clue about the shuffled image, which is desired in any encryption method. This score is simply the average of the correlation of the pixels in adjacent rows and columns (a measure of the similarity of pixel values). More precisely, in our algorithm, we define a vector  $x$  containing all columns except for the last one, and a vector  $y$  containing all columns excluding the first column. Then, the correlation coefficient

of these vectors is computed. A similar coefficient is obtained for the rows. Finally, the average of these two coefficients is taken as the correlation score.

The scrambling process is performed as described in Algorithm-2.

**Algorithm-2:Encryption phase**

Input :image to be scrambled and the *operSeq*

**S1.** Generate the generalized matrix  $M_g$  by using the *operSeq* list

**S2:** Calculate the new positions of image pixels using Eq(4) iteratively.

**S3:**Repeat S2 until we reach the original image.

**S4:**Obtain the iteration number, which is called the period of encryption cycle,  $P$ , at which the original image is retrieved.

**S5:** Select an image which has the lowest correlation among its adjacent pixels in the loop S2 as an encrypted image.

**S6:**Obtain the iteration number of the selected encrypted image,  $S$ .

---

**Algorithm-3** describes the decryption process. First, the generalized transformation matrix needs to be regenerated using the algorithm stated in Table 4.1.

**Algorithm-3: Decryption phase**

Input : scrambled image, *operSeq* ,  $P$ , and  $S$

**S1.**  $M_g$  is constructed using the *operSeq* list

**S1:** Apply generalized ACM to the encrypted image starting at  $S$ .

**S2:**Repeat S1 ( $P-S$ ) times, at which the original image is retrieved.

---

At the decryption process, several parameters are required to be able to decrypt the encrypted image. These parameters namely are the *operSeq list*, period  $p$ , and the iteration number  $s$ . providing the wrong *operSeq* cause in generating the wrong transform matrix and this leads to an incorrect decryption of the original image. Images scrambled with a particular transform matrix cannot be restored using a different transform matrix.

Here is the process of applying the generated matrix  $M_g = \begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix}$  with a random *operSeq* list. The full period in which the original image will re-appear is 200, the best candidate scrambled form of the image is at iteration 175. Figure 4.2 illustrates

the encrypting phase while Figure 4.3 shows the decrypting phase with the correct parameters.



Figure 4.3 Encryption process a) original image; b) image at full period; c) Encrypted image at iteration 175.



Figure 4.4 Decryption process  
a) Encrypted image b) Recovered image.

It can be seen from the above figure that the encrypted image was decrypted in 25 iterations which is the remaining steps to reach again the original image ( iteration number =  $p - s$ ).

To test the robustness of our algorithm, we tried to hack the encrypted image using a transform matrix of which its elements are too close to that used for encryption. We used  $Mg = \begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix}$  instead of  $\begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix}$  and applied the decrypting algorithm for a full period, the original image could not be obtained at any of its iterations. The best looking images was at iterations 48 and 98 as can be seen in Figure 4.5.

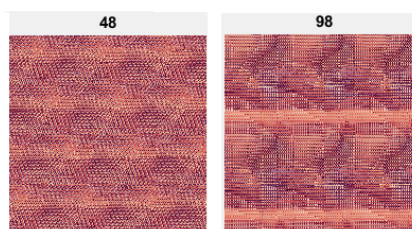


Figure 4.5 Decrypted image at iterations 48 and 98

## 4.2.2 Experimental Results for our Generalized ACM

To check the effectiveness of our proposed method, we have performed several tests on different color images with different sizes; the main focus was on square images of  $200 \times 200$  and  $400 \times 400$  pixels.

### 4.2.2.1 Adjacent Pixel Correlation Analysis

Correlation analysis measures the closeness of pixel values of image to its neighbor values. It is known that in digital images, any pixel is strongly correlated with its neighbors, in other words, for any ordinary image, each pixel is highly correlated with its adjacent pixels. Figure 4.6 shows the high correlation between adjacent pixels in the original image.

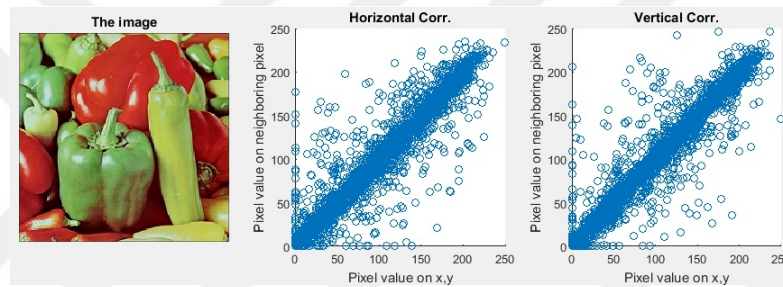


Figure 4.6 Correlation Analysis of original image

a) Peppers image b) Horizontal Correlation c) Vertical Correlation

In the encrypted image, we expect less correlation between adjacent pixels. The less correlated the encrypted image is, the more robust the encryption is.

Figure 4.7 shows some scrambled images at some iteration numbers  $s$ . Note that we will get the worst encrypted image at 100<sup>th</sup> iteration.

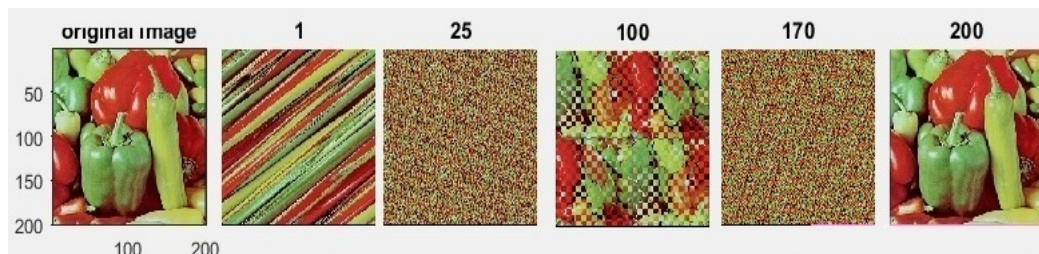


Figure 4.7 Peppers image at different iterations during scrambling process



In Table 4.2, it can be observed that neighboring pixels in the original image are highly correlated in both directions, while there is a little correlation between neighboring pixels in the scrambled image. Since there is no direct relationship between the quality of the scrambled image and the length of the iterations, a random selection of the iteration number may result in a highly correlated image. For example if we select the iteration number 100 to be the best, the correlation is higher than the lowest possible correlation. This is given in the last column of Table 4.2.

Table 4.2 Correlation values of original and encrypted image

Image	Original image		Encrypted image (at best iteration)		Image at iteration 100	
	Horizontal	Vertical	Horizontal	Vertical	Horizontal	Vertical
Lenna	0.9163	0.9520	0.3272	0.3151	0.4253	0.4250
Peng	0.9174	0.9571	0.0960	0.0977	0.2853	0.2855
Pepper	0.9384	0.9447	0.1472	0.1779	0.2863	0.2851

For different images, the best scrambled image is obtained at different iteration numbers, which are 175, 157 and 32 for the Lenna, Peppers and Peng images respectively.

To avoid selecting an unsuitable iteration number for encryption, we include the calculation of pixel correlation in each iteration and choose the iteration that gives us the least correlation among the adjacent pixels.

For the same image and a randomly given openSeq list, the best iteration number was 43 with correlation of 0.1469 and 0.1801 for horizontal and vertical neighbors respectively. Figure 4.8 illustrates the low correlation of adjacent pixels in the best encrypted image.

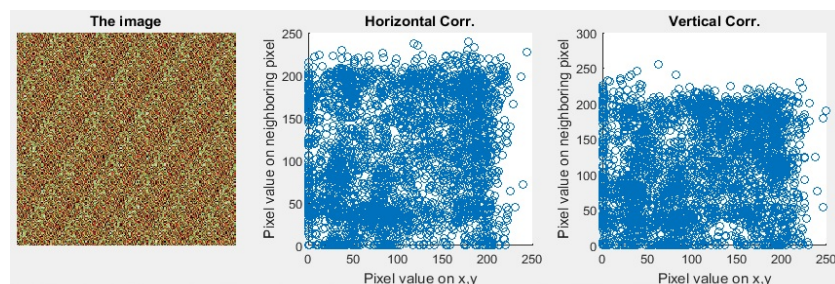


Figure 4.8 Correlation Analysis of encrypted Peppers image at iteration 157

a) Encrypted Peppers b) Vertical Correlation c) Horizontal Correlation

Another measurement is the Peak Signal to Noise Ratio PSNR. PSNR is often used as a quality measurement between two images to check the similarity between them. The higher the PSNR is, the more similarity.

In Table 4.3, low values of PSNR and high values of MSE reflect that there is a big difference between the original image and the encrypted image. In the same time the value of MSE was zero for the analysis of similarity between original and restored (decrypted) image that means the decrypted and original images are identical with no loss of data.

Table 4.3 Similarity analysis between original and encrypted images

Image	PSNR	MSE
Lenna	16.676	1489
Peppers	15.388	2095
Peng	11.289	4837

To show how the image is scrambled at different iteration numbers, a matrix  $Mg = \begin{bmatrix} 3 & 2 \\ 4 & 3 \end{bmatrix}$  was used which has a full period of 60 and a best scrambled image at iteration 52. The correlation scores for all iterations was plotted in Figure 4.9.

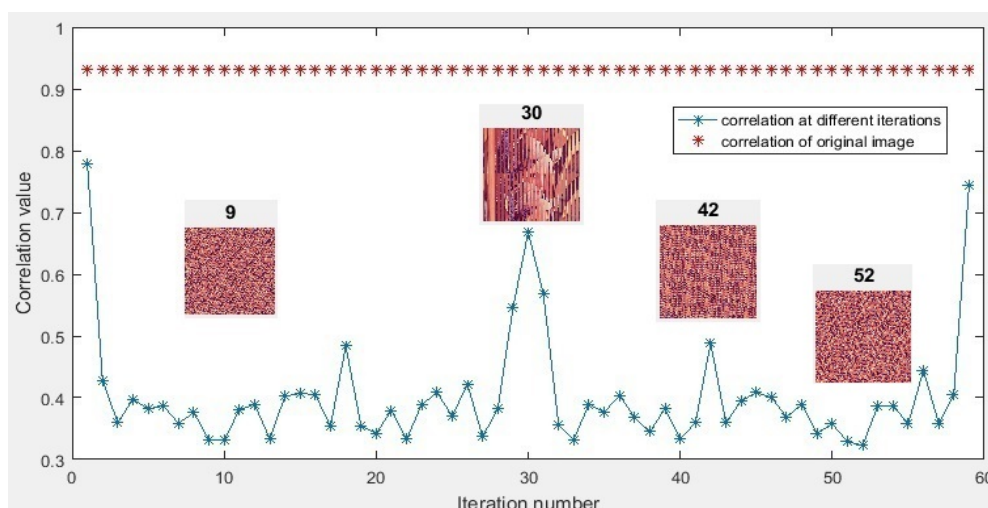


Figure 4.9 Correlation scores for different iterations

### 4.2.3 Robustness of the Proposed Method

To test the strength of the encryption for brute-force attacks, we try to recover the Lenna image from its scrambled version obtained in the previous section with 50 randomly created transformation matrices. Only the scrambled image is available and P, S and operSeq are not provided. The scrambled image was iterated for the full period for each test matrix to obtain several encrypted images with the hope that the obtained image will resemble the original image as close as possible. Therefore, we select the one of encrypted images with the highest correlation among its neighbor pixels. Figure 4.10 shows the maximum correlation value for each test matrix. None of these matrices leads to successful recovery of the original image. However, as seen, the highest correlation corresponds to the 26<sup>th</sup> matrix,  $M = \begin{bmatrix} 1 & 1 \\ 8 & 7 \end{bmatrix}$  which considered to be the best decryption of encrypted Lenna. Figure 4.11 illustrates the best possible recovered image using transformation matrix M. Although the obtained image seems to contain several patterns, one cannot decide content of the picture. This image has a correlation of 0.7218 and achieved at iteration 2.

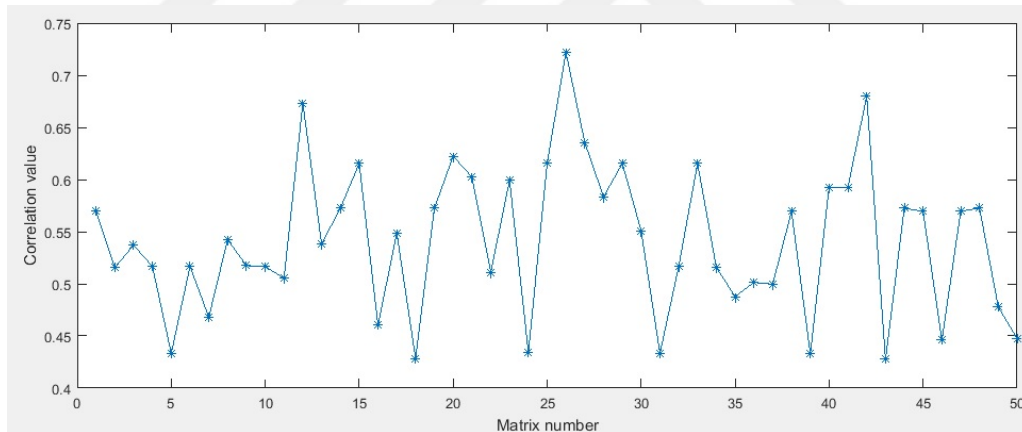


Figure 4.10 Best Correlation scores for different matrices

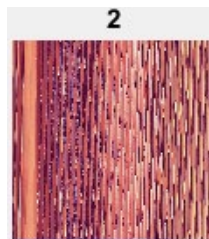


Figure 4.11 Best retrieved image

Table 4.4 shows a sample of 8 out of the 50 matrices used in the experiment.

Table 4.4. Correlation of neighborhood in matrix sample

Length of operSeq list	984	416	52	1020	244	1337	1222	710
Matrix#	1	5	12	24	26	37	45	50
Matrix	$\begin{pmatrix} 18 & 13 \\ 25 & 18 \end{pmatrix}$	$\begin{pmatrix} 1 & 10 \\ 2 & 19 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 4 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 8 \\ 4 & 11 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 8 & 7 \end{pmatrix}$	$\begin{pmatrix} 19 & 28 \\ 2 & 3 \end{pmatrix}$	$\begin{pmatrix} 18 & 13 \\ 25 & 18 \end{pmatrix}$	$\begin{pmatrix} 1 & 17 \\ 1 & 16 \end{pmatrix}$
Best Corr.	0.5695	0.4333	0.6732	0.4342	0.7218	0.4997	0.5695	0.4475

#### 4.2.4 Encrypting non-square images:

As Arnold cat map can be used only to encrypt square images, many researchers discussed the way to override this problem. Some researchers followed an approach of dividing the non-square image into several square divisions and after that they apply ACM for each of these divisions separately as in [69]. Tang et. al. also utilized the same idea but they divided the image into random overlapping square blocks [70]. Other researchers tends to reshape the or resize the image to NXN dimensions before scrambling it with ACM.

Both approaches have their disadvantages. The former consumes a lot of execution time for applying ACM several times one for each division at encryption and decryption processes. The later, causes the loss of image quality because of the interpolation process during the resize process.

We found a simple way of applying Arnold transform to encrypt rectangular images. In our approach, we tend to pad some rows or columns to the end of image matrix and encrypt the resulting image, at the other side we are decrypting the scrambled image then removing the added rows or columns to recover the original image.

This process is controlled by the following algorithm.

#### **Algorithm-4: Managing non-square images**

```
if rows = cols Set first pixel of R channel to Zero
if rows > columns
    n = rows - columns
    Add n rows
    Set first pixel of R channel to 1
    Store n in first pixel of G channel
if rows < cols
    n = columns - rows
    Add n columns
    Set first pixel of R channel to 2
    Store n in first pixel of G channel
```

Figures 4.12 and 4.13 show the encryption and decryption process for a non-square image.

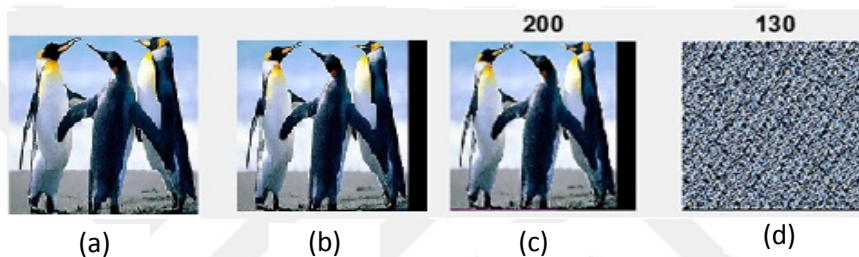


Figure 4.12 Encryption process of non-square image a) original image ;  
b) image with padded columns; c) image at full period; d) Encrypted image



Figure 4.13 Decryption process of non-square image a) Encrypted image ;  
b) Recovered image; c) Recovered image after reshaping;

#### **4.2.5 Advantages of the Proposed Method**

The proposed method has many advantages over the traditional Arnold cat map and its other variants. The advantages are summarized in the following points.

(1) In proposed method, all matrix coefficients are selected using several transformations controlled by an operSeq list. This feature is not found in traditional Arnold matrix which uses fixed set of matrix coefficients, and also in the improved variations in which there is at least one fixed coefficient.

(2) In proposed method all four matrix coefficients could be different and there are many alternatives to choose from. In the first variation of Arnold stated in (2), two coefficients are fixed to unity, and in the second variation stated in (3), the first coefficient ( $M(1,1)$ ) is fixed to unity and we have only 2 choice to select from while the third coefficient can be calculated by these two coefficients.

(3) In our approach, we calculate the correlation between each two adjacent pixels in the suggested encrypted image to decide at which iteration the process must stop to achieve the best encryption.

### 4.3 Steganography Part

In this part of the thesis we show the application of steganography in both spatial domain and transform domain and discuss the experimental results for each method.

#### 4.3.1 LSB Insertion Method

LSB Insertion is the most commonly used method in the spatial domain. The idea behind LSB substitution is that, while changing the values of least significant bits of a pixel does not affect the appearance of the image, so we can use these bits to store any data without losing the image quality. The MSB represents 50% of pixel value, and the LSB represents only about 0.39%.

Visually, the energy of each bit in the image pixel can be represented by the bit-plane splicing technique. In Figure 4.14, we show the eight bit planes of an 8-bit gray-scale. It is apparent that the two or three least significant bits do not encode much useful visual information of the image.

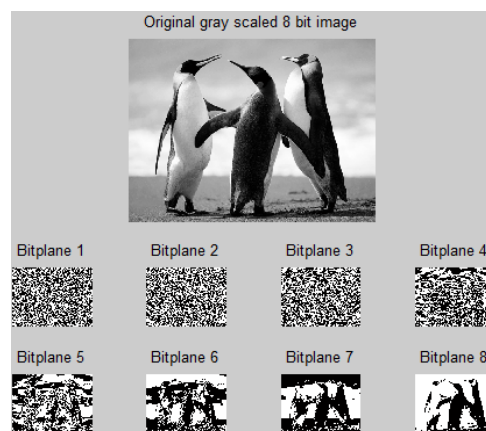


Figure 4.14 Bit-planes of a grey-scale image

• **1 LSB:**

Here we are embedding the MSB of the secret image into the LSB of the cover image (1 bit). Figure 4.15 shows the process.



Figure 4.15 Embedding one bit in LSB

Because of altering only one bit of the cover image, we are getting stego images of good quality, but in the other hand we will get a poor quality for the retrieved image.

• **4 LSB:**

Altering the four LSBs of the cover image dramatically decreases the quality of the stego-image, but producing a better quality for the extracted image as in figure 4.16.



Figure 4.16 Embedding four bits 4LSB

However, Experiments are done for 2 LSB and 3 LSB as well. Table 4.5 shows the results for all the four experiments.

Table 4.5 MSE and PSNR values for LSB method

Cover image	1 LSB		2 LSB		3 LSB		4 LSB	
	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
Peppers	0.1690	55.8527	0.9393	48.4098	3.1794	41.9319	17.6311	35.6827
Lenna	0.1665	55.9165	0.9145	48.5220	4.0186	42.0937	16.7701	35.8881
Bird	0.1671	55.9006	0.9140	48.5229	4.0027	42.1087	17.0152	35.8599
Sea	0.1657	55.9375	0.9081	48.5539	4.0285	42.0861	16.9391	35.8507

### 4.3.1.1 LSB insertion with encrypted secret image

To enhance the quality of the stego image, we encrypted the secret image before embedding it into the cover image. The encryption process was done easily by a simple procedure in which the secret image was XORed by a predefined key and that key was reused at the opposite process to extract the hidden image. The resulting stego-image was quite better. The values of PSNR and MSE also improved. Figure 4.17 shows the stego images resulted by the 4 LSB embedding with XOR encryption.

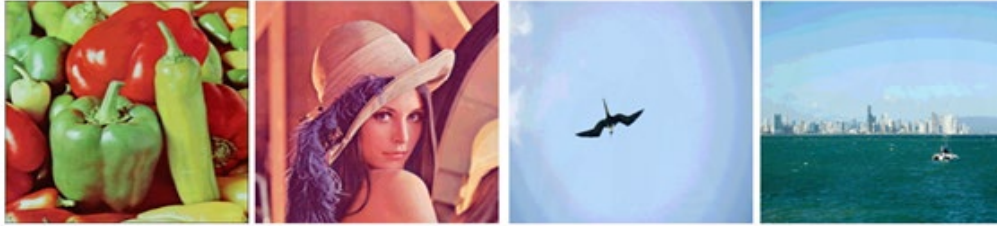


Figure 4.17 4LSB Embedding with XOR encryption

### 4.3.1.2 LSB -Lossless method

In this experiment, we altered only the 2 LSBs of the cover image. Actually, we are embedding all bit planes of the secret image inside the cover image that's why it is a lossless method. One To do so we used a cover image of size four times the secret image, the cover image was divided into four parts, then we take two bits (LSB) from secret image and put them in the two LSBs of part1; then we take the next two LSBs from the secret image and put them in the two LSBs of part 2; and so on. Figures 4.18 and 4.19 show the idea of our algorithm.



Figure 4.18 Divided Cover Image

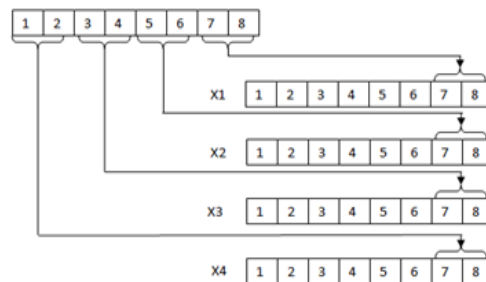


Figure 4.19 Lossless Embedding Procedure





Figure 4.20 Lossless embedding

The results of this implementation are shown in Table 4.6.

Table 4.6 PSNR values of stego images in lossless embedding

Cover image	MSE	PSNR
Peppers	0.8401	48.8877
Lenna	0.8328	48.9256
Bird	0.8348	48.9154
Sea	0.8379	48.8989

### 4.3.2 Embedding in Wavelet Transformation

As discussed in Chapter 2, there are many types of transform techniques. In this thesis, we concentrate on Wavelet Transformation because of its advantages over Cosine and Fourier Transformations.

While in DCT the image is divided into non-overlapping blocks, in Wavelet transform there is no need for this. Wavelet transformation has the advantage of higher compression ratio over DCT, and it enables the transformation of the whole image and introducing inherent scaling.

There are many types of common wavelet functions. Haar, Daubechies, Symlets are some of them. In our proposed work, we used the haar transformation because of its simplicity, computaion speed, memory efficiency and its exact reversibility. Our experiments showed the power of this transform over others regarding to the obtained results of our method.

The 2-D Haar transform is computed by iterating difference and averaging in the horizontal direction and then in the vertical direction.

### 4.3.2.1 DWT

Our approach is to apply DWT to both cover and secret images then embedding the LL band of secret image into different sub-bands of cover image.

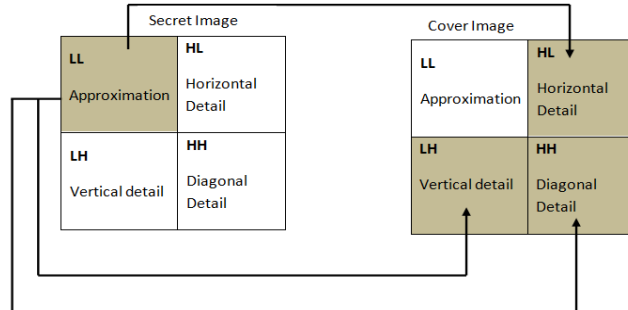


Figure 4.21 Embedding process using DWT

The secret data is scaled by a factor  $\alpha$  smaller than unit, and then is embedded within one of the above mentioned partitions. The reason for scaling is to decrease distortion of the stego image.

Figure 4.22 shows the resulting stego images using different values for  $\alpha$ .

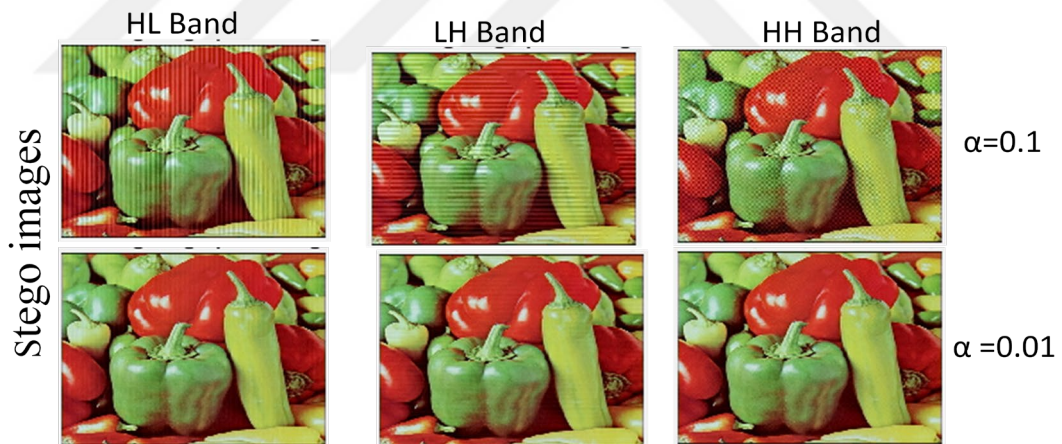


Figure 4.22 Stego images of DWT Embedding

Embedding secret image in the HH band gives the best quality for the stego image.

The PSNR and MSE values for comparing the cover and the stego image are shown in Table 4.7. Although the PSNR value for the stego image with  $\alpha = 0.001$  is the highest, the quality of the extracted image was very low and it will be ignored. In the following experimental results.

Table 4.7 PSNR values of stego images at different scaling factor  $\alpha$

Image	$\alpha = 0.5$		$\alpha = 0.1$		$\alpha = 0.01$		$\alpha = 0.001$	
	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE
Peppers	16.1192	1593	28.9235	83.9749	44.3209	2.4061	46.1829	1.5663

To extract the secret image from the stego image, a DWT transformation is applied and the image data is extracted from the HH band. Extraction was done in two different ways, the first is by applying the inverse DWT, and the second method is done by extracting the bits directly without applying inverse DWT. The extracted images are shown in the Figure 4.23.

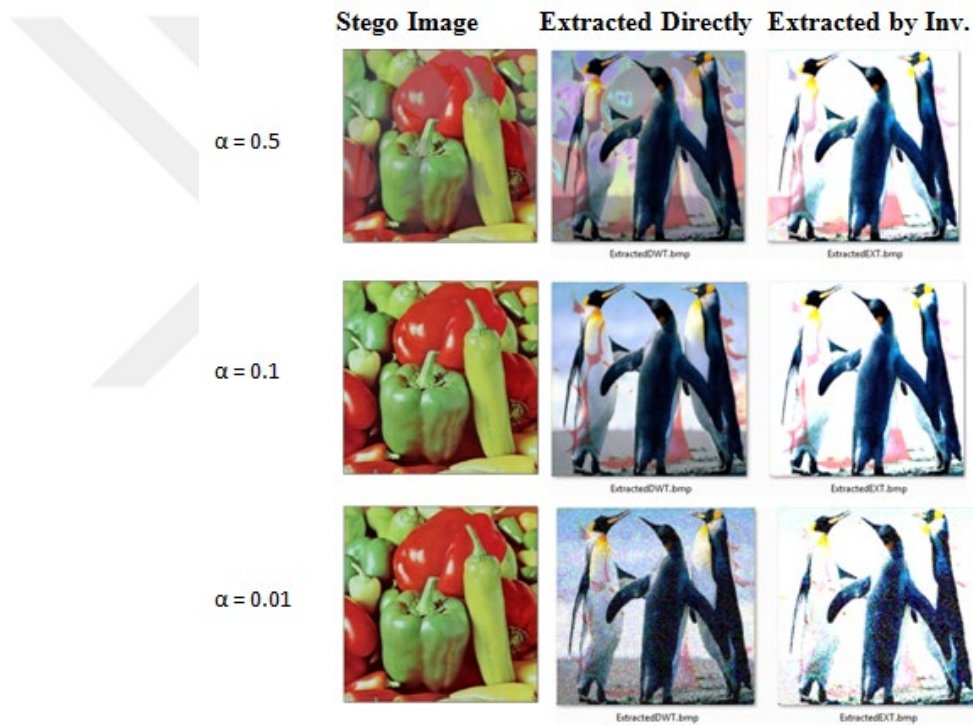


Figure 4.23 Stego and extracted images at different values of  $\alpha$

The PSNR and MSE values for comparing the Secret and the Extracted image are shown in Table 4.8.

Table 4.8 PSNR values for extracted images at different value of  $\alpha$

Image	$\alpha = 0.5$		$\alpha = 0.1$		$\alpha = 0.01$		$\alpha = 0.001$	
	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE
Inv. DWT	21.5140	55.39	26.6177	173.4607	22.6640	355.4674	9.3939	7476
Direct+resize	17.4109	1213	16.8547	1361	16.1811	1575	9.8410	6751

### 4.3.2.2 Stationary Wavelet Transform SWT

One of the main drawbacks of DWT method is that, it does not provide shift invariance because of the down-sampling of its bands. Down sampling a signal is achieved by reducing the sampling rate, i.e. removing some of the samples of the signal. This causes a major change in the wavelet coefficients of the image even for minor shifts in the input image. The shift variance of DWT causes inaccurate extraction of the transformed image.

Stationary Wavelet Transform (SWT) ) is similar to the DWT, the only difference is that, the signal is never sub-sampled and instead, the filters are up sampled at each level of decomposition[75]. We can say that SWT is essentially a DWT without down sampling. Figure 4.24 illustrates the first level decomposition of the SWT.

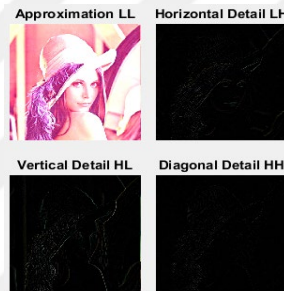


Figure 4.24 SWT First Level Decomposition

Figure 4.25 illustrates stego and extracted images generated by applying the SWT.

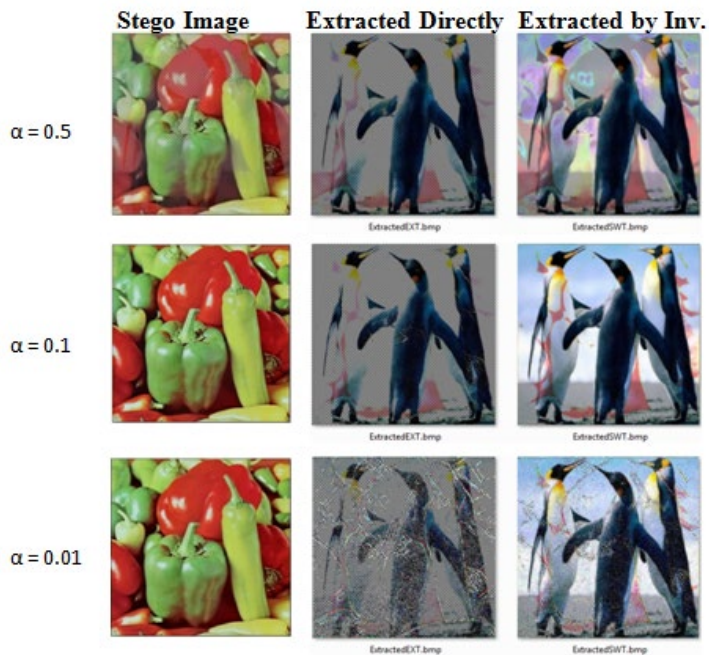


Figure 4.25 Stego and extracted images using SWT

Tables 4.9 and 4.10 show the PSNR and MSE values for both stego and extracted images.

Table 4.9 PSNR values for stego images at different values of  $\alpha$

Image	$\alpha = 0.5$		$\alpha = 0.1$		$\alpha = 0.01$	
	PSNR	MSE	PSNR	MSE	PSNR	MSE
Peppers	16.1332	1588	28.9766	82.9770	45.8713	1.6858

Table 4.10 PSNR values for Extracted images at different values of  $\alpha$

Image	$\alpha = 0.5$		$\alpha = 0.1$		$\alpha = 0.01$	
	PSNR	MSE	PSNR	MSE	PSNR	MSE
Inv. SWT	21.531	557.07	26.5222	176.290	18.9121	836.085
Direct	11.2017	4958	11.1120	5062	10.4460	5882

#### 4.3.2.3 Lifted Wavelet Transform LWT

Common wavelet transforms like DWT and SWT often have floating point coefficients. If the input consists of integers (unfortunately as in the case of images), the resulting output will not consist of integers. Thus the perfect reconstruction of the original image becomes difficult because the inverse wavelet transform becomes lossy.

With the introduction of Wavelet transforms that map Integers to Integers using Lifted scheme LWT, the output can be completely characterized by integers and thus exact decompression of the original data is achieved [71].

The LL sub-band in the case of LWT appears to be a close copy with smaller scale of the original image while in the case of DWT and SWT the resulting LL sub-band is distorted slightly. Figures 31 shows the decomposition of DWT, SWT and LWT.

Lifting scheme have the following advantages over conventional wavelet transform.

- It allows a faster implementation of the wavelet transform.
- It allows implementing reversible integer wavelet transforms. Floating point operations introduce rounding errors due to floating point arithmetic.

- Perfect reconstruction of images is possible for loss-less compression and steganography.
- Requires less memory.

Figure 4.26 illustrates the view differences in the LL-Sub-bands of the three transforms.



Figure 4.26 Comparison between DWT, SWT and LWT Decompositions

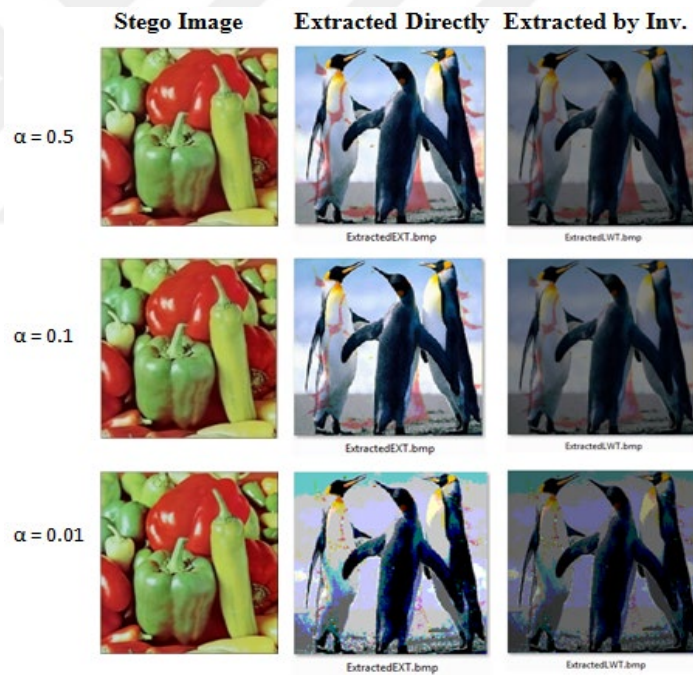


Figure 4.27 Stego and Extracted Images using LWT

Table 4.11 PSNR and MSE values for stego images with different values of  $\alpha$

Image	$\alpha = 0.5$		$\alpha = 0.1$		$\alpha = 0.01$	
	PSNR	MSE	PSNR	MSE	PSNR	MSE
Peppers	26.8522	135.219	39.8380	6.7916	47.5236	1.1510

Table 4.12 PSNR and MSE values for Extracted images with different values of  $\alpha$

Image	$\alpha = 0.5$		$\alpha = 0.1$		$\alpha = 0.01$	
	PSNR	MSE	PSNR	MSE	PSNR	MSE
Inv. LWT	14.3845	2417	14.5438	2321	14.3673	2412
Direct+resize	26.2211	185.389	27.5260	124.998	22.7135	358.55

#### 4.3.2.4 Combination of DWT, SWT and LWT

Recalling that among the three transformations the LL sub-band of LWT gives the best quality for the image, we experienced the combination of LWT/DWT and LWT/SWT such that the secret image is transformed by LWT and the cover image is transformed by DWT and SWT in sequence.

The stego images obtained by the two combinations are shown in Figure 4.28.







Method	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.01$
LWT in DWT			
LWT in SWT			

Figure 4.28 Stego images in different wavelet combinations

The extracted images (using inverse wavelet transform) by the two combinations are shown in Figure 4.29.




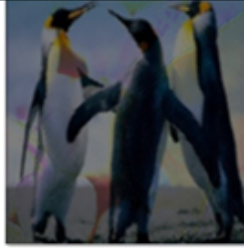

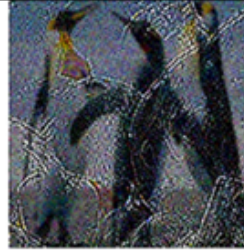
Method	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.01$
LWT in DWT			
LWT in SWT			

Figure 4.29 Extracted images by inverse in different wavelet combinations

The extracted images (directly) by the two combinations are shown in Figure 4.30.





Method	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.01$
LWT in DWT			
LWT in SWT			

Figure 4.30 Directly Extracted images in different wavelet combinations

The experimental results for all wavelet methods and their combinations are summarized in Table 4.13.



Table 4.13 PSNR values for stego and extracted images (peppers)

Method		Stego Image						Extracted Image					
		$\alpha = 0.1$			$\alpha = 0.01$			$\alpha = 0.1$			$\alpha = 0.01$		
Cvr	Scrt	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM
LWT	LWT	39.838	6.791	0.9916	47.523	1.151	0.9987	27.526	124.99	0.8387	22.713	358.55	0.4684
DWT	LWT	34.602	22.703	0.9734	45.640	1.774	0.9982	26.700	156.67	0.8004	19.588	715.58	0.2960
SWT	LWT	40.714	5.571	0.9932	53.073	0.3205	0.9997	10.661	5648	0.0816	10.046	6466	0.0187

Although the PSNR of SWT/LWT has the best value for the stego image, the extracted image of this combination has the poorest quality.

### 4.3.3 LWT with LSB Technique

In this experiment, we used LWT and LSB together. The MSBs of the secret image is embedded in the HH band of the transformed cover image. The stego images generated by applying an approach of embedding 4MSBs are shown in Figure 4.31.

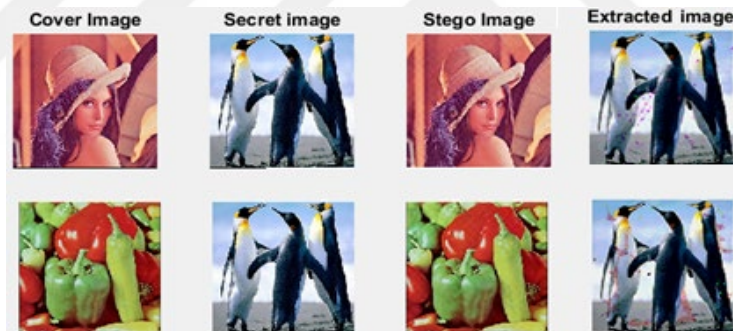


Figure 4.31 Stego images of LWT-LSB method (4 bits)

Table 4.14 PSNR values of stego images in LWT-LSB method

Image	PSNR	MSE
Lenna	38.6763	8.8452
Peppers	39.8557	6.7238

Table 4.15 PSNR values of Extracted images in LWT-LSB method

Image	PSNR	MSE
Lenna	24.9823	207
Peppers	23.9243	270

Again, here are the stego images generated by embedding 5 LSB in this approach.



Figure 4.32 Stego images of LWT-LSB method (5 bits)

As expected, embedding 4 bits results in better stego images, but the extracted image is better when embedding 5 bits as can be seen in Tables 4.16 and 4.17.

Table 4.16 PSNR of stego image using LWT-LSB method (5 bits)

Image	PSNR	MSE
Lenna	35.8962	16.7977
Peppers	36.6892	13.9703

Table 4.17 PSNR of Extracted image using LWT-LSB method (5 bits)

Image	PSNR	MSE
Lenna	25.4984	183.52
Peppers	24.2312	252.38

By applying a scaling factor  $\alpha=0.5$ , The quality of the stego image is increased as seen in Table 4.18 which reflects the results of 4 bit embedding.

Table 4.18 PSNR of stego images with scaling  $\alpha=0.5$

Image	PSNR	MSE
Lenna	39.7902	6.8439
Peppers	41.2362	4.8924

#### 4.3.4 Proposed Method: Fraction Based Embedding (FBE)

Our new technique is based on the following idea, instead of replacing pixels of the cover image with pixels of the secret image, why we do not try to add the pixel values of secret image to their corresponding pixels of the cover image?

Stego Pixel = Cover Pixel + Secret Pixel.

Two obstacles faced the addition operation of two pixel values are:

- 1- The summation of the two pixels may exceed 255 which is not recognized in image processing.
- 2- At the extracting phase it is impossible to retrieve the two values by knowing only their sum, for example;  $5+7=12$ , having the value 12 is not enough to retrieve 5 and 7. They may be 6 and 6 or 10 and 2; there are many possibilities.

Utilizing the nature of wavelet transformation which divides the image into 4 sub-bands with coefficient having real numbers, we found a solution to the above obstacles by reducing the pixel values of the secret image to very small values (fraction of one) and adding this fraction to the values located in the HH band of the transformed cover image.

Suppose the secret coefficient value to be embedded is 128, the process of our technique can be explained by the following steps.

- 1- Dividing the coefficient value by 10000 will result in a fraction value of 0.0128 in this case.
- 2- Suppose the HH sub-band coefficient value is 71.5000, adding both values will result in replacing the three right most digits (zeros) with 128, the new value of the coefficient will be 71.5128.
- 3- Repeat steps 1 and 2 for all coefficients, when finished apply inverse wavelet transformation of the four sub-bands to construct the stego image.
- 4- At the extraction phase, apply wavelet transformation to the stego image, take the rightmost 3 digits of coefficients and convert them back to integers to reconstruct the hidden image.

Since LWT contains only integers, so LWT could not be applied to the cover image. Only DWT and SWT can be used for the cover image. However LWT is applied to the secret image because it generates the best approximation sub-band and thus gives the best result for the Extracted image as discussed in Section 4.3.2.3.

SWT did not give good results because when we modify the coefficient values to have 4 digits in the fraction part, our 3 digits will be lost because when we apply the SWT at the extracting phase, all coefficients will have values rounded to only one digit in the fraction part. That is, if we save the value 71.5128, we will get only 71.5000 when we apply SWT on the stego image and thus our secret value (128) will be lost. Figure 4.33 shows a real 4x4 blocks of the coefficients during embedding and extracting phase. Note that the secret image is transformed by LWT (S-LL), while the cover image is transformed by SWT (C-HH).

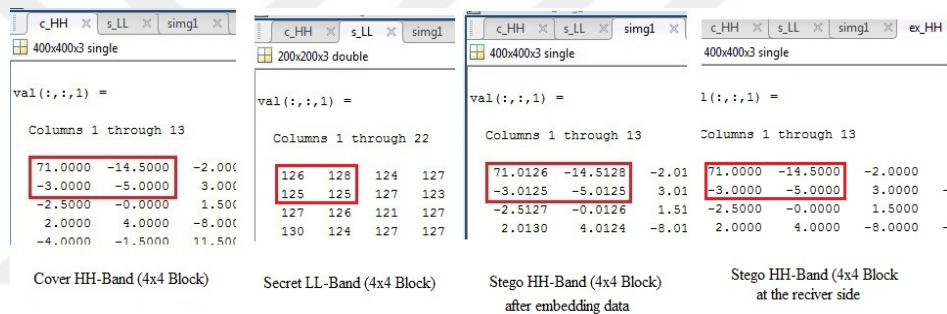


Figure 4.33 SWT and LWT coefficient blocks during embedding

Applying our technique with SWT transformation for the cover image results a distorted stego image and a fail to extract the secret image, see Figure 4.34.

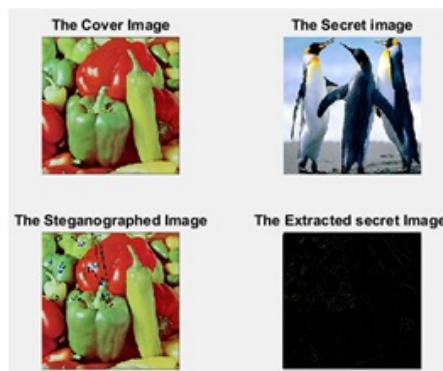


Figure 4.34 FBE in SWT Transform

Table 4.19 shows the results of embedding the peng image into four different cover images. As can be seen from the table, the similarity between the secret image and the extracted image is very low.

Table 4.19 Results for stego and extracted images in FBE method in SWT

Cover Image	Stego Image			Extracted Image		
	PSNR	MSE	SSIM	PSNR	MSE	SSIM
Peppers	39.3604	9.6714	0.9948	8.8558	8563	0.0169
Lenna	40.2580	11.4139	0.9950	8.8628	8550	0.0200
Barbara	40.8017	5.9728	0.9951	8.9630	8358	0.0209
Sailboat	42.6043	4.2054	0.9958	8.9109	8455	0.0269

However, DWT worked well and gave excellent results for both stego and extracted images. Figure 4.35 presents the experimental results of the method.



Figure 4.35 Stego and Extracted images using FBE

Block diagrams shown in Figures 4.36 and 4.37 illustrate the embedding and extracting process in the proposed FBE.

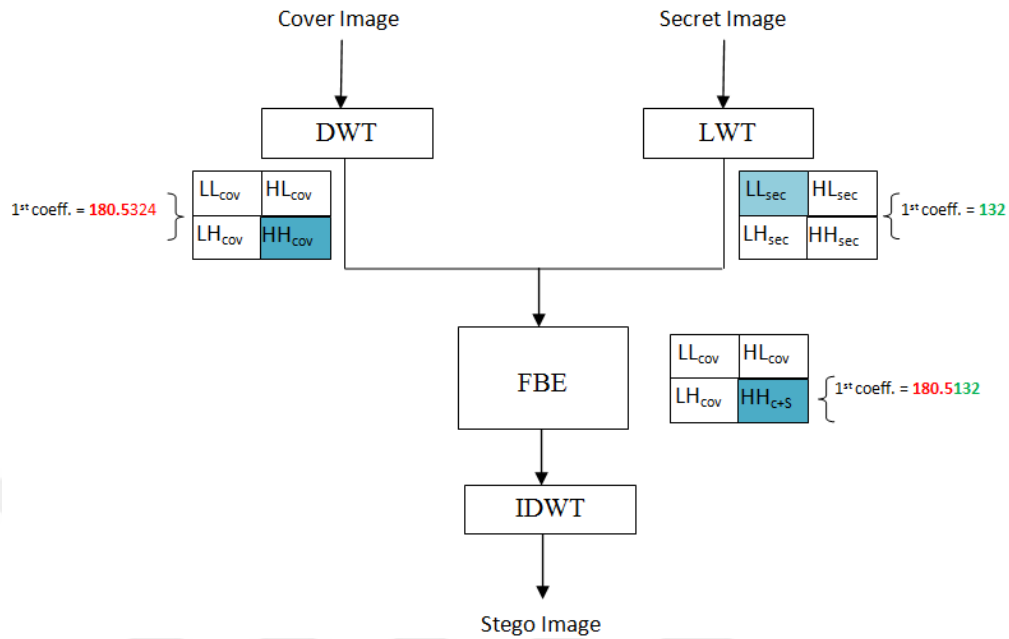


Figure 4.36 Block Diagram of FBE - Embedding process

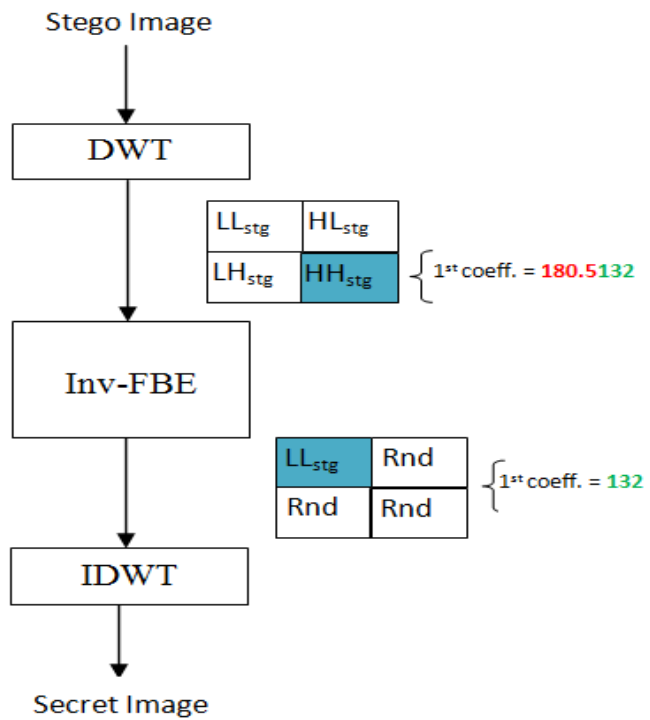


Figure 4.37 Block Diagram of FBE - Extracting process

#### 4.3.4.1 Experimental Results

The effectiveness of our proposed method is evaluated by several metrics. First, the metrics PSNR and MSE are used. These metrics and all other metrics based on MSE do not take the structural fidelity of the processed image into account. Because of this, we used the SSIM which is based on visible structures in the image by assessing the visual impact of three image characteristics which are structure, luminance and contrast.

SSIM is an improved version of the earlier universal image quality index and it gives a better indication of image quality [87].

The resulting values of SSIM are ranged between -1 and 1, values reaching 1 are the best because higher values mean that the cover image and the stego image are very close to each other.

The results of our method is excellent in both stego images and extracted images regarding to obtained values for PSNR and MSE . Table 4.20 is obtained by applying BMP images for both cover and secret images.

Note that the scores are obtained after reducing the size of the secret image into that of the extracted image.

Table 4.20 PSNR and SSIM values for stego and extracted images in FBE method

Cover Image	Secret Image	Stego Image			Extracted Image		
		PSNR	MSE	SSIM	PSNR	MSE	SSIM
Peppers	Peng	<b>67.1135</b>	0.0128	1.000	32.1924	52.2613	0.9352
Lenna	Peng	<b>66.8900</b>	0.0134	1.000	31.8542	61.1134	0.9258
Barbara	Peng	<b>68.3084</b>	0.0095	1.000	31.9701	54.7543	0.9355
Sailboat	Peng	<b>69.9147</b>	0.0068	1.000	31.2274	65.0899	0.9278
Peppers	Lenna	<b>67.1441</b>	0.0127	1.000	39.6509	37.1094	0.9942
Lenna	Peppers	<b>66.8759</b>	0.0135	1.000	31.7312	60.8708	0.9510

To investigate the effect of PSNR/MSE and SSIM, we made many experiments with different combinations of cover and secret images to see the difference between the traditional metrics PSNR/MSE and the SSIM Index.

Table 4.21 illustrates the PSNR, MSE and SSIM values for the Stego images in all of these experiments. We can see that the SSIM values are all ones except for the *Peng* cover image with a value of 0.9999. That means, the generated stego images are very close to the cover images which indicates the power of our method in generating high quality stego images. Note that the best PSNR values are those of *Lenna* and *Barbara* secret images in all other cover images.

Table 4.21 Resulting values for stego images

Cover Image	Secret Image	PSNR	MSE	SSIM
Peppers	Lenna	67.1441	0.0127	1.0
	Barbara	67.1441	0.0127	1.0
	Sailboat	67.1436	0.0127	1.0
	Peng	67.1135	0.0128	1.0
Lenna	Peppers	66.8759	0.0135	1.0
	Barbara	66.9438	0.0132	1.0
	Sailboat	66.9433	0.0132	1.0
	Peng	66.8900	0.0134	1.0
Barbara	Peppers	68.2680	0.0096	1.0
	Lenna	68.3771	0.0093	1.0
	Sailboat	68.3754	0.0093	1.0
	Peng	68.3084	0.0095	1.0
Sailboat	Peppers	69.8230	0.0069	1.0
	Lenna	70.0045	0.0066	1.0
	Barbara	70.0045	0.0066	1.0
	Peng	69.9147	0.0068	1.0
Peng	Peppers	68.7656	0.0087	0.9999
	Lenna	68.8594	0.0085	0.9999
	Barbara	68.8594	0.0085	0.9999
	Sailboat	68.8586	0.0085	0.9999

The same tests were conducted on the extracted images which were more distorted than the stego images. We included the extracted images in the tables to have a better view of the statistical results.



Tables 4.22 to 4.26 illustrate the PSNR, MSE and SSIM values for the extracted images in all of these experiments.

Table 4.22 Resulting values for extracted images from *Peppers* Image

Cover Image	Secret Image	PSNR	MSE	SSIM	Extracted Image
Peppers	Lenna	39.6509	37.1094	0.9942	
	Barbara	36.1726	15.7318	0.9790	
	Sailboat	37.2077	13.8917	0.9876	
	Peng	32.1924	52.2613	0.9352	

Table 4.23 Resulting values for extracted images from *Lenna* Image





Cover Image	Secret Image	PSNR	MSE	SSIM	Extracted Image
Lenna	Peppers	31.7312	60.8708	0.9510	
	Barbara	36.1726	15.7318	0.9790	
	Sailboat	37.0819	14.4336	0.9874	
	Peng	31.8542	61.1134	0.9258	

Table 4.24 Resulting values for extracted images from *Barbara* Image





Cover Image	Secret Image	PSNR	MSE	SSIM	Extracted Image
Barbara	Peppers	31.2605	91.2605	0.9478	
	Lenna	39.6509	7.1094	0.9942	
	Sailboat	37.1308	14.2530	0.9875	
	Peng	31.9701	54.7943	0.9355	

Table 4.25 Resulting values for extracted images from *Sailboat* Image






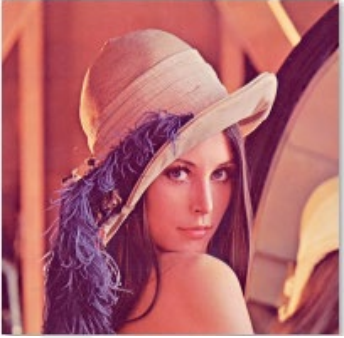


Cover Image	Secret Image	PSNR	MSE	SSIM	Extracted Image
Sailboat	Peppers	31.7459	80.2523	0.9527	
	Lenna	39.6509	7.1094	0.9942	
	Barbara	36.1726	15.7318	0.9790	
	Peng	31.9278	65.0899	0.9278	

Table 4.26 Resulting values for extracted images from *Peng* Image

Cover Image	Secret Image	PSNR	MSE	SSIM	Extracted Image
Peng	Peppers	31.8863	77.1802	0.9561	
	Lenna	39.6509	7.1094	0.9942	
	Barbara	36.1726	15.7318	0.9790	
	Sailboat	37.199	13.8917	0.9876	

As can be seen, the PSNR, MSE and SSIM values are fixed for *Lenna* and *Barbara* extracted images from all other cover images. Also, we can notice that the best PSNR and SSIM values are obtained with the *Lenna* extracted image, while the

worst values are of the *peng* extracted image. Figure 4.38 represents a graphical representation for SSIM values for all extracted images from all cover images.

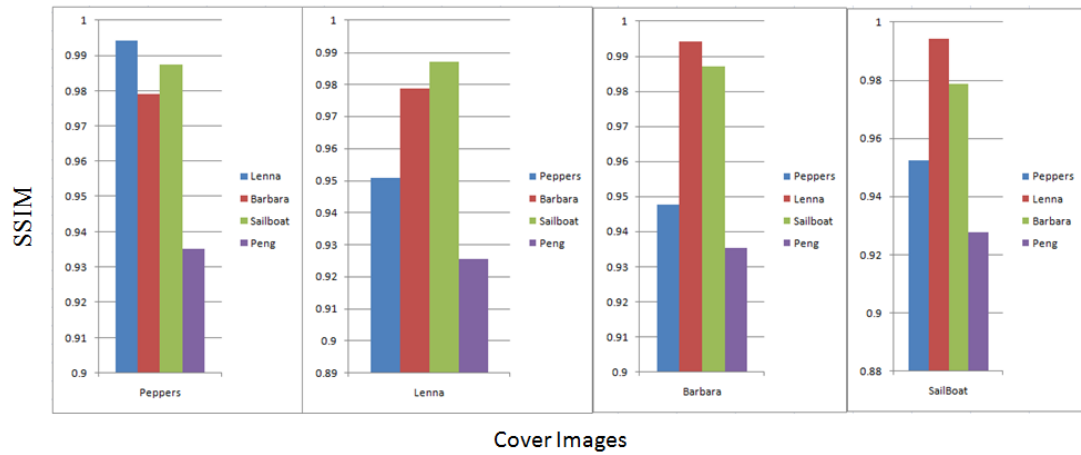


Figure 4.38 A graphical representation for SSIM values

Our approach works well regardless the type or the format of the used images. It works for BMP, JPEG color images and also works for gray scale images.

Figure 4.39 shows the stego and extracted images of the JPEG format without encryption, while Table 4.27 shows the results of our method by applying JPEG images.



Figure 4.39 Stego and Extracted Images using JPEG format

Table 4.27 PSNR values for stego and extracted using JPEG format

Cover Image	Secret Image	Stego Image		Extracted Image	
		PSNR	MSE	PSNR	MSE
Peppers	Peng	52.1850	0.4371	22.0653	465.425
Lenna	Peng	54.8787	0.2318	19.5058	768.768

By applying gray scale images, we get the following stego and extracted images as shown in figure 4.40.



Figure 4.40 Stego and extracted images in the gray scale level

Table 4.28 PSNR values of FBE method in the gray scale level

Cover Image	Secret Image	Stego Image		Extracted Image	
		PSNR	MSE	PSNR	MSE
Peppers	Peng	<b>61.2293</b>	0.0438	30.6183	56.3969
Lenna	Peng	<b>61.5329</b>	0.0436	29.7179	69.3891
Barbara	Peng	<b>61.9430</b>	0.0362	30.2581	61.2737
Sailboat	Peng	<b>64.3492</b>	0.0221	28.9718	82.3941

#### 4.3.4.2 Robustness against JPEG Compression

As stated earlier, our proposed method works well for many image formats. In Table 26, we presented the results of using JPEG images without compression. In this section we present the experimental results of using JPEG images with different levels of compression. We tested our method with 0, 25, 50, and 75% compression rates. By using JPEG images without compression (0% compression rate), we get good quality for the extracted secret images.

As can be seen in Table 4.27, the values of PSNR and MSE for the extracted images are 22.0653 and 465.425 respectively when saving the stego image without

compression. Figure 4.41 shows a comparison between the extracted secret images that were compressed with different compression levels with those without compression (compression ratio of 0%).



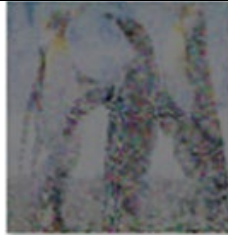
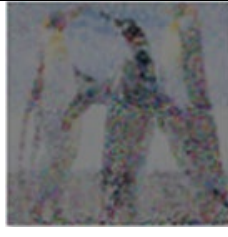

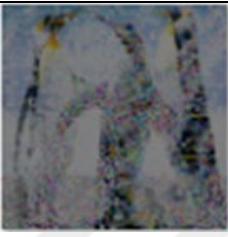










Cmp rate	Peppers	Lenna	Barbara	SailBoat
75 %				
50 %				
25 %				
0%				

Figure 4.41 Extracted images with different compression ratios

As can be seen, the extracted images are distorted compared to those images extracted from non-compressed stego images, and the PSNR values became worse. Note that we are embedding only the LL-band of the secret image, so we are actually retrieving the approximation of the secret image not the whole image. Table 4.29 shows the PSNR and MSE values of the extracted images.



Table 4.29 PSNR values for extracted images with Compression

Cover Image	Compr. rate of 75 %		Compr. rate of 50 %		Compr. rate of 25 %		Compr. rate of 0%	
	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE
Peppers	11.0944	5057	11.3813	4735	11.8717	4232	33.5269	29.3692
Lenna	11.0878	5066	11.4804	4629	11.9681	4139	33.3874	30.2550
Barbara	11.0329	5130	11.1914	4947	11.4766	4634	33.5132	29.4565
SailBoat	11.0915	5061	11.3042	4820	11.6024	4502	33.1405	32.1394

#### 4.3.4.3 Robustness against Gaussian noise

In this section we present the experimental results of adding Gaussian white noise to the stego images and its effect to the quality of the retrieved images. we applied different values for the mean and variance parameters for different degrees of noise.

Figure 4.42 illustrates the extracted images after applying Gaussian noise with the default value for mean ( $m=0$ ) with different values of the variance  $v$ .

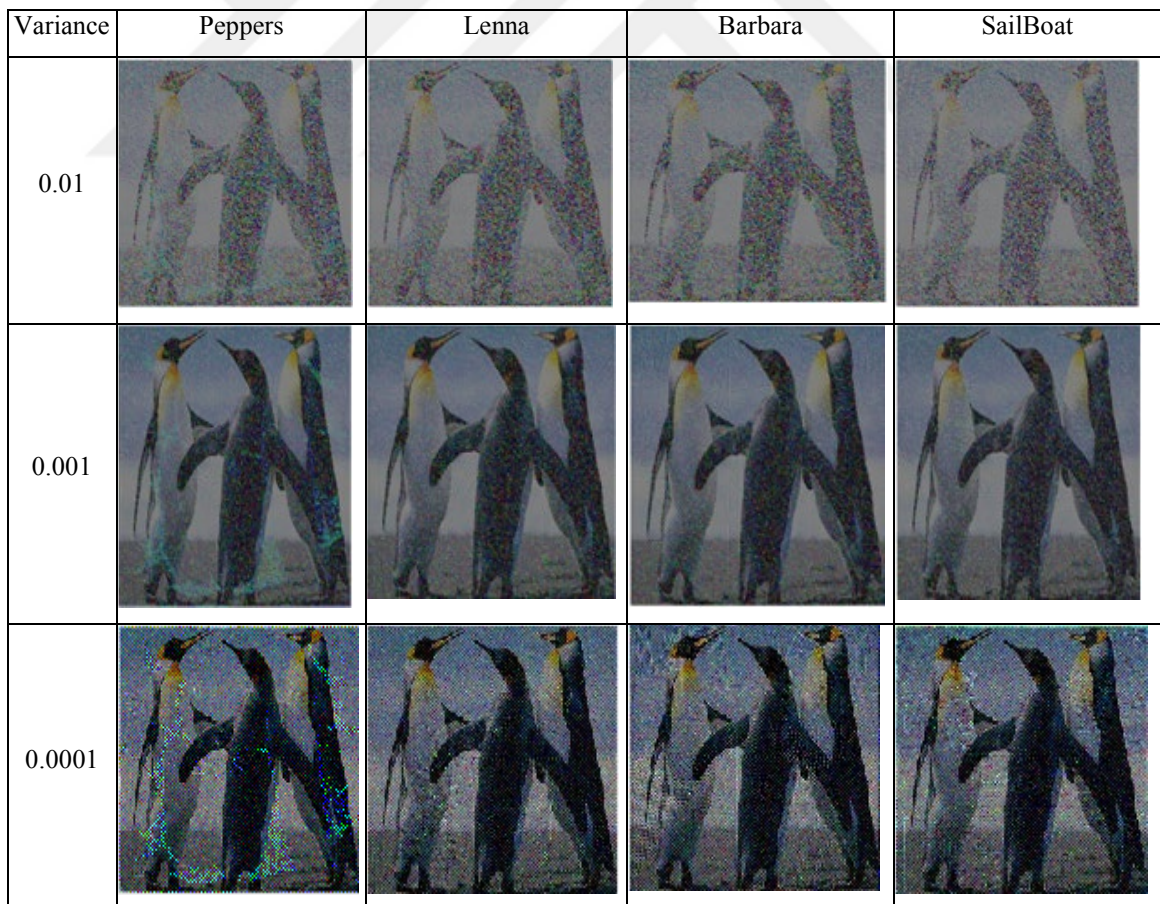


Figure 4.42 Extracted images with gaussian noise ( $m=0$ )

As can be seen, applying Gaussian noise to the stego image leads to extracting a noisy secret images. However, using lower values for the variance helps in extracting better secret images. Table 4.30 shows the PSNR and MSE values for the extracted images by applying different values for the Variance and mean equals to 0.

Table 4.30 PSNR values for the Extracted Images with Gaussian noise (m=0)

Cover Image	Variance= 0.01			Variance = 0.001			Variance = 0.0001		
	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM
Peppers	10.4926	5815	0.0240	12.4294	3745	0.0801	13.9883	2637	0.2494
Lenna	10.5563	5728	0.0241	12.5860	3604	0.0830	14.2931	2442	0.2752
Barbara	10.5413	5748	0.0235	12.5553	3627	0.0819	14.2340	2475	0.2611
SailBoat	10.5403	5750	0.0238	12.5575	3628	0.0825	14.2399	2473	0.2642

The same process was done using different value for the mean ( m= 0.5) and we got different results as can be seen in figure 4.43 and Table 4.31.

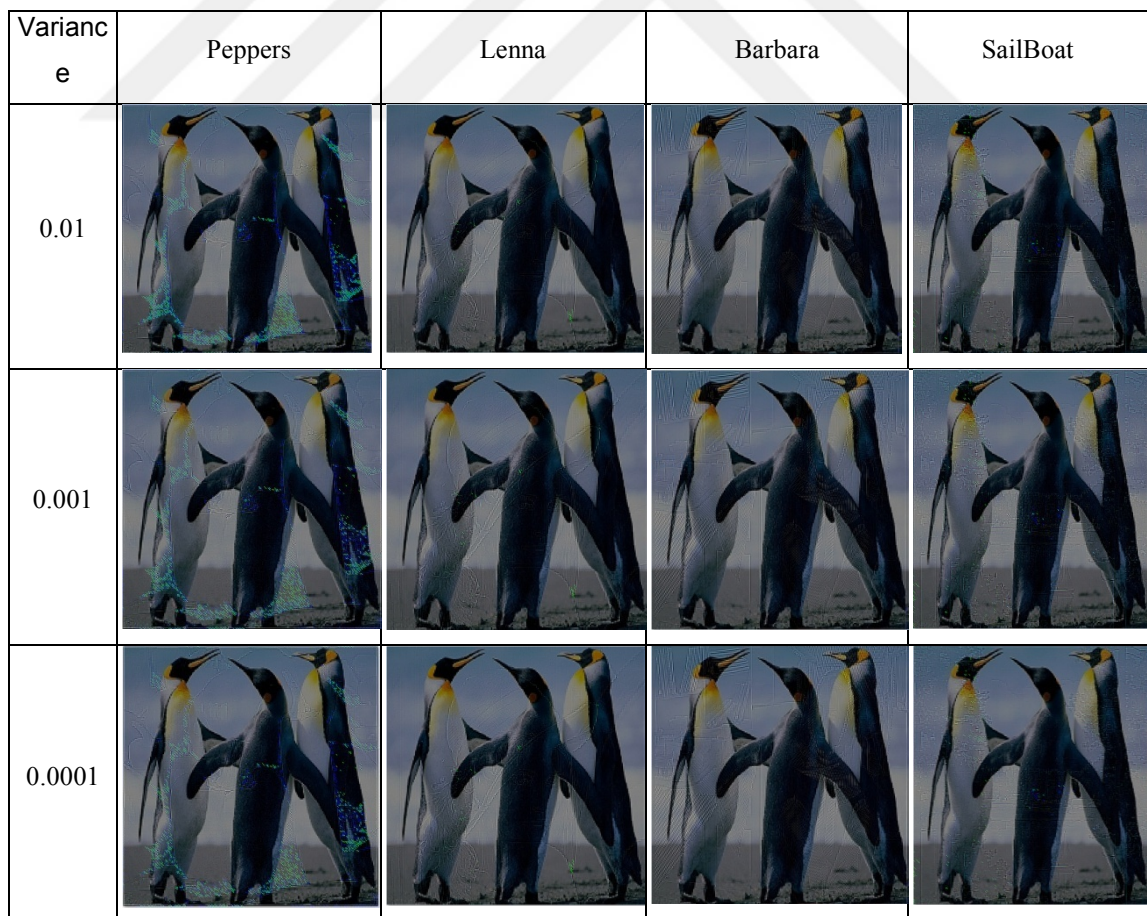


Figure 4.43 Extracted images with Gaussian noise (m=0.5)

As can be seen Figure 4.43, we notice that, by adjusting the value of mean to 0.5 we will be able to extract secret images with better quality compared to Figure 4.41.

Table 4.31 PSNR values for the extracted images with Gaussian noise (m=0.5)

Cover Image	Variance= 0.01			Variance = 0.001			Variance = 0.0001		
	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM
Peppers	14.3267	2452	0.4900	14.3774	2420	0.4918	14.3774	2420	0.4940
Lenna	14.7379	2208	0.5597	14.7380	2208	0.5598	14.7380	2208	0.5598
Barbara	14.6670	2244	0.4925	14.6669	2244	0.4926	14.6669	2244	0.4926
SailBoat	14.6738	2243	0.5098	14.6749	2242	0.5099	14.6749	2242	0.5100

#### 4.4 The Integrated Model

The secret image is 400x400 color image. First the image was encrypted by our Generalized ACM with the following parameters: key=695,  $Mg=[3 \ 2 \ ; \ 4 \ 3]$ , period=120, step(best iteration #)=8. Then The scrambled image is embedded into four cover images with the same size of the secret image. Figure 4.44 summarizes the result of embedding and extracting process.

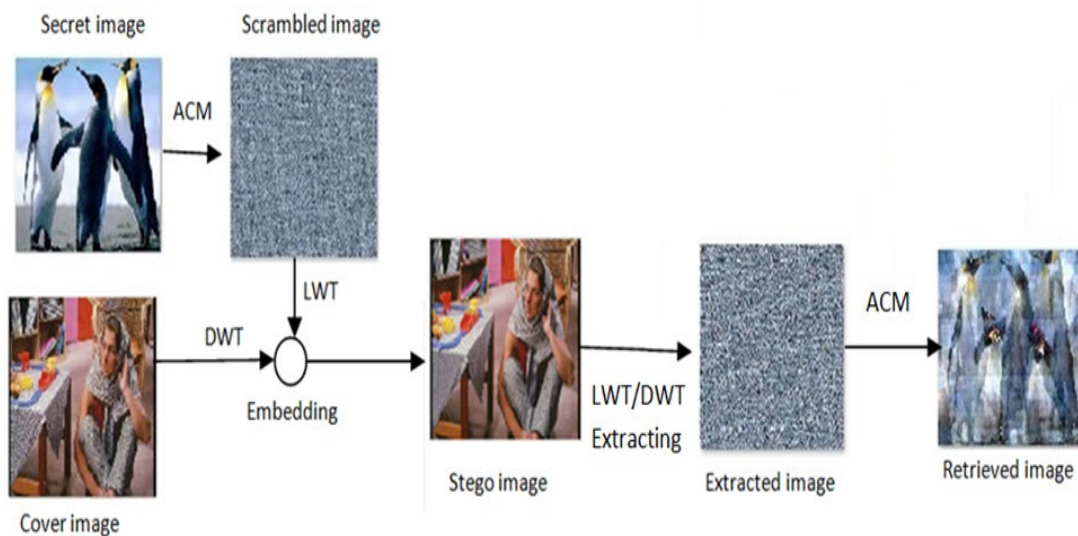


Figure 4.44 The integrated Model for embedding encrypted images

As can be seen from the above figure, the extracted secret image after decryption is distorted, the PSNR value is 17.1537 for the above image. That is because when the

secret image is embedded by any transform domain technique the extracted image will not be the same as the embedded image, so when we apply ACM to the extracted image it will not give the exact original image. This can be refined by embedding the four sub-bands of the secret image to get the best retrieved image, but of course this will decrease the quality of the stego image. However a quality balance should be done depending to our preferences according to the object of the hole process.



## CHAPTER 5

### DISCUSSION ON EXPERIMENTAL RESULTS

In this chapter we will discuss the experimental results obtained in this thesis and we will compare these results with the results obtained by other researchers in the literature. Recalling that our proposed method is a new technique that uses both DWT and LWT as a transform domain technique that can embed RGB images with different formats into another RGB image of the same size or bigger. It works also for gray scale images as well.

We used four standard images as cover images. These images are usually used in the field of steganography and image processing, these images are *Lenna*, *Peppers*, *Barbara* and *Sail-boat*. For the secret image we used one image, *Penguin*.

We have got good results for both stego images and extracted images according to PSNR, MSE and SSIM values obtained.

Some researchers made a standard for considering the acceptable quality of Stego images, that is the PSNR should be greater than 30db [44], [45]. PSNR values falling below 30 dB indicate a fairly low quality, i.e., distortion raised by the embedding process can be obvious, however, stego images having PSNR of 40 dB or above is considered to be of high quality [47]. Our results satisfy these limits, and outperform the results in the literature as can be seen in Table 5.1 and 5.2.

Table 5.1 PSNR values for stego and extracted images in FBE method

Cover Image	Secret Image	Stego Image		Extracted Image	
		PSNR	MSE	PSNR	MSE
Peppers	Peng	<b>67.1135</b>	0.0128	32.1924	52.2613
Lenna	Peng	<b>66.8900</b>	0.0134	31.8542	61.1134
Barbara	Peng	<b>68.3084</b>	0.0095	31.9701	54.7543
Sailboat	Peng	<b>69.9147</b>	0.0068	31.2274	65.0899
Peppers	Lenna	<b>67.1441</b>	0.0127	39.6509	7.1094
Lenna	Peppers	<b>66.8759</b>	0.0135	31.7312	81.8708

Table 5.2 compares the results of previous methods in the spatial domain with the results of our method.

Table 5.2 PSNR values for stego images in different methods

Method	Reference	PSNR	Notes
Pixel Value Differencing Method (PVD)	63	41.38	
Gray level modification method (GLM)	63	34.75	
Pixel Mapping Method (PMM)	63	54.15	
Pixel Position Modulus Method (PPMM)	65	33.9	
Least Significant Bit (LSB)	72	51.11	
LSB 1-2-3 method	38	42.69	Embed in 1 Red, 2 Green, 3 Blue
LSB-Edge based	50	65.66	Embeds Text message in image
2k correction-Edge based	73	38.66	
2k correction-Edge based	53	41.02	
DWT-PFM	63	35.39	
Peace Wise Linear Chaotic Map (PWLCM)	60	60.15	20% Embedding Rate
LSB-DCT	60	59.45	20% Embedding Rate
DCT	72	53.08	
DWT	72	46.78	
DWT-FFT-SVD	75	53.26	Images of YCBCR color space
DCT-DWT	79	54	Embedding in Red component
DWT-Block Matching	76	51.61	
DWT-Diamond embedding	77	52.11	Gray scale image
RDWT - QR Factorization	77	49.03	
DWT- Data Mining	77	73.31	Gray scale image Cover(512x512), hidden(64x128)
DWT-SVD	77	49.16	Gray scale image
LSB-DCT & 3-DES	78	55.29	Gray scale image Cover(512x512), hidden(128x128)
Tree-Based Parity Check TBPC	80	58.99	25% embedding rate
EdgeAdaptive LSB Matching Revisited ELSB-MR	82	56.87	25% embedding rate
Edge XOR Based (1 BPP)	85	57.99	25% embedding rate
Edge XOR Based (N BPP)	85	49.72	25% embedding rate
Edge Adaptive PVD	83	48.32	25% embedding rate
Tri-way Pixel Value Differencing	81	43.59	25% embedding rate
DCT with lucas sequence	84	41.24	25% embedding rate
Flipping the LSB	86	55.45	Transform domain
+/- embedding algorithm	86	54.47	Transform domain
F5 algorithm with matrix embedding	86	56.21	Transform domain
<b>Proposed Method (FBE)</b>	-	<b>67.71</b>	

It is clear that our results are better than results of the other methods in both spatial and transform domains.

Regarding the PSNR values for the extracted images, we obtained an acceptable results ( reaching 39.65 for *Lenna* embedded in *Peppers* ) in which the recovered secret image is fairly clear.

Combining our FBE method with our Generalized ACM, the secret image has a double security. Even in somehow any one extracts the secret bits from the stego image, he will receive only a scrambled image. Although the PSNR quality of the recovered secret image is of a low value, the recovered secret image is still perceivable.



## CHAPTER 6

### CONCLUSIONS AND FUTURE WORK

#### 6.1 Conclusions

We believe that hiding secret data inside another data is not enough to ensure that the secret data is safe. To improve the security, it is very important to use some type of encryption with the secret data before embedding takes place.

In this thesis, a novel image encryption algorithm based on Arnold Cat Map was introduced. We have also presented a scheme for embedding a color image within another color image with the same size in the transform domain using both DWT and LWT. The main contributions of this research lays in two parts and can be summarized as follows.

#### **Encryption Part:**

We established a way for the generalization of ACM that can be used instead of the traditional or the modified ACM variations. In our approach, the four coefficients of the  $2 \times 2$  transform matrix are unknown and there is no fixed structure for this transform matrix.

Also, we calculate the correlation between each adjacent pixels in the candidate encrypted image for each iteration during the period and selecting the iteration number in which the encrypted image has the lowest correlation value, this ensures the best encryption degree.

Experimental results in terms of analysis correlation show that the encrypted image obtained from our proposed algorithm retains small correlation values between adjacent pixels which means good encryption level.

#### **Steganography Part:**

We introduce a new technique in the transform domain, the Fraction Based Embedding Method FBE by combining DWT with LWT. Analysis tests like PSNR, MSE and SSIM has been performed on the proposed method to evaluate its strength.



The proposed method shows a significant quality increase for the stego image because a fraction form of secret data is embedded in the high frequency sub-band which is imperceptible to the Human Visual System and cause a little statistical artifacts to the image. Results show that our method gains a high level of security compared to the previous research in the literature.

All the experimental results obtained in this work, distinctly show the novelty of this approach.

Although we have got a very good results by applying our FBE method, adding noise or compressing the stego image caused a decrease in the quality of the extracted images. This because we embed secret data in the fraction part of coefficient values of the HH band, so any manipulation to image pixels will change the hidden data and thus this will be reflected to the quality of the extracted images.

## **6.2 Future work**

The scheme presented in this thesis is a framework for a more structured digital steganography scheme even with different techniques.

Two possible approaches can be applied using our FBE Technique to improve the quality of the method.

First, we can apply DWT to the cover image and LWT to the secret image, then we can embed at least 3 sub-bands (LL, LH, HL) of the secret image into the LH,HL, HH of the cover image to get the highest quality for the extracted image. However, embedding more sub-bands of the secret image will slightly decrease the PSNR for the stego image but the PSNR for the extracted image will be much better. According to our needs, a balancing policy should be followed.

The second suggested approach combines spatial domain and transform domain techniques. In the spatial domain, we can take the decimal value of each pixel byte of the secret image and embed it as a fraction in the HH sub-band of the transformed cover image by DWT. In this way we expect to have the best quality for the extracted image without degrading the quality of the stego image. This is because we are actually embedding all the 8 bits of secret data instead of embedding maximum of four bits as used by others.

## REFERENCES

- [1] H. Wang, S. Wang, “Cyber warfare: Steganography vs. Steganalysis”, *Communications of the ACM*, 47:10, October 2004.
- [2] S. Bellovin, M. Steven. “Frank Miller: Inventor of the One-Time Pad”. *Cryptologia* 35 (3): 203–222, 2011
- [3] W. Diffie, M. Hellman. “New Directions in Cryptography”, *IEEE Trans. Inf. Theory* IT-22 (1976): 644-654. Kahn, David. *The Code breakers*, rev. ed. New York: Scribner, 1996.
- [4] Y. L. Yang, N. Cai, and G. Q. Ni, “Digital image scrambling technology based on the symmetry of Arnold transform”, *Journal of Beijing Institute of Technology* Vol. 15, pp. 216-220, 2006.
- [5] K. Tsan Lin, “Image Encryption Using Arnold Transform Technique and Hartley Transform Domain”, *9th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2013.
- [ 6 ] V.L Arnold and A. Avez, “Ergodic problems of classical mechanics”, *Benjamin*, New York, 1968.
- [7] Z. Shang, H. Ren, J. Zhang. “A Block Location Scrambling Algorithm of Digital Image Based on Arnold Transformation”, *The 9th International Conference for Young Computer Scientists*, 2008, pp. 2942-2947.
- [8] C. Fu, G. Zhang, M. Zhu, L. Cong, W. Lei, “A Novel Parallel Image Encryption Scheme Using Chaos”, *IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, 2017
- [9] M. Brindha, “Periodicity analysis of Arnold Cat Map and its application to image encryption”, *Proceedings of the International Conference on Inventive Computing and Informatics (ICICI) IEEEExplore Compliant - Part Number: CFP17L34-ART*, ISBN: 978-1-5386-4031, 2017.

- [10] A. Soni, J. Jain, R. Roshan, "Image steganography using discrete fractional Fourier transform", *International Conference on Intelligent Systems and Signal Processing (ISSP)*, 2013, 97-100.
- [11] S. Katzenbeisser and F. Petitcolas, "Information Hiding Techniques for Steganography and Digital Watermarking", *Artech House Publishers*, 2000.
- [12] N. Rani, J. Chaudhary, "Text Steganography Techniques: A Review", *International Journal of Engineering Trends and Technology (IJETT)*, Vol. 4 Issue 7, July 2013.
- [13] J.C. Hernandez-Castro, I. Blasco-Lopez and J.M. Estevez-Tapiador, "Steganography in games: A general methodology and its application to the game of Go", *Computers and Security, Elsevier Science*, 25, 64-71, 2006.
- [14] H. Qi, W. Snyder, and W. Sander, "Blind consistency-based steganography for information hiding in digital media", in *Proceedings of IEEE International Conference on Multimedia and Expo*, vol. 1, Switzerland, 2002.
- [15] Y.K Lee, and L.H Chen, "High capacity image steganographic model", *Visual Image Signal processing*, 147:03, June 2000.
- [16] M. Buker, H. Tora, E. Gokcay, "Effect of Secret Image Transformation on the Steganography Process", *24th IEEE International Conference on Electronics, Circuits and Systems*, Georgia, Dec. 2017.
- [17] X. Liao, Q. Wen, and J. Zhang, "A steganographic method for digital images with four-pixel differencing and modified LSB substitution", *Journal of visual communication and image representation*, vol. 22, no.1, pp. 1-8, 2011.
- [18] T. Sharp, "An implementation of key-based digital signal steganography", *Lecture Notes in Computer Science, Springer Berlin Heidelberg*, vol. 2137, pp. 13-26, 2001.
- [19] J. Mielikainen, "LSB matching revisited", *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 285-287, 2006.
- [20] Z. Xia, X. Wang, X. Sun, Q. Liu, and N. Xiong, "Steganalysis of LSB matching using differences between nonadjacent pixels", *Multimedia Tools and Applications*, vol. 75, no. 4, pp. 1947-1962, 2016.

- [21] S. Bhattacharyya, A. Khan, I. Banerjee, G. Sanyal, "A Robust Image Steganography Method Using PMM in Bit Plane Domain", *International Journal of Computer and Information Engineering* Vol.:8, No:9, 2014.
- [22] D. Wu, W. Tsai, "A steganographic method for images by pixel-value differencing", *Pattern Recognition Letters*, Vol. 24, Issues 9-10, pp. 1613-1626, June 2003.
- [23] N. Provos, "Defending against statistical steganalysis", *10th USENIX Security Symposium*, 2001.
- [24] H. Kekre, T. Sarode, P. Halarakar, "Image Scrambling using R-Prime Shuffle", *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, (IJAREEIE), vol.2 Issue 8, pp. 4070-4076, August 2013.
- [25] H.B.Kekre, Tanuja Sarode, Pallavi Halarakar, "Image Scrambling Using R-Prime Shuffle on Image and Image Blocks", *International Journal of Advanced Research in Computer and Communication Engineering* Vol. 3, Issue 2, February 2014.
- [26] Q. Daong-Xu, Z. Cheng, H. You, "A new class of scrambling transformation and its application in the Image information covering", *Science in China(Series E)*, Vol. 43(3), pp. 304-312, 2000
- [27] Y. Zhang, Q. Wang, "A New Scrambling Method Based on Arnold and Fermat Number Transformation", *International Conference on Environmental Science and Information Application Technology*, 2009.
- [28] J. ZOU, R. Ward, "Introducing Two New Image Scrambling Methods", *IEEE Pacific Rim Conference on Communications Computers and Signal Processing (PACRIM 2003)*.
- [29] S. Bukhri, M. Anjum, I. Bajwa, S. Dillbar, "Chaos Image Encryption Followed By the Steganography Technique", *Sindh Univ. Res. Jour. (Sci. Ser.)* Vol.49(1), pp. 113-118, 2017.

- [30] P. Gupta, S. Singh, I. Mangal, "Image Encryption Based On Arnold Cat Map and S-Box", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 4, Issue 8, August 2014
- [31] M. Farajallah, Z. Fawaz, S. El-Assad, O. Deforges, "Efficient Image Encryption and Authentication Scheme Based on Chaotic Sequences", *The Seventh International Conference on Emerging Security Information, Systems and Technologies*, 2013.
- [32] N. Mohamed, M. El-Azeim, A. Zaghloul, "Improving Image Encryption Using 3D Cat Map and Turing Machine", *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 1, 2016.
- [33] A. Zhang, N. Zhou, L. Gong, "Color Image Encryption Algorithm Combining Compressive Sensing with Arnold Transform", *journal of computers*, Vol. 8, No. 11, November 2013.
- [34] D. Wang, C. Chang, Y. Liu, G. Song, Yunbo Liu, " Digital Image Scrambling Algorithm Based on Chaotic Sequence and Decomposition and Recombination of Pixel Values", *International Journal of Network Security*, Vol.17, No.3, PP.322-327, May 2015.
- [35] Gustavus J. Simmons, "The Prisoners' Problem and the Subliminal Channel", in *Proceedings of CRYPTO '83*, pp 51-67. Plenum Press, 1984.
- [36] E. Kawaguchi, R. Eason, "Principle and applications of BPCS-Steganography " , *The SPIE Conference on Multimedia Systems and Applications*, Boston, 1998.
- [37] P. Khairnar, V. Ubale, "Steganography Using BPCS technology", *International Journal Of Engineering And Science*, Vol.3, Issue 2, pp. 08-16, May 2013.
- [38] D. Rawat, V. Bhandari, "A Steganography Technique for Hiding Image in an Image using LSB Method for 24 Bit Color Image", *International Journal of Computer Applications (0975 – 8887)* Vol. 64– No.20, February 2013.
- [39] G. Sravanthi, B. Devi, S. Riyazoddin & M. Reddy, "A Spatial Domain Image Steganography Technique Based on Plane Bit Substitution Method", *Global Journal of Computer Science and Technology Graphics & Vision* Vol. 12 Issue 15, 2012

- [40] G. Kamau, S. Kimani, W. Mwangi, “An enhanced Least Significant Bit Steganographic Method for Information Hiding”, *Journal of Information Engineering and Applications*, Vol 2, No.9, 2012.
- [41] M. Juneja and P. Sandhu, “An Improved LSB Based Steganography Technique for RGB Color Images”, *International Journal of Computer and Communication Engineering*, Vol. 2, No. 4, July 2013.
- [42] B. Banik, S. Bandyopadhyay, “Image Steganography Using Bit Plane Complexity Segmentation and Hessenberg QR Method”, *In Proceedings of the First International Conference on Intelligent Computing and Communication*, 2017, pp. 623-633, Springer Singapore.
- [43] A. Kumar, M. Bansal, “Hiding Negative of an Image using Steganography Even Odd Algorithm for Security Purposes”, *IOSR Journal of Computer Engineering (IOSR-JCE)*, Vol. 16, Issue 1, pp. 70-75, Feb. 2014.
- [44] A. Krishna, S. Parimi, G. Manikandan, N. Sairam, “A Clustering Based Steganographic Approach for Secure Data Communication”, *International Conference on Circuit, Power and Computing Technologies [ICCPCT]*, 2015.
- [45] J. Chandrasekaran, G. Arumugam, D. Rajkumar, “Ensemble of Logistic Maps with Genetic Algorithm for Optimal Pixel Selection in Image Steganography”, *IEEE Sponsored 2nd International Conference on Electronics and Communication System, (ICECS)*, 2015.
- [46] A. Tyagi, R. Roy, S. Changder, “High Capacity Image Steganography based on Pixel Value Differencing and Pixel Value Sum”, *Second International Conference on Advances in Computing and Communication Engineering*, 2015.
- [47] S. Thepade, S. Thube, “Novel Color Image Steganography Using Vector Quantization Codebook Generation Methods LBG, TCEVR and KFCG”, *International Conference on Pervasive Computing (ICPC)*, 2015.
- [48] S. Islam, M. R Modi, P. Gupta, “Edge-based image steganography”, *EURASIP Journal on Information Security*, Vol 8, 2014.

- [49] S. Kaur, S. Singh, “A New Image Steganography Based on 2k Correction Method and Canny Edge Detection”, *International Journal of Computing & Business Research* ISSN: 2229-6166, 2012
- [50] N. Jain, S. Meshram, S. Dubey, “Image Steganography Using LSB and Edge Detection Technique”, *International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2231-2307, Vol.2, Issue-3, July 2012.
- [51] Y. Bassil, “Image Steganography based on a Parameterized Canny Edge Detection Algorithm”, *International Journal of Computer Applications* (0975 – 8887) Volume 60– No.4, December 2012.
- [52] R. Amirtharajan, S. Behera, M. Swarup, M. Ashfaaq, J. Rayappan, “Colour Guided Colour Image Steganography”, *Universal Journal of Computer Science and Engineering Technology*, Vol 1, 16-23, October 2010.
- [53] A. Kaur, S. Kaur, “Image Steganography Based on Hybrid Edge Detection and 2k Correction Method”, *International Journal of Engineering and Innovative Technology (IJEIT)*, Vol 1, Issue 2, February 2012.
- [54] W. Chen, C. Chang, T. Ngan Le, “High payload steganography mechanism using hybrid edge detector”, *Expert Systems with Applications*, 37 3292–3301, 2010.
- [55] S. Arora, S. Anand, “A Proposed Method for Image Steganography Using Edge Detection”, *International Journal of Emerging Technology and Advanced Engineering*, Vol. 3, Issue 2, February 2013.
- [56] V. Jadhav, P. Belagali, S. Soudagar, P. Patil, “Edge Adaptive Image Steganography using LSB Matching Revisited”, *IEEE Transactions on Information Forensics and Security*, Vol: 5 , Issue: 2 , 201 - 214, June 2010.
- [57] H. Hiary ,K. Sabri, M. Mohammed, A. Al-Dhamari, “A Hybrid Steganography System based on LSB Matching and Replacement”, *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 9, 2016.
- [58] M. KUMAR, M. YADAV, “Image Steganography Using Frequency Domain”, *International Journal of Scientific & Technology Research* , VOL 3, ISSUE 9, Sep. 2014.

- [59] A. Al-Ataby, Fa. Al-Naima, “A Modified High Capacity Image Steganography Technique Based on Wavelet Transform”, *The International Arab Journal of Information Technology*, Vol. 7, No. 4, October 2010.
- [60] M. Habib, B. Bakhache, D. Battikh, S. El Assad, “Enhancement using chaos of a Steganography method in DCT domain”, Fifth International Conference on Digital Information and Communication Technology and its Applications (DICTAP), 2015.
- [61] A. ElSayed, A. Elleithy, P. Thunga and Z. Wu, “Highly Secure Image Steganography Algorithm using Curvelet Transform and DCT Encryption”, *Long Island Systems, Applications and Technology*, 2015.
- [62] L. Guo, J. Ni, W. Su, C. Tang, and Y. Shi, “Using Statistical Image Model for JPEG Steganography: Uniform Embedding Revisited”, *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 12, Dec 2015.
- [63] S. Ash, S. Mukherjee, “A DWT Based Steganographic Method Using Prime First Mapping (PFM) ”, *Second International Conference on Advances in Computing and Communication Engineering*, 2015.
- [64] K. Sabyasachi, R. Ratnakirti, C. Suvamoy, “A DWT based Steganography Scheme with Image Block Partitioning”, *2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, 2015.
- [65] S. Mukherjee, S. Ash, G. Sanyal, “A Novel Image Steganographic Approach by Pixel Position Modulus Method (PPMM) ”, *2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2015.
- [66] S. Ahani, S. Ghaemmaghami, “Colour image steganography method based on sparse representation”, *IET Image Process.*, Vol. 9, Iss. 6, pp. 496–505, 2015.
- [67] A. Nag, S. Biswas, D. Sarkar and P. Sarkar, “A novel technique for image steganography based on DWT and Huffman coding”, *IJCSS*, vol. 4, no. 6, pp. 561-570, Jan. 2011.
- [68] A. Verma, R. Nolkha, A. Singh and G. Jaiswal, “Implementation of Image Steganography Using 2-Level DWT Technique”, *International Journal of Computer Science and Business Informatics*, Vol. 1, No. 1, May 2013.



- [69] L. Min, L. Ting, H. Yujie, “Arnold transform Based Image Scrambling Method”, *Proceedings of 3rd International Conference on Multimedia Technology(ICMT-13)*, Nov. 2013.
- [70] Z. Tang, X. Zhang, “Secure Image Encryption without Size Limitation Using Arnold Transform and Random Strategies”, *Journal of Multimedia*, Vol. 6, No. 2, April 2011.
- [71] R. Calderbank, “Wavelet Transforms that map integers to integers”, *Applied And Computational Harmonic Analysis* Vol. 5, pp.332–369, 1998.
- [72] S. Chandran, K. Bhattacharyya, “Performance Analysis of LSB, DCT, and DWT for Digital Watermarking Application using Steganography”, *International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, 2015.
- [73] J. Yu<sup>1</sup>, E. Yoon, S. Shin<sup>1</sup> and K. Yoo<sup>1</sup>, “A New Image Steganography Based on 2k Correction and Edge-Detection”, *Fifth International Conference on Information Technology: New Generations*, (itng 2008), Las Vegas, NV, 2008, pp. 563-568.
- [74] G. Nason, B. Silverman, “The Stationary Wavelet Transform and some Statistical Applications”, *In: Antoniadis A., Oppenheim G. (eds) Wavelets and Statistics. Lecture Notes in Statistics*, vol. 103. Springer, New York, NY, 1995.
- [75] S. Yadav, M. Dixit, “An Improved Image Steganography based on 2- DWT-FFT-SVD on YCBCR Color Space”, *International Conference on Trends in Electronics and Informatics*, 2017.
- [76] J. Kim, H. Park, J. Park, “Image Steganography Based on Block Matching in DWT Domain”, *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2017.
- [77] N. Surse, P. Vinayakray-Jani, “A Comparative Study on Recent Image Steganography Techniques Based on DWT”, *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, 2017, pp. 1308-1314.

- [78] G. Ardiansyah, C. Sari, “Hybrid Method using 3-DES, DWT and LSB for Secure Image Steganography Algorithm”, *2nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2017.
- [79] P. Sharma, A. Sharma, “Robust Technique For Steganography on Red Component Using 3- DWT -DCT Transform”, *Proceedings of the Second International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, 2018, pp. 1049-1054.
- [80] C. Hou, C. Lu, S. Tsai and W. Tzeng, “An optimal data hiding scheme with tree-based parity check”, *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 880-886, March 2011.
- [81] Y. Lee, J. Lee, W. Chen, K. Chang, J. Su and C. Chang, “High-payload image hiding with quality recovery using tri-way pixel-value differencing”, *Information Sciences*, Vol 191, pp.214–225, 2012.
- [82] W. Luo, F. Huang and J. Huang, “Edge adaptive image steganography based on LSB matching revisited”, *IEEE Transactions on Information Forensics and Security*,5(2),201–214, 2010.
- [83] J. Mandal, D. Das, “Steganography using adaptive pixel value differencing (APVD) of gray images through exclusion of overflow/underflow”, *In Second international conference on computer science, engineering and applications (CCSEA-2012)* , pp.93–102, 2012.
- [84] D. Prithwish, R. Supriyo and D. Atanu, “An Efficient Embedding Technique in Image Steganography Using Lucas Sequence”, *International Journal of Image, Graphics and Signal Processing*, Vol. 9, pp. 51-58, 2017.
- [85] H. Al-Dmour, A. Al-Ani, “A steganography embedding method based on edge identification and XOR coding”, *Expert Systems With Applications*, Vol. 46, pp. 293–306, 2016.
- [86] P. Malathi, T. Gireeshkumar, “Relating the embedding efficiency of LSB Steganography techniques in Spatial and Transform domains”, *Procedia Computer Science*, Vol. 93, pp. 878 – 885, 2016.

[87] W. Zhou, C. Alan Bovik, H. Sheikh, E. Simoncelli, “Image Quality Assessment: From Error Visibility to Structural Similarity”, *IEEE Transactions on Image processing*, Vol. 13, No. 4, pp. 600–612, April 2004.

