COMPARISON OF DIFFERENT ININFORMATION HIDING TECHNIQUES INTO VISUAL OBJECTS (STEGANOGRAPHY)

A MASTER'S THESIS

in

Computer Engineering

Atılım University

by

BASHIR MUJBER

DECEMBER 2015

COMPARISON OF DIFFERENT ININFORMATION HIDING TECHNIQUES INTO VISUAL OBJECTS (STEGANOGRAPHY)

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

OF

ATILIM UNIVERSITY

BY

BASHIR MUJBER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF COMPUTER ENGINEERING

DECEMBER 2015

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

_____

Prof. Dr. İbrahim Akman

Director

I certify that this thesis satisfies all the requirements as a thesisfor the degree of Master of Science.

_____

Prof. Dr. İbrahim Akman

Head of Department

This is to certify that we have read the thesis "Comparıson Of Dıfferent Inınformatıon Hıdıng Technıques Into Vısual Objects (Steganography)" submitted by "Bashir Mohamed Ali Mujber" and that in ouropinion it is fully adequate, in scope and quality, as a thesis forthe degree of Master of Science.

_____

Asst. Prof. Dr. Atila Bostan

Supervisor

Examining Committee Members

Assoc. Prof. Dr. H. Hakan MARAŞ _____

Asst. Prof. Dr. Gökhan Şengül _____

Asst. Prof. Dr. Atila Bostan _____

Date: December 8,2015

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Bashir Mujber

Signature:

ABSTRACT

COMPARISON OF DIFFERENT TECHNIQUES ININFORMATION HIDING INTO VISUAL OBJECTS (STEGANOGRAPHY)

Mujber, Bashir

M.S., Computer Engineering Department

Supervisor: Asst. Prof. Dr. Atila Bostan

December 2015, 91 Pages

Steganography is an important field of research in recent years to embed a range of data. It is the science that hide information in cover medium without being accompanied by any detectable effect or distortion in that medium. Nevertheless, most of the modern researches focus on hiding information in image according to its popularity. This thesis compares the efficiency of three techniques in embedding text in an image file. Although there are several others, the selected techniques in this study are Least Significant Bit (LSB) (1-LSB, 2-LSB and 3-LSB) with one or RGB color, Bit Plane Complexity Segmentation (BPCS) with various threshold values and Discrete Cosine Transform (DCT) with varying quality factors. The LSB and BPCS algorithms have experimented in spatial domain on Bitmap 24 bits format as cover image to generate a stego images while DCT algorithm is implemented in frequency domain on JPEG image format in which the stego image is transformed from spatial domain to the frequency domain and secret bits are concealed into the frequency modules of the original image.

The aim of this thesis is to carry out various types of image steganography techniques for purpose of identifying various principles of image steganography in terms of visual effectiveness and efficiency of the selected algorithms on different images visual characteristics. However, the algorithms that had been chosen for this purpose are discussed in details in this thesis. The visual affectivity of the stego images were measured by comparing the histograms of the stego and cover images. In order to implement a sound and unbiased discussion, we used MSE calculation in comparisons

In addition, we discussed the implementation of these algorithms in detail with an explanation on the obtained results in all cases.

Keywords: Steganography, Least Significant Bit, Bit Plane Complexity Segmentation, Discrete Cosine Transform, MSE, Histogram.

ÖZ

GÖRSEL OBJELERE BİLGİ GİZLEME (STEGANOGRAFİ) DEFARKLI TEKNİKLERİN KARŞILAŞTIRILMASI

Mujber, Bashir

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Asst. Prof. Dr. Atila Bostan

Aralık 2015, 91 Sayfa.

Steganografi son yıllarda verileri bir objeye gömme ve saklama konusunda önemli bir araştırma alanı haline gelmiştir. Steganografi ilgili ortamda fakedilebilir herhangi bir etki veya bozulma yaratmaksızın, kapak ortamda bilgi gizleme bilimdir. Bununla birlikte bir çok modern araştırma, populeritesine göre değişik görsellere bilgi gizleme konusuna yoğunlaşmıştır. Bu tez çalışmasında görsellere metin mesaj saklanmasında kullanılan üç farklı tekniğin etkinlikleri karşılaştırılmıştır. Literatürde başka tekniklerin de bulunmasına rağmen çalışmada seçilen üç teknik şunlardır. Tek renk ve RGB renk bileşimi ile En Önemsiz Bit (LSB), (1-LSB, 2-LSB and 3-LSB) alternatifleri, , değişik eşik değerleri ile Ayrılık Kosinüs Dönüşümü (DCT), farklı kalite katsayıları ile Bit Düzlem Karmaşıklığı Segmentasyonu (BPCS). LSB ve BPCS algoritmalarında altlık olarak mekansalkodlamalı24 bitlik Bitmap görüntüler kullanılmıştır.. Öte yandan DCT algoritmasında ise frekans kodlamalı JPEG görüntüler kulanılmıştır. Burada atlık oalarak kullanılan görüntüler mekansal kodlamadan frekens kodlamasına dönüştürülmüş ve gizli mesaj bu görüntüye gömülmüştür.

Bu tezin amacı değişik görüntü steganografi tekniklerinin uygulaması ve değişik görüntüler üzerinde her bir algoritmanın görsel etkinlik karşılaştırmasını yapmaktır. Bu bağlamda, bu tez çalışmasında, seçilen algoritmalar detaylı şekilde tartışılmıştır. Stego görsellerin etkinliği stego ve altlık görüntü histogramlarının karşılaştırılması ile ölçülmüştür. Güçlü ve tarafsız bir tartışma yürütebilmek adına, karşılaştırmalarda MSE hesaplama tekniği kullanılmıştır. Buna ilave olarak, bahse konu algoritmaların

uygulama detaylarııca tartışılmış, ve yaptığımız örnek çalışmalardan elde edilen her bir sonucun açıklamaları da sunulmuştur.

Anahtar Kelimeler: Steganografi, En Önemsiz Bit, Bit Düzlem KarmaşıklığıSegmentasyonu, Ayrık Kosinüs Dönüşümü, MSE, Histogram.

ACKNOWLEDGMENTS

Thanks to God the most compassionate and the most merciful. My Allah's mercy and peace upon our leader Mohammed, Who invites us to science and wisdom and members of his family and his followers.

I would to express my deep gratitude after God almightily in the completion of this research to my supervisor Asst. Prof. Dr. Atila Bostan who has suggested this thesis and gave me a lot his time. I am indebted for his suggestions and valuable remarks.

Also, I would like to express my gratitude to my parents and my sisters and brothers. Finally, I would like to thanks my wife for her help and encouragement and to everyone who helped in one way or another in bringing out this work.

My God bestow health and happiness to all of them.

Motivation

Since the process of encrypting information has been used for a long time in different methods and techniques, so it became a common method to send secret information from one place to another. In addition, the third party may receive the message and can temper the content, try to decrypt it or prevent its delivery. So, the focus is directed on science that prevents third parties to see the information during the transmission process and this science is known as steganography.

Steganography is considered as a good method for exchanging the secret information between the parties around the world through the internet without the fear of exposing the information that is inserted inside a medium. On the other hand, this science concerns on concealing information using the digital media such as images, texts, voice and video.

However, steganography through images is the most popular method because the images are the most available files through the internet. When using encryption the size of the message is not important but it is a big challenge when using steganography because in most of the techniques the size of the cover file (i.e. image) defines the message capacity.

Meanwhile, the effectiveness of the information concealment in a cover image is one of the key concepts -that image steganography should focus on.

Thesis Objective

The primary goal of this thesis is to study the effectiveness of selected image steganography techniques from the visual distortion viewpoint. The history of steganography in terms of its development and applications is also summarized in the study. In addition, we studied the internal structure of the images and their characteristics and types, through studying some of the image steganography techniques and by comparing their impact on the original images after the embedding process. Also, image steganography algorithms have been applied on images with different visual characteristics (indoor, outdoor and indoor with people) and we discussed the obtained results in details.

Thesis Organization

This thesis has been divided into five chapters as follows:

- **Chapter One:** Provides the definition and history of steganography and the difference between steganography and cryptography. Also, it gives a brief description on the steganography types according to the types of the media that works on it. In addition, we explained in details some of image steganography algorithms by explaining the structure of each algorithm and the embedding and extraction processes.
- **Chapter Two:** provides the research methodology, and describes the selected images and the text that has been used as a secret information in detail. Furthermore, we illustrated each algorithm according to its parameters, in addition to the tools and the applications that were used in this study.
- **Chapter Three:** We have presented the results of the study as the histograms of the stego and the original images. The results are presented for each of the algorithm that is explained in detail in chapter two.
- **Chapter Four:** We compared the original image histograms with that of the stego-images which were the results of the selected algorithms. We discussed the visual effectiveness of the algorithms in terms of the MSE calculation.
- **Chapter Five:** It includes a summary and conclusions. As well as, it reports the recommendations and the trends that can be used for future work in this field.

TABLE OF CONTENTS

ABSTRACT iii

ÖZ v

Motivation viii

Thesis Objective viii

Thesis Organization viii

TABLE OF CONTENTS x

LIST OF TABLES xiv

LIST OF FIGURES xv

LIST OF ABBREVIATIONS xviii

Chapter One 1

1. Introduction 1

1.1 What is Steganography 1

1.2 Steganography History 1

1.3 Steganography vs Cryptography 2

1.4 The Steganography Criteria 4

1.5 The Basic Model of Steganography 4

1.6 Steganography Types 5

1.6.1 Text 5

1.6.2 Audio 6

1.6.3 Image 7

1.6.3.1 Spatial Domain Techniques 8

1.6.3.2 Transform Domain Techniques 8

1.6.4 Video 9

1.7 Image Steganography 9

1.7.1 Image File 9

1.7.2 Image Types 9

1.7.3 Embedding/Insertion a Message into an Image 10

1.7.3.1 Spatial Domain 10

1.7.3.1.1 Least Significant Bit (LSB) 11

1.7.3.1.1.1 Embedding Steps: 13
1.7.3.1.1.2 Extraction Steps: 13
1.7.3.1.2 Bit Plane Complexity Segmentation (BPCS) 14
1.7.3.1.2.1 Bit Plane 14
1.7.3.1.2.2 Binary Gray Code 16
1.7.3.1.2.3 Complexity Measure 17
1.7.3.2.5.3.1 Measurement of Complexity Based on Black and White Border 17
1.7.3.2.5.3.1 Measures of Complexity Based on Number of Close Areas, β 18
1.7.3.1.2.4 Resource Blocks and the Conjugation Operation 19
1.7.3.1.2.5 Embedding Steps: 22
1.7.3.1.2.6 Extraction Steps: 23
1.7.3.2 Frequency Domain 23
1.7.3.2.1 DCT 24
1.7.3.2.2 Jpeg Compression 25
1.7.3.2.2.1 RGB to YCbCr Conversion 26
1.7.3.2.2.2 Sampling 27
1.7.3.2.2.3 Transformation 28
1.7.3.2.2.4 Quantization 28
1.7.3.2.2.5 DPCM (Differential Pulse Code Modulation) 30
1.7.3.2.2.6 Entropy Coding on the DC Coefficient 30
1.7.3.2.2.7 RLC (Run Length Coding) 31
1.7.3.2.2.8 Entropy Coding the AC Coefficients 32
1.7.3.2.3 Hiding in DCT Domain 33
1.7.3.2.4 Algorithms Based on LSB Substitution 34
1.7.3.2.4.1 JSTEG Algorithm 34
1.7.3.2.5 F Group of Algorithms 35
1.7.3.2.5.1 F3 35
1.7.3.2.5.2 F4 35
1.7.3.2.5.3 F5 36
1.7.3.2.5.3.1 Permutation Scattering 36
1.7.3.2.5.3.2 Matrix Coding 38
1.7.3.2.5.3.3 Embedding Steps 40
1.7.3.2.5.3.4 Extracting Steps 41
1.8 Steganalysis 42
1.8.1 Ciphertext-only Attacks: 43
1.8.2 Chosen Plaintext Attack: 43
1.8.3 Chosen Ciphertext Attack: 43
1.8.4 Chosen Stego Attack: 44
1.8.5 Stego Only Attack: 44
1.8.6 Known Cover Attack: 44
1.8.7 Known Message Attack: 45

1.8.8 The Targeted Steganalysis: 45
1.8.9 The Blind Steganalysis: 45
1.8.10 Semi-Blind Steganalysis: 45
1.9 Literature Survey 45
Chapter Two 48
2. Methodology 48
2.1 Cover Image Selection 48
2.2 Secret Message Selection 48
2.3 Steganography Algorithm Selection and Used Tools: 49
2.3.1 Stego-Image Generation by LSB 49
2.3.1.1 LSB in a Single Color 50
2.3.1.1.1 One LSB 50
2.3.1.1.2 Two LSBs 50
2.3.1.1.3 Three LSBs 51
2.3.1.2 LSB in Three Color (RGB) 51
2.3.2 Stego-Image Generation by BPCS 53
2.3.3 Stego-Image Generation by F5 54
2.4 Criteria for Assessing the Quality of the Image-Stego 60
2.4.1 Histogram 60
2.4.2 Mean Square Error 61
Chapter Three 63
3. Results 63
3.1 Substitution Technique (LSB Method) 64
3.1.1 One LSB 64
3.1.2 Two LSB 65
3.1.3 Three LSB 66
3.2 BPCS Techniques 67
3.3 DCT Techniques (F5 Method) 72
Chapter Four 76
4. Discussion 76
4.1 Substitution Technique (LSB method) 76
4.2 BPCS Method 80
4.3 DCT Technique (F5 Method): 81
Chapter Five 84

5. Conclusion 84
5.1 Conclusion 84
5.2 The Recommendation 85
5.3 Future Work 85
References 87

LIST OF TABLES

TABLE

1. THE CONNECTION BETWEEN THE EMBEDDING RATE AND CHANGE DENSITY . 39
2. DEPENDENCY (×) BETWEEN MESSAGE BITS xi AND CODE WORD BITS $a'j$ 40
3. THE ABBREVIATION OF USED HISTOGRAM TITLE 63
4. MSE RESULTS OF LSB METHOD WITH ONE COLOR..... 77
5. MSE RESULTS OF LSB METHOD WITH RGB COLOR 77
6. MSE RESULTS OF BPCS METHOD 80
7. MSE RESULTS OF F5 METHOD..... 82

LIST OF FIGURES

FIGURES

1. CRYPTOGRAPHY 3
2. STEGANOGRAPHY 3
3. THE EMBEDDING MODEL OF STEGANOGRAPHY 5
4. EMBEDDING WITH THE LSB METHOD 12
5. BIT PLANE LEVELS 15
6. BIT PLANE LEVELS 15
7. PBC AND CGC SYSTEM 16
8. FOUR BLACK PIXELS SURROUNDING A WHITE PIXEL 17
9. (A) ALL PIXELS IN THE IMAGE ARE WHITE (B) BLACK-WHITE CHECKER BOARD 18
10. CALCULATE A SUMMATION OF BLACK AND WHITE BORDERS 18
11. VALUES OF A AND B FOR SOME BLOCKS OF SIZE 8×8 19
12. RESOURCE BLOCKS OF 2 DIFFERENT 8-CHARACTER 20
13. EXPLAIN THE CONJUGATION PROCESS 21
14. BPCS TECHNIQUE 22
15. BASELINE JPEG ENCODER 25
16. COLOR TRANSFORMATION 26
17. THE 4:2:0 SAMPLING DIAGRAM 27
18. 4:2:2 SAMPLING METHOD 27
19. AN EXAMPLE OF DCT COEFFICIENTS FOR AN 8X8 BLOCK 28
20. QUANTIZATION TABLES 29
21. AFTER THE QUANTIZATION PROCESS 29
22. DPCM PROCESS 30
23. RELATIONS SIZE, AMPLITUDE AND NUMBER 31

24. Entropy Coding On the DC Coefficient 31
25. Zigzag Order Detour 32
26. RCL Process 32
27. Entropy Coding the AC Coefficients 33
28. Block Diagram of Embedding Process of JPEG Steganography 33
29. Block Diagram of Extraction Process of JPEG Steganography 34
30. JSTEG Algorithm 34
31. F5 Method Processes 36
32. Hiding a Random Walk 37
33. Hiding Permutation Scattering 37
34. Cover Image Selection A, B and C Respectively 48
35. One Bit in Blue Color 50
36. Two Bits in Blue Color 51
37. Three Bits in Blue Color 51
38. LSB Three Colors Embedded 52
39. LSB Method Interface 52
40. Implementation of F5 Program 59
41. Histogram of RGB Color 61
42. Original Bitmap Images Histogram 64
43. Histogram of One LSB One Color 64
44. Histogram of One LSB RGB Color 65
45. Histogram of Two LSB One Color 65
46. Histogram of Two LSB RGB color 66
47. Histogram of Three LSB One Color 66
48. Histogram of Three LSB RGB color 67
49. Histogram of BPCS with Threshold (5) 67
50. Histogram of BPCS with Threshold (10) 68

51. HISTOGRAM OF BPCS WITH THRESHOLD (15) 68
52. HISTOGRAM OF BPCS WITH THRESHOLD (20) 69
53. HISTOGRAM OF BPCS WITH THRESHOLD (25) 69
54. HISTOGRAM OF BPCS WITH THRESHOLD (30) 70
55. HISTOGRAM OF BPCS WITH THRESHOLD (35) 70
56. HISTOGRAM OF BPCS WITH THRESHOLD (40) 71
57. HISTOGRAM OF BPCS WITH THRESHOLD (45) 71
58. HISTOGRAM OF BPCS WITH THRESHOLD (50) 72
59. HISTOGRAM OF BPCS WITH THRESHOLD (55) 72
60. HISTOGRAM OF DCT WITH QUALITY (0)..... 73
61. HISTOGRAM OF DCT QUALITY (20)..... 73
62. HISTOGRAM OF DCT QUALITY (40)..... 74
63. HISTOGRAM OF DCT QUALITY (60)..... 74
64. HISTOGRAM OF DCT QUALITY (80)..... 75
65. HISTOGRAM OF DCT QUALITY (100)..... 75
66. MSE WITH LSB OF RED COLOR..... 78
67. MSE WITH LSB OF GREEN COLOR 78
68. MSE WITH LSB OF BLUE COLOR..... 79
69. MSE WITH LSB OF RGB COLOR 79
70. MSE WITH BPCS METHOD..... 81
71. GRAPH COMPARISON OF MSE RESULTS OF F5 METHOD 82

LIST OF ABBREVIATIONS

AC	Alternating Current Terms
BMP	Bitmap
BPCS	Bit Plane Complexity Segmentation
DC	Direct Current Term
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
JPEG	Joint Photographic Experts Group
LSB	Least Significant Bit
MSE	Mean Squared Error
PBC	Pure Binary Code

Chapter One

1. Introduction

1.1 What is Steganography

Steganography means verbatim which covers the procedure of writing, and is studied as one of the most crucial communication arts [1]. It contains two words, the first word is Steganos means "coverage" and the other word graph in which means “writing” in Greek. The main idea of it is to conceal the communication procedure without using encryption algorithms that make the process of communication non-understandable except for those who have the right keys [11].

Steganography inside the image is the process of developing a hidden message within the same or given image so that nobody can know what the message is or should not be able to detect its presence.

1.2 Steganography History

To understand Steganography we need to look into the history first where steganography appeared even before the actualization of the encryption, i.e. When the Greek historian Herodotus showed how information can be exchanged and secretly delivered through the apparently plain looking wax tablets in 474 BC, underneath which wood was scratched to write a secret message. Then, he explained by giving example of Demeratus a Greek living at that time in Persia in exile exported hidden warning messages to Greece about the invasion plan made by Persia against his country and in order to avoid the risk of them being revealed on the way to the city of Sparta, he put his message on wood, scratching the wood base and then covering with wax. The message transferred in form of apparently blank tablets to the city of Sparta and the allies got the message and in this way he warned them [1][14].

The other example can be seen when Histiaeus wanted to tell his allies the date of the revolution against their enemy. He shaved the heads of his servants who trust him and then a message in the form of a tattoo was written on their heads, after waiting for the time for the hair to grow back again, he then sent them to his allies across enemy territory without suspicion. As the shave appeared to ordinary travelers they reached

safely without any problem .When they met the leader of the allies they then revealed the messages by shaving their heads.

Steganography has evolved over time after it has the use of advanced techniques such as using of invisible ink to write messages, using certain substances to make the message disappear. Later the hidden message can only be detected using chemical reactions or even by using the heat. Carrier pigeons have been used to transport messages in the past as well that were in the form of microfilm after progress in photography [3] .This was followed by new developments in this area where lenses were used and films that provided the ability to decrease the size of the hidden messages. This technique was used in the World War II by the Germans, known as microdot [3]. Today the communications has increasingly evolved, as now the use of the electronic digital multimedia (such as audio, video and images) is means that steganography now is greatly used in computer files with digital data as the carrier.

1.3 Steganography vs Cryptography

The term Steganography (Figure 2) means “cover writing” while Cryptography (Figure 1) means “concealed writing”. Cryptography is the procedure of sending the message in distinctive forms so that only the involved people can demystify the message and then read it. The message that is sent without encryption is called the plaintext whereas the enciphered message is labeled as ciphertext. The procedure for changing the plaintext to a ciphertext is called encryption while the reverse operation (i.e., alteration of encrypted text to a plaintext) is called decryption. Encryption protects the contents of the message by encrypted it during transmission of data from the sender to the receiver. However, when the receiver receives the message he decrypts it and ensures from its integrity

Steganography works to conceal the message in plain view inside the data instead of encrypting it and does not require sending confidentially [3].

At the present time, steganography works on digital media as cover image and embedding digital media as secret message, the example for the used digital media are .wav, .gif, .bmp, .jpeg, .txt, .mp3, and .doc. Steganography is thought to be one of the most essential techniques to the future of the Internet in terms of privacy and security. The importance of steganography is highlighted because of the weakness in the encryption process and the desire to get the secrecy in the open systems. Lots of

governments have made laws in an effort to decrease the strength of encryption systems or completely prevented them; this may create unfortunately weak and breakable encryption algorithms in the Internet community [4]. Hence, the of steganography appears more than ever significant where the hidden message inside another file can be detected and read only by the involved entities or individuals and no one has the ability to read the message even with the knowledge of its existence. However, encryption and steganography do not provide the desired privacy and confidentiality, but that can be accomplished by utilizing both technologies to provide acceptable limits of privacy and confidentiality of anyone connected to the open systems [4].

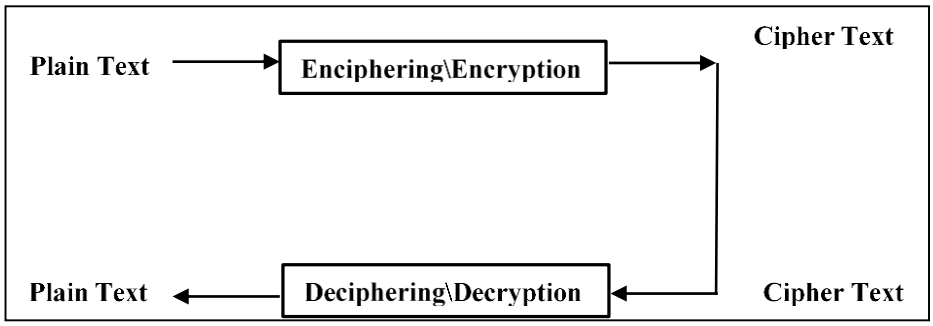


Figure 1: Cryptography

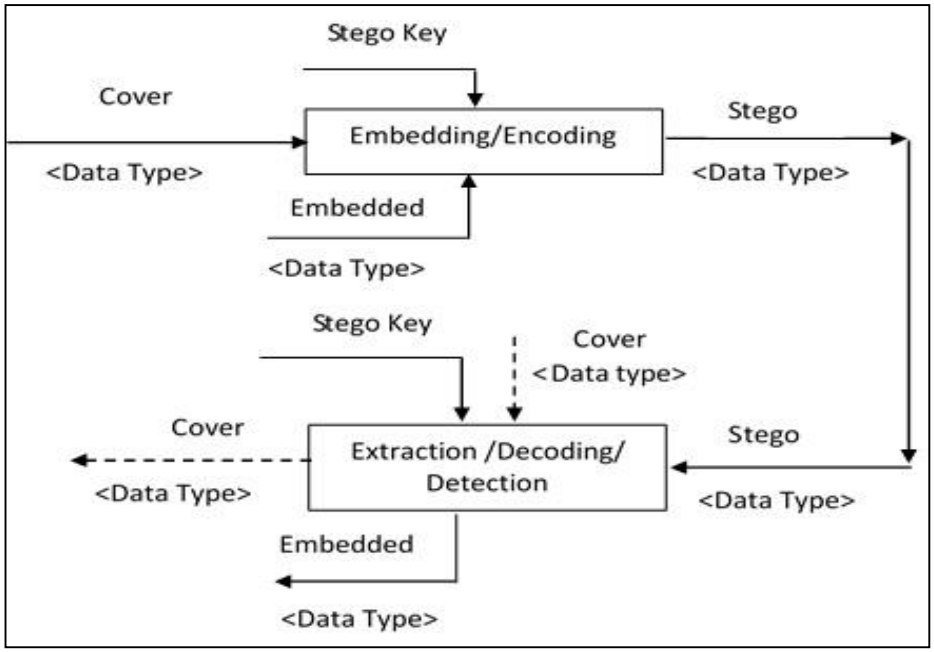


Figure 2: Steganography

1.4 The Steganography Criteria

In the application of steganography in a file, there are several criteria that must be considered. Those criteria are as follows:

1. Imperceptibility: The existence of the secret message cannot be perceived by human senses, both auditory and visual senses. For example, if the cover as a form of images, then making stego- images message insertion is difficult to distinguish by eye with its cover image. If cover is in the form audio (e.g., files mp3, wav, midi, and so on), then the ear senses cannot detect change to its audio stego.

2. Fidelity: Quality media reservoir has not changed much as a result of insertion. Such changes cannot be perceived by the senses. For example if cover is images, then creating insertion message in stego-image is difficult to distinguish by eye with its cover image. If its cover is a form of an audio, then the audio stego does not damaged ears and senses cannot detect such changes.

3. Recovery: The message that is hidden must be revealed back. Steganography is the purpose of hiding information, then at any time the hidden information must be taken back to be able to use more appropriate needs.

1.5 The Basic Model of Steganography

We will name the existing model in Figure 3 as the embedding model and is based on the discussion outcomes at the Workshop of Hiding Information in Cambridge [6]. The original data that have been entered called $\boldsymbol{cover}$ whereas data that will be concealed inside $\boldsymbol{cover}$ using the function $\boldsymbol{f_E}$ called $\boldsymbol{emb}$. The resulting data as the result of hiding process is called $\boldsymbol{stego}$. Later, the hidden data $\boldsymbol{emp}$* will be extracted by producing an output $\boldsymbol{cover}$* with the usage of the operation $\boldsymbol{f_E^{-1}}$. Practically, the extracted data that named $\boldsymbol{emp}$* should be equal to the data that has been hidden which is named as $\boldsymbol{emp}$ in the model. The participants in the workshop were not intent to design a model for evaluating the security of steganography systems. It was regarded just as a trial of including the dedicated knowledge in the process of information hiding in a more abstract way.

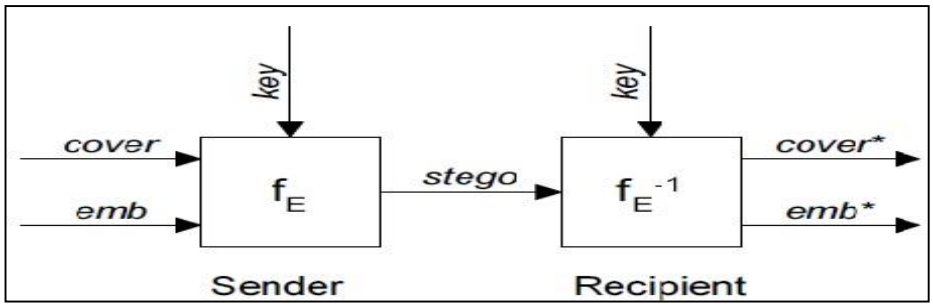


Figure 3: The Embedding Model of Steganography

f_E:	Embedding function of steganography
f_E^{-1}:	Extracting function of steganography
cover:	The embedded message (emp) will be covered inside it
emp:	The embedded message
key:	The parameter of f_E
stego:	The embedded message in a cover data

Therefore, the existing model in Figure 3 should not be used if you want to assess the security systems of steganography because there are not any explanations on the behavior of the $\boldsymbol{f_E}$ function nor the proficiency or knowledge possessed by the attacker.

1.6 Steganography Types

There are many techniques which can be applied for concealing information within the multimedia objects, with audio files, image or video object cover depending on the object type given below. Here we will discuss various methods and techniques that are frequently applied in text, image, audio or video of steganography.

1.6.1 Text

The procedure of hiding a text inside text can be accomplished by changing the text format or by changing certain properties in text elements such as letters. The goal of designing coding methods is to initiate changes that are unbreakable even if they contain noise [5][8].

These measures are not decodable reliably and visual changes in them are at a minimum level. File format or document which are on a computer file format describes

the contents of the document, and page layout or format using standard description languages like TeX, @off, PostScript2, etc.

Three following coding techniques describe a different approach instead of the form:

- **Line-Shift Coding:** This method is used for encoding the document by converting text lines positions perpendicular to encode document uniquely. The code that is embedded can be taken out later from the bitmap or the format file. In some specific cases this decoding system can be obtained without the requirement of the inventive image, since the original one is recognized to have constant line spacing between contiguous lines within a paragraph.
- **Word-Shift Coding:** This procedure is utilized to encode the document by converting text lines in a horizontal location to uniquely encode the document. This embedding procedure can be put on to either the bit map or to the file format of the page image. Decoding can be applied from the bit map or from the file format. This approach may be applied to documents that contain spaces between the neighboring variables. Variable space in text documents is frequently used to allocate white space while moderating text.
- **Feature Coding:** This coding procedure can be put on the bit map image or to the format file of the document. Images are then being tested depending on the features of the chosen text and these features are then changed first depending on the term. Decoding needs the actual image, or additional one, a measurement of the change in pixels at a feature. As there are many probable selections of text features; here, we select to alter upwards, vertical end lines - which is the tops of letters, b, d, h, etc. These ones i.e. end lines are changed by shortening or extending their distances by one (or additional) pixels, however otherwise these do not change the end line feature [5]. There are other forms of steganography text which are defined by Chapman et al. as the way to use the natural written language to conceal the undisclosed message [7]. Due to this variable spacing, decoding needs the initial image - or more specifically, the spacing between words in the un-encoded document[8].

1.6.2 Audio

The process of concealing information is computer-based audio steganography system, secret messages are embedded in digital sound [14]. Here a message is hidden

in the mode of a simple change in the binary sequence of an audio file. The existing system can hide messages in form of many files like AU, Wav, Mp3, etc. [6]. Concealing the message using this process is relatively harder than hiding the message using other media such as a digital image. There are several techniques and methods that are used for the very purpose of hiding information in this method, these methods come in the range from simple to most difficult that simply hide information in a pattern of a noise in audio file but also there is sophisticated methods more accurate working on advanced signal processing for concealing information technologies.

These methods range from the extent of simplicity and complexity not only hide text within the audio file in the form of noise but also there is sophisticated methods more accurate working on advanced signal processing for concealing information technologies.

The following is a listing of ways that are commonly chosen for the reason of audio steganography [6]:

- **Low bit encoding:** This procedure is somewhat as same as the LSB which is generally applied in the photos. This approach is often noted by the human ear so it is risky to use.
- **Spread spectrum:** This method works on adding a random voice to the information signal and that voice is spread all over the spectrum.
- **Echo data hiding:** This method uses the echoes in audio files and adds extra voice to that echo.
- **Differential phase variation**: In this method, the file is distributed into blocks using the embedded message.

1.6.3 Image

To embed specific messages or text inside an image that requires encrypting all message bits or inserting noisy areas which brings less attention so that we could hide our specific message within the region of huge natural colors variations [9]. It is equally possible that the hidden message is randomly distributed to various sections of the image. There are numerous methods that are utilized for the reason of hiding information within the image [9]:

1.6.3.1 Spatial Domain Techniques

- **Least significant bit insertion (LSB):** LSB is substitution method that uses specific k LSBs in each pixel to hide a secret message [10]. It is thought as one of the easiest ways to conceal a secret image in a specific image. Nevertheless, it is not difficult to disclose a stego-image that is created by using the LSB insertion technique.
- **Bit-Plane Complexity Segmentation**: The structural views can be incorporated into Spatial Domain techniques. This technique is first introduced by RO Eason and Eiji Kawaguchi in 1997 is a steganographic technique that utilizes the computation complexity of each bit plane to insert data. This technique is done by dividing the carrier medium into segments where each segment is measured according to defined threshold. Insertion of data is done on a segment that has a high complexity (noise like regions) [11]. Unlike the case with LSB insertion technique that uses only 1 bit low, with the data BPCS technique not only pasted on the LSB, but also the entire bit plane.

1.6.3.2 Transform Domain Techniques

If we involve information in the spatial domain method this information may be exposed to loss if it under goes any of the image processing techniques such as compression. To overawe this problem, we secure the information in the domain of frequency so that the secret information is inserted in the frequency values whereas the high frequency part is deleted. First, we put the conversion process on image and the data that is intended to be hidden, then the conversion coefficients values change accordingly. There are basically three transformation techniques [10]:

- **Fast Fourier Transform (FFT) Steganography Technique:** This procedure uses two dimensional of FFT. First, it converts the cover image to the transform domain and then the hidden bits are included on significant coefficients. Also, this technique includes complex mathematical formulas. Therefore, it includes more mathematical processes and requires more time than required for DCT steganography.
- **Discrete Cosine Transform (DCT) Steganography Technique:** This technique is applied to transform the two dimensional of the cover image, it

can also be derived from the FFT method but it requires less computational operations from FFT technique because it works only with the real values.

- **Discrete Wavelet Transform (DWT) steganography Technique:** This method works on the separation of high frequency components from the low frequency components and then replaces the high frequency part by the confidential data. The embedding amplitude of this procedure is far higher than the DCT steganography [23].

1.6.4 Video

The video file is a collection of concealed images with added voices to it, here the intended information is hidden in form of displayed images using image concealing techniques such as DCT[20].

1.7 Image Steganography

1.7.1 Image File

The digital image which is a rectangular table points or picture elements arranged in **M** rows and **N** columns. The symbol **M*N** is called the resolution of the image (although sometimes the term is applied for referring to the quantity of pixels per unit length of the image). Image points called pixels [13].

1.7.2 Image Types

The images can be divided according to the following [13]:

- **Two-level (or monochromatic) image:** In this case, all the pixels can have only two values, which are commonly called, black (binary unit or base color) and white (binary zero or a background color). Each pixel of the image is depicted by one bit, which is the simplest type of image.
- **Halftone images**: Each pixel of the image can have $\mathbf{2^n}$ values between 0 and $\mathbf{2^n - 1}$ denoting one of the $\mathbf{2^n}$ shades of gray (or other) color. The number $\boldsymbol{n}$ is usually comparable with the size of bytes, i.e., it is equal to 4, 8, 12, 16, 24 or more fold 4 or 8. The most important bits of the plurality of pixels form the most important bits of the image bit plane, or layer. Hence, the half-tone image with a scale of $\mathbf{2^n}$ levels is composed of $\boldsymbol{n}$ bit planes.

- **The color image:** There are several methods of specifying colors. Each color pixel comprises of three parts is called byte. Typical color models are RGB, HLS, and CMYK.
- **The image with continuous tone:** This kind of image can have many similar colors (or semitones). When adjacent pixels differ, it is nearly to determine the color only by the eye. As a result this kind of images may contain areas in which the eye color seems ever-changing. In this case, the pixel is depicted as (halftone or a color image). Continuous tone image can be understood from the process of making a photo from a camera, so before shooting the picture, that time period is called the continuous tone image.
- **Discrete-tone graphic:** (it is also called synthetic). Normally, in this case, the image is obtained by artificial means. It may be consisting only some colors or many colors, but there is no noise and spots of natural images. Examples of these images can provide pictures of artificial objects, machinery or equipment, a page of text, maps, pictures or images on a computer screen. (Not every artificial image necessarily has discrete tones. Computer generated image that should look natural, has a continuous tone, despite its artificial origin.) Artificial objects, texts, drawn lines have the mode of well-defined borders. They contrast sharply against the background of the rest of the image (background). Adjacent pixels in discrete-tone image are often single or strongly change their values. Such images do not compress data loss methods, as the distortion of only some pixels letters make it unreadable, transforms familiar mark completely indistinguishable. Here it is important to note that the discrete-tone images, usually carry a greater redundancy.
- **Images like cartoons:** This cartoon image is in which there are bigger areas of one color. At the same time adjoining areas, possibly are very different in its color.

1.7.3 Embedding/Insertion a Message into an Image

1.7.3.1 Spatial Domain

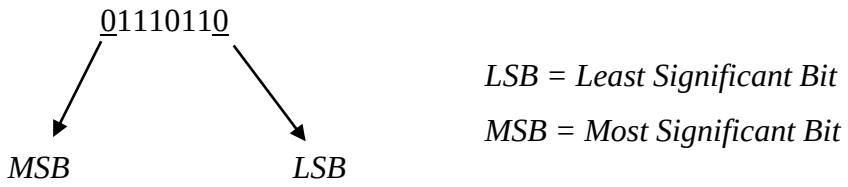
Spatial Domain techniques embed secret data pixel directly into cover image. The most popular data hiding method is, changing pixels' right most digits or last two, known as

(Least Significant Bit) LSB. Lossless image formats like .bmp, .png, and 8-bit gray-scale .gif are usable for LSB methods.

1.7.3.1.1 Least Significant Bit (LSB)

Any information within a digital system is represented in the form of bit sequences. So a value, for example the decimal number 54, represents binary notation positional as follows:

$$0 * 2^7 + 0 * 2^6 + 1 * 2^5 + 1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0 = 00110110$$



The bits associated with the base 2 with smallest exponent, are said least significant bits, since a change in their affects with a minimum change in the value represents the byte, while the bits associated with the base 2 with exponent larger are said to be the most significant bits for the opposite reason [15].

For example, changing only the least significant bit of the binary sequence 00110110 (decimal 54) will get the new sequence 0011011**1** corresponding to the decimal number 55 that will not change the physical appearance of the image; instead of changing the most significant bit, then the leftmost, the corresponding sequence will be **1**0110110, which is the number 182 in decimal [12].

As you can see, despite it performed changes only one bit in both cases, the result obtained is extremely different, with a slight change in the first compared to a substantial change in the second, for which the value of the information that the bytes represent after a change depend on an exponential relation with respect to the position of the bit under consideration. On this consideration many of the steganography algorithms are based on LSB to post confidential information within cover image, because these changes do not involve substantial losses in quality of cover image, which are not perceptible to the human visual system and can be attributed to noise, since it acts mainly on the least significant bits. Therefore, in most cases, the stego-resulting image is not distinguishable to the naked eye than the original one (cover

image), and it is still difficult to determine if any loss of quality detected are indeed attributable to the presence of a secret message inside, or rather linked to the processes of acquisition and image processing original. The mode of insertion of the data in the least significant bits varies depending on the use as a 24-bit or an 8-bit cover image. In the image consisting of 8-bit for example gray image, one bit of information is hidden in each pixel, these bits are simply overwritten consecutively with the message bits in the LSB method. The block image example of 12 pixels of this integration is shown in Figure 4 [15][21].

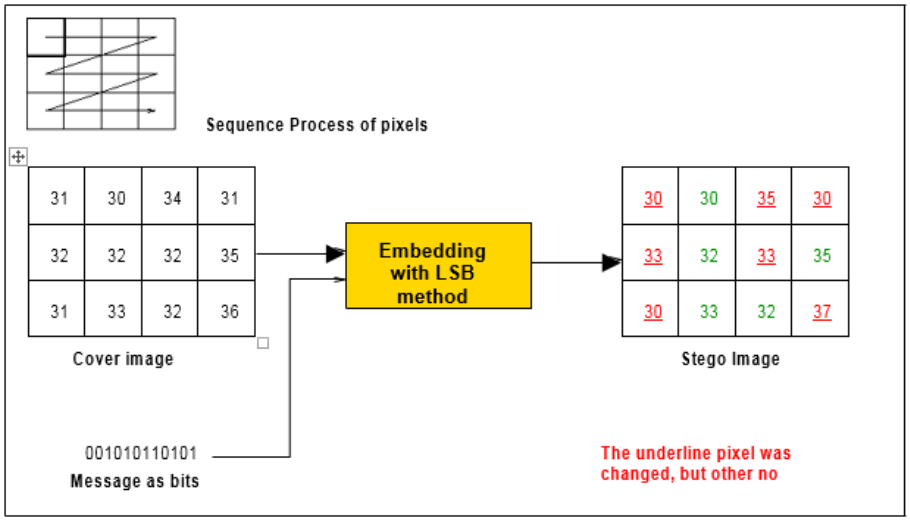


Figure 4: Embedding With the LSB Method

The maximum change in the gray level is in this method one, that when embedding a zero in an odd gray value whose amount is reduced by one, whereas by increased embedding a one in a straight gray scale its value by one, but embedding a zero in a straight gray value or embedding a one in an odd gray scale does not lead to change.

Within an image with depth of 24 bit color (8 bits for each primary color, Red, Green and Blue), the change of the last two bits in the bytes representing the each color of a pixel, it does not make significant changes to the color of the pixel itself, however, perceptible to the human eye, the images will be presented as a whole with the same characteristics. Assume that the original three pixels are as follows: [16]

(11101010 11101000 11001011) (R=234, G=232, B=203)

(01100110 11001010 11101000) (R=102, G=202, B=232)

(11001001 00100101 11101001) (R=201, G=37, B=233)

If we want to hide the alphabetic “J” that has the location 74 in the ASCII standard and has the follows binary representation “01001010”, by altering the channel bits of pixels.

(1110101**0** 1110100***1*** 1100101***0***) (R=234, G=**233**, B=**202**)

(0110011**0** 1100101***1*** 1110100**0**) (R=102, G=**203**, B=232)

(1100100**1** 0010010***0***11101001) (R=201, G=**36**, B=233)

In this situation, just 4 bits need to be changed to successfully embed the alphabetic. However, the result change that was conducted by using the LSB algorithm is very minute and therefore cannot be noticed by the human eye that means the message has been embedded well. The benefit of LSB algorithm, which is illustrated in Figure 4, is important because of its simplicity which is why it is so extensively used by many techniques [25].

1.7.3.1.1.1 Embedding Steps:

Step1: Read image and message.

Step2: Change the message to bits stream.

Step3: Decompose pixel into RGB values (one byte for R, one for G, one for B).

Step4: Replace the red LSB value with a message bit (it will only be one or zero)

Replace the LSB of green. Place this by the next highest bit.

Replace the LSB of blue. Place this by the next bit down.

Step5: Read the next pixel and decompose into RGB bytes.

Step6: Supplement the characters of the text file in each first section of next pixels by substituting it.

Step7: Continue iterating over pixels and their RGB values until all bit of message are inserted.

Step8: Write stego image.

1.7.3.1.1.2 Extraction Steps:

Step1: Read stego image.

Step2: Repeat over pixels.

Step3: Decompose pixel into RGB values (one byte for R, one for G, one for B)

Step4: Read LSB value from red then put it in the Vector

Get the LSB from green. Put it in the next position of vector

Get the LSB from blue. Put it in the next position of vector

Step5: Read the next pixel and decompose into RGB bytes.

Step6: Repeat step 4 and 5 until all pixels are read.

Step7: Change vector into array of 8 bits.

Step8: Convert each 8 bits into character.

Step9: Write message

1.7.3.1.2 Bit Plane Complexity Segmentation (BPCS)

The BPCS technique was invented by Eiji Kawaguchi and Richard O. Easonin [18] in 1998 in order to overcome the deficiencies or defects in the Least Significant Bit (LSB) that rely on manipulation technique. Whereas LSB works better with color and grayscale images, therefore, within its capabilities, which is severely crippled by limitation of capacity. The BPCS works on the idea that the highest bit value can be utilized to conceal the information provided on what appears the complex areas.

1.7.3.1.2.1 Bit Plane

An image with a color depth of N bits may be represented by N-bit planes, each of which can be seen as a single binary image [20]. In particular, it may induce an order that varies from the Most Significant Bit (MSB) to Least Significant Bit (LSB) The plans of the most significant bits contain information on the structure of the image, while gradually less significant provide smaller and smaller details . Note that only plans 7-3 contain significant data from the visual point of view, the image noise and capture errors are more noticeable in the bottom levels as can be seen in figure 5 and figure 6.

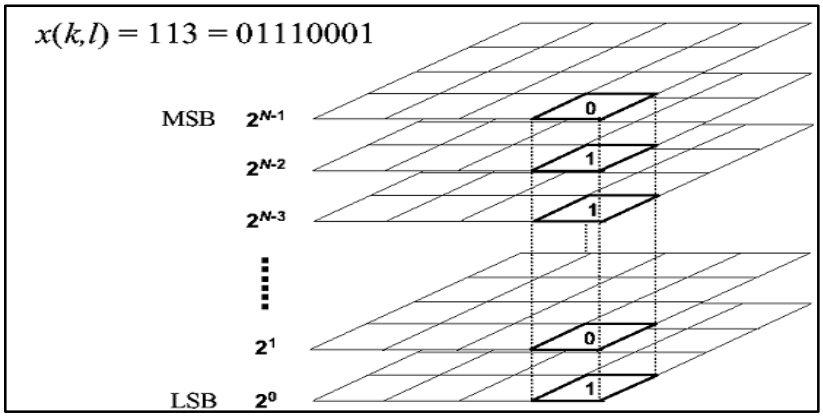


Figure 5: Bit Plane Levels

If the image of P consists of three colors, red, green, blue, then it can be shown

P = (PR1, PR2... PRN, PG1, PG2 ... PGN, PB1, PB2... PBN)

PRI: bit-plane to-i to red

PGI: bit-plane for the i-th green

PBI: bit-plane for the i-blue.

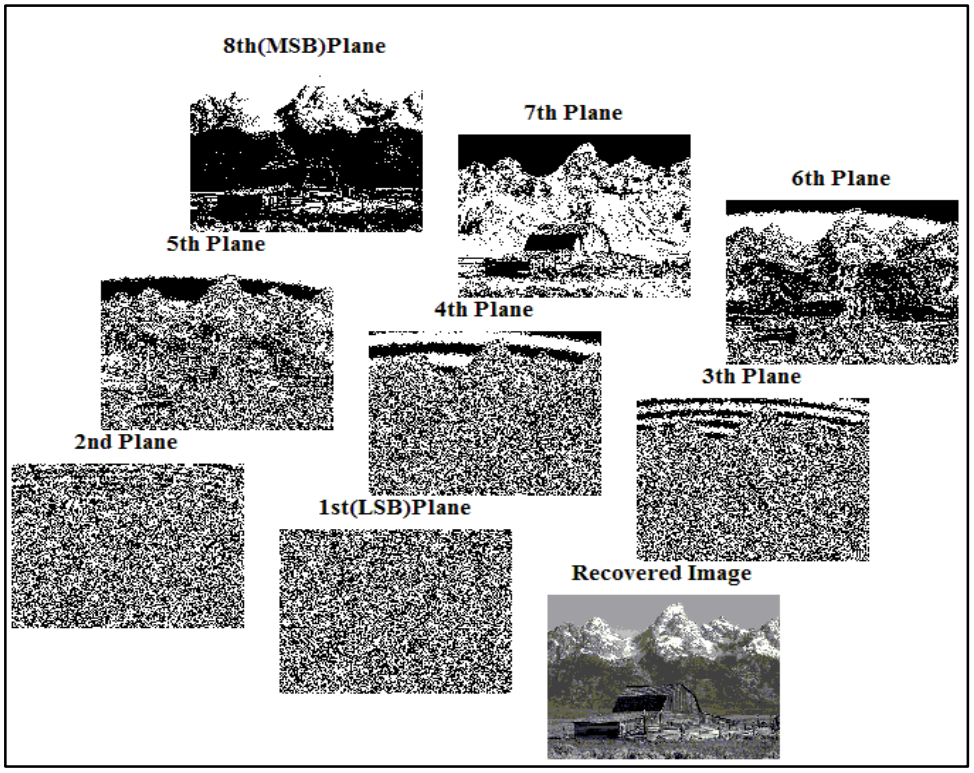


Figure 6: Bit Plane Levels

1.7.3.1.2.2 Binary Gray Code

Pure Binary Code (PBC) on the bit-plane provides the area greater for insertion. But PBC may have "Hamming Cliff" problems in which by small changes in color affects many bits of a color value [23]. Consider the 8-bit images, there are two consecutive pixels intensity values 127 and 128 accordingly. In PBC, 127 and 128 are indicated as 01111111 and 10000000. Both 128 represented as pixels seems to be identical to the human eye but very different in bit representation. This is named as the "Hamming Cliff" concept. If the embedded secret data, then there could be a possibility that could be 11111111, 01111111, 10000000 or 00000000 as in figure 7. There is a difference in the intensity of the gray level that is ignored by the human eye, after embedding the gray level difference is the 255 that is a black pixel appears dark while the other looks pure white pixels. These changes are easily visible to the human eye. This weakness is avoided by Canonical Gray Coding (CGC).

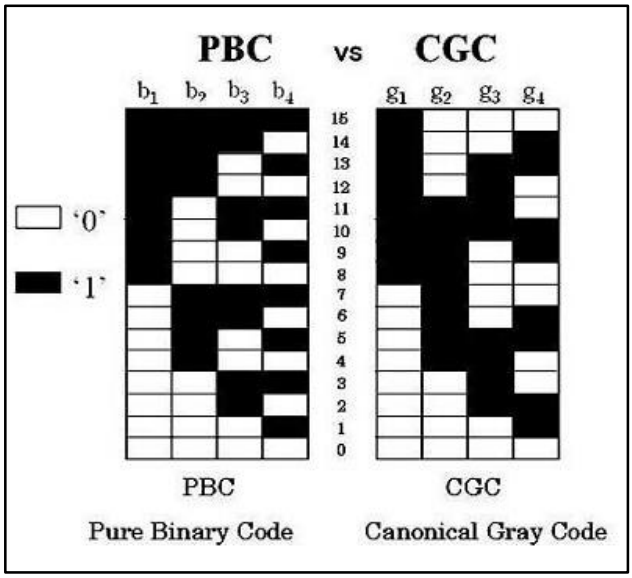


Figure 7: PBC and CGC system

Thereby, 127 is represented in binary form as 01111111 Current is represented as 01000000 in the CGC. Similarly, 128 is represented in the CGC as 11000000. Now, two pixels look the same, but differ only by one bit. It is exactly opposite to the PBC. So, CGC is not experiencing problems of "Hamming Cliff". After embedding, it can be 11000000 and 01000000 110000000 01000000 [16]. So, this change occurs within the pixels level of intensity that is not too flashy. Therefore, the CGC system is more worn than the PBC in BPCS system.

Here is the formula equation between the PBC and CGC of binary image (with $\otimes$ is Exclusive OR) the relation between b_i and g_i is as follows:

$$g_i = \begin{cases} b_0 & i = 0 \\ b_{i-1} \oplus & i > 0 \end{cases} \tag{1}$$

$$b_i = \begin{cases} g_0 & i = 0 \\ b_{i-1} \oplus g_i = g_0 \oplus \ldots \oplus g_i & i > 0 \end{cases} \tag{2}$$

Pixel values can be converted pure binary coding into pure binary converted coding.

1.7.3.1.2.3 Complexity Measure

Basic step one in steganography type of BPCS is to find areas in the cover image that can hide data gradually. There is no comprehensive definition available of complications or areas in the image. However, there are two definitions for measuring the different complexities by Niimi and Kawaguchi, One of them depends on the quantity of related areas that can be made use for find complex areas in the image, and the other rely on the length of the white and black border [4].

1.7.3.2.5.3.1 Measurement of Complexity Based on Black and White Border

This procedure works depending on the four border-related to neighboring each pixel. The total white and black border in the binary image is the sum of color switches along the columns and rows in the image [22]. To illustrate more let's take the given example, Figure: 8 which contains five pixels, white pixel encircled by four black pixels, from which we conclude that there are four color changes. The total border length is 4.

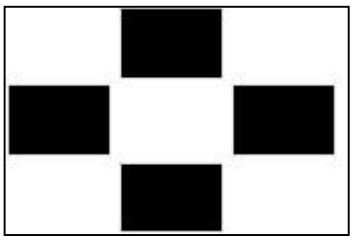


Figure 8: Four black pixels surrounding a White pixel

In Figure 9 (a), there is no color change across the columns and rows because it is made of only white pixels. Hence, the complete length of the border is zero. In Figure 9 (b), there are pixel wise black and white color changes. So, the total amount of color changes is, 24.Also it can be estimated by mathematical method to calculate the border.

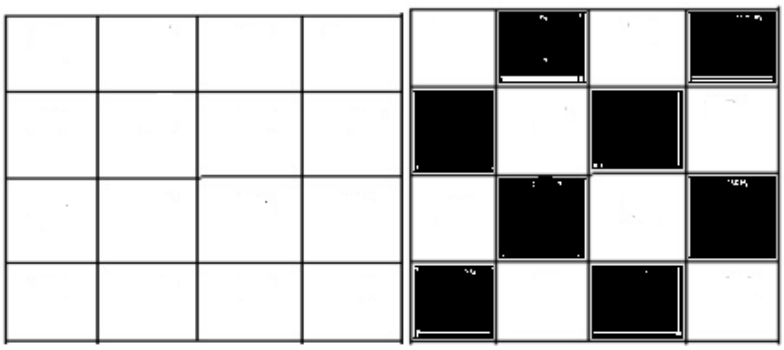


Figure 9: (a) all pixels in the image are white (b) black-white checker board

It can be explained how to calculate its borders by using a matrix of ones indicated to white color and zeroes indicated to black color ,which shown in figure 10 [22]

1	0	1	0	3
0	1	0	1	3
1	0	1	0	3
0	1	0	1	3
3	3	3	3	24

Figure 10: Calculate a summation of black and white borders

Hence, the formula to estimate the longest border limit for $(2^m x 2^m)$ binary image can be estimated by $2 * 2^m * (2^m - 1)$. So if we take Figure: 9 (b) it is $x\ 4 * 4, m = 2$ so for the longest border in the example of Figure 9 (b) is 24. It can be estimated in the following manner:

$$\alpha = \frac{k}{2 * 2^m * (2^m - 1)} \tag{3}$$

Here, where 'k' represents the total length, where α is the range between zero and one.

1.7.3.2.5.3.1 Measures of Complexity Based on Number of Close Areas, β

This procedure also rely on the principle of the four neighborhoods of a pixels [23].

$$\beta = \frac{m}{2^N * 2^N} \tag{4}$$

Where m here represents the quantity of connected areas in the $2^N * 2^N$ in the binary square image. Note that the ease of the β located at $[1/(2^N * 2^N),1]$ with maximum in the range that has been obtained for the checker board pattern and minimum that

has been obtained for the plain white or plain black image. Assuming that the image shaped is square of size $2^N * 2^N$, this measurement will suffer much of their applicability to all the images because the image is not often a square shape. To make this more comprehensive measurement it can be adapted to every size of the block of dimension $2^n * 2^n$.Where n here represent a range between 2 and 4 to any image of dimension $M * N$ on the condition that M and Nmust be dividable by 2^n and n values here are very large. If the n values are very small$(n = 1, 2)$, it provides very large spatial localization in regard to make the measurement complexity significantly meaningful. Example, if n = 3, therefore the measure complexity is utilized to every block of size $8 * 8$ exclusively in the image [23].

Figure 11 (a) and (b) shows the values of the α and β with two blocks templates with a size of $8 * 8$. Practically, we discover that the measure α greatly uniform or small at [0,1] while β measuring tends to possess a limited peak at 0.4 [2]. Figure 11 (c) shows the largest values for α and β that are obtained from the examining white and black border.

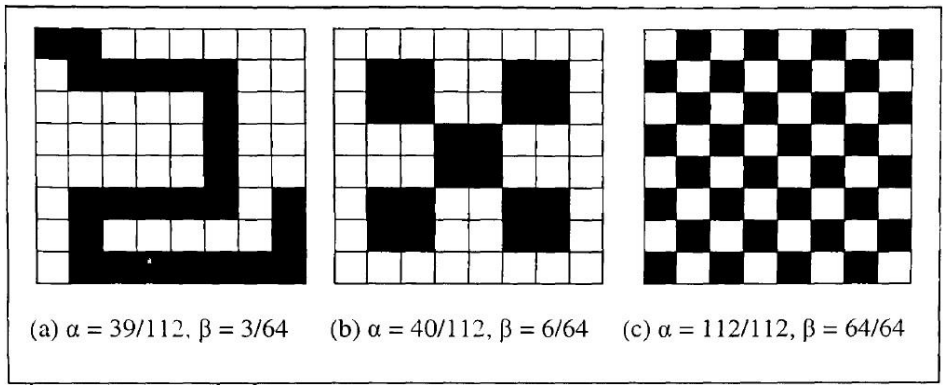


Figure 11: Values of α and β for some blocks of size 8×8

1.7.3.1.2.4 Resource Blocks and the Conjugation Operation

Once the image has been made of 24-bit it is divided into 24 bit-planes and the complexity α found for each of the exclusive blocks that has size of 8*8 for each 24 bit-planes, the complexity that belong to each block is compared then with the threshold α_0. If $\alpha > \alpha_0$, the block will be treated as complex enough and will be replaced by blocks of data. α_0 , generally uses this Standard value which is about 0.4 [29]. The complex blocks replaced by parts of data in the form of bit-planes called Resource Blocks or Data Blocks. The resource blocks is a parts of data obtained from ASCII file called resource file can be a Word document, text file or image. Each 8byte

block of a resource file forms an 8*8 resource block with the 8-bit binary representation of each byte which is forming the row of the 8*8 block. For example, a sequence of 8 characters from an MS word document 'This one' (8 characters includes the blank space) would form an 8x8 resource block as shown in Figure: 12 (a) and a block of 8 consecutive blank spaces would form a block as shown in Figure: 12 (b) where, 'T' is 01010100, 'h' is 01101000, blank space is 00100000 (32 in ASCII) and so on [23].

0	1	0	1	0	1	0	0	T
0	1	1	0	1	0	0	0	h
0	1	1	0	1	0	0	1	i
0	1	1	1	0	0	1	1	s
0	0	1	0	0	0	0	0	
0	1	1	0	1	1	1	1	o
0	1	1	0	1	1	1	0	n
0	1	1	0	0	1	0	1	e

(a)

0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0

(b)

Figure 12: Resource blocks of 2 different 8-character

After the resource file is split into a set of pieces each piece is equivalent to 8 byte and pitched to a binary resource blocks that consisting of 8*8, they are ready to replace the complex blocks in the form of bit-planes but there can be one problem; complex block represents the block that show noise and altering of this block will not be felt except if the altering results in the formation of the block is less complicated than the value of the noisy region of the threshold limit: α_0. Consider the block presented in Figure 12 (b) this block often resemble Word files blocks which contain many sections of empty spaces.

The complexity value of this block is 0.1429, which is much fewer than the value of α_0 that used normally. If this block has been change by a complex block in the mode of a bit-planes image, it shows a clear contradiction, particularly if it is in one of the upper order bit-planes. Similarly, the decoding unity will not distinguish the block also, supposing that only the complex blocks was replaced. Therefore, only the blocks that are complex have effective information. Thus, to overwhelm this problematic issue, the conjugation operation is presented.

Figure 11 (c) shows the most possible complex 8*8 blocks with possible amount value of 1. This block is pointed out as W_c and the value of its upper left is 1. The checker-

board pattern with a complexity value amount of 1, can form with the upper left value to be 0 and is indicated by: B_c. All the black and white blocks pointed out by B and W respectively. W_c is used for all future descriptions, while all of it would relate to $B_c$as well. TheW_c block has a distinctive property that when it is XORed (exclusive OR operation) with a non-complex block, P (say), of complexity say $\alpha_n < \alpha_0$, then the subsequent block, P*, has a complexity of $(1\alpha_n) > \alpha_0$. As with any XOR operation, the block P can be easily retrieved by XORing again with W_c. This operation of changing the complexity of a block by XORing with W_c is named the conjugation operation and is denoted b '*'. Figure 13(a) shows a non-complex (or simple) block, P (say), 13(b) is the perfectly complex block,W_c and Figure 13 (c) is the conjugated block, P*, obtained by XORing corresponding pixels in Figure 13(a) and (b) [23].

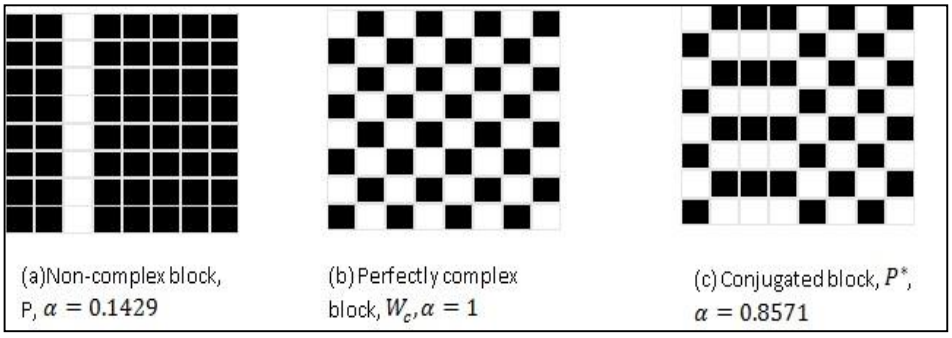


Figure 13: Explain the Conjugation Process

The 2 main factors of the conjugation process can be summed up as given below:

1) $\alpha(P*) = 1 - \alpha(P).$ (5)
2) $(P*) *= P.$ (6)

The first shows the property by the encoder to encrypt the uncomplicated resource blocks, whereas the second property is used by the decoder to recover the original block. It is possible that not all the resource blocks are complex. It is important to follow any blocks that have been conjugated.

All of this has been done with the help of conjugation map. On behalf of every 8 byte from the resource file one bit is added to the conjugation map to specify if the block has been conjugated. ‘1’ means that the resource block has been conjugated, while ‘0’ means that resource block has been inserted as it is. The conjugation map will embed after inserting the resource blocks wholly.

Again, it is probable that the conjugation map itself, when cast into 8x8 blocks, may not practice a complex block, and it becomes essential to identify a conjugation map from the conjugation map. There are two essential solutions to this problem:

1) Form a conjugated map to the conjugated map and include it in the some well-known complex sections in the image. Example, the LSB planes.
2) States the bites in the conjugation map in blocks of 63 bits (if necessary filling the last block with zero value).Cast them in block of 8x8, by making the value of the first bit (top left bit) zero. If the complex block is zero include it as it is, if it is not zero, then the block will be conjugated.

The conjugation operation will make the top-left bit as one automatically. The bit will be used by the decoder to realize that the block has been conjugated by the encoder.

This implementation customs the second method because it provides unlimited flexibility while dealing with images that are not essential to have complex LSB planes. In figure 14 the approximate flowchart explains the technique of BPCS.

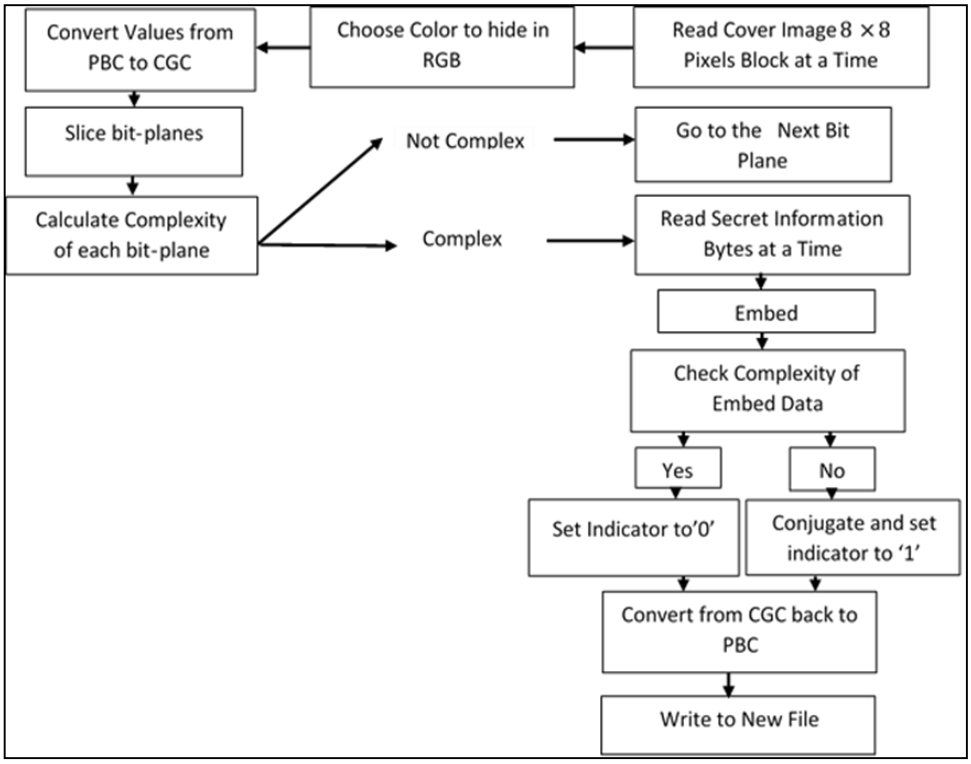


Figure 14: BPCS Technique

1.7.3.1.2.5 Embedding Steps:

Step 1: Read image and message.

Step 2: Decompose image into bit-planes. Then, convert it from pure binary code to Canonical Gray Coding system

Step 3: Segment all bit-planes into 8×8 blocks and put Each 8 characters of a message in a Block form.

Step 4: Using a Threshold to determine if the block is complex or non-complex for the image Blocks and the same threshold value is measured for each message block.

Step 5: Replace message blocks with image blocks, if complexity

Step 6: If it is not complex block, the message block is conjugated. Conjugation map is constructed from conjugated blocks.

Step 7: Convert all bit-planes into PBC code

Step 8: Write stego image

1.7.3.1.2.6 Extraction Steps:

Step 1: Read Stego image

Step 2: Decompose image into bit-planes. Then, convert it from pure binary code to Canonical Gray Coding system

Step 3: Segment all bit-planes into 8X8 blocks.

Step 4: Threshold to determine if the block is complex or non-complex for the image blocks.

Step 5: Extract message from complexity blocks of image.

Step 6: Extract conjugated blocks if necessary based on conjugation map information.

Step 7: write message.

1.7.3.2 Frequency Domain

French mathematician scientist named Joseph Fourier introduced the Fourier series at the beginning of 1980s which aims to represent the periodic signal time-continuous. The signal can be divided to a linear weighted sum of harmonically correlated complex exponentials. The sum presents the availability content of reference and is called spectrum. When the signal is not periodically its period becomes infinite. Image can be considered as a function of spatial degrees. This method (Fourier Transform) works to convert the image to a set of orthogonal functions, as well as to transform the spatial density of the image in its frequency range [21].

1.7.3.2.1 DCT

The Fourier series was formed originally while solving the problem of heat conduction and has a huge quantity of applications which underwent multiple changes out of those one later driven out of still some more which we want to mention here and which is Discrete Cosine Transform (DCT) [21]. Many of the algorithms goal is to compact the image and video files use the DCT algorithm to modify the image to its frequency domain and apply quantization for data compression. This helps to separate the image into parts or sub-spectral ranges while maintaining the image quality. One of the most well-known methods which employ a DCT procedure for image compression is the JPEG method [25].

The Fourier transform core is composite valued. The DCT is acquired by using only a real part of the Fourier composite core. If $f(x, y)$ symbolizes the image in the spatial domain and $f(u, v)$ represents an image in the frequency domain, then the general equation for a 2D DCT is [13]:

$$f(u,v) = C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} F(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right) \tag{7}$$

Where if $u = v = 0, C(u) = C(v) = \sqrt{\frac{1}{N}}$; else,$C(u) = C(v) = \sqrt{\frac{2}{N}}$.

The inverse DCT can be characterized as

$$F(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)f(u,v)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right). \tag{8}$$

A more acceptable method for showing the 2D DCT is with matrix products as$F = MfM^T$, and its inverse DCT is$f = M^T FM$, where F and f are $8 * 8$ data matrices, and M is the matrix as [21].

$$M = \begin{bmatrix} \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} \\ \frac{1}{2}cos\frac{1}{16}\pi & \frac{1}{2}cos\frac{3}{16}\pi & \frac{1}{2}cos\frac{5}{16}\pi & \frac{1}{2}cos\frac{7}{16}\pi & \frac{1}{2}cos\frac{9}{16}\pi & \frac{1}{2}cos\frac{11}{16}\pi & \frac{1}{2}cos\frac{13}{16}\pi & \frac{1}{2}cos\frac{15}{16}\pi \\ \frac{1}{2}cos\frac{2}{16}\pi & \frac{1}{2}cos\frac{6}{16}\pi & \frac{1}{2}cos\frac{10}{16}\pi & \frac{1}{2}cos\frac{14}{16}\pi & \frac{1}{2}cos\frac{18}{16}\pi & \frac{1}{2}cos\frac{22}{16}\pi & \frac{1}{2}cos\frac{26}{16}\pi & \frac{1}{2}cos\frac{30}{16}\pi \\ \frac{1}{2}cos\frac{3}{16}\pi & \frac{1}{2}cos\frac{9}{16}\pi & \frac{1}{2}cos\frac{15}{16}\pi & \frac{1}{2}cos\frac{21}{16}\pi & \frac{1}{2}cos\frac{27}{16}\pi & \frac{1}{2}cos\frac{33}{16}\pi & \frac{1}{2}cos\frac{39}{16}\pi & \frac{1}{2}cos\frac{45}{16}\pi \\ \frac{1}{2}cos\frac{4}{16}\pi & \frac{1}{2}cos\frac{12}{16}\pi & \frac{1}{2}cos\frac{20}{16}\pi & \frac{1}{2}cos\frac{28}{16}\pi & \frac{1}{2}cos\frac{36}{16}\pi & \frac{1}{2}cos\frac{44}{16}\pi & \frac{1}{2}cos\frac{52}{16}\pi & \frac{1}{2}cos\frac{60}{16}\pi \\ \frac{1}{2}cos\frac{5}{16}\pi & \frac{1}{2}cos\frac{15}{16}\pi & \frac{1}{2}cos\frac{25}{16}\pi & \frac{1}{2}cos\frac{35}{16}\pi & \frac{1}{2}cos\frac{45}{16}\pi & \frac{1}{2}cos\frac{55}{16}\pi & \frac{1}{2}cos\frac{65}{16}\pi & \frac{1}{2}cos\frac{75}{16}\pi \\ \frac{1}{2}cos\frac{6}{16}\pi & \frac{1}{2}cos\frac{18}{16}\pi & \frac{1}{2}cos\frac{30}{16}\pi & \frac{1}{2}cos\frac{42}{16}\pi & \frac{1}{2}cos\frac{54}{16}\pi & \frac{1}{2}cos\frac{66}{16}\pi & \frac{1}{2}cos\frac{78}{16}\pi & \frac{1}{2}cos\frac{90}{16}\pi \\ \frac{1}{2}cos\frac{7}{16}\pi & \frac{1}{2}cos\frac{21}{16}\pi & \frac{1}{2}cos\frac{35}{16}\pi & \frac{1}{2}cos\frac{49}{16}\pi & \frac{1}{2}cos\frac{63}{16}\pi & \frac{1}{2}cos\frac{77}{16}\pi & \frac{1}{2}cos\frac{91}{16}\pi & \frac{1}{2}cos\frac{105}{16}\pi \end{bmatrix}$$

The DCT parameters that are obtained show the connection between the original $8 * 8$ block and the corresponding DCT base image.

1.7.3.2.2 Jpeg Compression

JPEG (Joint Photographic Experts Group) is a standard international compression technique that has been established to support various types of applications for continuous-tone images [25]. JPEG standard has two basic methods for the compression and that meet the unique needs of every application. The DCT-based method is states for lossy compression, and the predictive method states for lossless compression [13]. The simple lossy technique is called baseline that rely on DCT-based methods. The lossy technique is extensively applied today and it is considered as an efficient method for a wide variety of applications. In the next field, we will provide the JPEG standard and we will focus on the baseline method as shown in (Figure 14).

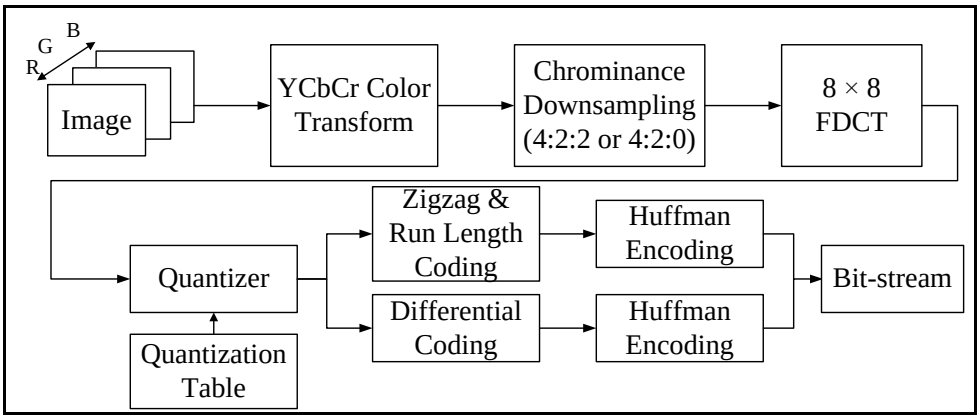


Figure 15: Baseline JPEG Encoder

JPEG coder is shown in Figure 15. JPEG encoder performs the following steps [21], [19], [20]

JPEG decoder performs the inverse steps to bring back the original image. However, the third step of encoding causes loss of data, so it is not possible to fully restore the image to its original form. The following briefly describes each of these steps coding [25].

1.7.3.2.2.1 RGB to YCbCr Conversion

RGB color is not an efficient way to store data because of mutual repetition between components. YCbCr is a practical estimate for perceptual uniformity and color processing, where the primary colors that conform roughly to Green, Blue and Red are being treated into perceptually useful information. By doing this, succeeding image/video processing, storage and transmission can do operations and present errors in perceptually meaningful methods. In the YCbCr color model the presence of a pixel is known by its brightness (Y), its blueness (Cb) and its redness (Cr). The Y element, called luminance, is weighted sum of R, G and B elements. The weight depends on human eye sensitivity as in Figure 16. The Cb and Cr components, called chrominance signify how much blue and red is in that color respectively [13].

According to CCIR Rec.601 [26], with little modifications in JPEG, color space conversion bases can be presented on three linear equations:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299000 & 0.587000 & 0.114000 \\ -0.168736 & -0.331264 & 0.500002 \\ 0.500000 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \qquad (9)$$

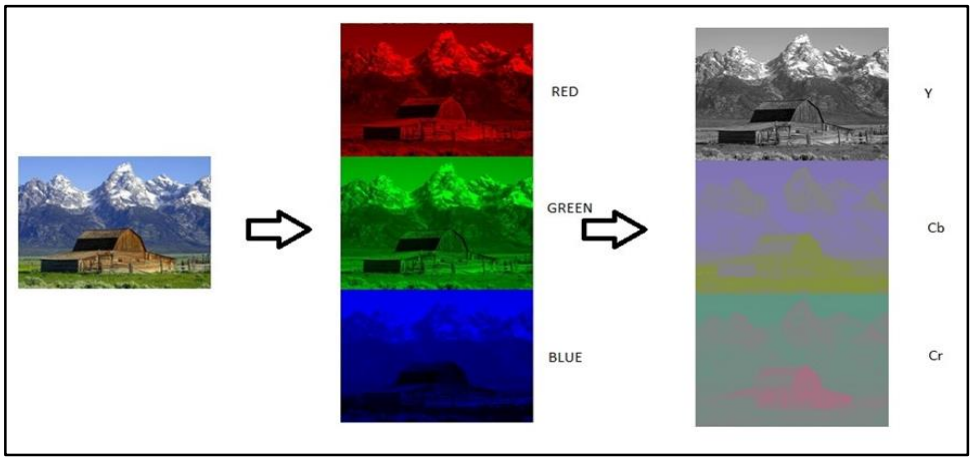


Figure 16: Color Transformation

1.7.3.2.2.2 Sampling

A human eye is less sensitive to the color image as compared to its sensitivity to the brightness of the image. Thus in YCbCr color space, Chroma components contain less details as compared to the details of Y components. Practically, the reduction of accuracy from Chroma components limits the amount of its data without any loss of optical quality perception. In the compression of JPEG Chroma data components are squeezed by taking samples.

- **4:2:0**

  The 4:2:0 sampling method, the chrominance components is reduced by taking the average value of each 2×2 block. That means for four samples of the luminance, there is just one sample of chrominance (Cb and Cr individually). Figure17, given below, shows the 4:2:0 sampling chart.

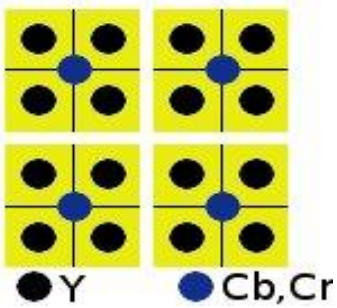


Figure 17: The 4:2:0 Sampling Diagram

- **4:2:2**

  In this sampling method, Chroma components are reduced in horizontal dimension only. That means there is just one chrominance sample (Cb, Cr individually) for two luminance samples. Figure18 shows the sampling method of 4:2:2.

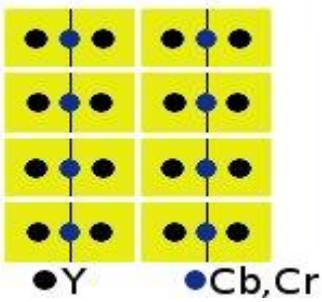


Figure 18:4:2:2 sampling method

- **Grayscale**

  This method works to remove chrominance data definitively. There is just one element of luminance.

1.7.3.2.2.3 Transformation

After averaging image elements, divide the original image into blocks of size $8 * 8$ and apply the 2D DCT transformation which is mentioned above to each of block image. DCT coefficients are calculated according to the formula $F = MfM^T$ which is the result shown in figure19 [13].

48	39	40	68	60	38	50	121
149	82	79	101	113	106	27	62
58	63	77	69	124	107	74	125
80	97	74	54	59	71	91	66
18	34	33	46	64	61	32	37
149	108	80	106	116	61	73	92
211	233	159	88	107	158	161	109
212	104	40	44	71	136	113	66

(a) f (x, y): 8×8 values of luminance

251	118	-13	6	-2	6	-1	0
279	-68	-8	-7	-1	4	-4	-1
-51	-14	34	-14	5	0	-1	0
27	5	-10	8	-7	4	-5	1
-22	-7	14	-9	4	-2	1	1
-3	15	-18	15	-6	2	-1	2
7	-9	6	-6	4	0	0	2
3	7	-9	3	0	-2	-1	0

(b) F (u, v): 8×8 DCT coefficients

Figure 19: An example of DCT coefficients for an 8x8 block

The results of a 64 elements DCT transform are 1 DC coefficient and 63 AC coefficients. The DC coefficient denotes to the average color of the 8×8block. The 63 AC coefficients characterize color change in a block crossways. (DC presents a constant voltage whereas AC voltage differs depending on a sinusoidal curve and these names i.e. DC and AC come from electrical engineering) In JPEG compression, AC and DC coefficients are encoded differently.

1.7.3.2.2.4 Quantization

It is a fundamental step in the process of compression. Here the feature of the representation in the frequency domain is disparate to what happened in the spatial domain before DCT where not each direction takes the similar significance for optical quality of image. Eliminating the components of high frequency shrink the level of point but the whole structure remains as it is, because it is dominated by components of lower-frequency.

The quantization can be defined by the following formula:

$$F_q(u,v) = Round\left(\frac{F(u,v)}{Q(u,v)}\right) \tag{10}$$

The standard of JPEG offers the example of the quantization table that has been used through good results of 8-bit of each section, luminance and chrominance images. The quantization table in standard of JPEG for luminance and chrominance components is given in figure 20(a and b) respectively:

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(a) Luminance quantization table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

(b) Chrominance quantization table

Figure 20: Quantization tables

Numerous encoders use the example of quantization tables, but the values are not always supposed to be perfect. Encoder could use another quantization table and may be it can be an optimal quantization table by first examining the image. An example of luminance quantization using default quantization table is shown in Figure 21.

32	6	-1	0	0	0	0	0
-1	-3	0	0	0	0	0	0
-1	0	1	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 21: After the Quantization Process

After the quantization process, most values of the equation will be equal to zero as showed in Figure 21.

1.7.3.2.2.5 DPCM (Differential Pulse Code Modulation)

Normally, the adjacent blocks with high values are similar therefore; the coefficients of DC are of the differential encoding using this technique as shown in Figure 21. This shows that the value of every DC coefficient is defined as the value of the previous DC coefficient and it records the two coefficients difference between each other. This is necessary to remember only the value of first DC coefficient and differences of neighboring DC coefficients. DC coefficients between the blocks of one another are not much different; this can be seen by looking at the following Figure 22 which will be described in more detail to understand this even better:

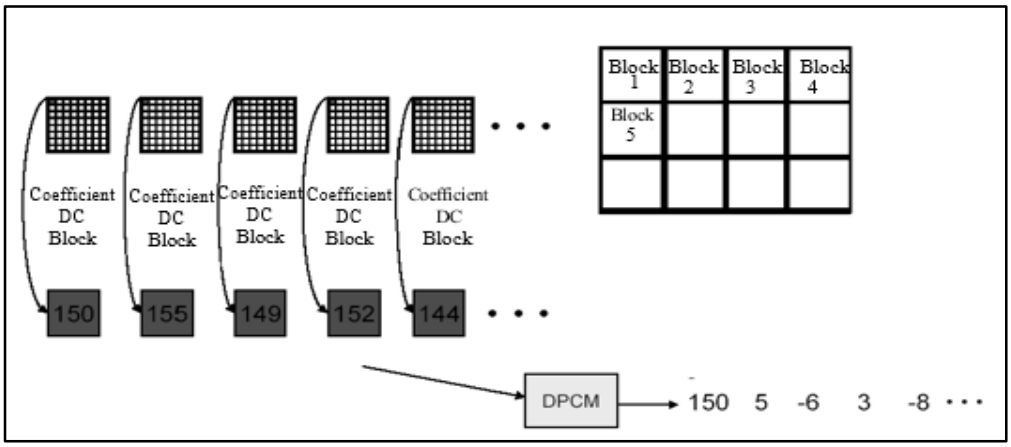


Figure 22: DPCM Process

1.7.3.2.2.6 Entropy Coding on the DC Coefficient

DC coefficients that have gone through a DPCM stage is then compressed by utilizing Huffman compression. Each code of DPCM that belong to DC is represented by Size and Amplitude, where size is the number of bits needed to represent the coefficient and amplitude represents the actual bits. For instance, the values in figure 22 that resulting from DPCM stage that are 150, 5, -6, 3 and -8 will be converted to (8, 10010110), (3, 101), (3, 001), (2, 11), (4, 0111)[25]. Moreover, Size represents the Huffman code because smaller Sizes are often appear whereas Amplitude is not Huffman code, its value can transformation extensively [25]. Thus Huffman coding has no appreciable benefit. The following table in Figure 23 demonstrates the relationship between size, amplitude and numbers:

SIZE	AMPLITUDE	NUMBER
1	0.1	-1.1
2	00,01,10,11	-3,-2,2,3
3	000,...,011,100,...,111	-7,...,-4,4,...,7
4	0000,...,0111,1000,...,1111	-15, ..., -8, 8, ..., 15
...	...	...
10	0000000000,..., 0111111111, 1000000000, ..., 1111111111	-1023,..., -512, 512,..., 1023

Figure 23: Relations Size, Amplitude and Number

As seen above that the amplitude can declare the absolute value of the numbers in DPCM, which means the amplitude can contain positive and negative numbers (in the form of one of its positive complement). In stage compression Entropy Huffman coding experiences only its size alone, because the change in size is not too far away, while the amplitude varies greatly. This is further described in the following Figure 24.

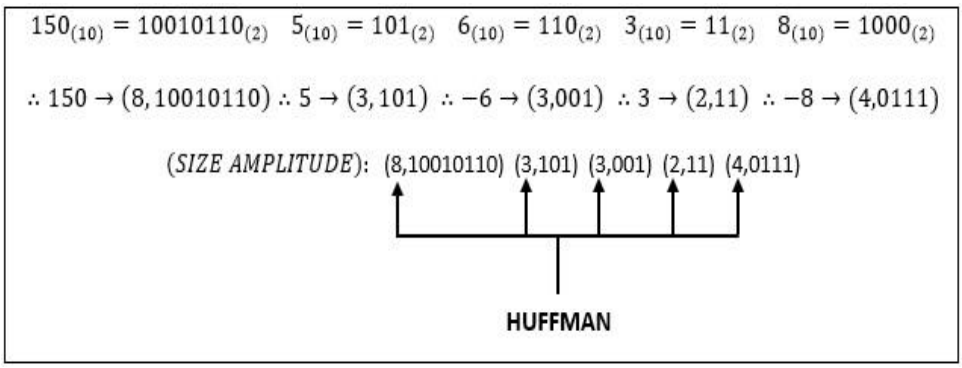


Figure 24: Entropy Coding On the DC Coefficient

1.7.3.2.2.7 RLC (Run Length Coding)

At this stage each AC coefficients together in each block enter the stage of Entropy Encoding. Here RLC technique is used because the AC coefficients have many sequential values, the order value average length is of zero average as seen in Figure: 20 which show an example of the block which has been quantized by DCT coefficients [19]. It demonstrates that the zero values adjacent from the left / top right / bottom, therefore, to make the value 0 more and more. The sequential orders of the coefficient changes in the form of a zig-zag as described in further detail in the following Figure 25:

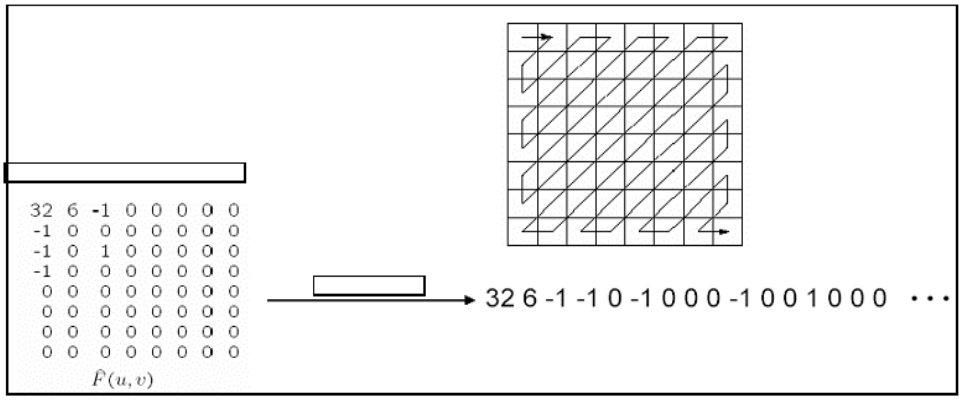


Figure 25: Zigzag Order Detour

After converting the value order, the value of AC transforms into two pairs (run-length, value), where run-length is the number of consecutive 0 and value is the value of nonzero located afterwards. In this case, the DC coefficient is not taken into consideration in the RLC Figure 26.

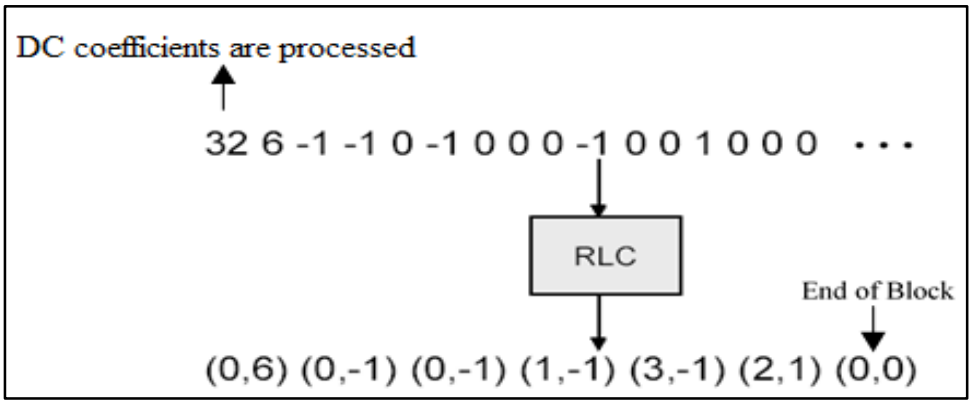


Figure 26: RCL Process

1.7.3.2.2.8 Entropy Coding the AC Coefficients

AC coefficients that have gone through the RLC stage after the compressed using Huffman compression, previously the two of rows (run-length, value) were changed into triple form (run-length, size, value), just the same as the DC coefficient . In this case, it suffered only run-length Huffman compression. To further describe this, it is shown in the following Figure 27.

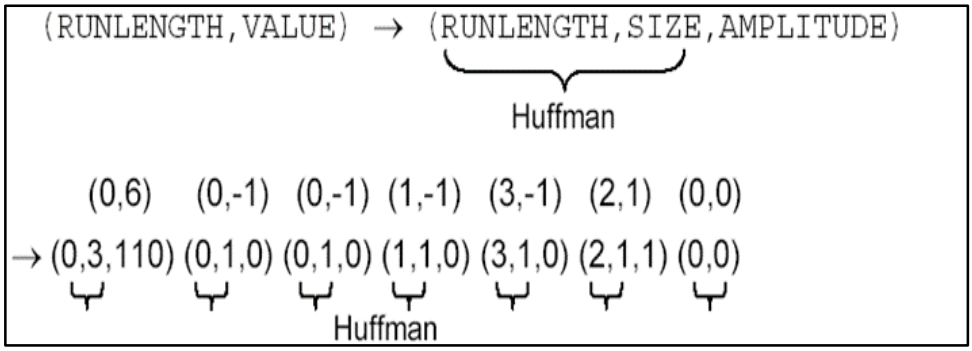


Figure 27: Entropy Coding the AC Coefficients

1.7.3.2.3 Hiding in DCT Domain

Stenographic methods that hide the message in the spatial domain are not suitable for working with image formats that use compression process with the loss of data. Therefore, the advanced steganography algorithm is adapted to images of JPEG format. The specificity of these algorithms is reflected in the fact that it operates in the DCT, which can be classified into stenographic techniques of transform domain. The principle of using the JPEG encoder to conceal a data, between step quantization and lossless compression is performed to hide secret messages. In other words, the message is concealed in the quantized DCT coefficients. Figure 28, 29 shows the sequence of steganography algorithm steps. In the process of retrieval, it is enough to use the Huffman decoder to form the DCT coefficients in which the message is hidden. Steganography algorithm is implemented as a part of the work to hide messages using only component that defines the brightness of the image such as Y component. Because of the encoding process given in the previous section, most of the capacity allocated to hide this message is just the Y component. Therefore, it do not lose much capacity if the Cb and Cr components are ignored in the hiding.

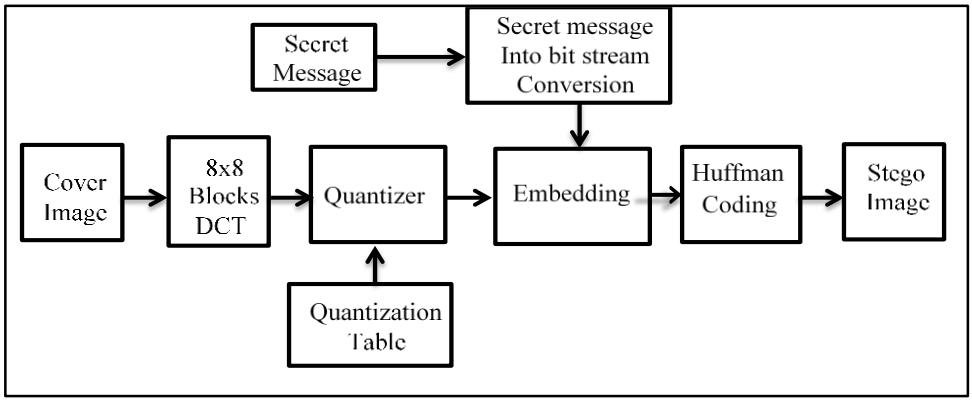


Figure 28: Block Diagram of Embedding Process of JPEG Steganography

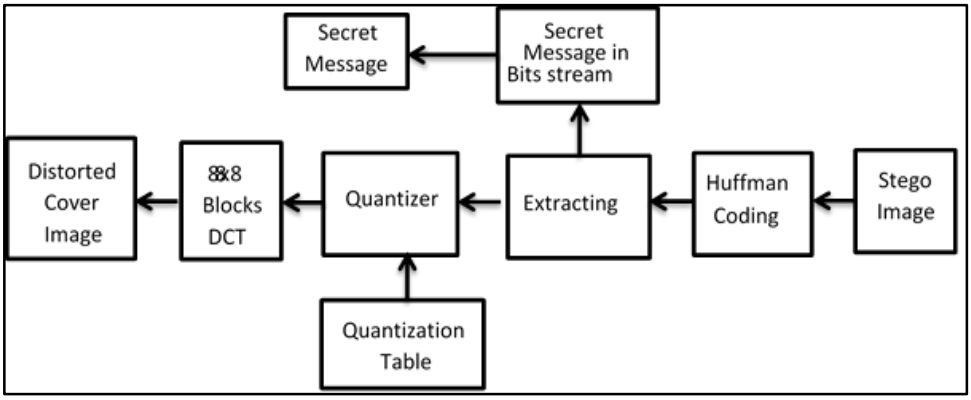


Figure 29: Block Diagram of Extraction Process of JPEG Steganography

1.7.3.2.4 Algorithms Based on LSB Substitution

To hide the message in the DCT domain, it can use LSB substitution method. The most steganography algorithms of this type are JSTEG and outguess.

1.7.3.2.4.1 JSTEG Algorithm

It is one of the simplest steganographic algorithms that work with JPEG images. In fact, JSTEG is LSB method that operates in the quantized DCT domain with time to avoid the DCT coefficients whose value is 0 or 1. The zeros related to the frequencies that are when ignored, the hidden message may be lost. The ones which are not used because of the process of LSB substitution can have the value 0. JSTEG algorithm sequentially goes by the DCT coefficients, without using steganographic key. In Figure 30 JSTEG algorithm is implemented in the context of work.

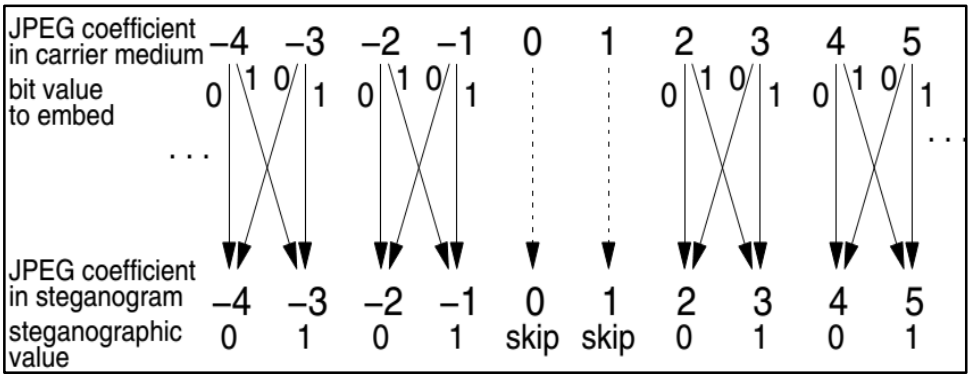


Figure 30: JSTEG Algorithm

1.7.3.2.5 F Group of Algorithms

F group steganographic algorithm is not similar to the LSB substitution method, but the insertion is done by changing the absolute values of the DCT coefficients. This group consists of F3, F4 and F5 algorithms respectively.

1.7.3.2.5.1 F3

F3 algorithm is a modification of JSTEG algorithm [27]. There are two significant differences [28] :

First: when the least significant bit DCT coefficient is not same to or equal to the bit messages that are hiding in this coefficient, DCT coefficient does not change or alter the least significant bit, but in fact it reduces the absolute value. When concealment is used and the DCT coefficients whose value is equal to 1, however it exempts the DCT coefficients whose value is 0.

Second: When the hiding takes into account, the procedure of compression (shrinkage) that occurs by inserting the bit 0 in the DCT coefficient whose value is -1 or 1. Then the new value of the DCT coefficient is equal to zero. Since the receiver cannot distinguish the resulting compression of zero, which does not contain bits of messages, you will reset the message to the next DCT coefficient.

1.7.3.2.5.2 F4

F4 algorithm is in numerous ways similar to F3 algorithm, and since it is different from the way of inserting bits of the message in the corresponding AC coefficient [27]. When F3 algorithm to reduce the coefficient of absolute value if its least significant bit does not match, it intended message bits.[28] If the observed AC coefficient labels with ACcoef and message bits, then inserted F3 algorithm can be defined in these terms:

$$\mathrm{AC_{coef}} = \mathrm{AC_{coef}} - \mathrm{sgn}(\mathrm{AC_{coef}}) \cdot [\mathrm{LSB}(\mathrm{AC_{coef}}) \oplus \mathrm{bit}] \tag{12}$$

In F4 algorithm, positive coefficients are treated in the same way as the F3 algorithm. In other words, it reduces the value of the coefficient messages only if the least significant bit of the coefficient does not match the bit messages. In condition of

negative coefficients rule applies inversely: essence of the message increases the value of the coefficient only if the least significant bit of the coefficient corresponds to message bits. The above can be defined in these terms:

$$AC_{coef} = AC_{coef} - sgn(AC_{coef}) \cdot [LSB(AC_{coef}) \oplus bit \oplus (AC_{coef} < 0)] \quad (13)$$

1.7.3.2.5.3 F5

Basically, F5 algorithm is similar to an F4 algorithm [27] [28]. The improvement can be observed in the use of two techniques: permutation scattering and matrix coding. The method of hiding is shown in the diagram given in Figure 31.

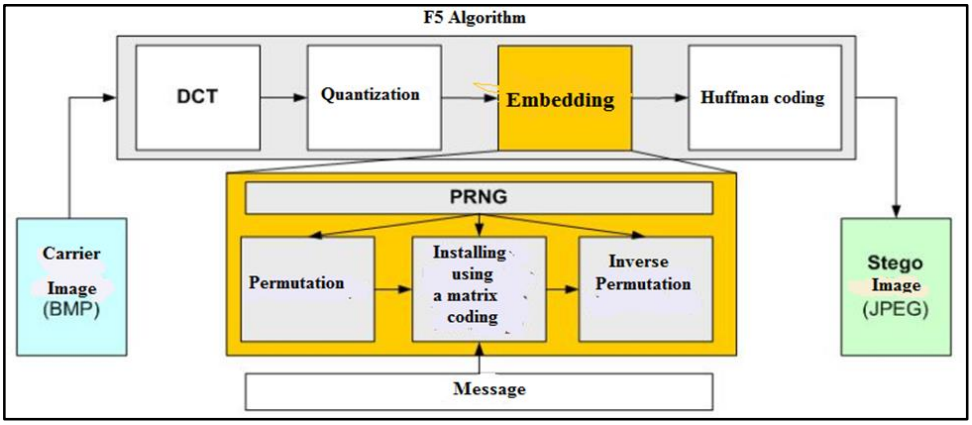


Figure 31: F5 Method Processes

1.7.3.2.5.3.1 Permutation Scattering

This technique prevents the steganalysis attackers. Its hiding speed greatly depends on the way it hides especially when hidden message size image has capacity holders. If a relatively large message is hidden then the work of random walks to search for free coefficients is longer. This procedure of hiding can significantly slow down at the end because it scans the image more. Therefore, the F5 algorithm uses Combinatorial scattering techniques. Figure 32 here illustrates a random walk technique, and in Figure 33 illustrates combinatorial scattering technique [28].

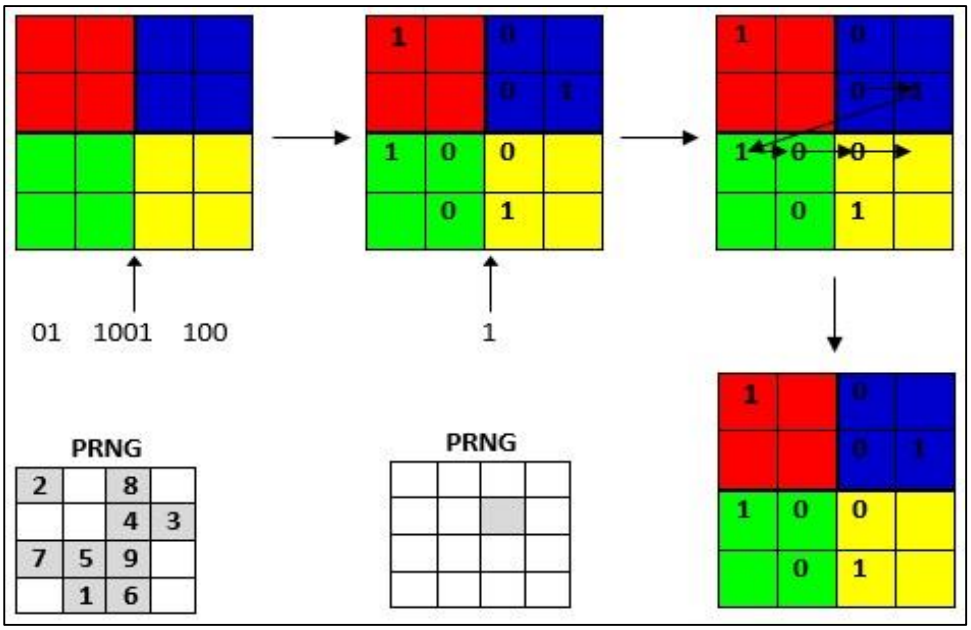


Figure 32: Hiding a Random Walk

In fact, the F5 algorithm is hiding messages in sequence, but before inserting, permuted the image, which depends on steganography key. After loading is done, using permutations inverse, ensure the message bits, even though, stored sequentially, it scattered throughout the image. Since inserting does not do random walks to search for free coefficients, but the message hides sequentially, this way of hiding will not increase the size of messages causing a significant slowdown in the process of hiding. As regards to smaller messages, this way of hiding is somewhat slower.

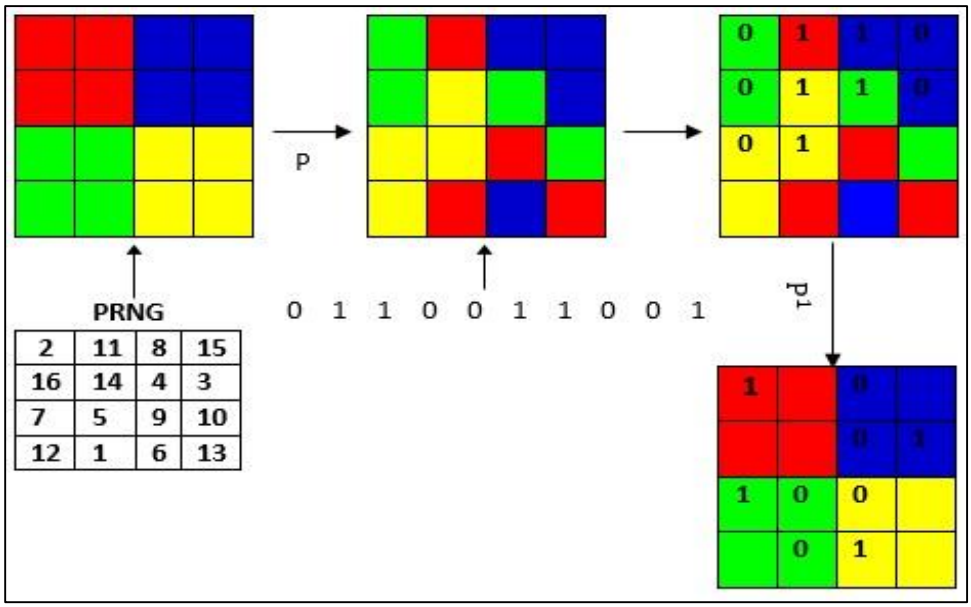


Figure 33: Hiding Permutation Scattering

1.7.3.2.5.3.2 Matrix Coding

Matrix encoding is a new technique which has been developed by Ron Crandall [29] to improve the embedding efficiency. The first application of matrix encoding is probably F5 algorithm. If most of the capacity has been used in the steganography then the matrix encoding reduces the required number of changes. If we assume that there is a secret message, distributed single values distributed separately in locations that are meant to change nearly half of the message but not the rest.

The embedding efficiency is 2 bits if the matrix encoding which is not used. Due to the shrinking that results from the F4 algorithm, the embedding algorithm is even a bit less, e.g. which is 1.5 bits per change (the shrinkage happiness when the change is occurred without embedding anything). As an example, if want, we can embed a very short message includes just 217 bytes (1736 bits) , F4 changes 1157 places in the Expo image. F5 embeds the similar message by using matrix encoding with only 459 changes (through an embedding proficiency is 3.8 bits per change) [28].

The following[42]example shows us in detail what would happen if we have embedding 2 bits x_1, x_2 in three modifiable bit places a_1, a_2, $a_3$altering one place as the greatest. We may encounter the following four cases [28]:

$$x_1 = a_1 \oplus a_3, x_2 = a_2 \oplus a_3 \Rightarrow \text{Change nothing} \quad (14)$$

$$x_1 \neq a_1 \oplus a_3, x_2 = a_2 \oplus a_3 \Rightarrow \text{Change} a_1 \quad (15)$$

$$x_1 = a_1 \oplus a_3, x_2 \neq a_2 \oplus a_3 \Rightarrow \text{Change} a_2 \quad (16)$$

$$x_1 \neq a_1 \oplus a_3, x_2 \neq a_2 \oplus a_3 \Rightarrow \text{Change} a_3 \quad (17)$$

In the above four cases we have not changed more than one bit. Generally, we have a CODEWORD a with$n$ adaptable bit places for $k$secret of message bits$x$. Let $f$be a hash function that extracts k bits from a code word. Matrix encoding makes it possible for us to find a suitable modified code word a' for every a and x with $x = f(a')$, such as the distance of Hamming [42]

$$d(a, a') \leq d_{max} \quad (18)$$

This code is denoted by a triple order (d_{max}, n, k): a code word with n places will be changed into not more than d_{max}places to embed k bits. F5 implements matrix encoding only for$d_{max} = 1$. For,$(1, n, k)$ the code words length is $n = 2^k - 1$. with neglecting shrinkage, we get a change density [42]:

$$D(k) = \frac{1}{n+1} = \frac{1}{2^k} \tag{19}$$

And then the embedding rate is:

$$R(k) = \frac{k}{n} = \frac{1}{n}.Id(n+1) = \frac{k}{2^k - 1} \tag{20}$$

By using the density change and the rate of the embedding we can define the embedding efficiency $W(k)$. It indicates the average number of bits that can we embed per change [42]:

$$W(k) = \frac{R(k)}{D(k)} = \frac{2^k}{2^k - 1}.k \tag{21}$$

The embedding efficiency of $(1, n, k)$ code is permanently greater than$k$. Table: 1 demonstrates that the rate decreases by the increasing of the efficiency. Therefore, we achieve great efficiency with only very short messages. Table 2 gives the dependences between the message bits $x_i$and the changed bit places$a' j$. We assign the dependences with the "binary coding" of$j$ to column$a' j$. Thus, like this we can control the hash function very quickly.

$$f(a) = \oplus_{i=1}^{n} a_i.i \tag{22}$$

K	N	Change density	Embedding rate	Embedding efficiency
1	1	50.00%	100.00%	2
2	3	25.00%	66.67%	2.67
3	7	12.50%	42.86%	3.43
4	15	6.25%	26.67%	4.27
5	31	3.12%	16.13%	5.16
6	63	1.56%	9.52%	6.09
7	127	0.78%	5.51%	7.06
8	255	0.39%	3.14%	8.03
9	511	0.20%	1.76%	9.02

Table 1: The Connection between the Embedding Rate and Change Density

$f(a')$	a'_1	a'_2	a'_3
x_1	×		×
x_2		×	×

$f(a')$	a'_1	a'_2	a'_3	a'_4	a'_5	a'_6	a'_7
x_1	×		×		×		×
x_2		×	×			×	×
x_3				×	×	×	×

Table 2: Dependency (×) Between Message Bits $x_i$and Code Word Bits a'_j

We find the bit place

$$s = x \oplus f(a) \tag{23}$$

For that we should change .the code of the word resulting in

$$a' = \begin{cases} a, & if\ s = 0 (\Leftrightarrow x = f(a)) \\ (a_1, a_1, \dots, a_s, \dots, a_n) & otherwise \end{cases} \tag{24}$$

We can find an optimal parameter $k$for every message to embed and everycarrier medium providing sufficient capacity, so that the message just fits into the carrier medium. For instance, if we need to embed a message with 1000 bits into a carrier medium with a capacity of 50000 bits, then the necessary embedding rate is $= 1000 : 50000 = 2\%$. This value is between $R(k = 8)$ and $R(k = 9)$ in Table 1. We choose$k = 8$, and are able to embed $50000 : 255 = 196$ code words with a length$n = 255$. The $(1, 255, 8)$ code could embed $196.8 = 1568$ bits. If we chose $k = 9$ instead, we cannot embed the message completely [28].

1.7.3.2.5.3.3 Embedding Steps

In detail here are the following steps of the algorithm F5:

Step 1: Start a JPEG compression, then stop after the quantization stage and counting the DCT coefficients of the image.

step2: Generate randomly a powerful number (From the point of cryptography). The number also can randomly resurrected by a secret key included.

Step 3: Starting a permutation, with two parameters, namely the generation of random numbers Step (2) and the coefficients of the image counting Step (1)

Step 4: Determine the value of the parameter k by capacity of image and the length of the secret message.

Step 5: Calculate the code word length (byte placeholder bits from the secret message), where n = 2k- 1, where n is the length of a code word, and k is from step (4).

Step 6: Perform embedding secret messages with a (1, n, k) matrix encoding:

a. Fill a buffer with n nonzero coefficients.

b. Perform hashing against this buffer.

c. Add k subsequent bits of the message into a hash value (per bit, with XOR operation)

d. If the result of (c) is 0, then the buffer has not changed. In addition, the results of

(C) Must index buffer, the absolute value of the element is reduced by one (Decrementing).

e. Do a testing or diminution of value (shrinkage) which can eliminate the value of the message is entered into the buffer, for example like produce a value of zero.

If that is true, then we have to adjust the buffer (eliminating the possibility of zero by taking other non-zero coefficient).

If it does not happen diminution of value, then proceed to the examination of a new coefficient which is right after the current buffer. If there are still bits of secret messages that have not been entered, repeat step (a).

Step 7: Continue JPEG compression (Huffman coding, etc.).

1.7.3.2.5.3.4 Extracting Steps

Secret message extraction is done in a manner equivalent to the method for embedding data.

Step 1: start Huffman decoding, then after, dezig-zag processes counting the DCT coefficients of stego image.

Step 2: Generate randomly a powerful number (From the point of cryptography). The number also can randomly resurrected by a secret key included.

Step 3: Starting a permutation, with two parameters, namely the generation of random numbers (Step 2) and the coefficients of the image counting (Step 1)

Step 4: Extract the message length k of the status word.

Step 5: Calculate n = 2k- 1

Step 6: With that information and the array of coefficients of the image, read n positions of the array, which are given by the permutation, in order to extract the k bits that were embedded in those positions.

Step 7: The k bits that were embedded we just need to group them into bytes and write them to a text file.

1.8 Steganalysis

It is the science that deals with the disclosure of the messages that are concealed within the images and are prepared by stego-systems. It is a fast developing science as well and fairly fresh where majority of the periodicals which are focused on it, have been published in the last ten years. The main purpose of steganalysis is to detect if the image contains a concealed message or not. Work and research done in this field contributes to find a statistical characteristics of images, where it is considered a success if it can figure out the presence of a concealed message within the image with more accuracy than of random guessing. In addition, it is also trying to calculate the length of inventory text or one hidden in the image and the type of algorithm used as well as the content of the message, also trying to find out the type of secret key. Therefore, the attack using steganalysis tries to reach one of the characteristics that have been mentioned above (the secret key used, the type of the inserted algorithm and the length of the message) and each one of them will lead to the other. The primary purpose of steganalysis is to decode the undisclosed message after the extraction and not only to be sure of the presence or absence of hidden text within the image. Therefore, in the absence of any knowledge of steganography techniques and the secret key used, it is impossible to decipher the hidden text, it needs to be made sure of its existence or it may take very long time. Thus, the message length or extent of their impact on the image features may give very important information to steganalysis. The main ability of the attacks is the quality to distinguish between stego image and the

cover for that (as previously mentioned) finding the stego image which may lead to the extraction of the message. Attacks for the sake to find the secret message use a number of ways, including code analysis and dictionary attacks. Nevertheless, Neils Provos [23, 24] in 2001 used his steganalysis software StegDetect [21] to test a large models of images that he downloaded by using a web crawler from eBay and Usenet. His attack was by using the distributed dictionaries on the doubted stego images that percentages are very small compared to the images that he has tested but he could not find any confidential messages.

Steganalysis despite playing a role in reading the contents of the message, it is possible for it to destroy or disable the message, on reversing the code analyzer which decrypts the encrypted message, steganalysis detects the presence of hidden message.

Scientists S. Jajodia and Neil F. Johnson [1] tried to classify the attacks that could have been waged by steganalysis to eliminate or to modify the message depending on the information available, with the attacks that may be launched by the cryptanalysis. The comparison based on, for the encryption between any possible parts of the plain text with parts of the ciphertext, as for steganography between the stego image and any possible parts of the message. The message concerning steganography may be encrypted or not, in case it is encrypted we can apply the code analysis techniques. If we classify the attack methods of steganography we have to put in the other side the cryptanalysis techniques. There are many types of attacks that belong to cryptanalysis:

1.8.1 Ciphertext-only Attacks:

In this case, the cryptanalysis will have the cipher text only, in some cases will have a piece of a plain text and this is used in finding the plain text.

1.8.2 Chosen Plaintext Attack:

In the most appropriate case, attacks where the cryptanalysis will have the plain text which is a part of the ciphertext, this will be analyzed numerous times and then the real text will be identified which is the actual one send by the sender.

1.8.3 Chosen Ciphertext Attack:

The cryptanalysis has the ciphertext with the encryption algorithm. So, when one while decrypt the ciphertext and looks for a match in the plain text and he knows and can find the user’s secret key that has been used in the encryption process. The challenges

faces steganalysis is different from the challenges that are faced by cryptanalysis, where steganalysis detects the ciphertext and analyze it, whereas the cryptanalysis just examine the ciphertext. Quite often, the parallel attacks of finding the message are available to steganalyst but in the field of steganalysis due to the fact that we have more than just the plain text and ciphertext there are the challenges and their dispute which make these classifications difficult, for that they have been characterized to several forms imposing that steganalyst has stego pattern on minimum.

1.8.4 Chosen Stego Attack:

This type of steganalyst is responsible of the type of the algorithm that may as well be used to make the stego medium. The basis of the work of such kind of attack depends on the attack by the chosen ciphertext, also in case of steganography it is very difficult to do so. Theoretically putting it, when one tries making a new steganography mediums for matching the valid sounds that have been intercepted is seemingly easy, but in fact, completing of this procedure can be extremely hard because here not only the medium is unknown but the cover of the hidden message is unknown as well. The most realistic attack can be carried out by steganalyst on steganography image is to make systematic use for the specific algorithm and attack the image that has been intercepted because in this case the steganography algorithm and the medium are unknown.

1.8.5 Stego Only Attack:

In this kind of attack the steganalyst has only information on the medium of the steganography and does not have any other information. This attack is considered the toughest type of attack and is same as the attack to use of the ciphertext only that is launched by cryptanalysis. The method to do that is through by all means possible of attack in the modern steganography algorithms.

1.8.6 Known Cover Attack:

The attack is launched by knowing the medium of the cover of the steganography no matter what the medium is. Then, the steganalyst finds the contrast between the two mediums and then see the type of the algorithm that can be used. This type of attack is similar to the attack which uses plain text only and that can be launched by the cryptanalysis.

1.8.7 Known Message Attack:

This kind of attack is used when the concealed message of steganography is detected by the steganalyst. Thereafter, the steganalyst tries to analyze the stego image for future attacks. Even when the hidden message is known by it, it will be very hard to find.

In spite of all these method, it is not much used because the primary purpose of steganalysis is to ensure the presence or absence of the secret message and not decode it. There are another kinds of attacks that are less theoretical and more practical as follows:

1.8.8 The Targeted Steganalysis:

This attack is applied in case for knowing the encryption algorithm use, where it works on the algorithm that is being used there only. The technique works on certain type of steganography and occasionally restricts the formation of the image i.e. Where and when the statistics of the encrypted image found through careful analysis and study of the embedding algorithm. The result of this technique is very accurate but not flexible because there is no path to be extended to make other embedding algorithm.

1.8.9 The Blind Steganalysis:

This method has been formed to work on every kind of images and all the embedding algorithms. The method works on the principal of learning between the statistical attributes of the pure image and the stego image and also gives the differences between them,-where the principal 'learns' to work by training the machine on large database of images. It is considered more flexible but less accurate than the former one.

1.8.10 Semi-Blind Steganalysis:

This method works on a range of different stego systems. It rely on the type of the embedding range that is used (spatial or transformational) [30].

1.9 Literature Survey

In 2001 Uruba I. applied system to the colored image by using LSB method. The system compares the value for each ASCII character with palette location of the image if the value is equal to the palette value that is compensated in another location [31]. Hamami A. in 2001designed an efficient system to check the image as if to see it contains a secret message or not. Through the system has extracted the secret message,

which may be in the form of text or image, and when it fails to extract the hidden message or keep it from traffic, the system destroys the hidden text [32]. Fridrich J. and Goljan M. In 2003 described steganography where they took large loads of gray scale images and added a few amplitude of the noise to the image pixels with certain specifications. Susceptibility noise distribution is arbitrary so the parties related to it have the possibility to hide the noise in conformity with the output noise of the devices themselves [33].

From here we can determine that this would be better to embed which uses arbitrary processes in terms of providing best safety such as added noise, fix amplitude of the image and the LSB. Alawy S., in 2004, provided a sophisticated way to hide text within the image of the type JPEG without stirring suspicions knowing that the quality of concealment are not affected by compression of the image. That image kind of JPEG which has converted each block of the masses to the image in 64 laboratories using DCT, then these transactions are quantized using the quantization table [34].

After that, every two coefficients of each block have been quantized using DCT to hide the text [14]. Ashraf A., in 2004, established a paper that makes a comparison between eight multiple-precision libraries based on various criteria including ease of use and ease of transport and performance. The performance for each library was evaluated, according to the original system which takes into account the relative use of encryption primitive operations and performance [35].

The purpose of this study is to evaluate the performance of private libraries and the statement of their suitability for the implementation of large groups of public keys coding systems such as RSA, DSA, charts and curved oval. Then choose the best library based on the required performance and the materials used for the implementation of the required time[35].

Davidson L. and Paul G., in 2005, defined a mechanism for the mobilization of the message site based on the method of restoring the image. When a message is concealed within the image, the image energy will increase effectively.

They define two energy-colored images and a gray one to measure the probability of each pixel of energy, Results indicated that steganography has energy outweigh the energy of their counterparts. According to them if the image has been divided into small areas, it can be further developed using the results of various means in which

Kohanon maps self-organized or assembly spatial technique are used and then apply method to demonstrate the extreme values of each region or part [36]. Ibrahim A., 2007, suggested a work that has not used LSB-1 cover but used instead a new way of steganography depending on the spatial domain to hide data and this work used LSB-3 in order to increase durability. In regard to reduce the difference between the existing steganography cover and the cover itself LSB-1, 2 was modified according to one bit from the bits of the message. Also, there has been steganography key used to re-ordering the message bits before embedding them [37]. Naji A. and Zaidan A., 2000, designed a very strong method; it used AES method to encrypt data before hiding it and then use a key of 128 bits, and then hide the information in a file extension of EXE. It is not very easy for any hacker to guess the hidden information in a file extension of EXE, after retrieving information; the hidden blade is decrypted [38].

Chapter Two

2. Methodology

This chapter discusses the methodology used for the study of the steganography implementation and analysis algorithms to digital images; here we present a series of experiments conducted in order to evaluate the embedding efficiency of the algorithms have been developed during the course of this work. For this, we used special programs if found, otherwise, we have developed our program for further implementation that must be chosen. Moreover, performance analysis and effectiveness will also be discussed further.

2.1 Cover Image Selection

We have selected the image carefully for that it is complicated and of multi characteristics, it is expected here that it has an effect in power and capacity, all images have the same size 300*200 on two formats BMP, JPEG, as given in the following: We examined if the characteristics of the cover image has impact on the concealment efficiency of stego.

Figure 34((A) Group_of_student: here the dominant color is the blonde color regarding the hair, but the color of the jackets is black which makes it also a dominant color in this picture. (B)Living_room_home_house, it is obvious that the dominant color is green color of the trees and the field which has a light green color. (C)Spring_sunshine_may the dominant color is pink color, although it is complicated to tell the exact color.).

Figure 34: Cover Image Selection A, B and C Respectively

2.2 Secret Message Selection

As it is known, in modern steganography methods that the secret data consists of any digital files like a text, an image, audio, video and the executive file, etc. However, in this research the text file was chosen in making our experiment and the reason for that

is the objective which is, to use the maximum insertion capacity of every method which is not expected in few of them and also depends on formats and scenes of chosen image, that leads to use a concealed message that is different in size for every experiment. We used a text message since it is not important for our study whether the message is text, voice or another image, it is just a data, a collection of 0's and 1's… Therefore, with regard to, 'Vocabulary as a reflection of life wisdom '[39] we have chosen the below mentioned paragraph given in bold, as the concealed message for our experiment, and it is kept in a text file to be used with the suggested techniques. Emphasize the repetition of the same text to fill up the total capacity of the cover image which differs depending on the steganography technique used.

"Comprehend the environment in terms of discrete objects and events as a result we can say that the world consists of a multitude of uniquely defined objects and events They can be further organized into classes as groupings based on the criterion of similarity or shared characteristics The mental construction which comprises the criterion of similarity and which subsequently enables the classification of objects is called the concept In other words it stands for or represents a common set of attributes of an object or event" [39**].**

2.3 Steganography Algorithm Selection and Used Tools:

As we mentioned previously in the first chapter that there are numerous methods for hiding information inside the image, in both image and transform fields, therefore the following method was adapted (LSB and BPCS) for the image domain, and the F5 for the frequency domain, the advantages of using these steganography methods is to see the highest percentage to replace secret data on the cover considering low visual artifacts. These methods were tested for the change in the special parameters for each unit separately. We inspected if the parameters have impact on the concealment efficiency of the technique from here, we utilize the whole comprehensible capacity for each method through repeating the selected text which we choose previously in the cover images.

2.3.1 Stego-Image Generation by LSB

The advantage of using LSB steganography is not needed to a complex calculation for the purpose of the data hiding in the image as well as it is considered one of the most prevalent methods to hide data in the spatial domain of the image. These

methods are simple to implement, however, over time the tool Internet offers the possibility to identify and extract the information in any bit plane. Therefore there is a need to make a program which permits us to withhold information in an image using proper to the purpose of the research, so we develop program to adjust some variation of parameters and getting the final result of the application as stego image file. Based on the inherent characteristics of the human eye and images in this part, there are some of the more classical algorithms used in steganography, elucidating the operation of each.

2.3.1.1 LSB in a Single Color

2.3.1.1.1 One LSB

This is the first algorithm and also the simplest. Basically it consists of changing the least important bit of the color bands (R, G or B) of an image cell array in consideration to enter the message utilizing the space of one bit per pixel to store the message.

Besides the already mentioned low computational cost that is characteristic of LSBs algorithms, specifically in this regard we can quote high fidelity between the original image and the stego-image. The change of just one bit ensures a great difficulty to note the "naked eye" the contrast between them in Figure 35. Here, one bit stored in accordance to the method described in this Section.

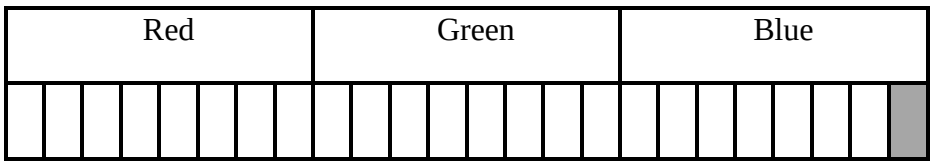


Figure 35: One Bit in Blue Color

2.3.1.1.2 Two LSBs

A slight variation of the one LSB, the only difference, as its name suggests, is to use two least significant bits of the color bands (R, G or B) separately in the image.

The subtle difference is in the ability to the hide message stored twice, and the change in the figure is also a bit sharper, but still generally imperceptible to the "naked eye", as we can observe in Figure 36, where two bits of message stored in it.

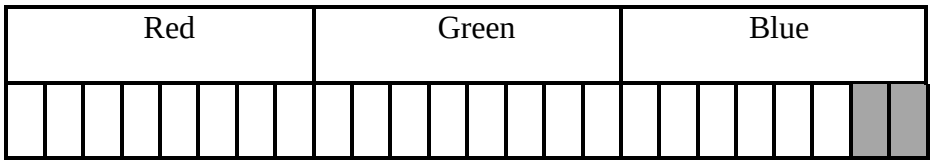


Figure 36: Two Bits in Blue Color

At various stages of this work, we repeat the original image without hidden messages only to facilitate visual comparison of the picture with the message. Also, the image will be stored in BMP format to stop any kind caused by coding formats that use data compression (like JPG).

2.3.1.1.3 Three LSBs

In general, the three LSB bits boost the quantity of bits to be used. This causes a boost in storage capacity over a stego-image quite modified from the original. To illustrate, Figure 37 presents an example of inserted three bit in one color in pixel.

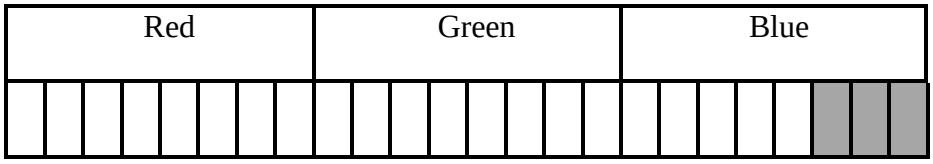


Figure 37: Three Bits in Blue Color

2.3.1.2 LSB in Three Color (RGB)

Still quite similar to previous algorithms, with only a modification:

The least important bit to be modified is switched between bands cyclically. For example, a message to store only four bits in the image the algorithm record the first bit in the R band, the second bit in the G band, and B in the third quarter turn to band R in next pixel. Thus, a switchover is produced in order pixels to be modified to achieve thus imposing a difficulty more for steganalysis to detect and get the message correctly. Figure 38 shows the one, two and three bits of secret data using the cyclical LSB algorithm.

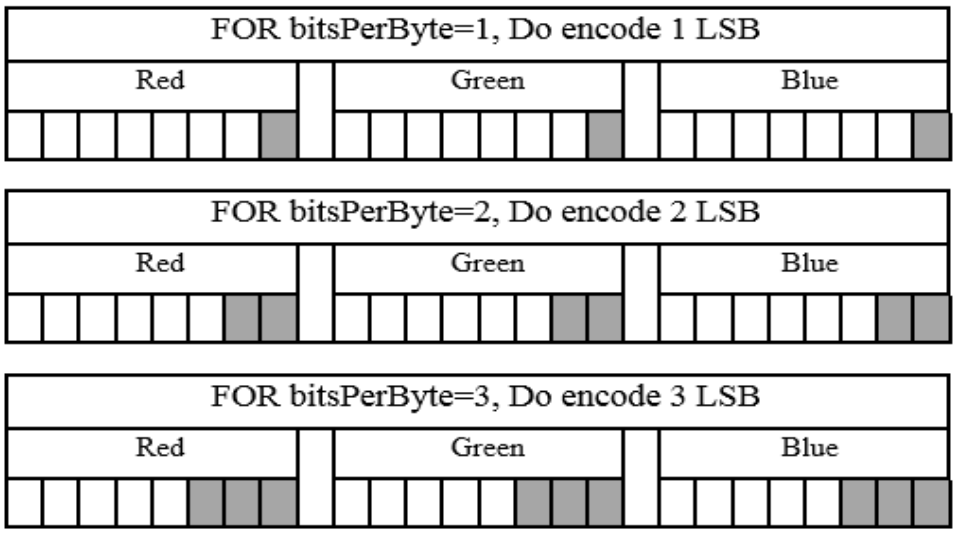


Figure 38: LSB Three Colors Embedded

The algorithms have been advanced using MATLAB and each allows the variation of parameters such as number of bits or color component to modify [41]. The techniques were developed initially in the shape of scripts, wherein each was implementing a method (LSB bit 1, etc.). Subsequently, aiming facilitating experimentation and obtaining the results, a graphical interface was created (GUI) using the guide Matlab. This interface provides a facility to choose the parameters of each experiment. In Figure 39 one can notice the GUI.

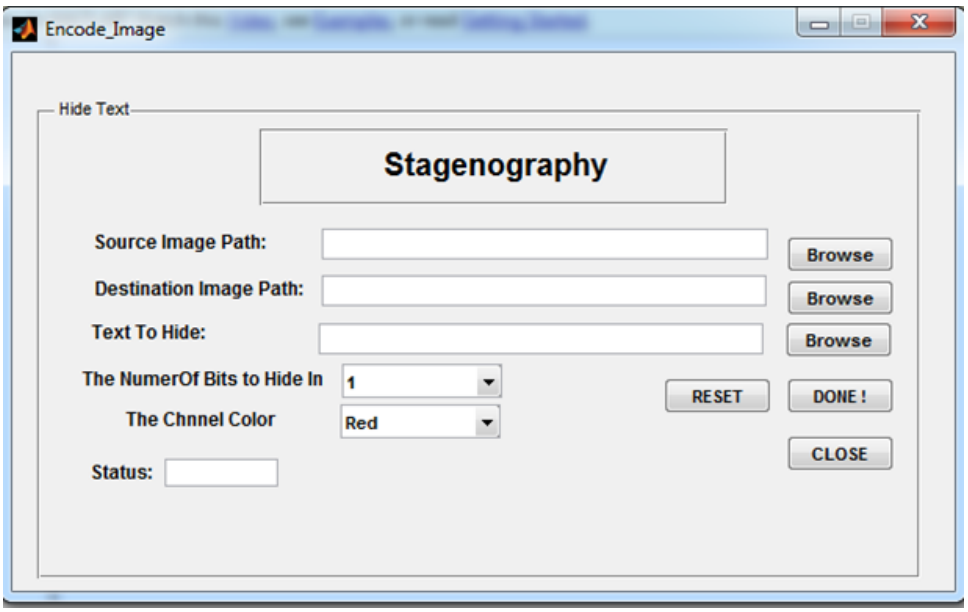


Figure 39: LSB Method Interface

With regard to the parameters of each algorithm, we varied the use of each one of them in regards to the number of LSB bits to be modified in the image in consideration to

making it possible the analysis of the impact which this change does not only through use of the similarity indices, but also by our visual system. In choice of which color component (R, G or B) to modify we opted for, where possible, modify each of color separately and all of them because of the difficulty inherent to the human being to visualize this component. Therefore, we sought to make changes in the images less possibly noticeable to our eyes.

2.3.2 Stego-Image Generation by BPCS

BPCS steganography is a technique which has a large capacity concealment. A case has been discussed previously, each bit-plane of the BMP image format can be segmented into a set of informative regions are parts of the image that are visually distinguishable. Therefore, these regions cannot be absolutely modified for steganography purposes and noisy regions are complex and may be altered to include secret information without any affect to the image quality as a whole. In the images with natural scenes, it happens that regions are characterized by having an informative coefficient α low complexity while the noisy regions possess a high α. Therefore, given a fixed threshold of complexity namely $\alpha 0$ parameter dependent variable from the image, regions of the bit-plane having $\alpha<\alpha 0$ are defined information and thus cannot be modified, while those with $\alpha \geq \alpha 0$ are classified as noisy and, therefore, capable of being altered by the insertion in them of secret data without causing noticeable deterioration of the quality of cover. Therefore, this method based on the threshold in concealing information, and to get variable values for the threshold in these experiments of ours by using Qtech-Hide&View v011 tool for BPCS [40]. This software was created and developed by Eiji Kawaguchi with the members of "Steganography Research Group" associated with Kyushu Institute of Technology, Japan (KIT-TEGROUP) for academic purposes. The Information embedding in this tool contains parameters which are illustrated below:

- Cover image (The format of container images, BMP, PNG or JPG (standard, or progressive data Type of DCT method is because of JPEG). Image size yield in the range of 128x128 ~ 3.200x3.200 pixels).
- Secret message
- Complexity threshold which has a range starting from 0 value to 55.

- Stego image (the format of the output image (stego image) is as PNG file, BMP file or as it will specify one of the JPG).

In these experiments, we depended on using different values for the threshold starting from value 5 until 55 to promote our expected objectives in this project so as to study the level of insertion capacity of the concealed information and the effect of the same on the different used image.

2.3.3 Stego-Image Generation by F5

As already mentioned, F5 steganography algorithm was introduced by German researchers Westfeld and Pfitzmann in 2001. The objective of their research is to create concepts and practical embedding method for JPEG images that will provide high steganographic capacity without compromising security. In the embedding process, message length and the number of non-zero coefficients non-DC utilized to establish the best embedding matrix that minimize the amount of modification of the cover image. Embedding Matrix has basically three parameters (c, n, and k), where c is the number of changes per group coefficient n, and k is the number of bits embedded. In their paper, the authors defines a simple matrix embedding (1, 2k-1, k) by making use of a "hash" function that generates k bits when applied to 2k- 1 coefficient. F5, in the version 11, is written in Java [28] and thus platform-independent. Both the algorithm and the source code to the implementation are freely translated that has six input in the program as follows:

- Quality factor Q of stego image,
- Input file (TIFF, BMP, JPEG or GIF)
- The name of the output file,
- The file possessing the hidden message,
- A user password to use as seed for the pseudorandom number generator random (PRNG)
- The comments to be included in the header of the JPEG file.

In chapter one, in accordance to F5 algorithm steps, here we repeat it with the procedure of its program to explain and understand this method in practice:

1. JPEG compression start and stop after quantization.

2. Initializing a random number generator with the key derived from the password.

3. Initialize the permutation with two parameters: the random number generator and the number of coefficients.

```
F5Random random = new F5Random(passWord.getBytes());

Permutation permutation = new Permutation(DctCoeffCount, random);
```

4. Copy the coefficients of the image into an array and calculate the parameter *k* using the capacity of the image and the length of the secret message.

5. Calculate the code word length n = 2 k – 1.

```
for (i=0; i<DctCoeff.Count; i++)   {

 if (i%64==0) continue;

 if (DctCoeff [i]==1) _one++;

 if (DctCoeff [i]==-1) _one++;

 if (DctCoeff [i]==0) _zero++;  }

 _large = DctCoeffCount - _zero - _one –DctCoeffCount%64;

_expected = _large+ (int)(0.49*_one);

  for (i = 1; i < 8; i++)   {

int usable, changed;

 n = (1 << i) - 1;

usable = ((_expected * i) / n) - ((_expected * i) / n) % n;

usable /= 8;

if (usable == 0) break;

if (usable < byteToEmbed + 4) break;  }   k = i - 1;

  n = (1 << k) - 1;
```

In code above, the expression 0.49*_one is because of the said estimated loss shrinkage rate [42]. Moreover, the line usable = _expected * i / n - _expected * i / n % n; which is (i / n) the embedding rate, R (k) (formula (20) in chapter1). In simple words, the number of possibly available bits (_expected) times the embedding rate (i / n), gives the number of bit we can embed.

6. Embed the hidden message with the code (1, n, k) using arrays:

a. Take the first byte of the hidden message and prepare to embed it.

```
embeddingLoop:
do {
kBitsToEmbed = 0;
 // get k bits to embed
for (i=0; i<k; i++) {
if (availableBitsToEmbed==0) {
// If the byte of embedded text is empty, we will get a new one.
try{
if (embeddedData.available()==0) {
isLastByte = true;
break;}
byteToEmbed = embeddedData.read();
 byteToEmbed ^= random.getNextByte();
} catch (Exception e)
{e.printStackTrace(); break;}
availableBitsToEmbed=8;}
nextBitToEmbed = byteToEmbed & 1;
 byteToEmbed >>= 1; availableBitsToEmbed--;
kBitsToEmbed |= nextBitToEmbed << i;}
```

b. Fill a buffer with *n* coefficients of the different image of 0. Also check that second 1 or -1 is not used.

```
do {
j = startOfN; one = startOne;
 for (i = 0; i < n; j++) {
if (j >= DctCoeffCount) {
// in rare cases the estimated capacity is too small
System.out.println ("Capacity exhausted.");
break embeddingLoop;}
shuffledIndex = permutation.getShuffled(j);
 // skip zeroes
if (DctCoeff [shuffledIndex] == 0) continue;
 //  skip every second 1 or -1
if (Math.abs(DctCoeff [shuffledIndex]) == 1) {
if ((++one & 1) == 0) continue;}
codeWord[i++]=shuffledIndex;  }
```

c. Generate a hash value of *k* bits. Add the following *k* bits of the message hash with a xor.

```
endOfN = j;
hash = 0;
for (i = 0; i < n; i++) {
if (DctCoeff [CodeWord[i]] > 0) {
extractedBit = DctCoeff [CodeWord[i]] & 1;}
else { extractedBit = 1 - (DctCoeff [CodeWord[i]] & 1); }
if (extractedBit == 1) {
hash ^= i + 1;}  }
```

d. If the sum is zero, the buffer remains unchanged. If not, we add or subtract one on that image position based on whether the coefficient is positive or negative, in consideration to decrease the absolute value.

```
i = hash ^ kBitsToEmbed;
if (i == 0) break;
// embedded without change
 i--;
if (DctCoeff[CodeWord[i]] > 0) {
    DctCoeff[CodeWord[i]]--;
} else
{ DctCoeff[CodeWord[i]]++;}
```

e. Check during the process we have not produced a zero. If happened, adjust the buffer eliminating zero, i.e., repeat step (6) from the same coefficient. If we have not produced any zero coefficients continue with the following buffer. If there is still a message to embed, continue from step (6).

```
if (DctCoeff [CodeWord[i]]==0) {
_thrown++; }
} while (true);
startOfN = endOfN;
startOne = one;
  } while (!isLastByte);
```

7. Continue to JPEG compression once the modified coefficients have been incorporated into the image.

This is the procedure followed when n> 1. When n = 1 , for embedding the message is a modified LSB, i.e. embedding bits like LSB, but takes into account the permutation used in F5 and checks used for the coefficients are different from 0 and also checks that 1 or -1 is not used.

Secret message extraction is done in a manner equivalent to the method for embedding data. Extract the message length and k of the status word. With that information and the array of coefficients of the image we read n positions of the array, which are given by the permutation of F5, in consideration of extract the k bits that were embedded in those positions.

The example in Figure 40 below, to embed a plantext.txt file size (528 KB) with test.jpg file size (300*200) and save it under steg.jpg file with quality 80:

```
C:\f5>java Embed test.jpg stego.jpg -e plaintext.txt -q 80
DCT/quantization starts
Image Size (300 x 200)
Got 94848 DCT AC/DC coefficients & one=9380 & large=9783
Expected capacity: 14379 bits
Expected capacity with
Default code: 1797 bytes (efficiency: 1.5 bits per change)
(1, 3, 2) code: 1198 bytes (efficiency: 1.7 bits per change)
(1, 7, 3) code: 770 bytes (efficiency: 2.1 bits per change)
(1, 15, 4) code: 478 bytes (efficiency: 2.6 bits per change)
(1, 31, 5) code: 286 bytes (efficiency: 3.1 bits per change)
(1, 63, 6) code: 165 bytes (efficiency: 3.5 bits per change)
(1, 127, 7) code: 95 bytes (efficiency: 4.1 bits per change)
Permutation starts
Embedding of 4256 bits (528+4 bytes) using
(1, 7, 3) code 1999 coefficients changed (efficiency: 2.1 bits per change)
961 coefficients thrown (zeroed)
4256 bits (532 bytes) embedded
Starting Huffman Encoding.
```

Figure 40: Implementation of F5 Program

After testing, the program gives us the expectation of capacity in cover image which can be used to insert a secret message with list of n and k codes before inserting a secret text ,therefore , the (1,n=7,k=3) code using to embed rely upon the size of text.

The intrusive image format for this program is JPEG and the parameter used here is the image quality, we have experimented this method by using images of different qualities, from 20 to 100, to study its effect on capacity with our expectations that any difference in the image has an effect too.

2.4 Criteria for Assessing the Quality of the Image-Stego

The processing and analysis of images is always a question for their quality. So also is the shorthand processing for embedding messages in image containers occurs some distortion of images that must be assessed. Image quality can be assessed in different ways and in connection with various tasks. In one approach, the quality is regarded as a characteristic of the image, and determines its own properties (statistical, structural, and semantic). In this case, the quality criteria are either subjective (determined by the human eye), or based on image characteristics: shape and parameters of the brightness distribution, the width of the spatial spectrum, etc. Moreover, objective criteria used in assessing the quality of the images are criteria to get a computed image characteristic difference signal between two images: a real and some ideal, or it may be the original and transformed. They are called difference metric distortion. Using these criteria it allows to evaluate the quantitative changes of brightness levels of image distortion when creating transformations (filtering, data compression, etc.) that is substantially the quality of the conversion means - algorithm or system. It is extremely necessary in the construction of algorithms and image processing systems and algorithms for evaluating quality. The two popular measurement tools which are used widely for this purpose are image histogram and mean square error introduced below:

2.4.1 Histogram

The histogram is a significant statistical characteristic of data. In various image processing applications, the histogram is generally utilized as the basic characteristic to present the distribution of the intensity, color, and texture parameters of images. As a statistical feature, the histogram is equally not sensitive to translation and rotation of objects. Meanwhile, it is a standardized and compressed data storage type that can save much space. Because of these advantages and along with the same, the histogram is

used mostly in image segmentation, registration, tracking, and especially in the image retrieval field that involves a large amount of data. The following formula calculates the histogram measurement where its parameters are k is the maximum pixel value in an image, m is the pixel value, and n is total repetition of n (histogram) in image [41].

$$n_i = \sum_{i=1}^{k} m_i \tag{25}$$

The example in figure 41 below, for an image recorded in the RGB, palette permits you to view the distribution of brightness in the channels Red, Green, Blue.

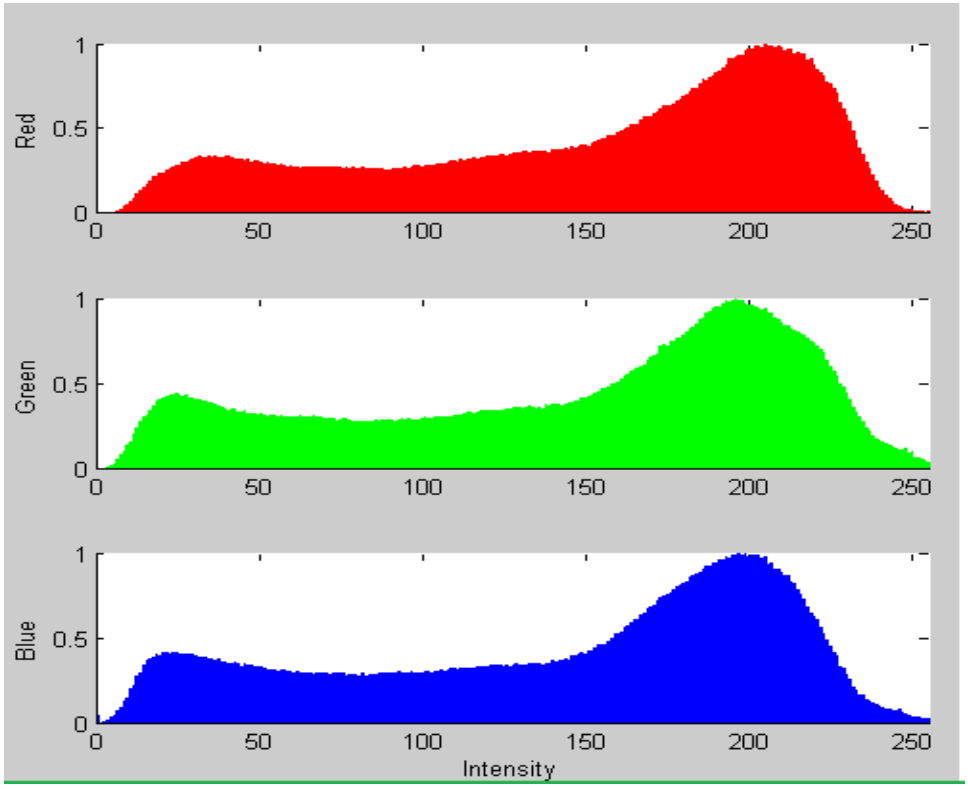


Figure 41: Histogram of RGB Color

2.4.2 Mean Square Error

The most popular distance tools for analysis of the level of distortions that are introduced into the cover image at the time to hide the information, therefore, MSE can be utilized to examine the quality of the stego-images as well. MSE is the ratio of

sum of the square of the differences in the pixel values between the corresponding pixels of the two images over total pixel number. MSE can be calculated if two images dimensions are equal. If two images are identical MSE value will be 0. Formula 26 shows how to calculate MSE value. X and Y are images with same dimensions. **m** and **n** are the dimensions of images [43].

$$MSE(X, Y) = \frac{1}{(m.n)} \sum_{i=1}^{i=m} \sum_{j=1}^{j=n} [X(i, j) - Y(i, j)]^2 \tag{26}$$

Chapter Three

3. Results

This chapter will present all Histogram results that have been obtained by running the algorithms, with a different group of parameters for each set, in relation to modification of the carrier image after insertion of the secret data. Based on these results we can say that the histogram analysis must proceed very carefully because it is still only for hiding the program proposed by the maximum amount of information embedded as it can be demonstrated minimal or major differences between the original and the stego images histograms. As for the other results as stego images and the data table of histograms, they were copied in a CD which is attached to this thesis report.

Note: The abbreviations were used to label the results of each image histogram that is shown in table 3 with a logical form to suit the size and to decrease repetition according to the following arrangement:

Abbreviations	The words
B	Histogram of blue band of the image
C	Chosen color of LSB (1 or3) color
G	Histogram of green band of the image
GS	Group_of_student
LR	Living_room_home_house
ORG	Original image
Q	Quality
R	Histogram of red band of the image
RGB	Histogram of red, green and blue band of the image
SP	Spring_sunshine_may
STEG	Stego-image
T	Threshold

Table 3: The Abbreviation of Used Histogram Title

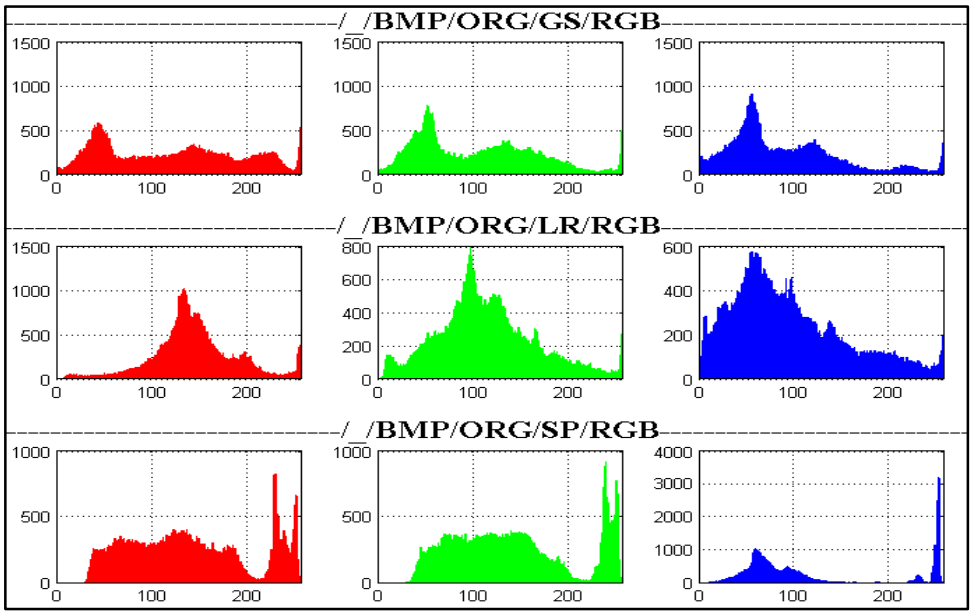


Figure 42: Original Bitmap Images Histogram

3.1 Substitution Technique (LSB Method)

Here, the results of the histograms were collected for the stego images that have one color modulation in one form, and RGB colors stego images. The one least significant bit result is shown in the figures 43 and 44, two least significant bit result in the figures 45 and 46, and three least significant bit result in the figures 47 and 48.

3.1.1 One LSB

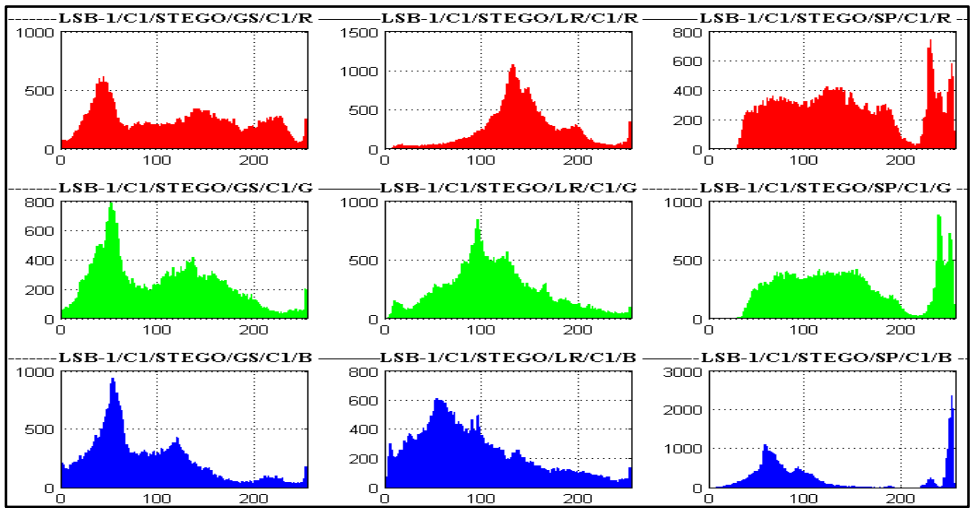


Figure 43: Histogram of One LSB One Color

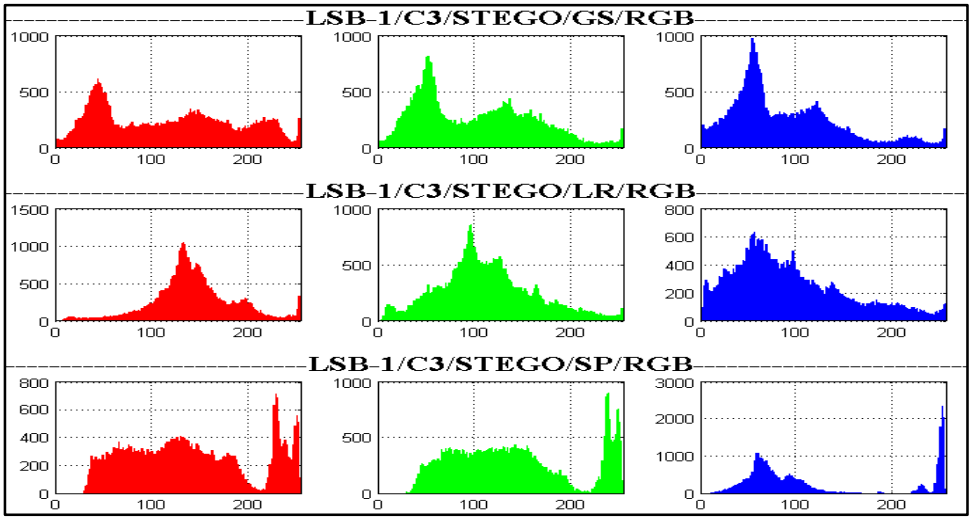


Figure 44: Histogram of One LSB RGB Color

3.1.2 Two LSB

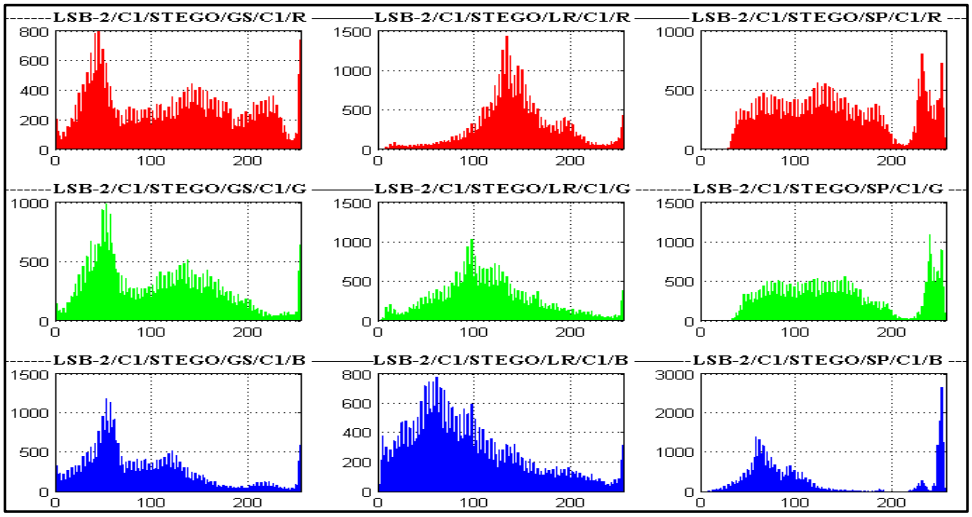


Figure 45: Histogram of Two LSB One Color

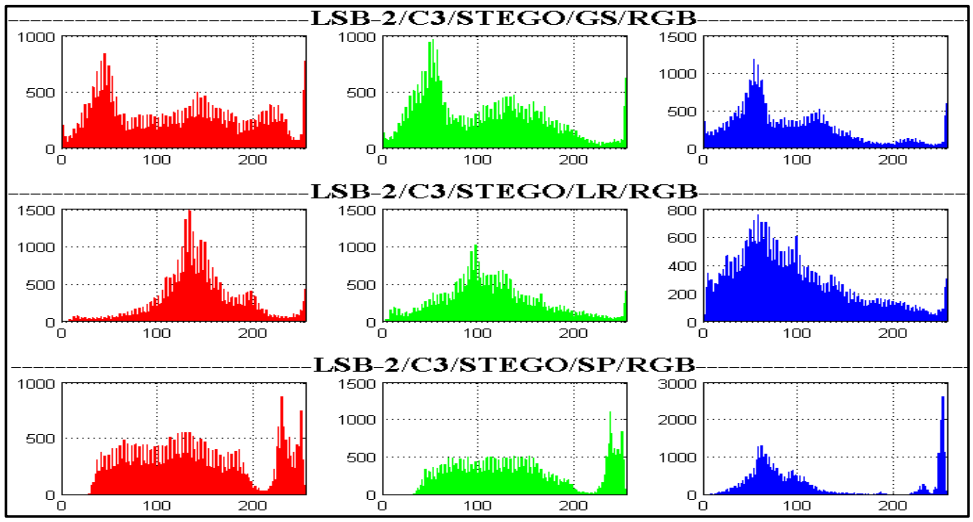


Figure 46: Histogram of Two LSB RGB color

3.1.3 Three LSB

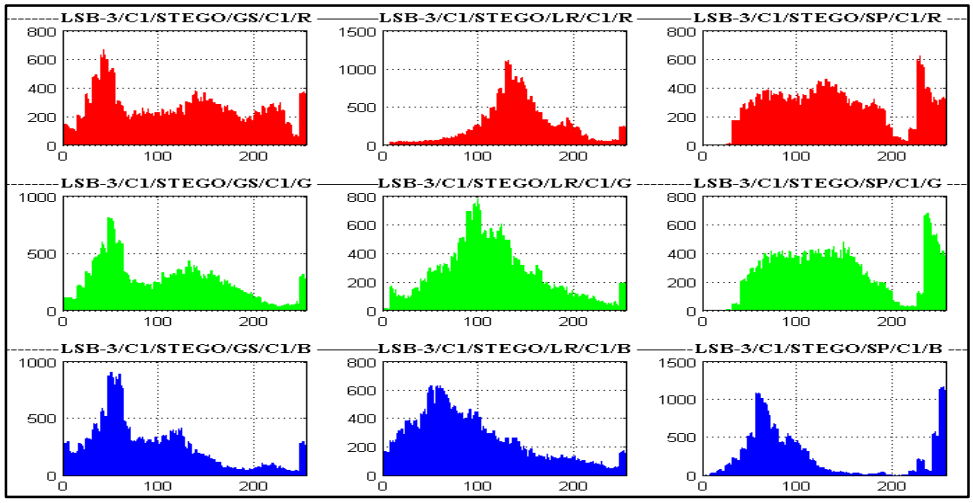


Figure 47: Histogram of Three LSB One Color

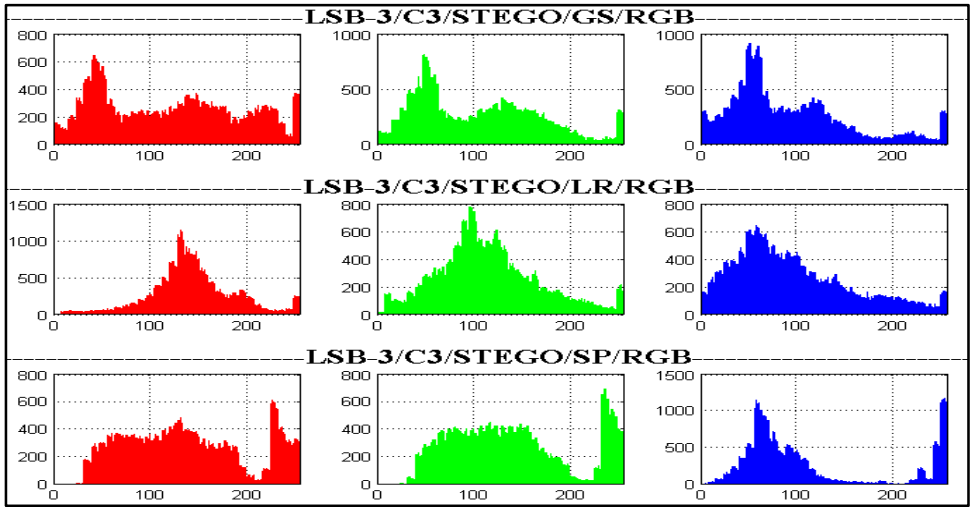


Figure 48: Histogram of Three LSB RGB color

3.2 BPCS Techniques

The results of stego images shown in the histograms below from the figure 49 to 59 depend on the value of the threshold, starting from the value 5 up to value 55.

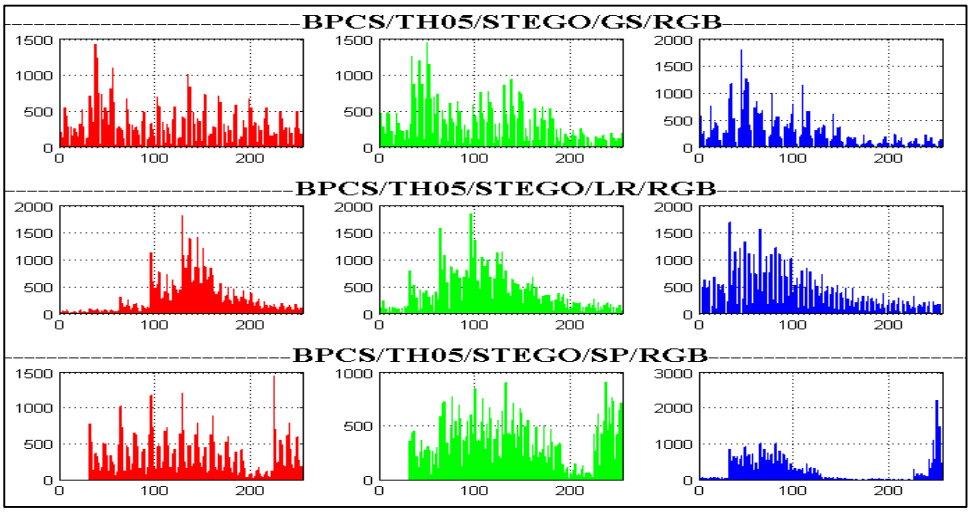


Figure 49: Histogram of BPCS with Threshold (5)

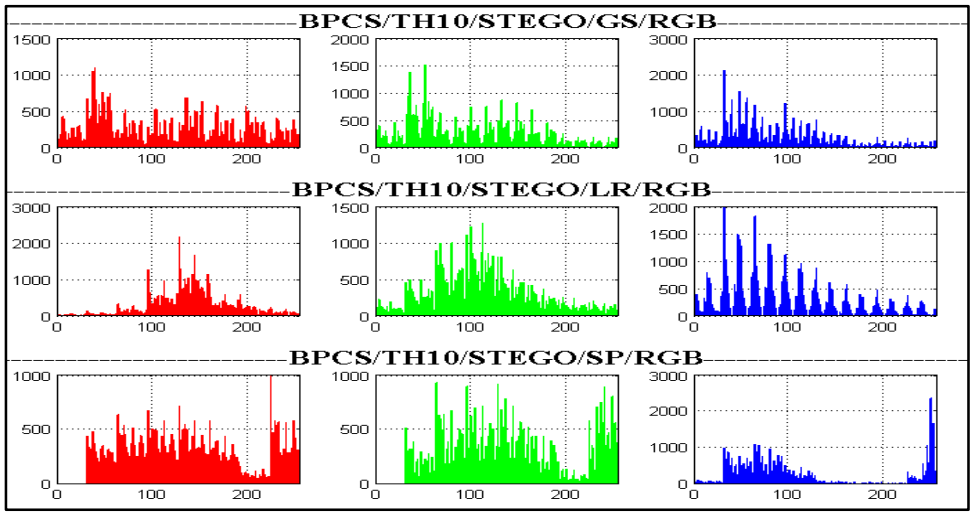


Figure 50: Histogram of BPCS with Threshold (10)

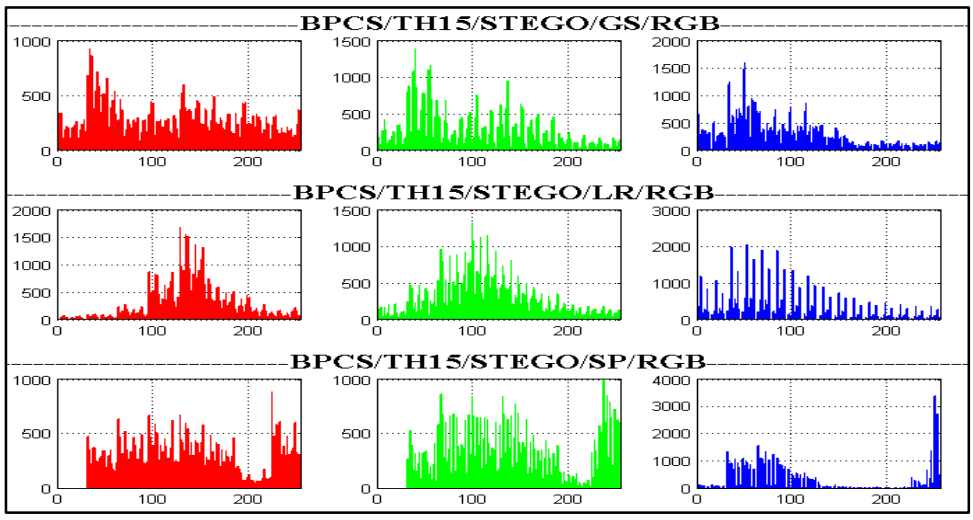


Figure 51: Histogram of BPCS with Threshold (15)

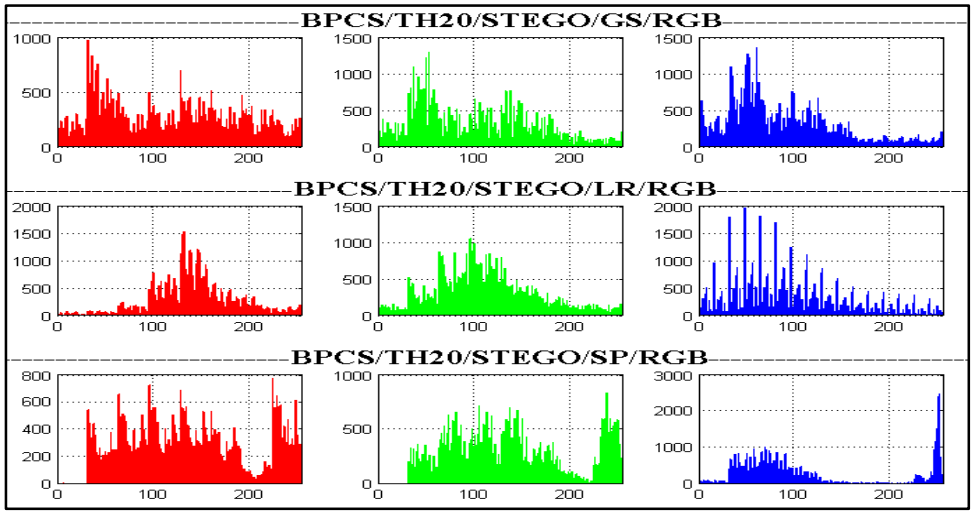


Figure 52: Histogram of BPCS with Threshold (20)

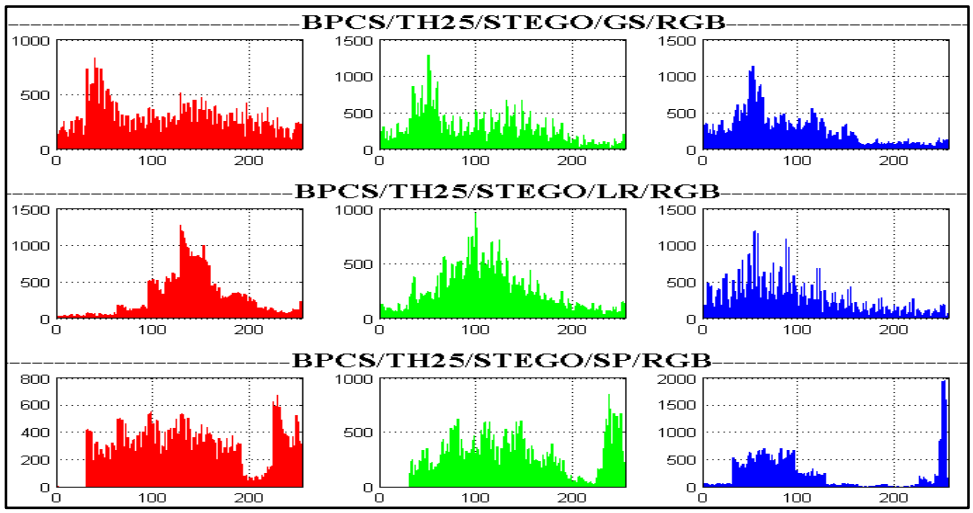


Figure 53: Histogram of BPCS with Threshold (25)

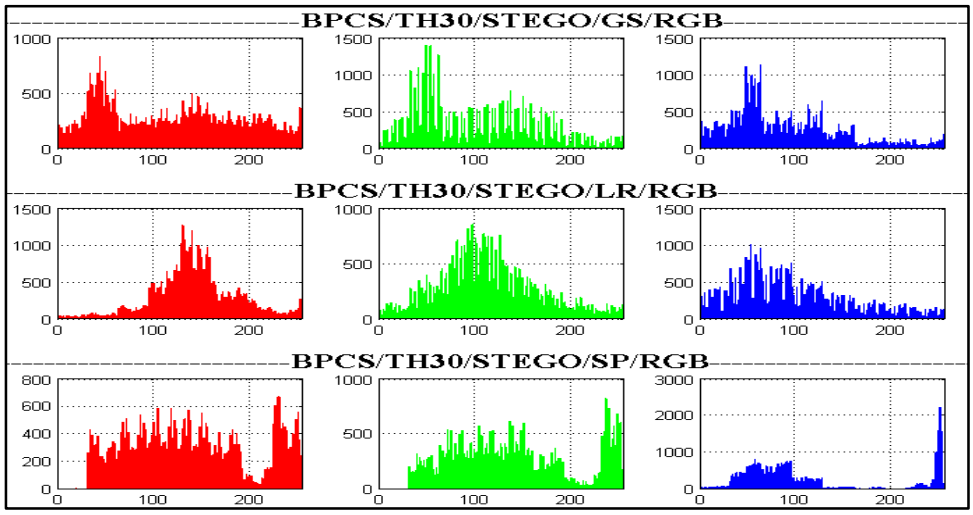


Figure 54: Histogram of BPCS with Threshold (30)

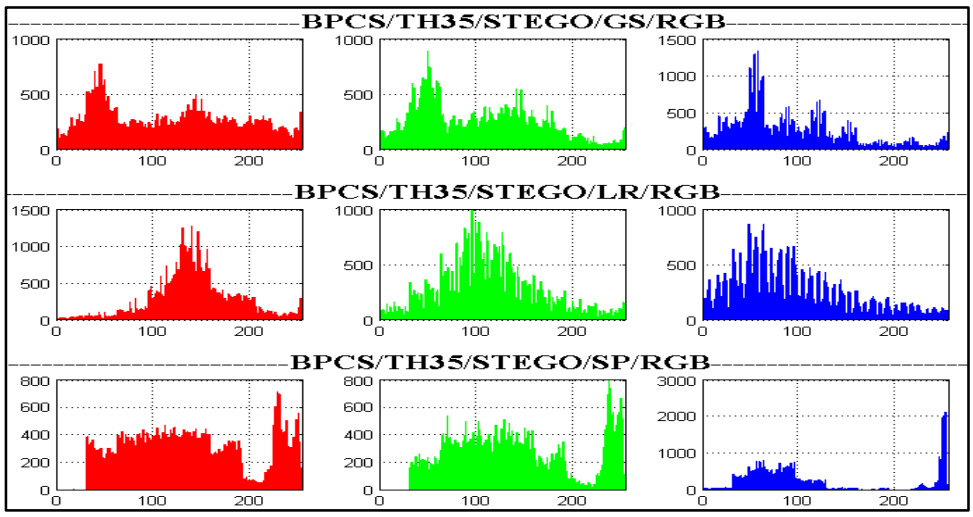


Figure 55: Histogram of BPCS with Threshold (35)

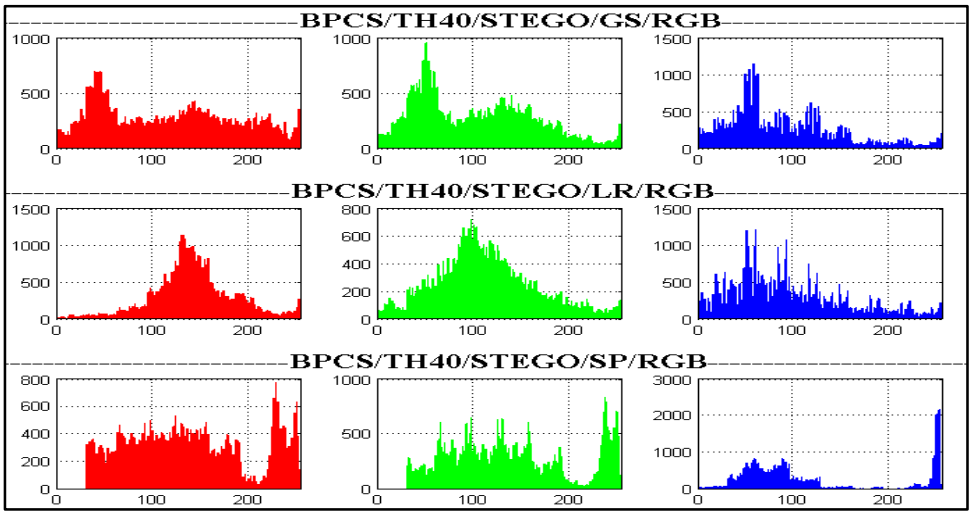


Figure 56: Histogram of BPCS with Threshold (40)

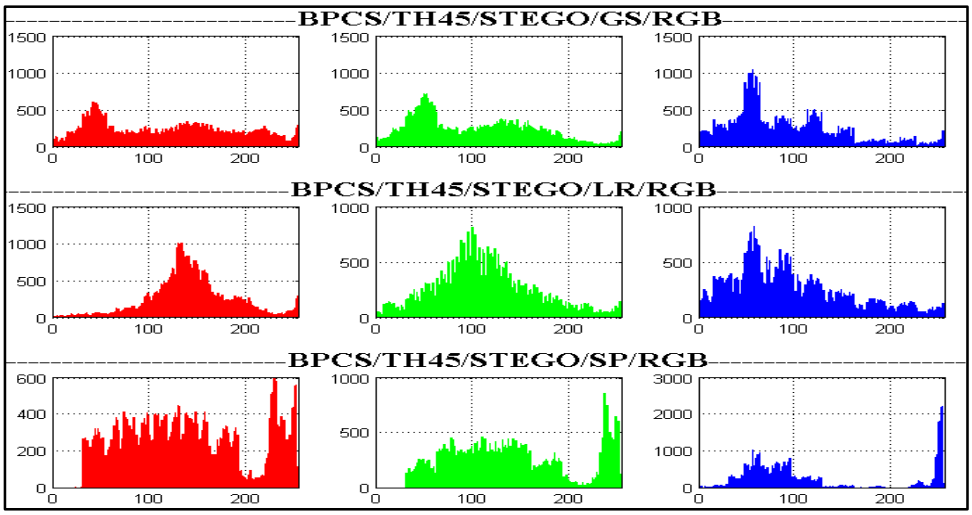


Figure 57: Histogram of BPCS with Threshold (45)

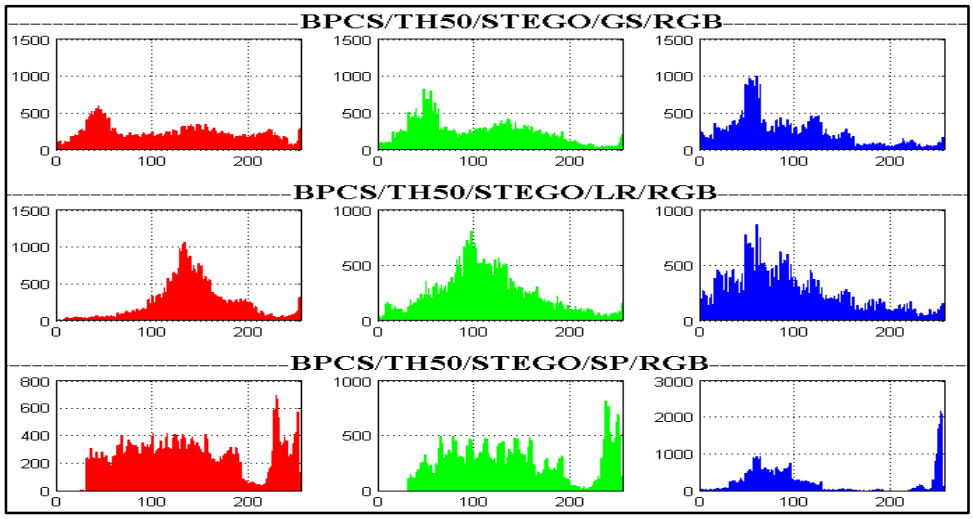


Figure 58: Histogram of BPCS with Threshold (50)

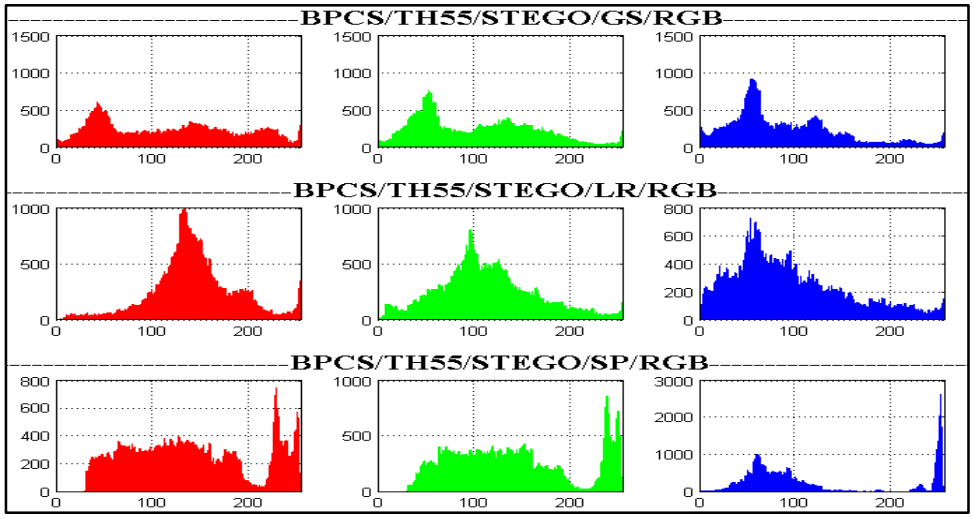


Figure 59: Histogram of BPCS with Threshold (55)

3.3 DCT Techniques (F5 Method)

With reference to the previous chapter, the vector that was counted for the experimenting and appliance of this method is the quality of the image, which means the cover image and stego image quality is the same for each experiment. Here, the results of the histogram of the original image versus the results of the histogram of the

stego image is shown on the left which shows quality values from 0 to 100 with 20 steps, these results are given in the figures from 60 to 65.

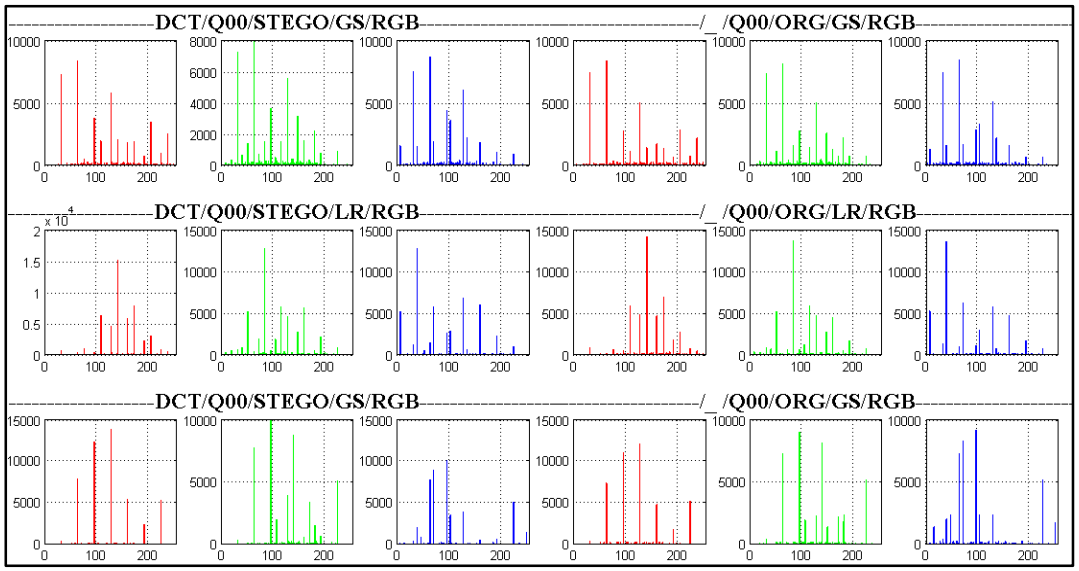


Figure 60: Histogram of DCT with Quality (0)

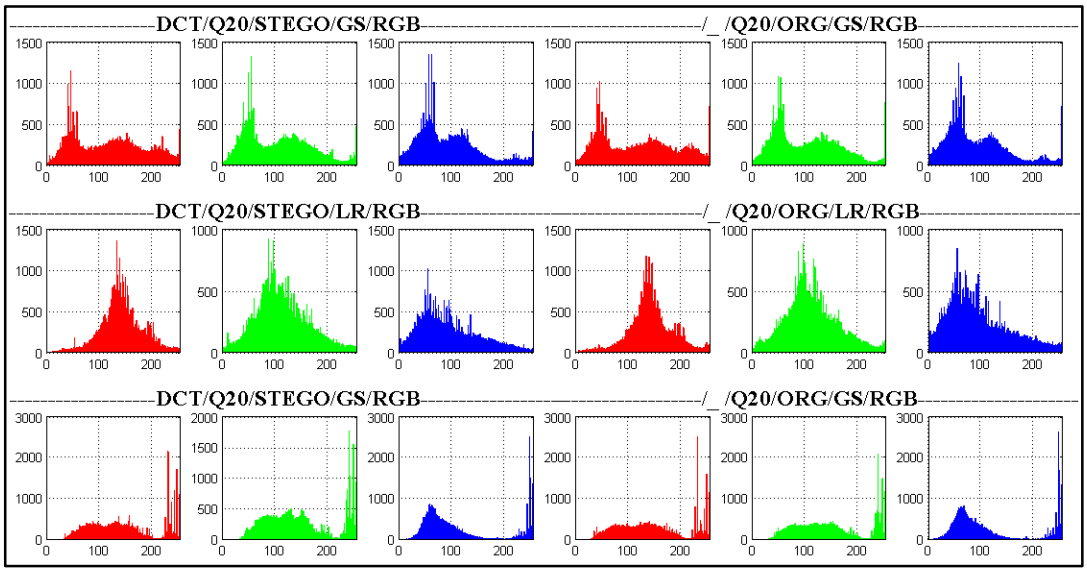


Figure 61: Histogram of DCT Quality (20)

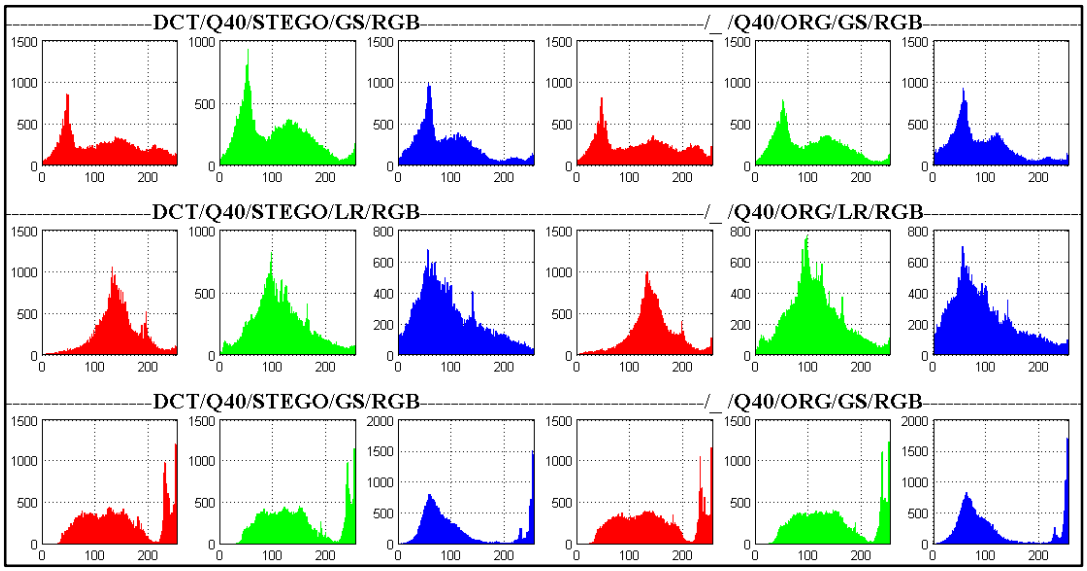


Figure 62: Histogram of DCT Quality (40)

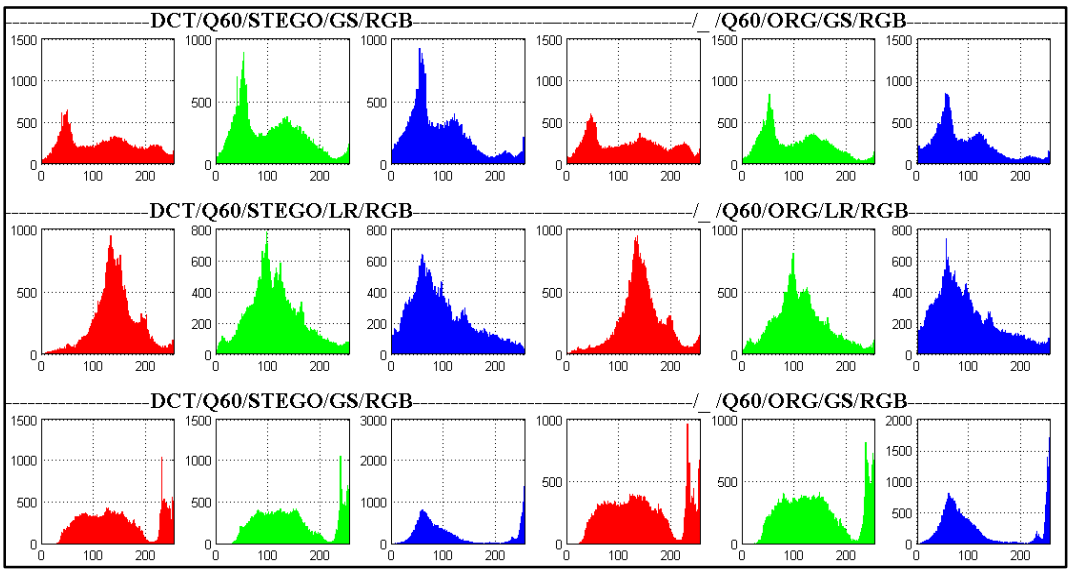


Figure 63: Histogram of DCT Quality (60)

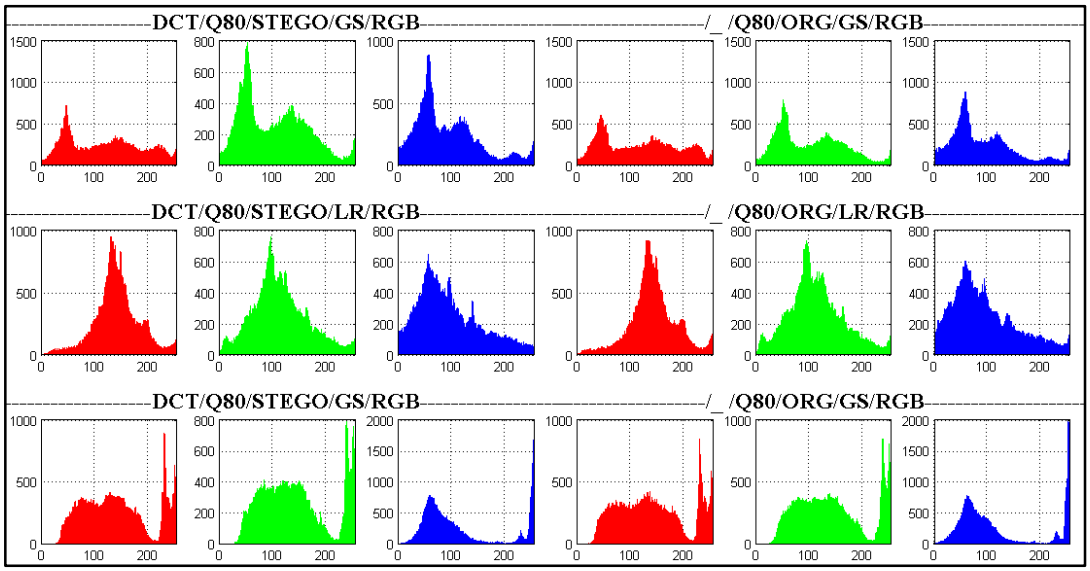


Figure 64: Histogram of DCT Quality (80)

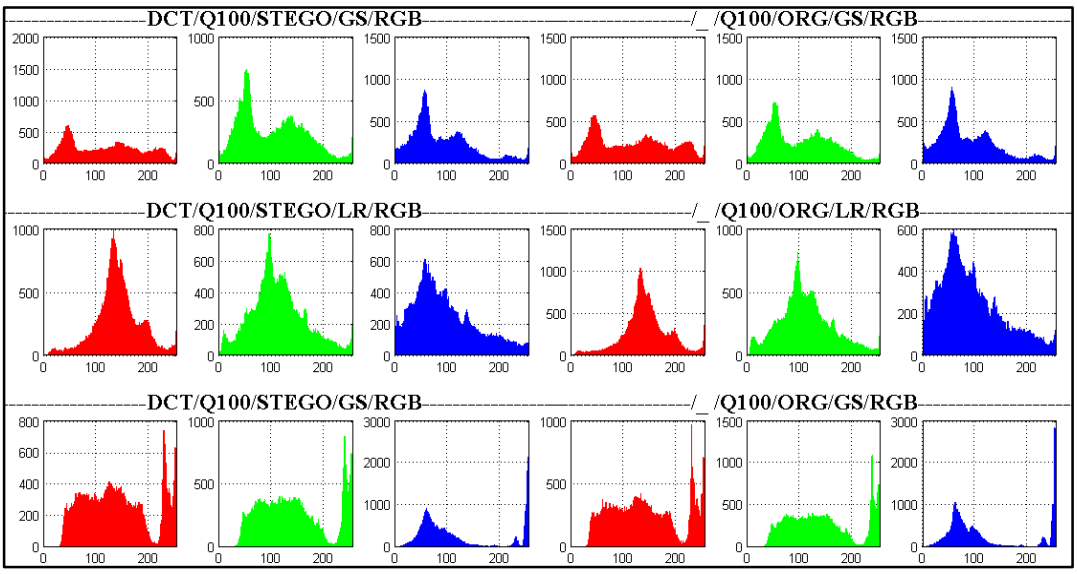


Figure 65: Histogram of DCT Quality (100)

Chapter Four

4. Discussion

In this chapter we discuss the results that have been calculated using MSE parameter that are applied to the original images with the resulting images (stego-images) which are obtained from the concealment processes by using the algorithms that is proposed in this research.

In the beginning, we have presented the MSE results that have been taken by applying LSB algorithm on the original images when using one[1], two[2] or three[3] bits, as well as the use of single color (red, green or blue) or three colors (RGB).

Later, we offered the MSE results that are obtained through the use of BPCS algorithm that are applied on the original images which were based on the threshold value of BPCS algorithm.

Finally, we have exposed the results of MSE parameter that have been taken by using the DCT algorithm based on the quality parameter of DCT algorithm.

"Although we are aware of the subjectivity of the results, we report our sense of vision on the stego images as well. Interested readers can find the stego images in the attached CD to have their own sense of comparison."(As mentioned previously as well).

4.1 Substitution Technique (LSB method)

In Table4, we have presented the results of MSE parameter that have been obtained using LSB algorithm. Through Table4, one can notice that the error values of a single color that is experimented on three images (as can be seen) are increasing with the increasing number of bits insertion, while, the increasing error values for all the stego-images are considered as slight which is between 0.0881% to 1.9801% in the average. However we found it imperceptible with the human eye. Nevertheless, the RGB error rate values as described in Table 5 are higher than the error rate for a single color and it is also considered as a small percentage and cannot be observed by the Human Visual System as well.

[1] 1-LSB=8th bit of byte (color).
[2] 2-LSB=7th and 8th bits of byte (color).
[3] 3-LSB=6th, 7th and 8th bits of byte (color).

LSB	Group_of_student			Living_room_home_house			Spring_sunshine_may		
Bits	Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
One	0.0903%	0.0888%	0.0898%	0.0891%	0.0896%	0.0901%	0.0886%	0.0881%	0.0884%
Two	0.4447%	0.4459%	0.4437%	0.4346%	0.4458%	0.4431%	0.4454%	0.4356%	0.4345%
Three	1.9801%	1.9563%	1.9532%	1.9009%	1.938%	1.9334%	1.8638%	1.8985%	1.7408%

Table 4: MSE Results of LSB Method with One Color

LSB	Group_of_student	Living_room_home_house	Spring_sunshine_may
Bits	RGB	RGB	RGB
One	0.2686%	0.2679%	0.2650%
Two	1.3345%	1.3231%	1.3194%
Three	5.8868%	5.7983%	5.4970%

Table 5: MSE Results of LSB Method with RGB Color

As observed in Figure 66, Figure 67 and Figure 68, the error rate values are almost equal in all of the resulting images when 1-LSB and 2-LSB on one color are applied. Whereas, we also note that Spring_sunshine_may resulted with lower error rates with respect to the other two images when compared with their MSE results in all four LSB variants. The researcher thinks that the characteristics of the images used might have effect on the results.

In order to find more accurate details of the results and their differentiation we need to use other images with different characteristics, and that is outside the limits of this research.

In Figure 69, it is clear that the error rates of the stego images in 1-LSB and 2-LSB are equal for all the images. While the error ratio is differentiated to some extent when we use 3-LSB, especially in the Spring_sunshine_may image where the error values are lesser than the other two images. The rationale for this effect may be the image characteristics (as for the differentiations in Figure 66, 67 and 68 as well).

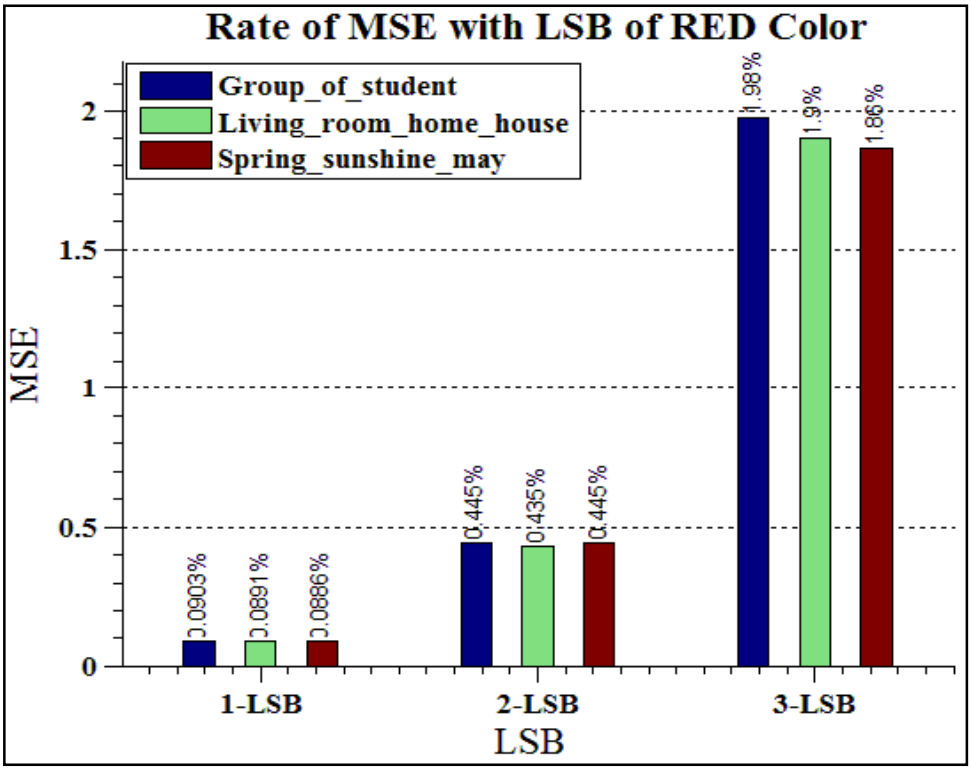


Figure 66: MSE with LSB of Red Color

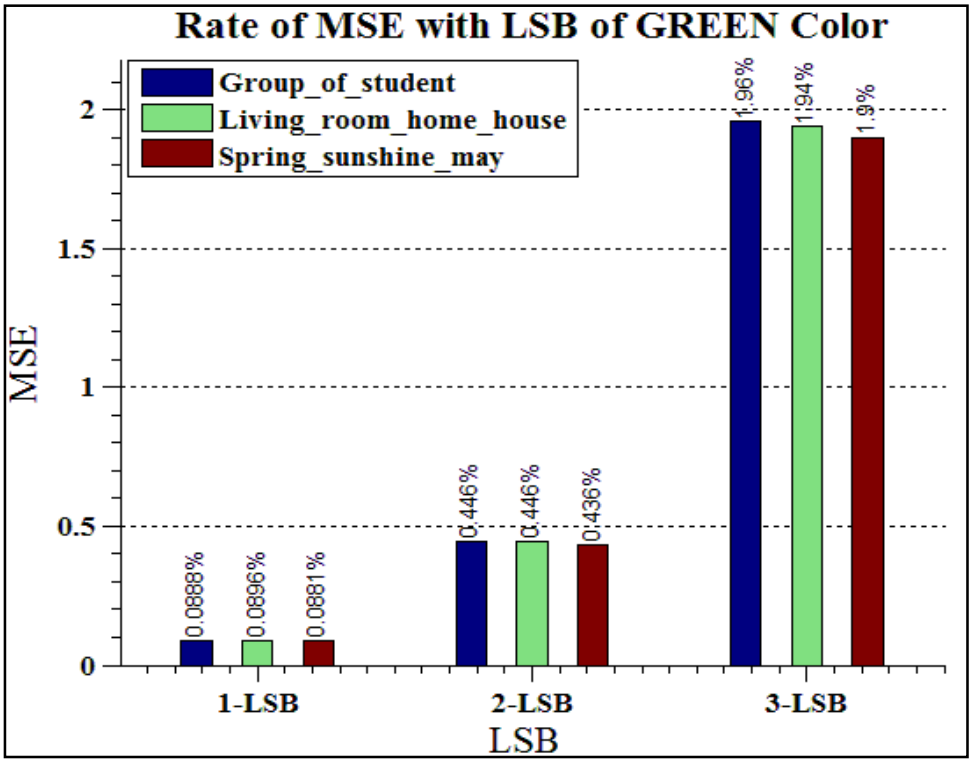


Figure 67: MSE with LSB of Green Color

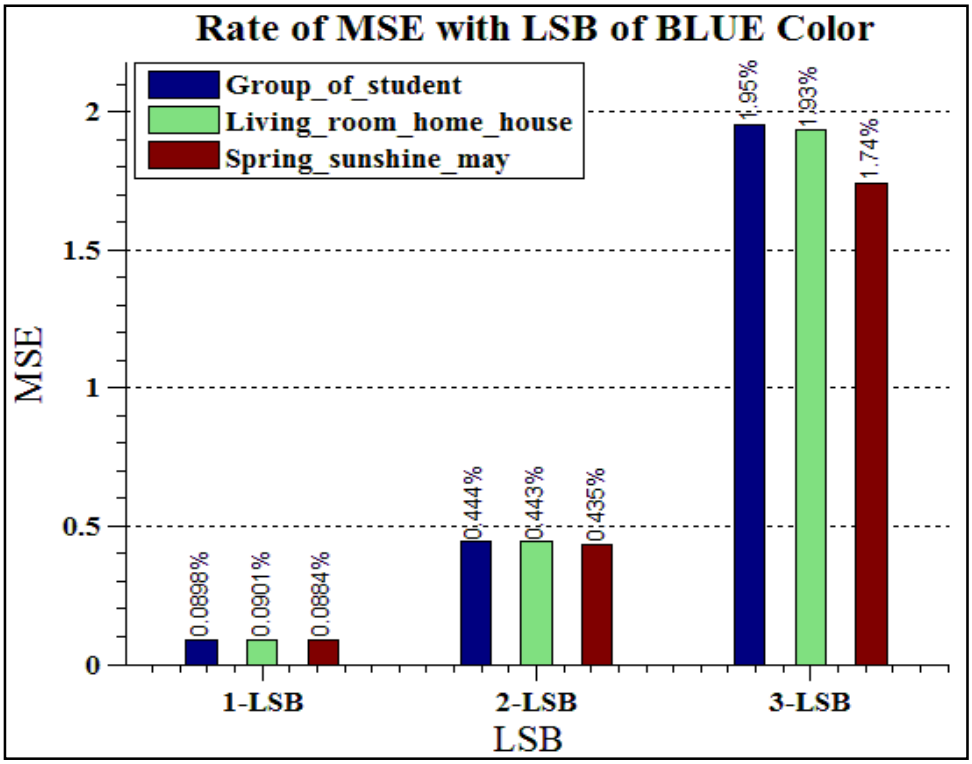


Figure 68: MSE with LSB of Blue Color

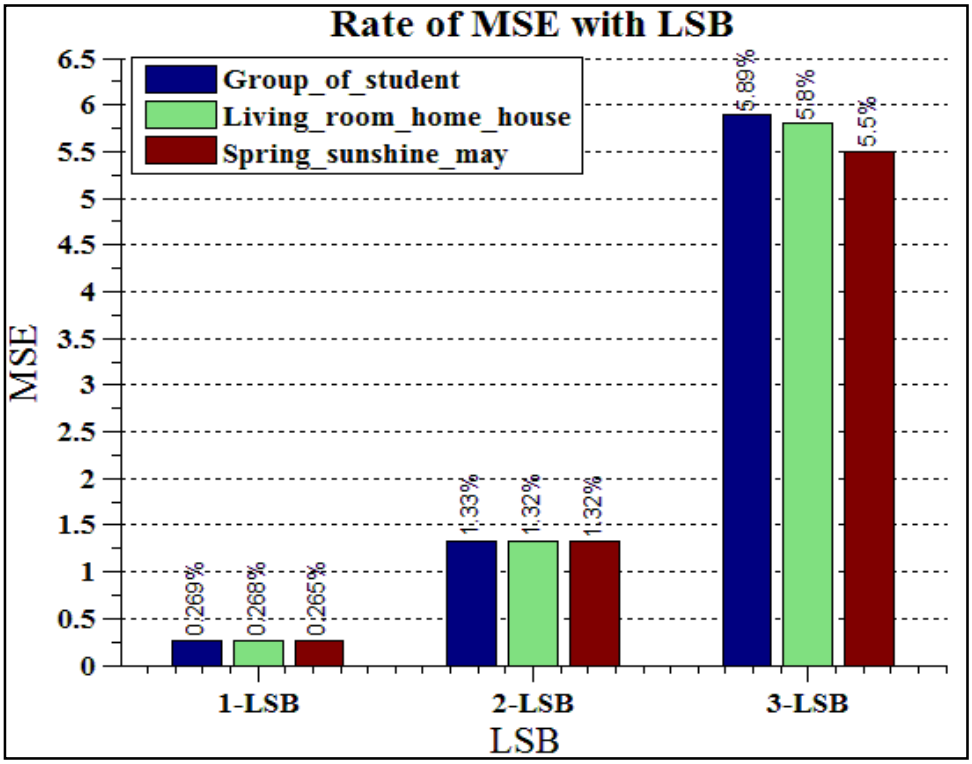


Figure 69: MSE with LSB of RGB Color

4.2 BPCS Method

As noted in Table6 the results of MSE that are acquired by applying this algorithm is inversely proportional with the increasing value of the threshold, for example, the error rate values is higher for all the resulting images when the threshold value is five and the error value decreases as the threshold value increases. The reason of this result is due to the characteristics of this algorithm. Since the algorithm inserts the message in more noisy sectors of the image as the threshold increases. This means any increase in the threshold value of this algorithm leads in finding the most complicated region in the image.

Threshold	Group_of_student	Living_room_home_house	Spring_sunshine_may
5	65.7564%	68.5856%	60.8435%
10	65.122%	67.0347%	60.0367%
15	62.909%	61.3456%	57.3337%
20	61.1615%	59.0472%	54.6625%
25	50.8781%	49.916%	48.0202%
30	43.9612%	41.4487%	42.6388%
35	37.7359%	38.5485%	42.7025%
40	33.5039%	30.8883%	39.6302%
45	27.3793%	24.4784%	35.1589%
50	19.8973%	17.3382%	29.6809%
55	11.1635%	10.2082%	20.3199%

Table 6: MSE Results of BPCS Method

In Figure70, we note that the error rate is decreasing gradually in all the images (with respect to threshold values). On the other hand, for the image of Spring_sunshine_may the error rate gradually decreased up to threshold 30, after this point the error rate slightly increased until the threshold 35. From this point on, the error rate declined in parallel with the other images. This observed discrepancy between thresholds 30 and 35 may stem from the visual characteristics of Spring_sunshine_may image. On the other hand, in order to be safe from a mishandling in the experiment, we conducted message insertion with BPCS method several times on Spring_sunshine_may image. However, our experimental setting was not designed to test this diversification. Thus to explain this effect more sophisticated experiments are needed to be conducted.

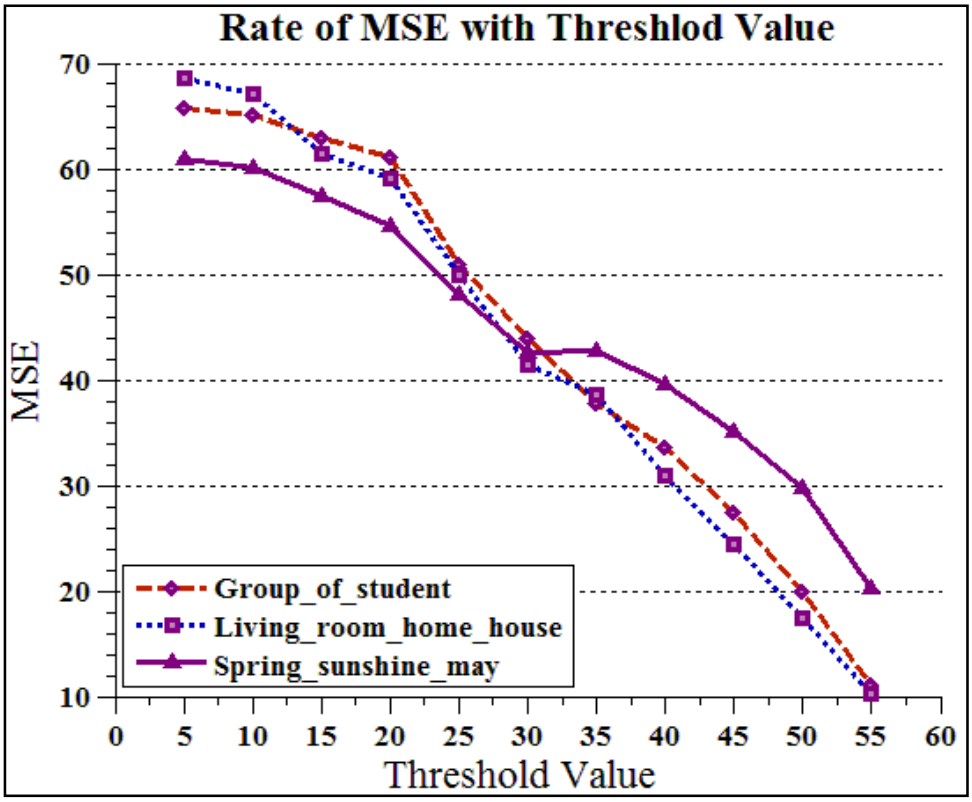


Figure 70: MSE with BPCS Method

4.3 DCT Technique (F5 Method):

The Table 7 shows the MSE values for the stego images except for "0" quality. We excluded the zero quality results from the discussion, because the images were distorted to the point that the cover image cannot be recognized by the human eye, and we used the results only to sequence the quality values of the algorithm that started from zero quality and increased each time by 20 steps until 100. It is noted that the error values decreases wherever the quality value in the algorithm increases. This is due to the expected result of the algorithms that are looking for the best places in the image for the purpose of inserting the data in them.

Figure 71 shows the MSE results of stego images for six different quality values in sequence. It can be noted that for the images Living_room_home_house and Group_of_student, there are clear decline in error values as the quality of the images increase. For Spring_sunshine_may image, as observed, the error value remains almost constant between the quality values 20 to 80.This difference is contrary to the results of error rates of other images. According to the researcher this effect may be due to the characteristics of this image as well. Therefore, as long as more focused

experiments with different cover image characteristic are not conducted we cannot prove or refute this suggestion.

Quality	Group_of_student	Living_room_home_house	Spring_sunshine_may
0	33.8045%	25.396%	16.6332%
20	38.501%	31.2491%	35.0562%
40	30.7981%	26.4137%	32.9387%
60	25.7269%	22.0955%	32.1279%
80	20.246%	17.2534%	32.9285%
100	5.8947%	4.4901%	13.6744%

Table 7: MSE Results of F5 Method

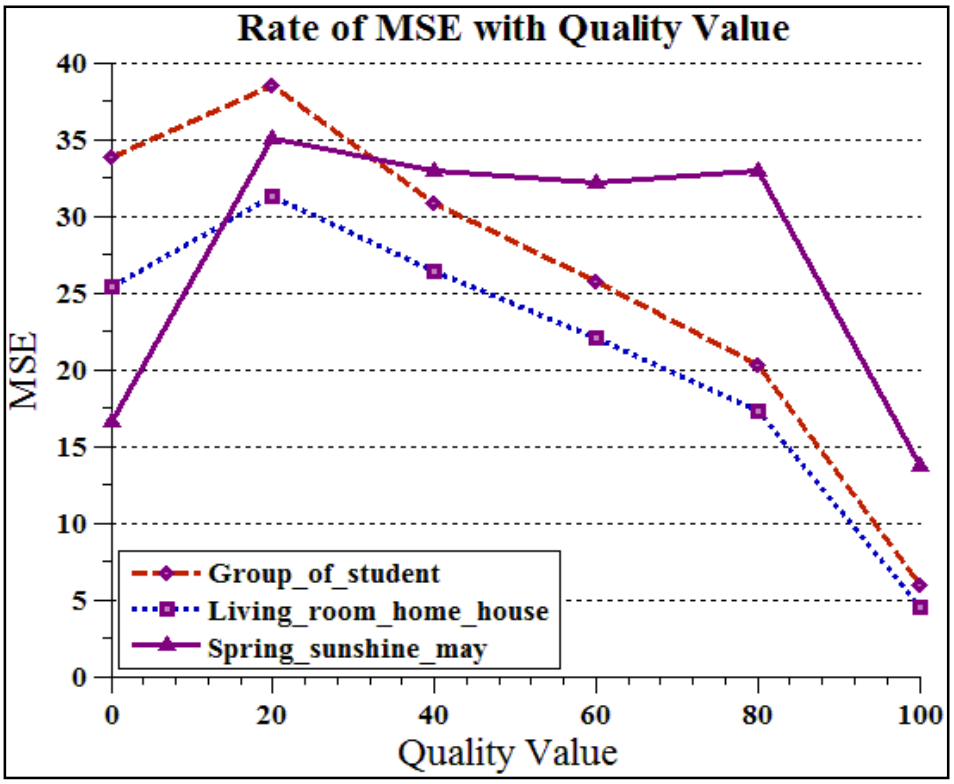


Figure 71: Graph Comparison of MSE Results of F5 Method

As a summary from the above results, we note that the error rate values that resulting from LSB is less than that of BPCS and DCT techniques. In other word, the visual distortion of images that resulting from LSB technique is almost non-existent if when compared with the amount of image distortions in BPCS and DCT techniques.

In other dimension, the LSB technique works on simple part of the image as compared to the other techniques which work on all parts of the image, this leads that LSB cannot be compared to the other used techniques such as BPCS that has high and complex capacity where the information that hidden inside it cannot be detected. Also, LSB cannot be compared to DCT technique because DCT is complex algorithm and there is difficulty to detecting the information inside it but its capacity is less than the capacity of LSB.

Through the obtained results, we note that different images have distinctive effects on BPCS and DCT techniques. In other words, BPCS and DCT are effected via the visual characteristics of the images. This change is due to that both algorithms are interacting with characteristics of the images to look for the best regions in order to hide the data.

Chapter Five

5. Conclusion

5.1 Conclusion

Steganography is an important tool in hiding the information either during transfer or in storage. It was discovered a long time ago as a cryptographic method. Since then, it has been used to hide large amounts of data securely and reliably to some extent. The techniques that have been used in this research are the most important and commonly used techniques in steganography, namely LSB, BPCS and DCT.

LSB is the easiest one to use for concealments and extraction operations with respect to the other two methods in this study. The advantage of BPCS method is its ability to store large amount of data which exceeds the amount of data that can be stored using the LSB method. At this research both LSB and BPCS are experimented in spatial domain with the bitmap images of 24 bit color.

The last method used in this study is DCT which is the most complicated one in complexity when compared with the LSB and BPCS methods. Because it relies heavily on the frequency domain representation and it uses JPEG-image compression. However, the amount of data that can be stored by using DCT method is less than the amount of data that can be stored by using the other two methods and all this has been proven by the practical experiments that were conducted in this research. The main objective of this research is to study the information hiding efficiency of selected steganography algorithms on the images with different visual characteristics.

According to the results that were obtained in MSE analysis and by visual comparisons on histograms of the original and the stego-images, one can perceive the distinctive effects of different algorithms and also varying influence of the images with different visual characteristics. However, by looking to the results of MSE parameter that we have obtained, it is clear that the information hiding efficiency of LSB algorithm is not effected by different visual characteristics of the cover images which were in bitmap format if we use 1-LSB, 2-LSB and 3-LSB.

In using BPCS there are two major factors that affect the results obtained. First one is the threshold value and the other is the properties of the image used. In order to express

more strict results on the effect of image characteristics in BPCS algorithm it is better to test it significantly wide variety of images with different visual characteristics.

On the other hand, from the results that we have obtained it is clear that the error rate values of DCT algorithm is less than the error rate values of BPCS algorithm with respect to the increasing quality factor values of the DCT algorithm. Nevertheless, the effects of DCT algorithm that when it been applied it cause double compression on the stego image and that accuse decreasing the size of the stego image because the more of AC coefficients of DCT algorithm are decreasing to zeros and most of non-zero AC coefficients are decreased. Therefore, this existence of these errors is due to the rate number of the coefficients that been changed that initiated by the F5 method embedding to the number of the all non-zero AC coefficients in the cover image as compared to the stego image which has the same quality factor value.

5.2 The Recommendation

Through the experiments that we have done, we suggest to use different types of secret messages like images, audio video or even documents in order to compare the effect of the hidden information on the stego-images. Also, in this work we used pure steganography in order to examine its effect we recommend using stego key in the embedding operation as well. On the other hand, in order to improve and verify the conclusions in this study, it would be better to conduct these experiments with larger number of cover images of different visual characteristics.

5.3 Future Work

Through our study we observed that there is a correlation between three properties that are perceptually, embedding capacity and robustness. New technologies must be developed to manage the three properties at a high level. Depending on this we proposed the following:

- Use of wavelet transform technique to increase the embedding capacity that works on maintaining the robustness of stego-image.
- It can be noted that most of steganography researches so far were towards embedding algorithms that generating images be close to the cover images as close as possible. All algorithms study the behavior of the cover images while they are cancelling the bit stream of the images. Therefore, our idea can be taken advantage of “Visual Cryptography”, which encrypts the message by

distributing the decryption key to different images, for example, the key message can be divided into appropriate combination of these images.

References

[1] Johnson, Neil F., and Sushil Jajodia. "Exploring steganography: Seeing the unseen." Computer 31.2 (1998): 26-34.

[2] Singh, Kh Manglem, et al. "Hiding secret message in edges of the image." Information and Communication Technology, 2007. ICICT'07. International Conference on. IEEE, 2007.

[3] Thampi, Sabu M. "Information hiding techniques: A tutorial review." ISTE-STTP on Network Security & Cryptography, LBSCE (2004).

[4] Dunbar, Bret. "A detailed look at Steganographic Techniques and their use in an Open-Systems Environment." Sans Institute 2002 (2002): 1-9.

[5] Brassil, Jack T., Steven Low, Nicholas F. Maxemchuk, and Lawrence O. Gorman. "Electronic marking and identification techniques to discourage document copying." Selected Areas in Communications, IEEE Journal on 13, no. 8 (1995): 1495-1504.

[6] Jayaram, P., H. R. Ranganatha, and H. S. Anupama. "Information hiding using audio steganography–a survey." The International Journal of Multimedia & Its Applications (IJMA) Vol 3 (2011): 86-96.

[7] Chapman, Mark, George I. Davida, and Marc Rennhard. "A practical and effective approach to large-scale automated linguistic steganography." In Information Security, pp. 156-165. Springer Berlin Heidelberg, 2001.

[8] Singh, Hitesh, Pradeep Kumar Singh, and Kriti Saroha. "A survey on text based steganography." In Proceedings of the 3rd National Conference, pp. 26-27. 2009.

[9] Morkel, Tayana, Jan HP Eloff, and Martin S. Olivier. "An overview of image steganography." In ISSA, pp. 1-11. 2005.

[10] Cheddad, Abbas, Joan Condell, Kevin Curran, and Paul Mc Kevitt. "Digital image steganography: Survey and analysis of current methods." Signal processing 90, no. 3 (2010): 727-752.

[11] Shih, Frank Y. Digital watermarking and steganography: fundamentals and techniques. CRC Press, 2007.

[12] Gupta, Shailender, Ankur Goyal, and Bharat Bhushan. "Information hiding using least significant bit steganography and cryptography." International Journal of Modern Education and Computer Science (IJMECS) 4, no. 6 (2012): 27.

[13] Salomon, David. Data compression: the complete reference. Springer Science &Business Media, 2004.

[14] Katzenbeisser, Stefan, and Fabien Petitcolas. Information hiding techniques for steganography and digital watermarking. Artech house, 2000.

[15] Chan, Chi-Kwong, and Lee-Ming Cheng. "Hiding data in images by simple LSB substitution." Pattern recognition 37, no. 3 (2004): 469-474.

[16] Hariri, Mehdi, Ronak Karimi, and Masoud Nosrati. "An introduction to steganography methods." World Applied Programming 1, no. 3 (2011): 191-195.

[17] Hong-Juan Zhang, Hong-Jun Tang,"A Novel Image Steganography Algorithm Against Statistical Analysis", Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007.

[18] Kawaguchi, Eiji, and Richard O. Eason. "Principles and applications of BPCS steganography." In Photonics East (ISAM, VVDC, IEMB), pp. 464-473. International Society for Optics and Photonics, 1999.

[19] Krenn, Robert. "Steganography and steganalysis." Retrieved September 8 (2004): 2007.

[20] Gonzalez, Rafael C., and Richard E. Woods. "Digital image processing 3rd edition." (2007).

[21] Fridrich, Jessica. Steganography in digital media: principles, algorithms, and applications. Cambridge University Press, 2009.

[22] Image, Gray-Scale. "Effectual Data Secrecy using Complexity Segmentation and Differential Expansion on." IIJCS:Volume 2, Issue 11, November 2014.

[23] YESHWANTH SRINIVASAN, B. E. "High capacity data hiding system using BPCS steganography." PhD diss., Texas Tech University, 2003.

[24] Bandyopadhyay, Samir K., et al. "A tutorial review on steganography." International conference on contemporary computing. Vol. 101. 2008.

[25] ITU, Information technology -- Digital compression and coding of continuous-tone still images: Requirements and guidelines, Recommendation T.81, 1993.

[26] ITU-R Recommendation BT.601-2, Encoding Parameters of Digital Television for Studios(1982-1986-1990), [formerly CCIR Rec. 601-2] (Geneva: ITU, 1990)

[27] Bateman, Philip, and Hans Georg Schaathun. "Image steganography and steganalysis." Department Of Computing, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, Surrey, United Kingdom, 4th August (2008).

[28] Westfeld, A. (2001, January). F5—a steganographic algorithm. In Information hiding (pp. 289-302). Springer Berlin Heidelberg.

[29] Crandall, Ron. "Some notes on steganography." Posted on steganography mailing list (1998).

[30] Morkel, Tayana, Jan HP Eloff, and Martin S. Olivier. "An overview of image steganography." ISSA. 2005.

[31] Uruba, I. "Hiding text in image." PhD diss., M. Sc. Thesis, University of technology, Computer Science Department, 2001.

[32] Al-hamami, M. "Information hiding attack in Image." PhD diss., M. Sc. thesis, Iraqi commission for computer &Informatics, Informatics Institute for Postgraduate Studies, 2002.

[33] Fridrich J. and Goljan M., "Digital image steganography using stochastic modulation", Department of Electrical and Computer Engineering, SUNY Binghamton, Binghamton, NY, 13902-6000, USA,2003.

[34] Alawy S., "Robust Information Hiding Techniques Using JPEG" University of Almusstnsry, Ms.cthesis in computer Science, 2004.

[35] Abusharekh, Ashraf M. Comparative analysis of multi-precision arithmetic libraries for Public Key Cryptography. 2004.

[36] Davidson, Ian, and Goutam Paul. "Locating secret messages in images." In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 545-550. ACM, 2004.

[37] Ibraheem, Aiad, and A. Sada. "Hiding data using LSB-3." J. Basrah Researches (Sciences) 33, no. 4 (2007): 81-88.

[38] Naji A. and Zaidan A., "Cryptography and Steganography", IJCSNS International Journal of Computer Science and Network Security, Electrical and Computer

Engineering Department, International Islamic University Malaysia, 53100 Gombak, Kuala Lumpur, Malaysia. VOL.9 No.5, May 2009.

[39] Balogová, B. ‘Vocabulary as a reflection of life wisdom’. Élan Vital in the area of intergenerational relationships. Proceedings of the conference with international participation 15.01.2010 in Presov. Faculty of Prešov University, Prešov. 2010. pp. 182-188. ISBN 978-80-555-0198-7.

[40] Noda, Hideki, Michiharu Niimi, and Eiji Kawaguchi. "Steganographic Methods Focusing on BPCS Steganography." In Intelligent Multimedia Data Hiding, pp. 189-229. Springer Berlin Heidelberg, 2007.

[41] Gonzalez, Rafael C., Richard Eugene Woods, and Steven L. Eddins. Digital image processing using MATLAB. Pearson Education India, 2004.

[42] Fridrich, Jessica, Miroslav Goljan, and Dorin Hogea. "Steganalysis of JPEG images: Breaking the F5 algorithm." In Information Hiding, pp. 310-323. Springer Berlin Heidelberg, 2003.

[43] Wang, Z., Bovik, A. (2009), “Mean Squared Error: Love it or Leave it? ANew Look at Signal Fidelity Measures”, IEEE Signal Processing Magazine Vol.26, 98-117.