Republic of Iraq
Ministry of Higher Education & Scientific Research
University of Baghdad
College of Engineering

RGBA Image Steganography Based on AES Technique

A Thesis
Submitted to the College of Engineering
in the University of Baghdad
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
in Electronics and Communications / Computer Engineering

By
Nawar Sa'ad Erhaieyym

Supervised By

Dr. Firas Ali Sabir

Ramadhan
1436

July
2015

4 وأنزل الله عليك الكتاب والحكمة وعلمك ما

لم تكن تعلم وكان فضل الله عليك عظيما 4

(النساء 113)

صدق الله العلي العظيم

Committee Certification

We certify we have read this thesis entitled ***"High Capacity Steganographic Method Based upon Image"*** and as examining committee, examined the student ***(Nawar Sa'ad Erhaieyym)*** in its contents and that, in our opinion it meets the standard of a thesis for the degree of Master of Science in Electronics and Communications / Computer Engineering.

Signature:
Name: Prof. Dr. Nasser N. Khamiss
Date: / / 2015
(Chairman)

Signature:
Name: Assist. Prof. Dr. Sadiq J. Abou-Loukh
Date: / / 2015
(Member)

Signature:
Name: Dr. Buthaina M. Omran
Date: / / 2015
(Member)

Signature:
Name: Dr. Firas Ali Sabir
Date: / / 2015
(Supervisor)

Approved by the Dean of College of Engineering

Signature:
Name: Prof. Dr. Ahmed A. Mohammed
(Acting Dean)
Date: / / 2015

Supervisor Certificate

I certify that the preparation of the thesis entitled ***"High Capacity Steganographic Method Based upon Image"*** was done by Mr. (***Nawar Sa'ad Erhaieyym***) under my supervision in the Electronics and Communications / Computer Engineering Department in the University of Baghdad in partial fulfillment of the requirements for the degree of Master of Science in Electronics and Communications / Computer Engineering.

Signature:
Name: Dr. Firas Ali Sabir

Date: / / 2015
(Supervisor)

Dedication

This thesis is dedicated, with deepest love and everlasting respect, to numerous precious persons.

To my dearest mother for her continuous love, support , encouragement and sincere prayers which helped me to achieve my dream.

To my beloved Wife for her love and support and everything she did to help me and stand with me in all situations.

To my family, my sisters, my brother, my mother in law , my father in law for their support and patience throughout these stressful years .

To my friends, with all their support and help and encouragement.

To the soul of my dearest father which give me the inspiration to walk throw my path

Acknowledgment

Firstly, all praise is due to Allah without his immeasurable blessings and favours none of this could have been possible. During the course of this study, there have been numerous people who have provided me with guidance, inspiration and support for the completion of this work. I would like to acknowledge ***Prof. Dr. Tarik Zeyad Ismaeel*** *for his support and guiding me to accomplish this research. I am also indebted to my supervisor* ***Dr. Firas Ali Sabir*** *and* ***Dr. Yamaan Ismaeel Majeed*** *for their continuous encouragement and support at all stages of this study. Their involvement proved vital to the completion of this work. I would like to thank them for their useful and valuable advice, for their insightful and encouraging comments and suggestions throughout this study.*

In addition, I would like to express my thanks to the head and the staff of the Electronics and Communications Engineering Department and the head and the staff of the Computer engineering Department at the University of Baghdad and my colleagues for their help and support.

Also I would like to thank ***Assistant prof. Arjun Ramachandra Nichal*** *for his help, consultations and support during this work*

To all of you, Thank you.

Nawar Sa'ad Erhaieyym

June 2015

Abstract

This work presents techniques of a steganography system based on the Least Significant Bit (LSB) substitution, the secret message is an image (grayscale or color). In this work, Alpha channel is added to the cover image (RGB.jpg, where RGB stands for Red, Green, and Blue) to increase the bit-depth of the cover image which become RGBA.png image. Bit-Plane Slicing (BPS) technique applied to the secret image for compressing and decreasing the size of secret data to embed.

To increase the security, two encryption models were applied namely, Simple Private Key (SPK) method, and Advanced Encryption Standard (AES) method, where SPK represent a simple XOR encryption, while AES represent the current encryption standard used today.

Results and comparative studies revealed the effectiveness of the proposed technique in generating the stego images. The size of the secret message can be the same size of the cover image while the distortion of the stego image is very low and the stego image is closed related to the cover image. Results also show that in the embedding capacity of 100%, the normalized cross-correlation is very close to one and the PSNR is up to 38.199 dB for the grayscale secret image and 38.155 dB for color secret image in SPK model, while PSNR is up to 38.408 dB for grayscale image and 38.226 dB for color image in AES model.

In addition, a comparison with many steganography approaches based on Wavelet domain and spatial domain is done; the proposed steganography provides embedding capacity and quality of the stego image with PSNR value higher than the comparable methods. This work is programmed and simulated using the MATLAB language.

Contents

Abstract I
Contents II
List of Abbreviations V
List of Symbols VII
List of Figures IX
List of Tables XII

Chapter One: Introduction

1.1 General Overview 1
1.2 Motivation 6
1.3 Literature Survey 7
1.4 Aim of the Thesis 10
1.5 Thesis Organization 10

Chapter Two: Theory of Steganography and Encryption

2.1 Introduction 12
2.2 Theory of Steganography 12
2.2.1 Main Components of Steganographic Systems 15
2.2.2 The Basics of Embedding 17
2.2.3 Steganographic Protocols 17
2.2.4 Types of steganography 18
2.2.5 Steganography Techniques 21
2.2.6 Steganalysis 23
2.2.7 Steganography Attacks 24
2.3 Digital Imaging Concepts 25
2.3.1 Images and pictures 25
2.3.2 Color representation 25

2.3.3 Image definition 26
2.3.4 Image File Formats 27
2.4 Alpha Channel 28
2.4.1 Pre-multiplied Alpha 31
2.4.2 Non-pre-multiplied alpha 32
2.4.3 Alpha channel creation 33
2.4.4 RGBA Pictures 34
2.5 Bit-Plane Slicing 34
2.6 Advanced Encryption Standard (AES) 38
2.6.1 Inner Workings of a Round 40
2.6.2 AES Key Expansion 47
2.6.3 Equivalent Inverse Cipher 49

Chapter Three: The Proposed Steganography System

3.1 Introduction 50
3.2 The Transmitter Side 50
3.2.1 Preparation of the Cover Image 52
3.2.2 Preparation and Decomposition of Secret Image 54
3.2.3 Encryption of the secret image 59
3.2.4 Proposed Embedding Algorithm 62
3.3 The Receiver Side 69

Chapter Four: Results and Discussion

4.1 Introduction 73
4.2 Measures of Quality of the Stego Images 74
4.2.1 Peak-Signal-to-Noise-Ratio (PSNR) 74
4.2.2 Mean Square Error (MSE) 74
4.2.3 Normalized Cross-Correlation (NCC) 75
4.2.4 Average Difference (AD) 75

4.3 The Results 76
4.3.1 Results of SPK model 76
4.3.2 Results of AES model 82
4.4 Discussion of the results 88
4.4.1 The capacity of the proposed system 88
4.4.2 The invisibility of the proposed system 92
4.4.3 The security of the proposed system 95

Chapter Five: Conclusions and Suggestions for Future Work

5.1 Conclusions 97
5.2 Suggestions for Future Work 98

References 100

List of Publications 107

List of Abbreviations

AD	Average Difference
AES	Advanced Encryption Standard
BBS	Blum Blum Shub Pseudo Random Number Generator
bKGD	Background Chunk
BPCS	Bit-Plane Complexity Segmentation
BPS	Bit-Plane Slicing
DCT	Discrete Cosine Transform
DD DT DWT	Double Density Dual Tree Discrete Wavelet Transform
DES	Data Encryption Standard
DWT	Discrete Wavelet Transform
FIPS PUB	Federal Information Processing Standards Publication
GIF	Graphics Interchange Format
IDEA	International Data Encryption Algorithm
INP	Interpolation by Neighboring Pixels
ISB	Intermediate Significant Bit
JPEG	Joint Photographic Experts Group
LSB	Least Significant Bit
MSE	Mean Square Error
MP3	MPEG-1 or MPEG-2 Audio Layer III
MPEG	Moving Picture Experts Group
NCC	Normalized Cross-Correlation
NIST	National Institute of Standards and Technology
OSI	Open Systems Interconnection model
PNG	Portable Network Graphics

PSNR	Peak Signal to Noise Ratio
RC4	Rivest Cipher 4
RGB	Red, Green, Blue color space
RGBA	Red, Green, Blue, Alpha color space
RHC	Rational Embedding Capacity
RSA	Rivest-Shamir-Adleman cryptosystem
SPK	Simple Private Key
SS	Spread Spectrum
SSIS	Spread Spectrum Image Steganography
TIFF	Tagged Image File Format
tRNS	Transparency Chunk
XOR	Exclusive OR operation
YCbCr	the luminance component, the blue-difference, and red-difference color space
YIQ	Luminance, in-phase, quadrature color space

List of Symbols

The symbol	The Discretion
aA	the amount of color contributed by a pixel
a_a	The length the first secret string
A	The matrix of MixColomn
A	The original cover image
A^{-1}	The matrix of InvMixColomn
b	Coverage of the pixel of the new geometrical object
bB	The color contributed to the new geometrical
b_b	The length of the second secret string
C	The composite color
c_c	The length of the third secret string
$C(i,j)$	The number of pixels of cover image
g	The complex function of Key expansion
G	The secret grayscale image
H	The height of the stego image and the cover image
K	The number of channel
len	The length of sec_{total}
L	the number of the gray scale levels in the two images (stego and cover)
m	The width of the image
N	The number of LSB bits can be used for embedding
n	The length of the image
p	The pixel value
Pl_k	Bit-Plane of the image

$\boldsymbol{R}$	The secret color image
$\boldsymbol{Rcon}[\boldsymbol{j}]$	The round constant
$\boldsymbol{S}$	The state array
$\acute{\boldsymbol{S}}$	The state array after the mix columns operation
$\boldsymbol{S}(\boldsymbol{i},\boldsymbol{j})$	The number of pixels of secret image
$\boldsymbol{St}(\boldsymbol{i},\boldsymbol{j})$	The number of pixels of stego image
W	The width of the stego image and the cover image

List of Figures

Figure (1.1)	Trade-off between embedding capacity, undetectability, and robustness in data hiding.	4
Figure (2.1)	A data hiding example.	13
Figure (2.2)	A model of the steganographic process.	13
Figure (2.3)	The different embodiment disciplines of Information Hiding.	15
Figure (2.4)	General Principle of Image Steganographic System.	16
Figure (2.5)	Pixels and bit representation of a greyscale image with bit depth 8.	26
Figure (2.6)	Pixels and bit representation of a 24-bit colour image using the RGB colour model.	27
Figure (2.7)	Bit-plane representation of an 8-bit image.	35
Figure (2.8)	An 8-bit fractal image.	35
Figure (2.9)	The eight bit planes of the image in Figure (2.8). The number at the bottom, right of each image identifies the bit plane.	36
Figure (2.10)	Bit-plane slicing representation.	37
Figure (2.11)	The overall structure of AES algorithm.	39
Figure (2.12)	Data structures in the AES algorithm.	40
Figure (2.13)	Substitute Bytes Stage of the AES algorithm.	41
Figure (2.14)	ShiftRows stage.	43
Figure (2.15)	MixColumn stage.	44
Figure (2.16)	Key expansion pseudocode.	47
Figure (2.17)	AES key expansion.	48
Figure (2.18)	AES encryption round.	49

Figure (3.1)	The main block diagram at the transmitter side of the SPK model.	51
Figure (3.2)	The main block diagram at the transmitter side of the AES model.	52
Figure (3.3)	The cover image.	53
Figure (3.4)	The resulted image if the value of Alpha is (Case 1) all ones, (Case 2) all zeros.	54
Figure (3.5)	Grayscale secret image.	55
Figure (3.6)	A color secret image.	57
Figure (3.7)	Encryption of Secret image using SPK.	59
Figure (3.8)	The encryption of secret image using AES algorithm.	60
Figure (3.9)	the secret image and the encrypted secret image.	61
Figure (3.10)	Example of embedding sequence.	62
Figure (3.11)	Flow chart of hiding process for SPK model.	65
Figure (3.12)	Flow chart of hiding process for AES model.	68
Figure (3.13)	The Main block diagram of the reciever side using SPK model.	69
Figure (3.14)	The Main block diagram of the reciever side using AES model.	69
Figure (3.15)	Flow chart of the extraction process using SPK model.	71
Figure (3.16)	Flow chart of the extraction process using AES model.	72
Figure (4.1)	Original image, secret image, stego-image, and Stego image of SPK-G256.	77
Figure (4.2)	Original image, secret image, stego-image of SPK-G256-2.	78
Figure (4.3)	Original image, secret image, stego-image of SPK-G512.	79

Figure (4.4)	Original image, secret image, stego-image of SPK-R256.	80
Figure (4.5)	Original image, secret image, stego-image of SPK-R256-2.	81
Figure (4.6)	Original image, secret image, stego-image of SPK-R512.	82
Figure (4.7)	Original image, secret image, stego-image of AES-G256.	83
Figure (4.8)	Original image, secret image, stego-image AES-G256-2.	84
Figure (4.9)	Original image, secret image, stego-image AES-G512.	85
Figure (4.10)	Original image, secret image, stego-image of AES-R256.	86
Figure (4.11)	Original image, secret image, stego-image AES-R256-2.	87
Figure (4.12)	Original image, secret image, stego-image of AES-R512.	88
Figure (4.13)	The cover image (Lena).	89
Figure (4.14)	Comparison of capacity between proposed method and other methods.	90
Figure (4.15)	The cover image, the secret image, the stego image, and the recovered secret image.	91
Figure (4.16)	The stego images obtained from embedding process with changing the number of bits used to embed.	93
Figure (4.17)	The comparison of PSNR between proposed method and other methods.	94
Figure (4.16)	Encrypted secret image.	96

List of Tables

Table (2.1)	Summary of differences among watermarking, fingerprinting and steganography.	14
Table (2.2)	The s-box.	42
Table (2.3)	The inverse s-box.	42
Table (4.1)	The results of embedding (256*256) gray-level secret image into (512*512) RGB cover image- SPK model.	76
Table (4.2)	The results of embedding (256*512) gray-level secret image into (512x512) RGB cover image – SPK model.	77
Table (4.3)	The results of embedding (512*512) gray-level secret image into (512*512) RGB cover image – SPK model.	78
Table (4.4)	The results of embedding (256*256) color secret image into (512*512) RGB cover image – SPK model.	79
Table (4.5)	The results of embedding (256*512) color secret image into (512*512) RGB cover image – SPK model.	80
Table (4.6)	The results of embedding (512*512) color secret image into (512*512) RGB cover image – SPK model.	81
Table (4.7)	The results of embedding (256*256) gray-level secret image into (512*512) RGB cover image – AES model.	82
Table (4.8)	The results of embedding (256*512) gray-level secret image into (512*512) RGB cover image – AES model.	83

Table (4.9)	The results of embedding (512*512) gray-level secret image into (512*512) RGB cover image – AES model.	84
Table (4.10)	The results of embedding (256*256) color secret image into (512*512) RGB cover image – AES model.	85
Table (4.11)	The results of embedding (256*512) color secret image into (512*512) RGB cover image – AES model.	86
Table (4.12)	The results of embedding (512*512) color secret image into (512*512) RGB cover image – AES model.	87
Table (4.13)	comparison of capacity percentage between the proposed method and other methods.	90
Table (4.14)	The results of embedding (512*512) color secret image the same image.	91
Table (4.15)	The differences between SPK and AES in the effecton the proposed system.	95

Chapter One

Introduction

1.1 General Overview

In the digital world, data is the heart of computer communication and global economy. To transfer data securely, the concept of data hiding has enticed researchers to develop creative methods to protect data from falling into wrong hands. Digital data can be transferred over computer networks from one point to another without any errors and interferences. The spread of digital media raised concern over the years about the possibility of attacking and manipulating the data by unauthorized persons. As a result of internet spread around the world, the motivation of hiding secret information on different digital multimedia carriers and secure communication is raised. Techniques for data hiding are increasing continuously with more advanced methods. The security of data has become a big concern due to the increasing of data communication over computer networks. Therefore, to protect information from unauthorized access and manipulation, data confidentiality and integrity are required [1].

To establish a communication through a public channel, cryptography has been introduced as a mechanism for setting up a secured logical channel through an insecure physical channel. However, cryptography has some drawbacks, notably that the just existence of an encrypted data may be motive suspicions in itself. If an unauthorized person intercepted an encrypted message in a secret communication, he might suspect that the message should be valuable, since there is someone cost an effort to encrypt this message in the first place [2].

There is another disadvantage of cryptography, which is the constraints that have been forced by some countries, these constrains consider to be a major drawback when cryptography techniques is to be used by remote users. A number of governments have embarked rules to enforce limitations on the strength and complexity of cryptographic techniques or completely prohibit it. This is mostly due to law enforcement anxiety of not being able to obtain intelligence by intercepting the data [3].

There are alternative techniques that can improve upon these constraints should be inspected. Steganography is one of these techniques that tries to protect confidential data from unauthorized persons [4].

Steganography is the science of hiding data in an imperceptible way in a cover media file. The word "Steganography" comes from Greek origin, which means "covered or hidden writing". The prime goal of steganography is to conceal the presence of the secret information in the cover media file. Traditional techniques of steganography involve the use of microdots, invisible inks, etc. Nowadays, steganography methods attempt to make use of the digital media like images, audio, video, etc. [5].

Steganography and cryptography are two members in the security systems family. Cryptography scrambles information by applying specific cryptographic techniques for transforming the secret information into ambiguous form. On the other side, Steganography conceals the secret information so it cannot be discovered. A cipher text message may cause suspicions on the recipient part, while the "invisible" message which generated by steganographic techniques will not. Any party involving in the secret communication session can always use a cryptographic system to encrypt the message before embedding it into the cover media to achieve more security. However, once the existence of hidden data is suspected or

detected, the concept of steganography is breaking dawn, even though the secret content of the message is not extracted or decrypted.

According to N. F. Johnson [6]: "Steganography's niche in security is to supplement cryptography, not replace it. If a hidden message is encrypted, it must also be decrypted if discovered, which provides another layer of protection ". However, the major challenges of an effective steganography system are [5]:

1. **Security of the hidden information**: For the purpose of avoid stirring the suspicions of eavesdroppers, while shunning the suspicious inspection of algorithmic detection, the hidden information should be invisible both statistically and perceptually.
2. **Capacity of the carrier**: Steganography aims at hiding communication and thus commonly requires adequate embedding capacity. On the contrary of watermarking, which requires a limited embedding capacity to embed only a small amount of information. Requirements for higher capacity and secure communication are often contradictions. A trade-off has to be sought depending on the specific application algorithms.

Watermarking is another technique of data hiding in digital media, and it is defined as "the process of embedding auxiliary information into a digital cover signal with the aim of providing authentication information"[5].

Steganography and watermarking differ from each other in many aspects including specification, purpose, detection, and extraction techniques. The major difference between watermarking and steganography is that the objective of the watermarking focus on protecting the carrier signal, while the embedded information provides copyright protection. On the other side, the objective of steganography is to protect

the embedded information, and the carrier media acts as an innocuous incognito chosen arbitrarily by the sender due to its technical suitability. Additionally, the presence of the watermark is mostly visible to the recipient, and any endeavor to remove or manipulate the embedded information makes the carrier useless. The fundamental key requirement of steganography is algorithmic and perceptual undetectability. On the contrary, than it is in the watermark, robustness is not crucial in steganography [5].

The key difference between Steganography and Watermarking in terms of the three requirements of capacity, undetectability, and robustness shown in Figure (1.1).

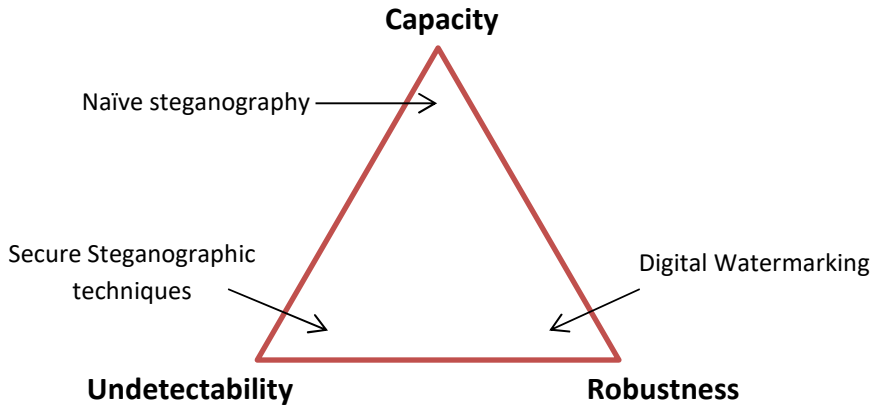


Figure (1.1): The trade-off among embedding capacity, undetectability, and robustness in data hiding systems [5].

Based on the embedding domain, digital steganography techniques are classified into spatial domain and transform domain. In spatial domain method, the secret message is embedded directly into the cover image by changing its pixel values (e.g. LSB method). General vantages of spatial domain technique are [7]:

1) The degradation of the cover image is low.
2) The embedding capacity is high.
3) Low implementation Complexity.

In the transform domain, this technique tries to encode secret information bits into the image coefficients. Analogous techniques can also provide large embedding capacity for steganography.

There are many applications for data hiding; some of these applications are [8]:

• **Advanced data structures**. Data structures can be devised to conceal unplanned information without violating the compatibility with older software. For example, if further information about photos is needed, then it could be included in the photos themselves. The information will be transmitted included in the photos, but it will not conflict with old software that does not aware of its existence.

• **Medical imagery**. Doctors and hospitals can combine exams, imagery, and information of their patients. When a doctor analyzes a radiological exam, the information of the patient is hidden in the image, reducing the possibility of wrong diagnosis and/or fraud.

• **Military agencies**. Military operations can be based on concealed and protected information. Even with encrypted information, the detection of a secret signal in a Battlefield can lead to the instant recognition and attack of the engaged parties in the process of communication.

• **Intelligence agencies**. Intelligence agencies are concerned with developing and using these techniques, with the identification of their drawbacks to increase their abilities to detect, track, and prevent tracking of hidden messages.

1.2 Motivation

The spread of the internet around the world results in raising the concern of information security while the access to the confidential or classified information is theoretically available to any unauthorized party that using the internet. Therefore, the data must be secured to prevent any unauthorized person to access it and break its confidentiality.

Cryptography scrambles the secret information in a way that makes it inapprehensible to any unauthorized party. While there are some countries restrict the use of cryptography, curb the development, and even circulation of cryptography systems and algorithms, steganography rises as a solution to exchange secret information and confidential data between people or companies or organizations.

The use of encryption is always in a race with the methods of breaking the cryptography system, where the existence of the scrambled code revealed the importance of the message or else why is someone make an effort to decrypt it in the first place, therefore the attacker will make every effort to decrypt this message, while the steganography conceals the very existence of the secret message. Thus, an unauthorized party cannot even know that a secret communication taken place.

The motivation of developing steganography systems is raised and become the major concern for military and intelligence agencies to protect secret information and to hide the communication process, on the other hand, preserving intellectual property rights of digital media is also raise as a great concern.

The major challenge of the use of the steganography systems is the capacity of the carrier media, while digital images, particularly those using JPEG format are the most widely used files for the steganography, there are

capacity limitations for hiding large size of secret information. Generally, the size of the embedded message is limited to the size of the cover image. Furthermore, embedding secret data in a cover image may change or modify some characteristics of the cover image and cause a noticeable distortion and therefore attracts the eavesdropper's attention. Therefore, the capacity of the steganographic system and the imperceptibility of the stego image are the most important factors in the image steganographic systems.

1.3 Literature Survey

In this section, an overview of some of the relevant works in the field is reviewed.

- **In 2010**, R. English [9], proposed the implementation of the Bit Plane Slicing (BPS) technique and provided a comparison of hiding capacity and effectiveness with the simple 4 least significant bits algorithm. The comparison showed that BPS is a much more effective technique of achieving a 50% hiding capacity and PSNR of up to 39.9129 dB.
- **In 2010**, C. W. Lee, and W. H. Tsai [10], proposed a new technique using PNG image as a cover image and the information sharing technique. The shares generated by utilizing the polynomial function coefficients Shamir's threshold as a carrier of the secret data. These shares embedded into Alpha channel of the cover image. The white noise resulted in the stego image is removed by changing the share values into appropriate value. Where choosing the value of 4 to the share results in data hiding capacity of 1,048,576 and PSNR value of 28.68 dB, while the share value of 2 results in data hiding capacity of 524,288 and PSNR of 40.34 dB.
- **In 2012**, S. Parah et al. [11], proposed a high capacity steganography method based on Intermediate Significant bit (ISB) substitution. The

secret data is partitioned into relatively decreasing length blocks; each block is encrypted by a secret key and embedded in the cover image. Different cover images used, where the results show that the capacity of the proposed system may reach 31.25% while the PSNR value is 36.08 dB.

- **In 2012**, P. Rudramath, and M. R. Madki [12], proposed an improved BPCS based steganography technique in which different bit-planes are processed differently, with adjusting high threshold on the most significant bit-planes and low threshold on the LSBs. The secret data is a text message encrypted by RSA algorithm. The histogram test is applied to this method and show that the system has an acceptable stego image quality. No PSNR test performed.
- **In 2012**, C. F. Lee, and Y. L. Huang [13], proposed a high capacity steganography method using Interpolation by Neighboring Pixels (INP) on maximum difference values to increase the effectiveness of the system. The proposed technique provides high capacity and good quality of stego image, where the payload can reach to 2.28 bit/pixel while the PSNR value is 20.49 dB.
- **In 2012**, A. Nichal, and S. Deshpande [14], proposed a steganography technique based on JPEG2000 compression. Redundancy evaluation approach is used to increase the hiding capacity. The proposed system implemented in JPEG2000 compression encoder and generated a stego stream, which normally decoded. The experimental results show that the embedding capacity is 13021 bits and the PSNR is 37.263 dB.
- **In 2012**, N. Batra, and P. Kaushik [15], proposed a steganography technique based upon dividing the carrier image into non-overlapping blocks of 16x16 pixels and a modified quantization table of 16x16. The proposed method is compared with Jpeg-Jsteg method in term of embedding capacity and imperceptibility. The

experimental results show that the proposed system can reach up to 69632 hiding capacity and the PSNR of 58.212 dB. It also had been found that increasing the size of the quantization table increases the capacity.

- **In 2013**, S. Sharma, and U. Kumari [16], proposed a steganography technique based upon LSB substitution and RGBA cover image and YIQ color space. The proposed technique reduced the size of secret color image by converting the color space of the secret image from RGB to YIQ by applying a transformation function. The modified secret image then embedded into the four channels of the RGBA cover image. Results showed that the proposed method has a high hiding capacity reach up to 100%, and PSNR value is up to 36.6 dB.
- **In 2013**, H. F. Ahmed, and U. Rizwan [17], proposed a steganography method based upon BPS and embedding multiple secret images in one cover image. The proposed method manages to hide up to seven secret images into the cover image while the PSNR value is 27.485 dB.
- **In 2013**, W. W. Zin [18], an image steganographic technique is presented by combining cryptographic and steganographic techniques. This system uses LSB-based data embedding technique to hide the encrypted message. Before embedding the secret message, RC4 algorithm is also used for message encryption. In this system, BBS (Blum Blum Shub) Pseudo Random Number Generator is used for generating the random sequences and then the secret messages can be hidden in PNG image file by using random sequences. The proposed system used perceptual test to show its strength and there is no statistical test applied.

1.4 Aim of the Thesis

The aims of this work are summarized, as follows:

- To design a data hiding technique in digital image by using AES algorithm.
- The primary aim of this work is to increase capacity of the steganographic system to the maximum value and improve the quality of the stego image.
- To study the cover image channels used in embedding and the effect of using each channel on the quality of the stego image.
- To study the effect of Alpha channel on the cover image and the embedding process.
- To get an acceptable quality of the extracted message, beside the acceptable quality of stego image should.

1.5 Thesis Organization

The thesis is organized into five chapters as follows:

- **Chapter One: -** gives an introduction about steganography and a brief description of related literatures works and aims of the work.
- **Chapter Two: -** gives a brief of steganography theory, its types, the types of attacks, its technique, basic model for steganographic system. It also gives a background of, image theory, its types, the color space systems, color channels concept, and Alpha channel principle. Bit-Plane Slicing concept and a background on AES algorithm also described in this chapter.
- **Chapter Three: -** describes and discusses the proposed steganography system, its basic idea and the main algorithms.
- **Chapter Four: -** presents the results obtained from the statistical tests of the system and discuss these results. This chapter also

includes a comparison with other conventional steganography techniques.

- **Chapter Five**: - concludes the advantages and provides suggestions for future work.

Chapter Two

Theory of Steganography and Encryption

2.1 Introduction

This chapter gives an overview of steganography concepts, types, and techniques, and concepts of digital images, including image structures, image storing technique, and color representation. In addition, this chapter discusses the concept of two important terms used in this thesis, which are Alpha Channel, and Bit-Plane Slicing. The encryption method used in this work also discussed and explained.

2.2 Theory of Steganography

Steganography means storing specified data in such a way that conceals the data presence. Paired with current communication techniques, hiding information in images is a methodology for transmitting information using innocuous carriers to hide their presence. For human eyes, information generally made of known forms, such as images, e-mails, sounds, texts, etc. [19]. Figure (2.1) illustrates an example of data hiding.

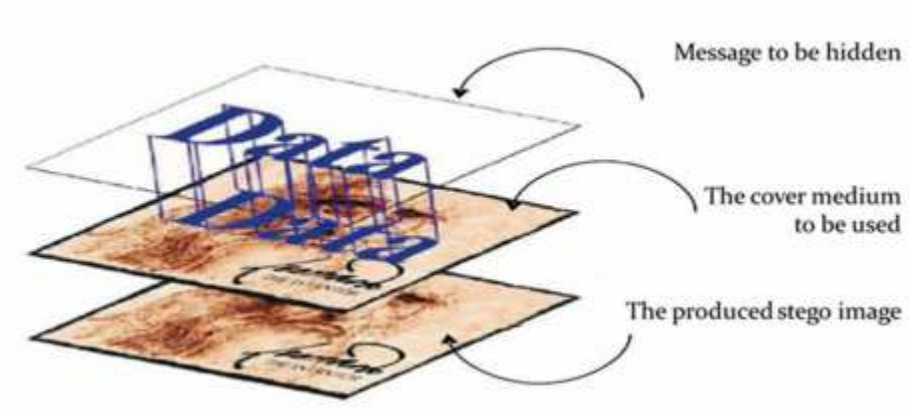


Figure (2.1) Data hiding example [19].

The process of understanding steganography based upon image requires a study background of the presentation and facilities of digital images. The process of embedding sometimes uses a secret key, known as the stego key, and without the existence of this key, it is so difficult or even impossible for any unauthorized person to detect and extract these hidden messages. Once the cover objects have messages or secret information embedded in them, they are called stego objects [4]. Figure (2.2) shows a general model for image steganography.

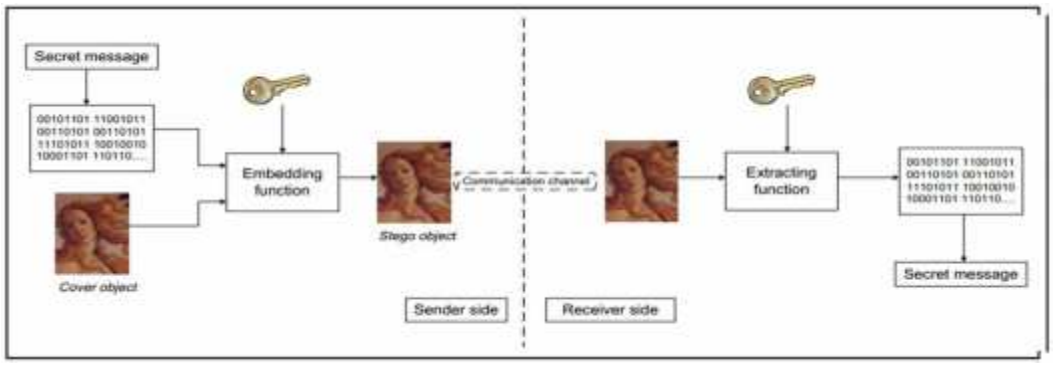


Figure (2.2) A model of the steganographic process [4].

A sender embeds a message into a cover carrier, by using a transformation process to the secret message, then manipulating a subset of the bits that represent a part of the cover object to form the final stego image or object. These images or objects will be transmitted over a communication system, to their destination. At the destination part, the

embedding procedure is reversed to reveal the hidden messages. If there was a secret key used to encrypt the secret data, both the sender and the receiver must know this key before sending the stego message. The difference between cryptography and steganography must be distinct and clear, however, there are some techniques are closely related to steganography were the differences are not clear [4].

There are two techniques that are closely related to steganography and place in the same branch of messages hiding. These two technologies are watermarking and fingerprinting. These technologies are used to protect the intellectual property. Therefore, the three technologies differ in purpose of usage, robustness, and embedding capacity of the covers [20]. The differences among these three technologies are summarized in Table (2.1).

Table (2.1) Summary of differences among watermarking, fingerprinting and steganography [21].

Method	Steganography	Fingerprinting	Watermarking
Purpose	Transmission of secret messages without raising suspicion	Protect intellectual property rights by identifying parties who break licensing agreements	Protect intellectual property rights
Perceptual invisibility	Crucial for embedded information not to be perceptual	Desirable, but not crucial	Desirable, but not crucial
Robustness against hostile removal, destruction or counterfeiting	Desirable, but not crucial	Crucial not to be able to remove embedded information	Crucial not to be able to remove embedded information
Large hiding capacity	Very important since it might be necessary to transmit large messages	Not important since copyright signatures are generally small	Not important since copyright signatures are generally small

Figure (2.3) illustrates the main components of information hiding field. This work is concerned with image steganography and does not discuss other types of steganography, like text or audio. The shaded rectangles indicated that the main interest of this thesis.

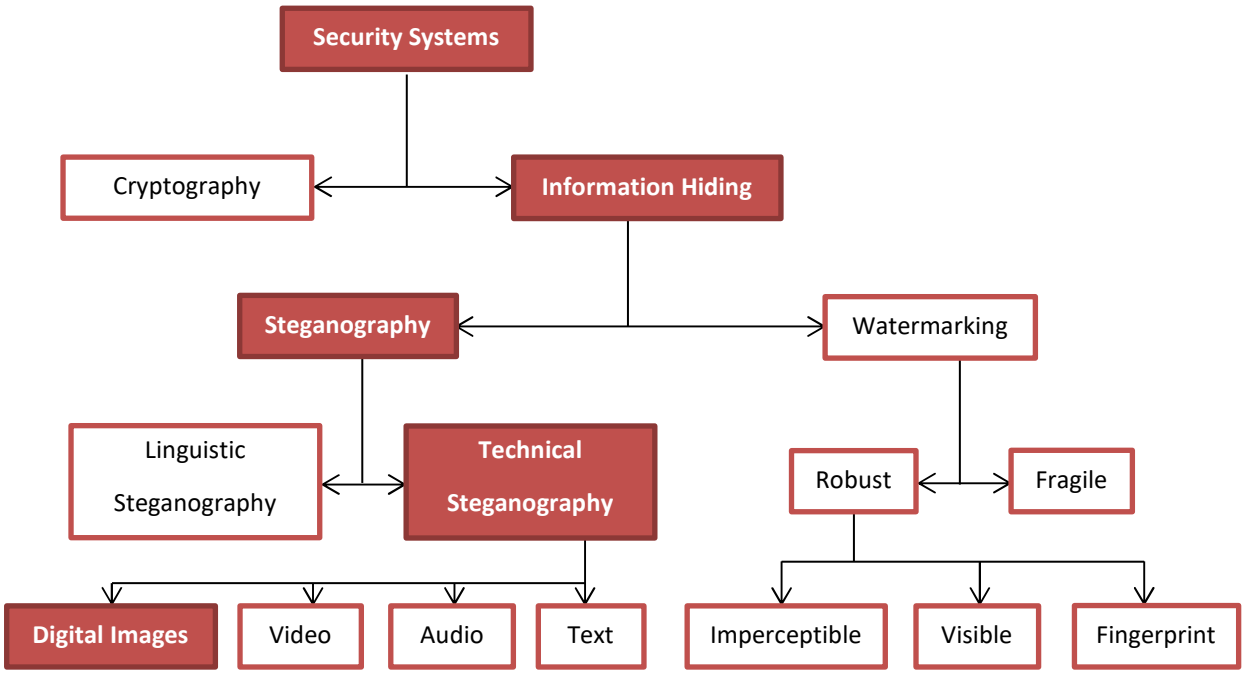


Figure (2.3) The different embodiment disciplines of Information Hiding[22].

2.2.1 Main components of steganographic systems

There is one common principle in steganographic systems. The sender (Ali), who intends to send a secret information (i) to the receiver (Omar), chooses an innocuous cover object (c) randomly. After that, Ali embeds the secret information (i) into the cover object (c) and may be with the use of a secret key (k). As a result, Ali generates a stego object (s) which must be semi-identical to the cover object (c) to any unauthorized party. Ali then, transmits the stego object (s) to Omar through a specified communication system. The objective of the system is to prevent Samir (a third person) from extracting the concealed message (i). On the other hand, Omar extracts the embedded secret information (i) since he knows the

embedding process and the secret key (k). Only the sender and the authorized receiver should know the stego key. Thus, most of steganography systems induce users to include a stego key or password in the embedding process. Figure (2.4) shows the general concept of image based steganography systems. [23]

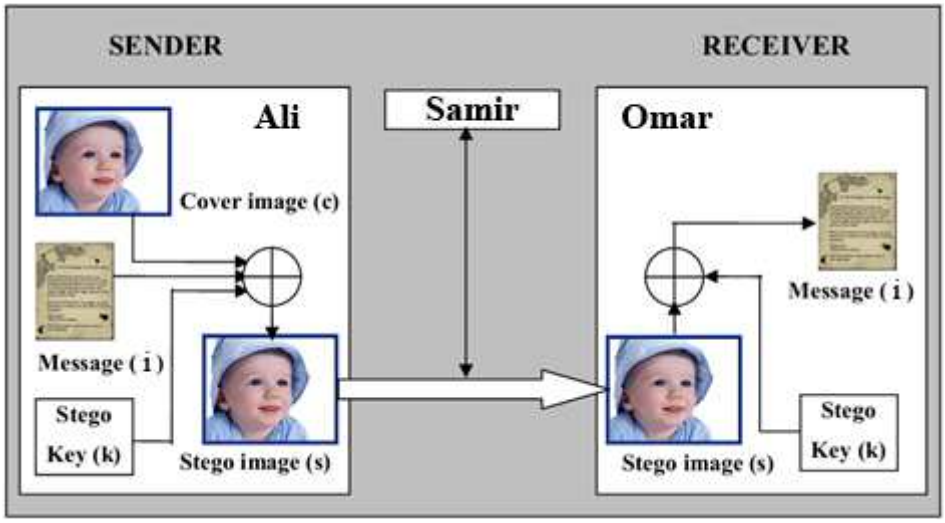


Figure (2.4) General Principle of Image Steganographic System [23].

Sometimes, an attacker (like Samir) will suspect in a stego object and detect how the messages were embedded, but he will be unable to reveal the hidden information. This process is called a secure steganography process because the secret information cannot be read unless the attacker knows the stego key. Thus, these secret keys must be chosen as rigid as possible for the purpose of preventing system attackers from decrypting the secret encrypted information using all possible available stego keys. Therefore, the security of the steganography system must satisfy Kerckhoffs's principle [23].

According to that, the measurements of security of steganography systems should be based on the situation that attackers have full knowledge of the embedding and extracting processes. However, in the case of detection, attackers only need the secret key to evident that the secret

transmission is taken place and to reveal the secret information. If the secret key that used in embedding process and the secret key that used in extraction process are both same, then the steganography algorithm is said to be symmetric. While, if these two stego keys are different for embedding and extraction process, then the steganography algorithm is said to be asymmetric [24].

2.2.2 The basics of embedding

There are three different key requirements for data hiding systems related with each other, which are capacity, security, and robustness [25].

- **Capacity**. It is the size of the data that the data hiding system can embed successfully without generating noticeable perceptual distortion in the carrier media and in order to be useful in embedding the secret information, the hiding capacity must be as high as possible.

- **Security**. The data hiding system is considered a secure if the embedded data irremovable if the attackers do not have full knowledge of the embedding algorithm.

- **Robustness (Visibility)**. The embedded data is considered a robust if its existence can be reliably detected with the image modification nit destroying.

2.2.3 Steganographic protocols

Three types of steganographic protocols are available, which are[26]:

1. Pure steganography:

A Steganography algorithm is called a pure steganography system when it does not involve the prior exchange of some secret information (like a stego key). Both sender and receiver should have access to the embedding and extraction algorithm, while the algorithm must be private.

2. Secret key steganography

With the use of pure steganography, no information is required to fulfil the transmission process. This is not very secure in practice. It must be assumed that, the intruder has a full knowledge of the sender and recipient algorithm that used for message transfer.

Theoretically, the intruder can extract data out of every stego object transmitted between sender and receiver. Therefore, the security of a steganography algorithm should be based on some secret information exchanged by the sender and receiver, which is the stego key. Without knowing this key, no one can extract the secret data from the stego object.

3. Public key steganography

Public key steganographic systems rely on the use of two keys, one private and one public. The public key is stocked in a public database, while the public key is used in the embedding process; the secret key is required to rebuild the secret message.

2.2.4 Types of steganography

In digital media files, redundant bits are defined as: "the bits of an object that provide resolution far greater than necessary for the information use and rendering". For example, some image files can provide more than 16 million different colors, whilst the human eye can only recognize approximately 10 million color levels. Thus, the redundant bits of a file are those bits that can be changed without any noticing. In steganography, file formats with a high level of redundancy is desired since redundant bits can be replaced with a secret object without being noticed [4].

1. Text steganography

Historically, hiding data in texts is the most significant technique of steganography. One approach that is used to conceal a secret message in texts is called a null cipher where the last letter of every word of a cover text string is used to hide a letter of the secret message [27].

Book cipher is another text steganography technique. A story or a journal is used as a cover file. A code that contain a string of indicators to characters is shared between the sender and receiver. For instance, the cipher code "222221" might mean page 22, line number 22, and the 21st character. Revealing the secret message relies only on obtaining knowledge of the secret code and the book. Because of the evolution of the internet and digital file format, the world became less interested in book ciphering. In the digital world, small changes to font style, size, and boldness, and line spacing, and other text formatting operations can be used in steganography process. The current text-based steganography programs use additional white spacing or extra tabbing at the end of a line [4].

Although a number of different methods can be defined for hiding data in text [53], text-based steganography using digital files became less popular because text files have a very little degree of redundancy [28].

2. Image steganography

Digital images have a large amount of redundancy, thus, images represent the most appropriate carrier used in steganography. Due to sharing images frequently on websites, e-mail attachments, etc. steganography on images is also the most common form of steganography[4].

Given that, images are ideal carriers, and common information media, this thesis focuses solely on steganography based upon image in subsequent chapters and algorithms.

3. Audio/Video steganography

Audio compression is at most rely on research that has been introduced in the biological characteristics of the human ear, especially on the amount of information that can be removed from the audio file without being noticed by human ear [29].

These characteristics can also be exploited for audio-based steganography by hiding data in audio objects without noticing the differences. Steganography based upon audio is less common method than the image steganography because the larger size needed for meaningful audio file [30].

Video files, in general, can be considered a set of still images and sound tracks, therefore both image and sound steganography methods can be used for video-based steganography. Further advantages of video-based steganography are that video files can hide a large amount of information. A drawback of video-based steganography is the large size of a video file that is not transmitted regularly over normal transmission systems [4].

4. Protocol steganography

Protocol steganography means: "the technique of embedding information within the volatile data created in network transmissions" [27].

A network packet parts are headers, user data, and trailers. All the packets sent over a network using the OSI model, have the same structure. Hidden channels where steganography can be used exist in OSI layers [31].

2.2.5 Steganography techniques

Steganography techniques are divided into six categories as follows[23]:

1. Substitution techniques

For a certain cover file, it is significant to detect some regions or data that can be modified without having any major changes in this cover file. Thus, a secret file can be embedded in by replacing the insignificant bits of a cover object file with the bits of secret message, without generating any serious artifacts to the cover file, so the recipient can extract the secret file if he has knowledge of the accurate embedding position [23].

In general, digital covers have a large amount of insignificant bits i.e. least significant bits (LSB). In steganography system based upon substitution technique, the bits of the secret message replace the LSBs of cover file without making huge changes in the cover object. In addition, the LSB technique works in spatial domain since it embeds the secret bits directly in the cover object. Due to the simplicity and the speed of LSB technique, it is the most popular technique used for digital image steganography.

2. Transform domain techniques

This technique is more complicated than LSB technique for hiding data in an image file. Many types of transformations are used on the image before embedding data in it [32].

It has been noted that hiding data in the frequency domain of a signal could be much more robust than hiding process operating in the time domain. Transform domain techniques embed secret files in significant areas of the cover image, which makes them more robust against attacks, like compression, cropping and other image processing than the LSB

approach. One of the most transforms used is the Discrete Cosine Transform (DCT). Other types of use transforms would be the Wavelet Transform and Contourlet Transform [26].

3. Spread spectrum techniques

Spread spectrum (SS) steganography technique is defined as "the process of spreading the bandwidth of a narrowband signal across a wide band of frequencies". In this technique, the frequency domain of the cover file is considered to be a transmission channel and the secret message as a signal that is to be transmitted over that channel. Due to the spread of the secret message over a wide frequency band, this technique is relatively robust against any manipulation or removing to the secret message [23].

In data hiding, two specific types of SS are mainly used: direct sequence and frequency hopping. In direct sequence, the secret signal is transmitted by a constant called chip rate, modulated with a pseudorandom signal and inserted into the cover object. In frequency hopping, the frequency of the cover signal is changed in a way that it hops quickly from one frequency to another [26].

4. Statistical techniques

These techniques hide only one bit of secret information into the cover object. This is called 1-bit steganography technique. If "1" is hidden in a cover object, some statistical features (e.g. entropy and probability distribution) of this cover object should be changed majorly to indicate the presence of the secret message. However, if the embedded bit is "0", the cover object is left unchanged [23].

5. Distortion techniques

The majority of the steganography systems are blind, which means that a recipient does not need the original cover object to perform the

extraction of the secret message from the stego object. However, if a distortion method is used, the recipient needs the original cover object to be used in recovering the secret information. For a recipient, the hidden file is the difference between the stego object received and the original cover object [23].

Using this method, a stego object is generated by creating a sequence of modifications and changes in the cover object. This sequence of modifications is chosen to match the secret information needed to be sent[32].

Most text steganography systems are of the distortion type. For instance, a document layout or words order might give an indication to the presence of information.

6. Cover generation techniques

Unlike all former hiding techniques given above, some steganographic mechanisms create a digital object only in order to being a cover for secret transmission, such as in creating fractal images as cover images each of which is individually defined by a group of fractal parameters like type, formula, scale, location, color space, etc. This group of parameters can be stored in a specific file, which can be transmitted very simply. Thus, the cover image can be perfectly restored [26].

2.2.6 Steganalysis

Steganalysis is "the science of detecting the existence of hidden messages in stego systems". The objective of steganalysis is to discover if an image contains secret information or not. There are three types of steganalysis techniques. These are [8]:

1. **Aural attacks**. They about eliminating the significant bits of a digital content for the purpose of activate human's visual inspection for any strange content.

2. Structural attacks. Hiding information may change the file format of the cover. Often, these changes make the cover pattern easily detected in the structure of the file format. For example, it is not preferable to hide data in images of GIF (Graphics Interchange Format) file format. In this type of format, an image visual structure exists to some degrees in all of bit layers of an image because the color index that represents 2^{24} colors uses only 256 values.

3. Statistical attacks. Digital pictures of natural scenes have featured statistical behavior. With the suitable analysis, one can identify whether or not a digital image has been modified, making the changes mathematically detectable. In this case, the main objective of steganalysis is to gather sufficient statistical evidences indicating the existence of hidden data in images.

2.2.7 Steganography attacks

Steganographic attacks include detecting, extracting and destroying hidden messages of the stego object. There are many types of attacks rely on the information available for analysis. Some of these are, as follows [26]:

- **Steganography only attack:** here only stego object is available for analysis.
- **Known carrier attack:** here both of the original cover object and stego object are available.
- **Known message attack:** In this case, the hidden message is known for attacker.

- **Known steganography attack:** The cover object, stego object as well as the steganography procedure are known.

2.3 Digital Imaging Concepts

Some principles in digital imaging field are considered In order to perfectly understand how data is embedded in digital images. These are:

2.3.1 Images and pictures

Human being relies heavily on vision to understand the world around him. One does not only look at objects to identify and distinguish them, but one can observe differences.

Humans have evolved their visual skills to be very accurate: one can identify a face in instantly, distinguish colors, and can deal with large amount of visual data rapidly [33].

Humans are interested in snapshots (single images). Though that image processing can deal with variations in scenes, however, an image is a single picture, which represents something. It can contain a picture of a person, an animal, a nature scene, a microphotograph of an electronic circuit, or a result of X-ray imaging.

2.3.2 Color representation

According to the trichromatic color theory: "each color that the human eye can perceive can be obtained from three basic colors. The three basic colors are: red, green and blue". Every digital color is made of a linear mixture of red, green and blue components. This is called the RGB color space, with the value of each color indicated by R, G, and B [34].

Another common color model is the YUV color space; also called luminance/chrominance model with luminance Y denotes the measured

linear combination of the RGB channels, while chrominance U and V indicate color information [35].

2.3.3 Image definition

In computer applications, an image is: "a collection of numbers that constitute different light wavelengths in different pixel in the image". Individual points are called pixels, the pixels shape a rectangular area where each pixel is located, and its color is assigned [4].

The number of bits in a color model, called the bit depth, which indicates the amount of bits used for each pixel. For instance, if the bit depth of an image is 8, that means there are 8 bits used to represent the color of each pixel and a total number of displayed colors are 256. Figure (2.5) shows an example of a grayscale image with bit depth 8 that can represent 256 different levels of gray.

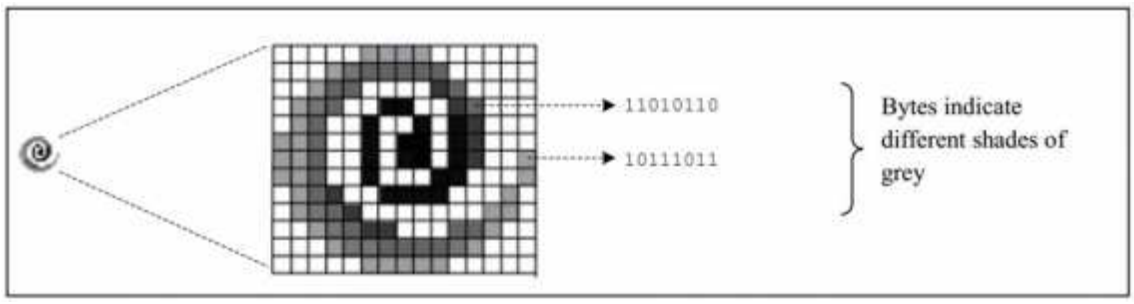


Figure (2.5) Pixels and bit representation of a greyscale image with bit depth 8 [4].

Digital color images, also called truecolor images, are usually represented with bit depth 24 and use the RGB color model. Each one of the three-color components (red, green and blue) is represented digitally by 8 bits [4]. Figure (2.6) shows the use of 24-bit images.

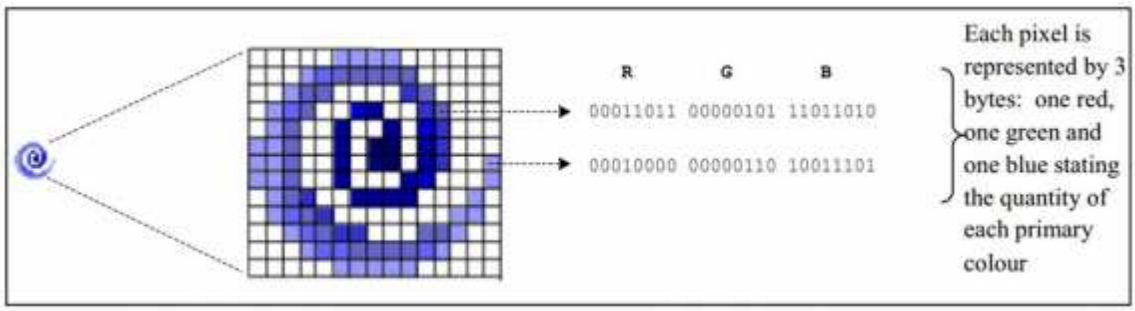


Figure (2.6) Pixels and bit representation of a 24-bit color image using the RGB color model [4].

In a given pixel, there can be 256 different amount of red, green, and blue, adding up to about less than 17-million combinations. In comparison, a quality offset printing press can provide about 4000 colors, a traditional film photograph can contain about 6-million colors, and the human eye can identify approximately 10-million colors. This means that a large amount of redundancy is created which can be used by the steganography algorithm [4].

2.3.4 Image file formats

The methods that used to store images differs mostly in the digital image representation and the method used in compression. The image file format typically relies on the intended use of the image [36].

Image formats can be categorized into two domains: spatial domain and frequency domain. In spatial domain, an image is represented as an intense rectangular grid of pixels. In frequency domain, an image is represented as mathematical models based on compression mechanisms to allow for a larger rate of compression.

2.4 Alpha Channel

In a full color rendering of an element, the RGB components represent only the color. For the purpose to place the element over any chosen background, a factor used for mixing is applied to all pixels to control the process of linear interpolation of colors of foreground and background. Generally, there is no way to encode this component as part of the color information. For anti-aliasing purposes, this mixing component required to be of comparable resolution to the color channels. This channel is called Alpha channel, where Alpha of 0 indicates no coverage, 1 to means full coverage, and the fractions representing partial coverage [37].

There are two methods to deal with the alpha of a pixel. One interpretation relies on the geometry half of the world and the other relies on the imaging half. Geometers consider "pixels" are geometrical areas intersected by some objects. So, alpha will be the percentage coverage of a pixel by an object. Imagers consider pixels are continuum point samples. So, alpha is the opacity provided at each sample. The imaging model will be taken, because a geometric picture must be reduced to point samples to display [38].

Consider the following geometrical scheme: there is a "pixel" with an area covered by an opaque geometrical element has a color value *A*. Thus, the value of color contributed by the object opaque area is *aA*. Hence, the color spread over the pixel and rise as a single new color representing the entire area, i.e. the color *aA* is a point sample. Now consider another opaque geometrical element with a color *B* added to the original pixel and disregard any other geometrical elements in that pixel.

Assume that the *B* object has pixel coverage of *b*. So, the quantity of color contributed by that area is *bB*. Again, this is a point sample representing the color of the second element. To conceptually merge the

contributions of the two objects in the pixel area, now the geometry model is used. The object *B* is allowed only (*1-b*) percent of the pixel area to be transparent to elements behind it. The actual geometry of the two elements is simply neglected here and consider that the pixel is allowing (1-b) times the color *aA* from behind *bB* to appear. This is added to the color due to the top element *bB*. So the total color of element with color *B* over element with color *A* is [39]:

The total color = $bB + (1 - b)aA$ …(2.1)

The equation (2.1) may be wrong if the second object exactly coincided with the first. The color of the beneath object will not have any contribution in the final color. Therefore, the model that used is an approximation in combining two images without the possibility to know how to determine the alpha at a point. There is no possibility to know whether a point sample with a partial opacity results from a partially transparent area or from an opaque area partially cover the area represented by the point sample [39].

Alpha channel, indicating transparency information on a pixel, can be used and calculated in different types of images such as in grayscale and true color PNG (Portable Network Graphics) images.

The value zero for Alpha represents full transparency, and the value of (2^x-1) (where *x* is the bit depth) for Alpha represents a full opacity of a pixel. Intermediate values for Alpha represent partial transparency of pixels that can be merged with a background image to generate a composite image.

Alpha channels can be included with images that are represented by either 8 or 16 bits for each sample, samples represented by less than 8 bits in an image cannot use Alpha channels. Both Alpha samples and image

samples are represented by the same bit depth. For each pixel, the Alpha sample is stored immediately after the RGB sample.

The alpha value for each pixel does not affect color values stored for that pixel. This known as "unassociated" or "non-pre-multiplied" alpha. (Another common method is to store sample values pre-multiplied by the alpha sample. This results in; the image is composited against a black background). It is worth to note that PNG does not use pre-multiplied alpha.

No storage cost is needed for transparency control in a full alpha channel. In a color-index image, an alpha value can be assigned for every palette entry. In grayscale and colored images, a single pixel value can be indicated as being "transparent". These methods follow the *tRNS* (Transparency) ancillary chunk type. The *tRNS* chunk: "indicates that the image uses simple transparency: either alpha values associated with palette entries (for indexed-color images) or a single transparent color (for grayscale and true color images). Since the simple transparency is not as elegant as the full alpha channel, it requires less storage space and is sufficient for many situations". When no alpha channel nor *tRNS* chunk are exist, the pixels in the image are considered to be fully opaque [40].

To determine if the pixel is occluded by a transparent red object one must assign (1,0,0,0.5) to that pixel: the 0.5 denotes the coverage and the (1,0,0) indicates the red color. The reason to dismiss this proposal is that all compositing operations will use multiplying the 1 value in the red channel by the 0.5 value in the alpha channel to calculate the red color contribution of this element in this pixel.

To avoid this situation of multiplication, there is another solution, which is storing the pre-multiplied value in the color component, so that (0.5, 0, 0, 0.5) will denote a full red element half covering a pixel [37].

The quadruple (R, G, B, A) identifies that the color (R/A, G/A, B/A) covers the pixel. A quadruple (R, G, B, A) where the alpha value is less than a color component indicates a color out of the [0,1] interval. For the representation of ordinary elements, the value of 0 for Alpha at a pixel, in general, makes the color components to be 0.

Therefore when alpha is 1 the RGB channels represent the true colors, 0 alpha makes RGB colors to be black, and fractional alpha makes RGB colors to be linearly darkened colors. If one uses an RGB monitor to view a scene, shadowed edges of RGBA elements thus present their anti-aliased nature.

It is significant to notice the differences between two key pixel representations [37]:

Black = (0,0,0,1);

Clear = (0,0,0,0);

The first pixel is an opaque black; the second pixel is Transparent.

2.4.1 Pre-multiplied alpha

For compositing color *C* generated by setting a pixel with color *B* with alpha *b* over a pixel with color *A* with alpha *a*, the value of *C* will be [39]:

$$C = bB + (1-b)aA = bB + aA - baA \quad \ldots(2.2)$$

For each pixel, there are three multiplications for each color component. Since the multiplications in computer graphics are costly, earlier, the researchers of (Lucasfilm) and (Pixar) found a reduction for this formula using one multiplication if the alphas were pre-multiplied times the color in an image.

Hence, if the color channels of image *J* included, not color A, but instead of that weighted color *aA*, and the same case for image *I*, then the equation (2.2) can be reduced to [39]:

$$C' = B' + (1 - b)A' = B' + A' - bA' \quad \ldots(2.3)$$

Where the letters with primes indicate that the colors have been pre-multiplied by their corresponding alphas values.

2.4.2 Non-pre-multiplied alpha

PNG image format uses "unassociated" or "non-pre-multiplied" alpha. This indicates that images with independent transparency masks can be stored lossless. "Pre-multiplied alpha", stores pixel values pre-multiplied by the alpha fraction; which means that the image is actually composited against a black background. This means that the image data will be lost[40].

Some images with pre-multiplied alpha can be transformed to PNG by division of the sample values by alpha, excluding when alpha is 0. The images will appear good if shown by a viewer that deals with alpha in an appropriate way, and will not look very good otherwise.

The same model used for composite color can be used for composite alpha. The opacity of the pixel that partially covered by the first geometric element is *b*, and the opacity for the second geometric object is a. Note that, the geometry of the model only allows (*1-b*) of the lower light filter to be influential. So the composite alpha will be as follows [39]:

$$g = b + (1 - b)a = b + a - ab \quad \ldots(2.4)$$

This is true for either cases, pre-multiplied or not.

2.4.3 Alpha channel creation

The alpha channel can be considered either as a layer that temporarily conceals transparent objects in an image, or it is a method for forming a non-rectangular image. In the first definition, the color values of fully transparent pixels must be kept for another usage. In the second definition, there is no useful data in the transparent pixels. In this case, in order to obtain better compression, fully transparent pixels must all contain the same color value [40].

It should be kept in mind the potentiality that a decoder will override transparency control, therefore, the colors assigned to each transparent pixel must be accepted background colors.

If a full alpha channel is not required, or compression efficiency is not fulfilled, the *tRNS* transparency chunk is also available.

If the background color of the image is known, it should be written in the *bKGD* (Background) chunk (The *bKGD* chunk identifies a default background color to composite the image against. The viewers may not have this chunk; so they can use another background). The unused screen area can be filled with the *bKGD* color by decoders that ignore transparency.

When there are pre-multiplied alpha data in the original image, it must be converted to PNG's non-pre-multiplied format by division of each sample value by the corresponding alpha value, then multiply the result from division process by the maximum amount for the image bit depth, and then taking the closest integer. It should be noted that valid pre-multiplied data the result of the division must always be in the range from 0 to 1. In case the alpha value is 0, the result is black (zeroes).

2.4.4 RGBA pictures

In complex animation, many full background pictures may have an Alpha of 1. Among foreground objects, the color values roll off in step with the alpha channel, which large transparent areas.

Mattes are: "colorless stencils used for controlling the compositing of other elements, which have 0 in their RGB elements". The procedure of offline storage of RGBA pictures must provide the natural data compression for dealing with the backgrounds RGB pixels, foregrounds RGBA pixels, and *A* pixels of mattes [37].

There are some problems in computation with those RGBA pictures. Storing of the color components pre-multiplied by the Alpha might affect the color resolution, mostly when the value of Alpha is near zero.

However, due to forming the picture will need that multiplication any case, storing the product obliges only a very small loss of precision. The computation of the color extraction is a difficult job. So (R/A, G/A, B/A) must recovered and, as alpha approaches to zero, the precision falls down sharply.

2.5 Bit-Plane Slicing

Consider that every pixel in an image is represented by 8 bits, it can be imagined that the image is composed of eight 1-bit planes, from bit-plane 0 which is the least significant bit plane to bit-plane 7 which is the most significant bit plane [41].

Figure (2.7) illustrates these ideas, Figure (2.8) shows an 8-bit fractal image, and Figure (2.9) illustrates the various bit planes for the fractal image.

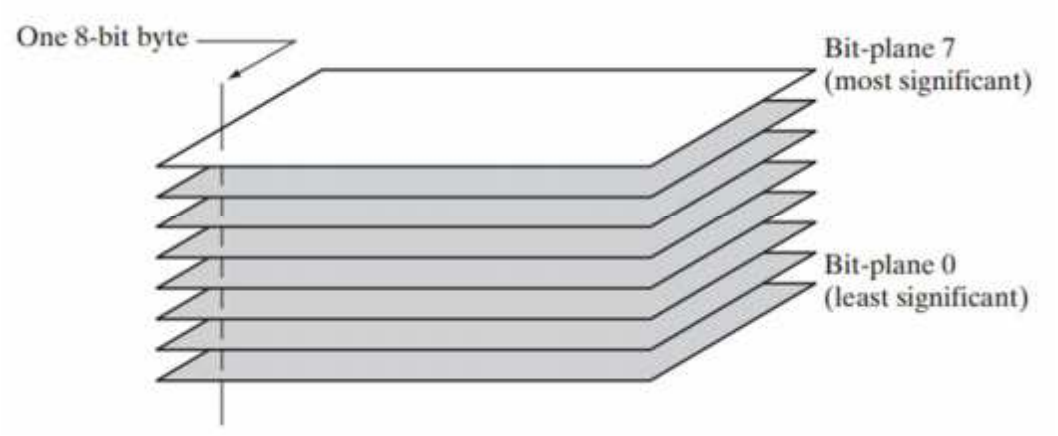


Figure (2.7) Bit-plane representation of an 8-bit image [41].

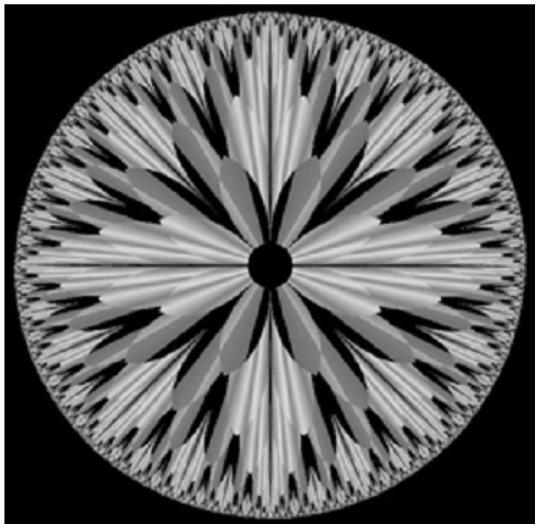


Figure (2.8) An 8-bit fractal image [41].

However, the higher-order planes, especially the top four, consist of the majority of the visually significant information. The other bit planes participate to more subtle details in the image.

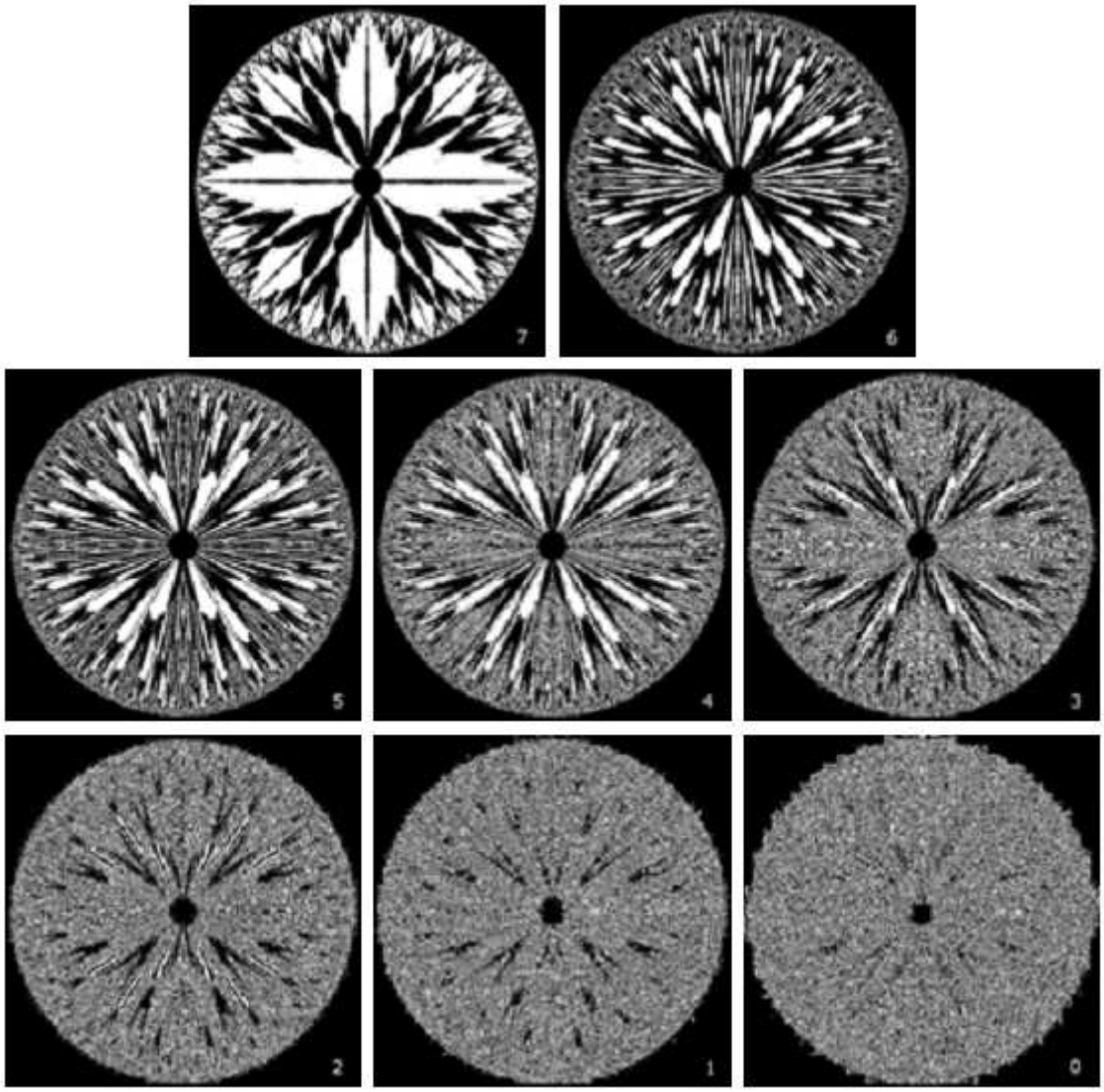


Figure (2.9) The eight bit planes of the image in Figure (2.8). The number at the bottom, right of each image identifies the bit plane [41].

The bit slicing procedure is useful for analyzing the relative significance played by each bit in an image. This procedure helps in identify the sufficiency of the number of bits used for quantization of each pixel. For bit-plane extraction for an 8-bit image, it is easy to show that the binary image for bit-plane 7 can be gained by processing the input image with a Thresholding gray level transformation function that is, first, used to identify all values in an image between 0 and 127 to one value (for example, 10); and, second, is used to identify all values between 129 and

255 to another (for example, 225). The binary image for bit-plane 7 in Fig. (2.9) was obtained by this procedure [41].

Suppose that the image size is 4 by 4 and the bit depth of each pixel is 4 bit, then the minimum pixel value of the image is 0 and maximum pixel value of the image is $2^4 - 1$ i.e. 15 as shown in Figure (2.10). So on the basis of these pixels it has 4 bit-planes; LSB, 2nd, 3rd and MSB bit-planes.

15	9	11	12
13	11	10	1
0	12	9	4
5	15	13	12

(a) Original Image having size 4x4

1111	1001	1011	1100
1101	1011	1010	0001
0000	1100	1001	0100
0101	1111	1101	1100

(b) Binary pixel representation

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

(c) LSB Bit-plane

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

(d) 2nd Bit-plane

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

(e) 3rd Bit-plane

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

(f) MSB Bit-plane

Figure (2.10) Bit-plane slicing representation

2.6 Advanced Encryption Standard (AES)

The new methods for standard encryption were Triple DES (3DES) and International Data Encryption Algorithm (IDEA) [42]. These techniques were too slow and IDEA was not free to be implemented because of patents [43].

A new algorithm created, Vincent Rijmen and Joan Daemen were developed. It was named Rijndael. This standard was known as Advanced Encryption Standard (AES) and up-to-date it is still the standard for encryption [44]. To date, there are no attacks better than brute-force known against AES [45].

The AES is a symmetric block cipher, like DES. However, AES differs from DES in number of aspects. The Rijndael algorithm allows for a variety of block and key sizes and not only the 64 and 56 bits as is it in DES. The range of blocks and keys in the range from 128, 160, 192, 224, and 256 bits and need not be the same [46]. However, for block size the AES standard indicates that the algorithm can accept only 128 bits and a choice between the three keys (128, 192, and 256) bits. According to the version used, the name of the standard is changed to AES-128, AES-192 and AES-256 respectively. There is another difference between AES and DES, that the structure of AES is not feistel. In the feistel structure, half of the data block is modified by the other half of the data block and then swapping between the two halves is done. In this case, the whole data block is parallel processed through every round by applying substitutions and permutations [47].

Many of AES parameters rely on the key length. For instance, if the key size used was 128 bits then the number of rounds is 10, while it is 12 for 192 bits and 14 for 256 bits. Currently, the most popular key size is the 128-bit key. The entire structure of AES can be shown in Figure (2.11).

The input is a single block of 128-bit size for both decryption and encryption. The input matrix known as the **in** matrix. This block is copied into a **state** array, which will be modified at every stage of the algorithm and then will be produced to an output matrix as shown in Figure (2.12). Both the plaintext and key are depicted presented as a 128-bit square matrix constructed from bytes. This key is then expanded into an array of key schedule words (the **w** matrix). Bytes within the **in** matrix are ordered by column. The same is applies to the **w** matrix [47].

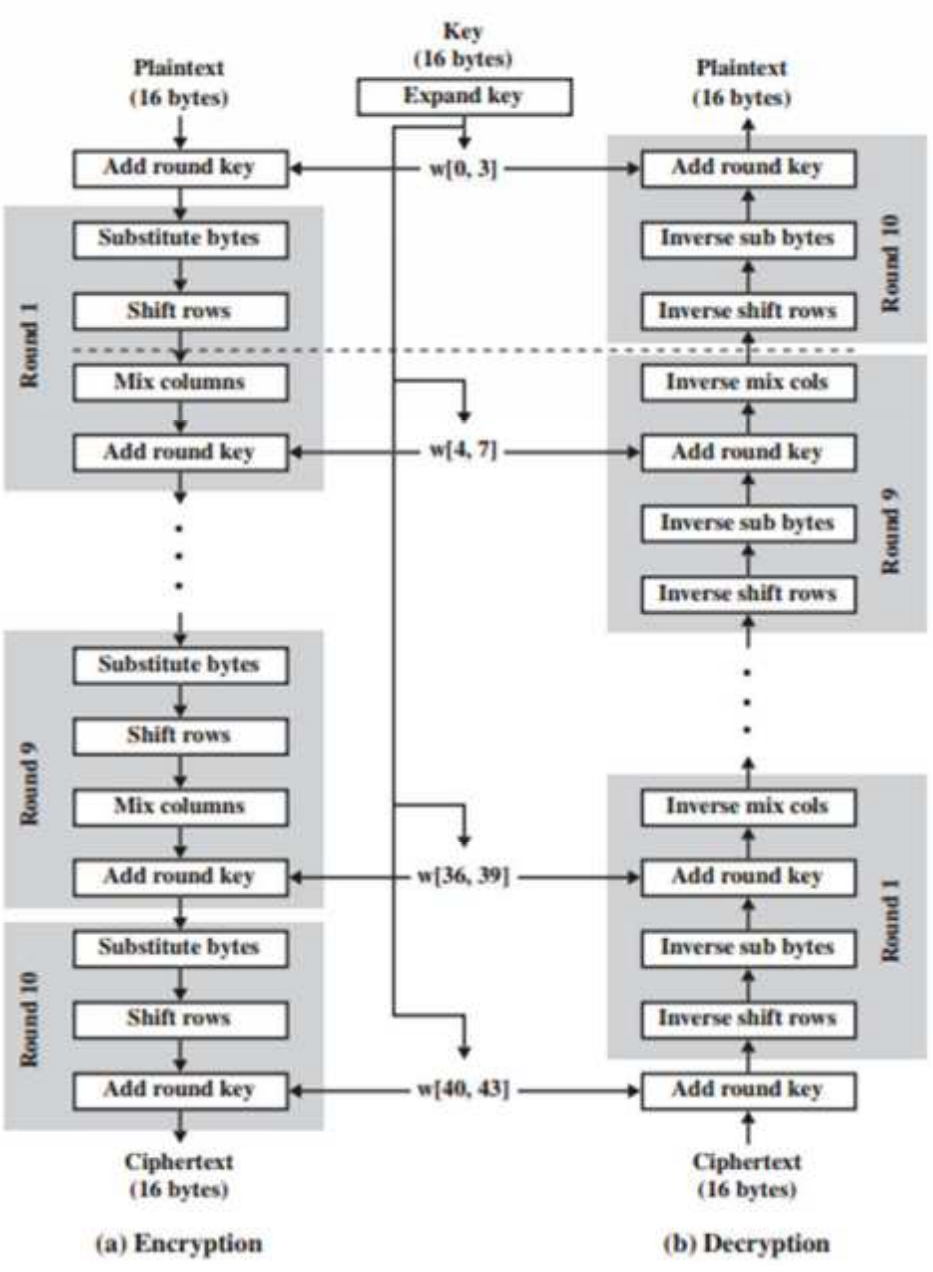


Figure (2.11) The overall structure of AES algorithm [47].

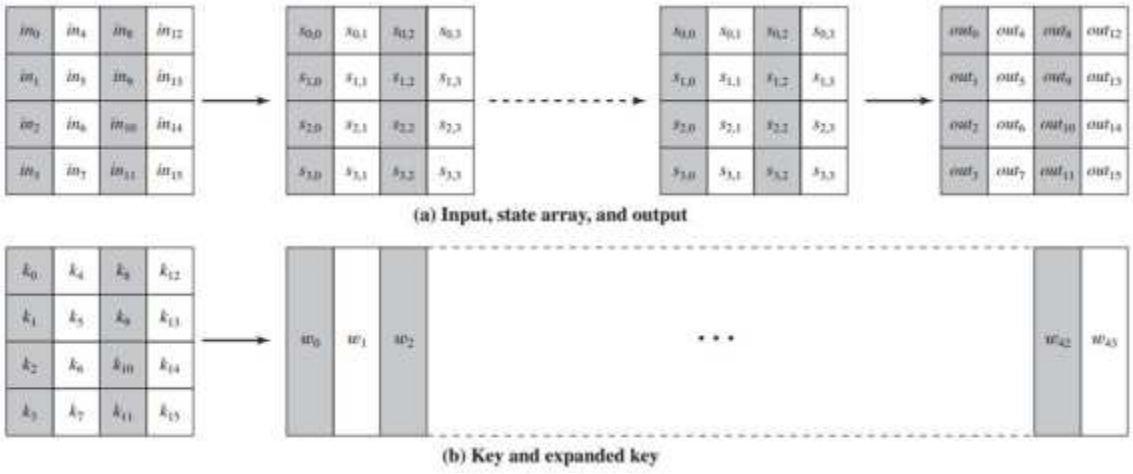


Figure (2.12) Data structures in the AES algorithm [47].

2.6.1 Inner workings of a round

The (Add round key) is the first stage in the algorithm followed by 9 rounds consist of 4 stages and a 10^{th} round consist of three stages. This is applied for encryption phase, for the decryption phase, stage of a round is the inverse of its counterpart in encryption phase. The four stages are as follows [48]:

1. Substitute bytes
2. Shift rows
3. Mix Columns
4. Add Round Key

The tenth round does not contain the Mix Columns stage. The first nine rounds of the decryption algorithm contain the following steps:

1. Inverse Shift rows
2. Inverse Substitute bytes
3. Inverse Add Round Key
4. Inverse Mix Columns

Again, the tenth round does not contain the Inverse Mix Columns stage.

1. Substitute bytes

This stage, called SubBytes, is a lookup table employing a 16×16 matrix consisted of byte values called an s-box. This matrix contains all the potential combinations of an 8 bit sequence (2^8 = 16×16 = 256). However, the s-box is not just a random permutation of these combinations [48].

The leftmost hexadecimal part of the byte is used to indicate a specific row of the s-box and the rightmost hexadecimal part indicates a column. For instance, the byte {95} (curly brackets represent hexadecimal values in FIPS PUB 197) indicates row 9 column 5 which found that it contain the value {2A}. This value is then used to update the state matrix. Figure (2.13) illustrates this process [47].

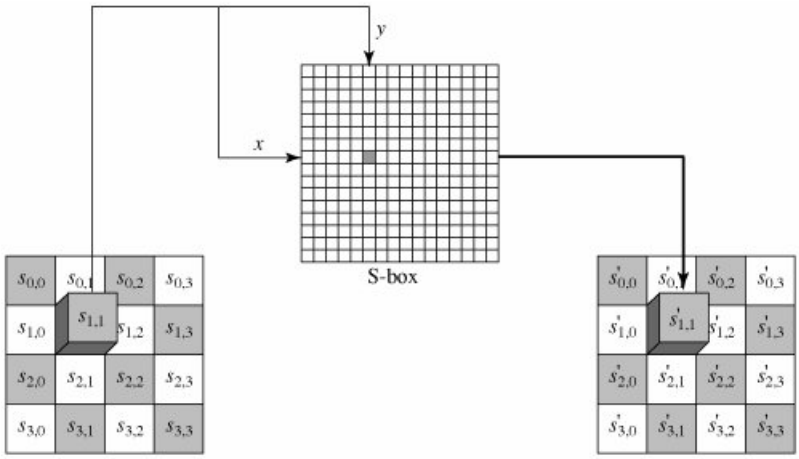


Figure (2.13) Substitute Bytes Stage of the AES algorithm [47].

The Inverse substitute byte stage, also called InvSubBytes, uses an inverse s-box. It is desired to select the value {2A} and gets the value {95}. Table (2.2), and Table (2.3) show the two s-boxes and it can be evidenced that this is the case indeed [47].

The s-box is designed to have immunity to known cryptanalytic attacks. Generally, AES developers endeavor to a design that has a minimum correlation between input bits and output bits, and the characteristic that the output cannot be considered as a simple

mathematical function of the input [49]. Moreover, the s-box has no fixed points (s-box (a) =) and no counterpart fixed points (s-box(a) = $\bar{a}$) where $\bar{a}$ is the bitwise compliment of a. The s-box should be invertible if decryption is to be possible (Is-box[s-box (a)] = a) in addition, it should not be its self-inverse where s-box(a) ≠ Is-box(a) [47].

Table (2.2) The s-box [47].

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	3B	52	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	**2A**	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Table (2.3) The inverse s-box [47].

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	**95**	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

2. Shift row transformation

This stage called ShiftRows is shown in figure (2.14). This is no more than simple permutation. It works as follows [47]:

• The first row of state matrix is not changed.

• The second row is circular shifted 1 byte to the left.

• The third row is circular shifted 2 bytes to the left.

• The fourth row is circular shifted 3 bytes to the left.

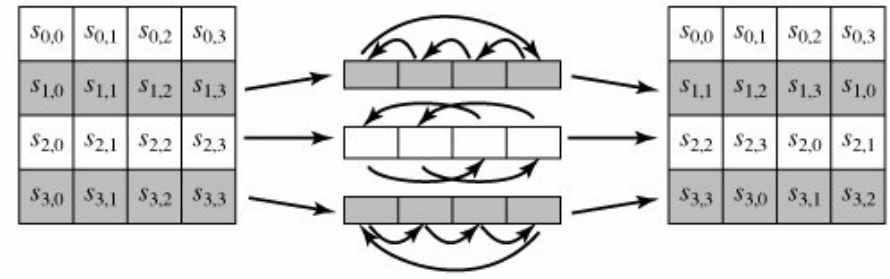


Figure (2.14) ShiftRows stage [47].

The Inverse Shift Rows stage, called InvShiftRows, fulfils these circular shifts in the opposite direction for each of the last three rows (the first row was unchanged to begin with) [48].

Recall that case is considered as an array of four-byte columns, where the first column indeed represents bytes 1, 2, 3 and 4. Thus, a one-byte shift is a linear distance of four bytes. The transformation also guarantees that the four bytes of one column are spread out in four different columns [47].

3. Mix column transformation

This stage called MixColumn is a substitution process using arithmetic of GF(2^8). Each column is treated separately. Each byte of a column is transformed into a new value, which is a function of all four bytes in the column. Figure (2.15) shows the MixColumn stage [49].

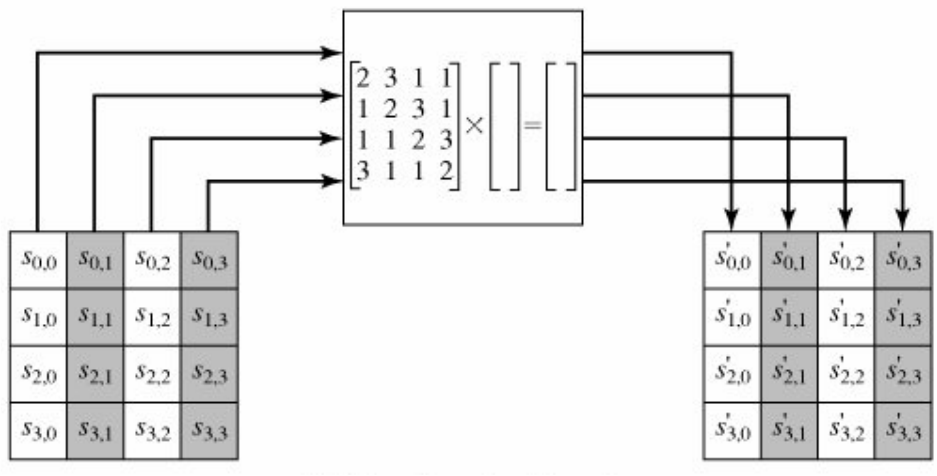


Figure (2.15) MixColumn stage [47].

The mapping process can be determined by the following matrix multiplication on state [47]:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} \acute{s}_{0,0} & \acute{s}_{0,1} & \acute{s}_{0,2} & \acute{s}_{0,3} \\ \acute{s}_{1,0} & \acute{s}_{1,1} & \acute{s}_{1,2} & \acute{s}_{1,3} \\ \acute{s}_{2,0} & \acute{s}_{2,1} & \acute{s}_{2,2} & \acute{s}_{2,3} \\ \acute{s}_{3,0} & \acute{s}_{3,1} & \acute{s}_{3,2} & \acute{s}_{3,3} \end{bmatrix} \quad \dots (2.5)$$

Each element of the product matrix is the products sum of elements of one row and one column. In this case, the individual additions and multiplications are done using GF(2^8). The MixColumns transformation of a single column j ($0 \leq j \leq 3$) of state can be expressed as [47]:

$$\begin{aligned} \acute{s}_{0,j} &= (2 \bullet s_{0,j}) \oplus (3 \bullet s_{1,j}) \oplus s_{2,j} \oplus s_{3,j} \\ \acute{s}_{1,j} &= s_{0,j} \oplus (2 \bullet s_{1,j}) \oplus (3 \bullet s_{2,j}) \oplus s_{3,j} \\ \acute{s}_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus (2 \bullet s_{2,j}) \oplus (3 \bullet s_{3,j}) \\ \acute{s}_{3,j} &= (3 \bullet s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \bullet s_{3,j}) \end{aligned} \quad \dots (2.6)$$

Where • indicates multiplication in the finite field GF(2^8)

As an example, let us take the first column of a matrix to be $s_{0,0}$ = {87}, $s_{1,0}$ = {6E}, $s_{2,0}$ = {46}, $s_{3,0}$ = {A6}. This would indicate that $s_{0,0}$ = {87} is transformed to the value $s'_{0,0}$ = {47} which can be shown by solving out the first line of equation (2.6) with j = 0.

Thus:

$$(02 \bullet 87) \oplus \left(03 \bullet s_{1,j}\right) \oplus 46 \oplus A6 = 47$$

To show this is the case, each Hexadecimal digits can be represented by a polynomial:

$$\{02\} = x$$
$$\{87\} = x^7 + x^2 + x + 1$$

By multiplying these two together:

$$x \bullet (\mathrm{x}^7 + \mathrm{x}^2 + \mathrm{x} + 1) = x^8 + x^3 + x^2 + x$$

The degree that results from this operation is greater than 7 so it must be reduced it modulo an irreducible polynomial $m(x)$.

The designers of AES chose $m(x) = x^8 + x^4 + x^3 + x + 1$. So it can be seen that

$$(\mathrm{x}^8 + \mathrm{x}^3 + \mathrm{x}^2 + 1) mod (x^8 + x^4 + x^3 + x + 1) = x^4 + x^2 + 1$$

This is equivalent to (0001 0101) in binary. This procedure can be used to solve the other terms. Thus the result is:

$$\begin{array}{rl} & 0001\ 0101 \\ & 1011\ 0010 \\ & 0100\ 0110 \\ \oplus & 1010\ 0110 \\ \hline & 0100\ 0111 = \{47\} \end{array}$$

Modulo $m(x)$ is in fact a simpler way to do the multiplication. If it is multiplied by {02} then a 1-bit left shift is done followed by a conditional bit wise XOR with (00011011) if the leftmost bit of the original value

(prior to the shift) was 1. Multiplication by other numbers is really a repeated procedure of this method [47].

However, the important note is that a multiplication process has been reduced to a shift and an XOR operation. This is one of the reasons of indicates why AES is efficient [45].

The InvMixColumns is expressed by the following matrix multiplication [47]:

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} \acute{s}_{0,0} & \acute{s}_{0,1} & \acute{s}_{0,2} & \acute{s}_{0,3} \\ \acute{s}_{1,0} & \acute{s}_{1,1} & \acute{s}_{1,2} & \acute{s}_{1,3} \\ \acute{s}_{2,0} & \acute{s}_{2,1} & \acute{s}_{2,2} & \acute{s}_{2,3} \\ \acute{s}_{3,0} & \acute{s}_{3,1} & \acute{s}_{3,2} & \acute{s}_{3,3} \end{bmatrix} \quad \ldots(2.7)$$

This first matrix of equation (2.5) is seen to be the inverse of the first matrix in equation (2.7). If these are labeled as A and A^{-1} respectively and state before the mix columns operation labeled as S and after as $\acute{S}$, it can be seen that :

$$\boldsymbol{A S = \acute{S}}$$

Therefore:

$$\boldsymbol{A^{-1}\acute{S} = A^{-1}A S = S}$$

4. Add round key transformation

In this stage called AddRoundKey the 128 bits of state matrix are bitwise XORed with the 128 bits of the round key. The procedure is really a columnwise operation between the 4 bytes of a state column and one code-word (4 bytes) of the round key. This transformation is as easy as possible which advances in efficiency but it also affects each bit of state [47].

2.6.2 AES key expansion

The AES key expansion process takes as input a code-word key and generates a linear array of 44 words where each round exploits 4 of these words as depicted in figure (2.12). Each word consists of 32 bytes which means each sub-key is 128 bits long. Figure (2.16) illustrates pseudo code for producing the expanded key from the original key [47].

```
KeyExpansion (byte key[16], word w[44])
{
    word temp
    for (i = 0; i < 4; i++)    w[i] = (key[4*i], key[4*i+1],
                                       key[4*i+2],
                                       key[4*i+3]);

    for (i = 4; i < 44; i++)
    {
     temp = w[i - 1];
     if (i mod 4 = 0)    temp = SubWord (RotWord (temp))
                                ⊕ Rcon[i/4];
     w[i] = w[i-4] ⊕ temp
    }
}
```

Figure (2.16) Key expansion pseudocode [47].

The key is copied into the first four words of the expanded key. The rest of the expanded key is placed in four words to the same time. Every added word $w[i]$ depends on the following word, $w[i-1]$, and the 4 positions previous word $w[i-4]$. In 3 out of 4 cases, simple XOR will be applied. For a word whose position in the w array is a multiple of four, a more complex function is applied. Figure (2.17) illustrates the production of the first eight words of the expanded key using the symbol g to denote that complex function. The function g consists of the following sub-functions [49]:

1. **RotWord**: applies a one-byte circular left shift on a word. Where an input word [b_0, b_1, b_2, b_3] will be [b_1, b_2, b_3, b_0].

2. **SubWord**: performs a byte substitution on each byte of its input word, using the s-box.
3. The result of steps 1 and 2 is XORed with round constant, $Rcon[j]$.

The round constant is a word in which the three least significant bytes are always 0. Therefore, the effect of an XOR of a word with $Rcon$ is to only apply an XOR on the most significant byte of the word. The round constant is differs for each round and is represented as $Rcon[j] = (RC[J],0,0,0)$, with $RC[1]= 1$, $RC[j] = 2 \bullet RC[j-1]$ and with multiplication defined over the field GF(2^8) [47].

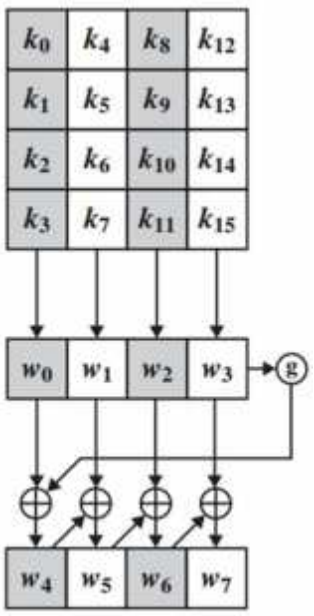


Figure (2.17) AES key expansion [47].

The key expansion was introduced to be impervious to known cryptanalytic attacks. The inclusion of a round-dependent round constant removes the symmetry, or similarity, among the way in which round keys are produced in different rounds [49].

Figure (2.18) gives a summary of each of the rounds. The ShiftRows column is illustrated here as a linear shift which introduces a better idea of how this representation helps in the encryption [47].

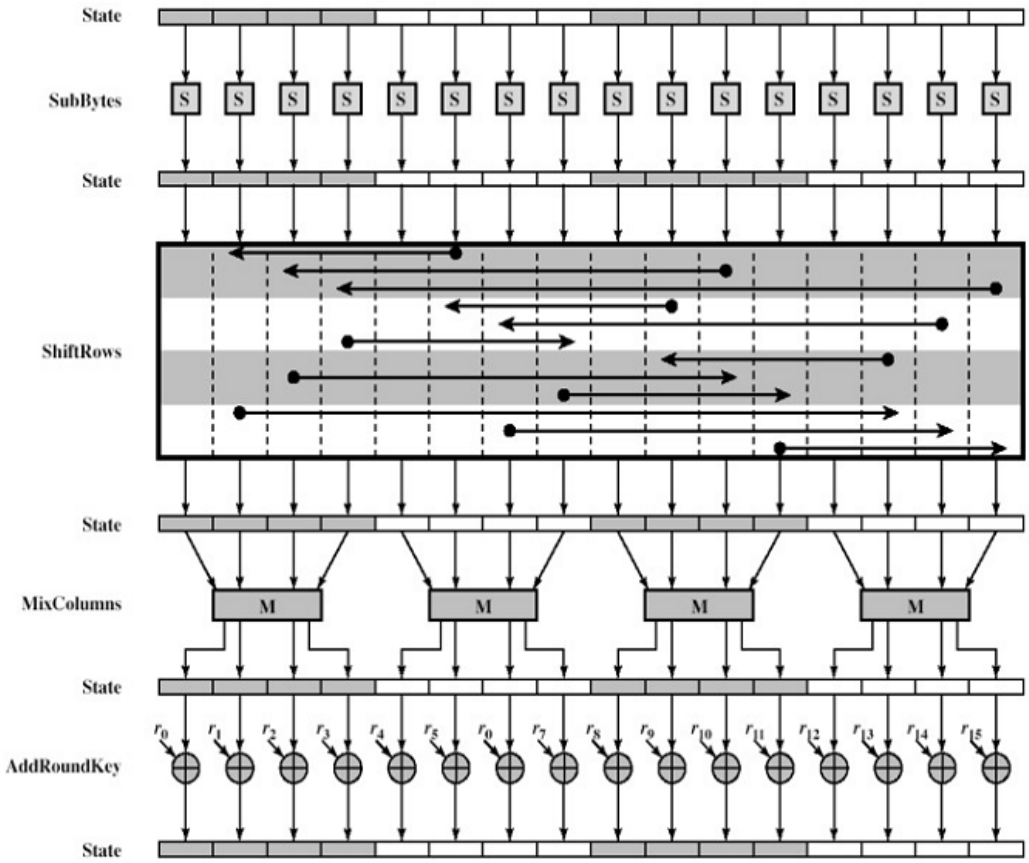


Figure (2.18) AES encryption round [47].

2.6.3 Equivalent inverse cipher

As shown in figure (2.14) the decryption operations are not identical to the encryption operations. However, the way of the key schedules is the same for both. This has the drawback that two individual software or firmware modules are required for applications that need the two operations. However, it is possible to develop an equivalent inverse algorithm. This means that decryption has the same structure as the encryption algorithms. However, to achieve this, key schedule must be changed [47].

Chapter Three

The Proposed Steganography System

3.1 Introduction

In this chapter, the proposed steganography system design is simulated, where the proposed algorithm relies on processing both of cover and secret image to reach the optimum results. The secret image is either gray or color image.

To add more security, the data to be hidden is encrypted using two types of encryption methods, first using a Simple Private Key (SPK) encryption, then using AES algorithm and compare the effectiveness of the two methods on the system.

In these algorithms, interest has been expressed in the quality of the extracted secret information beside the quality of the stego image, compared with the original cover.

3.2 The Transmitter Side

The algorithm is simulated in blind system which means that the receiver does not need the original cover image to extract the information hiding.

In this algorithm, the transmitter analyzes the cover (digital image) to determine the color channels and adds the fourth channel (Alpha channel) changing the extension of the image from JPG to PNG with non-pre-multiplied background channel. This transformation of image extension and bit-depth will prepare the cover image to accommodate the secret data.

In parallel, the secret image which is a gray image or color image, decomposed to a bit stream using Bit-plane slicing to compress it and decreasing the total amount of data to be embedded, also to convert the image to a sequence of bits that will be more controllable and easy to implement and embed. The data then encrypted to increase the level of security.

Then, the secret information is embedded using LSB embedding technique and spreading the message bit stream all over the cover image and in each channel of the cover image pixels (Red, Blue, Green, and Alpha).

The proposed system of the transmitter side is shown in the Figure (3.1) for SPK encryption algorithm and in Figure (3.2) for AES encryption algorithm:

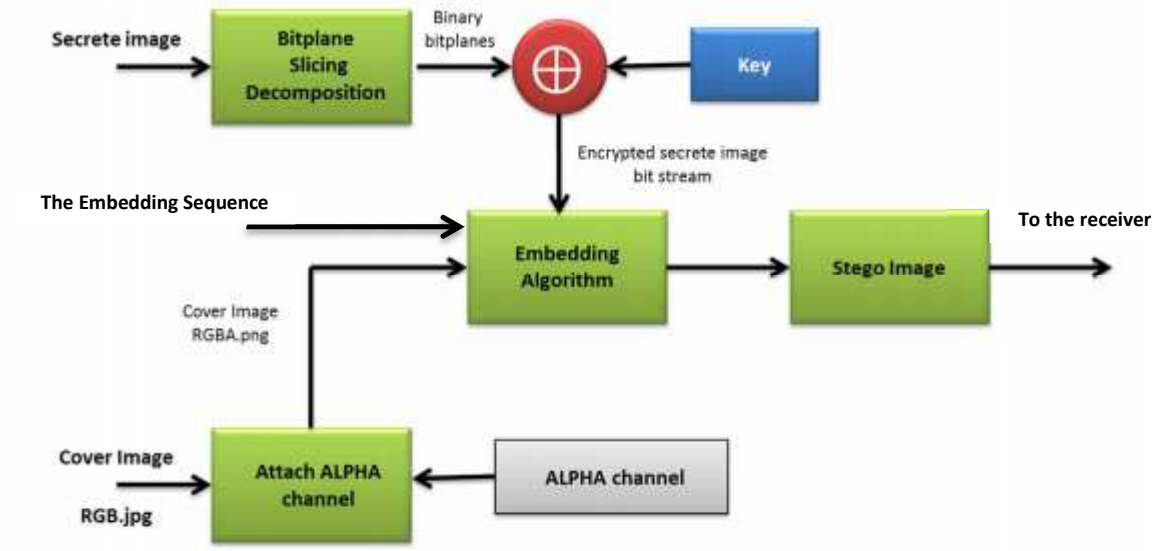


Figure (3.1) The main block diagram on the transmitter side of the SPK model.

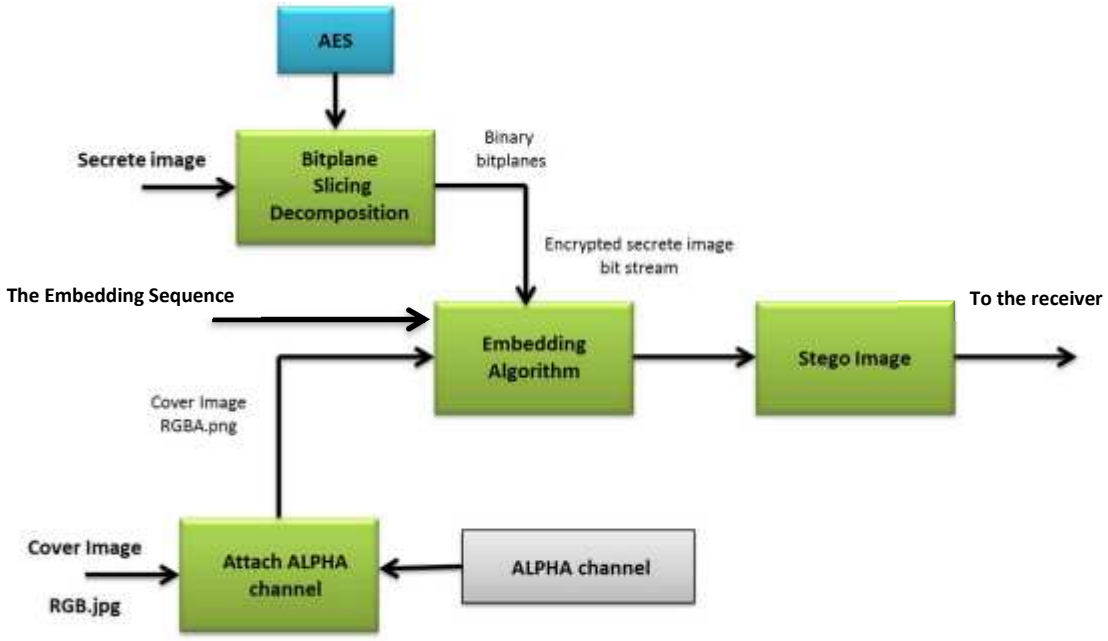


Figure (3.2) The main block diagram on the transmitter side of the AES model.

3.2.1 Preparation of the cover image

First in this system, the cover image should be selected carefully like choosing the cover with high details so when the least significant bits of pixels are replaced with the secret image bits; the cover image will not have a noticeable degradation.

Given cover image is a color image as in Figure (3.3).

Let A be an original color image having size $m * n * p$ represented as:

$$A = \left\{ x(i,j,k) \middle| \begin{matrix} 0 \le i < m, 0 \le j < n \;\; 0 \le k < p \\ x(i,j,k) \epsilon \{0,1,2,3,4, \ldots\ldots, 255\} \end{matrix} \right\} \quad \ldots(3.1)$$

Where, m is the length of the image, n is the width, and p is the bit depth of the image.

The value of k varies from 1 to 3

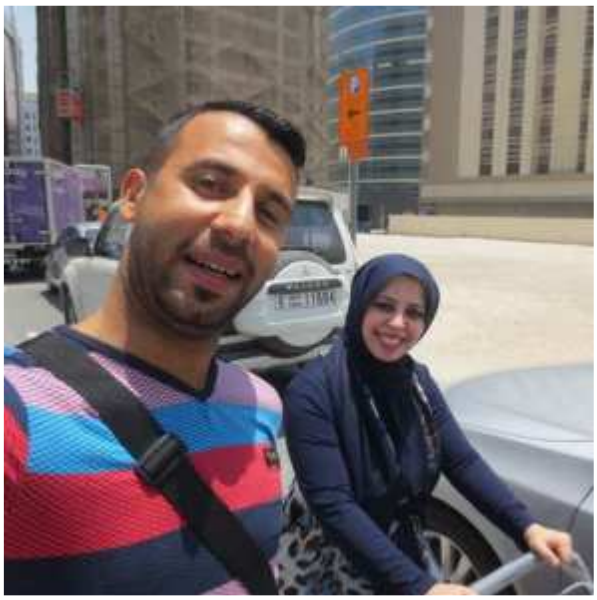


Figure (3.3) The cover image**.**

This image has the extension of JPG. It has 3 color channels (Red, Blue, Green). To add the fourth channel, it must be defined:

$$alpha = \left\{ x(i,j) \middle| \begin{matrix} 0 \le i < m, 0 \le j < n \\ x(i,j) = 255 \end{matrix} \right\} \quad \ldots(3.2)$$

The size of the alpha channel is exactly the same to that of cover color image.

There are two cases of selecting the value of Alpha channel:

Case 1: the image is a color image has a size of 512*512 and alpha channel comprises all ones then output will be a full color image.

Case 2: the image is a color image has a size of 512*512 and alpha channel comprises all zeros then output will be a gray image.

Figure (3.4) depicts the two cases of choosing the value of Alpha channel.

(Case 1) (Case 2)

Figure (3.4) The resulted image if the value of Alpha is (Case 1) all ones, (Case 2) all zeros.

In this work, the Alpha is chosen to be all ones, and this mean alpha channel will be a white plane acts as a transparent background of the image.

Preparation of the cover image
Input: selected cover image.jpg
Output: modified cover image.png
Step 1 Select the cover image with high details.
Step 2 Define 4th plane i.e. alpha channel with all ones.
Step 3 Attach alpha channel to the selected cover image.
Step 4 End

3.2.2 Preparation and decomposition of secret image

1. Grayscale secret image

Let G is a secret grayscale image has size $r * c$ represented as

$$G = \left\{ x(i,j) \middle| \begin{matrix} 0 \le i < r, 0 \le j < c \\ x(i,j) \in \{0,1,2,3,4, \dots \dots \dots, 255\} \end{matrix} \right\} \quad \dots(3.3)$$

Where, r is the length, and c is the width of the grayscale image

Figure (3.5) shows a grayscale secret image.

Figure (3.5) Grayscale secret image.

This 2D secret grayscale image is first passed through bit-plane slicing algorithm.

For grayscale images have 8 bit-planes, this can be represented as follows:

$$Pl_k = \left\{ x(i,j,k) \middle| \begin{matrix} 0 \le i < r, 0 \le j < c \\ x(i,j,k) \in \{0,1\} \end{matrix} \right\} \quad \ldots(3.4)$$

Where: $1 \le k \le 8$

The 8^{th} bit-plane contains more information than other planes, then for embedding process could only choose 8^{th}, 7^{th}, 6^{th} and 5^{th} bit-plane.

To convert all the 4 selected bit-planes into a 1D array as shown below:

$1 * (r * c)$, and represented as follows for each of 4 upper bit-planes:

$$sec_{array} = \left\{ x(1,j) \middle| \begin{matrix} 0 \le j < (r * c) \\ x(1,j) \in \{0,1\} \end{matrix} \right\} \quad \ldots(3.5)$$

To combine all four strings into 1D binary secret array:

$$sec_{total} = [sec_{array5}\ sec_{array6}\ sec_{array7}\ sec_{array8}] \quad \ldots(3.6)$$

Secret 1D array then divided into 4 parts.

Then by finding the length of the sec_{total} and the length of each divided string. Suppose this length is :

$$a_a = len/4 \quad \ldots(3.7)$$

$$b_b = a_a + a_a \quad \ldots(3.8)$$

$$c_c = a_a + a_a + a_a \quad \ldots(3.9)$$

The content of the 1st secret string is:

$$\text{sec}_{str1} = \text{sec}_{total}(1: a_a) \quad \ldots(3.10)$$

The content of the 2nd secret string is:

$$\text{sec}_{str2} = \text{sec}_{total}(a_a + 1: b_b) \quad \ldots(3.11)$$

The content of the 3rd secret string is:

$$\text{sec}_{str3} = \text{sec}_{total}(b_b + 1: c_c) \quad \ldots(3.12)$$

And the content of the 4th secret string is:

$$\text{sec}_{str4} = \text{sec}_{total}(c_c + 1: end) \quad \ldots(3.13)$$

Preparation of the grayscale secret image
Input: the grayscale secret image
Output: encrypted bit-stream
Step 1 Select the secret image with gray-level.
Step 2 Use Bit-slicing technique on the secret image and decompose the image into 8 bit-planes.
Step 3 Select the upper 4 bit-planes (MSE bit-planes).
Step 4 Convert each selected bit-planes into 1D array.
Step 5 Combine the 4 1D arrays into one 1D arrays of bit-stream.
Step 6 Divide the 1D bit-stream array by 4 parts.

Step 7 Find the length of 1D bit-stream array and the length of each divided arrays.

Step 8 End

2. Color secret image

Let R is a secret color image having size $r * c$ represented as:

$$R = \left\{ x(i,j,k) \middle| \begin{matrix} 0 \le i < m, 0 \le j < n \;\; 0 \le k < p \\ x(i,j,k) \epsilon \{0,1,2,3,4, \dots \dots \dots, 255\} \end{matrix} \right\} \quad \dots(3.14)$$

Figure (3.6) shows a color secret image.

Figure (3.6) A color secret image.

The secret image is an RGB image with 24 bit-depth, which means it has 24 bit-planes (8 bit-planes for each color channel r, g, and b). This representation can be treated as 3 monochrome images by extracting Red channel, Green channel, and Blue channel separately and apply Bit-Plane slicing on each of the three channels, where :

R_{red} = *channel* (: , : , 1);

R_{green} = *channel* (: , : , 2); …(3.15)

R_{blue} = *channel* (: , : , 3);

Using the formula (3.4) to apply bit-plane slicing:

$$Bitplane(R_{red}) = [\ pl_1, pl_2, pl_3, pl_4, pl_5, pl_6, pl_7, pl_8\ ];$$
$$Bitplane(R_{green}) = [\ pl_{11}, pl_{12}, pl_{13}, pl_{14}, pl_{15}, pl_{16}, pl_{17}, pl_{18}\ ]; \quad \ldots(3.16)$$
$$Bitplane(R_{blue}) = [\ pl_{21}, pl_{22}, pl_{23}, pl_{24}, pl_{25}, pl_{26}, pl_{27}, pl_{28}\ ];$$

Choosing the 4 upper planes from each channel (for embedding process), then converting the selected 2D bit-planes into 1D array as in formula (3.5) and combine all 12 arrays into 1D binary secret array :

$$sec_{total} = [sec_{array5}\ sec_{array6}\ sec_{array7}\ sec_{array8} sec_{array15}\ sec_{array16}$$
$$sec_{array17}\ sec_{array18} sec_{array25}\ sec_{array26}\ sec_{array27}\ sec_{array28}] \quad \ldots(3.17)$$

The 1D array sec_{total} is divided into 4 parts preparing to embed in the cover 4 channels.

Preparation of the color secret image

Input: the color secret image

Output: encrypted bit-stream

Step 1 Select the secret image with RGB color space.

Step 2 Extract R, G, and B channels from the secret image.

Step 3 Use Bit-slicing technique on each channel of the RGB secret image and decompose the image into 24 bit- planes.

Step 4 Select the upper 4 bit-planes (MSE bit-planes) from each group of bit-planes obtained from each channel. The total of selected bit-planes is 12.

Step 5 Convert each selected bit-planes into a 1D array.

Step 6 Combine the 12 1D arrays into one 1D arrays of bit-stream.

Step 7 Divide the 1D bit-stream array by 4 parts.

Step 8 Find the length of 1D bit-stream array and the length of each divided arrays.

Step 9 End

3.2.3 Encryption of the secret image

For more security, encryption is applied to the secret image. In this work, two methods were used separately for encryption.

1. Simple private key (SPK) encryption

In this method, a simple encryption algorithm using a private key. Secret key is a binary array with length varies for every secret string:

$$\text{sec\_}key = \left\{ x(1,j) \middle| \begin{matrix} 0 \le j < sec_{total} \\ x(1,j)\ \epsilon\ \{0,1\} \end{matrix} \right\} \quad \ldots(3.18)$$

Secret key and secret messages are XORed with each other's to produce encrypted secret strings as follows:

$$\begin{aligned} enc\_\text{sec}\_img1 &= \{\text{sec}_{str1} \oplus \text{sec } \_key\} \\ enc\_\text{sec}\_img2 &= \{\text{sec}_{str2} \oplus \text{sec } \_key\} \\ enc\_\text{sec}\_img3 &= \{\text{sec}_{str3} \oplus \text{sec } \_key\} \\ enc\_\text{sec}\_img4 &= \{\text{sec}_{str4} \oplus \text{sec } \_key\} \end{aligned} \quad \ldots(3.19)$$

Figure (3.7) illustrate the encryption process.

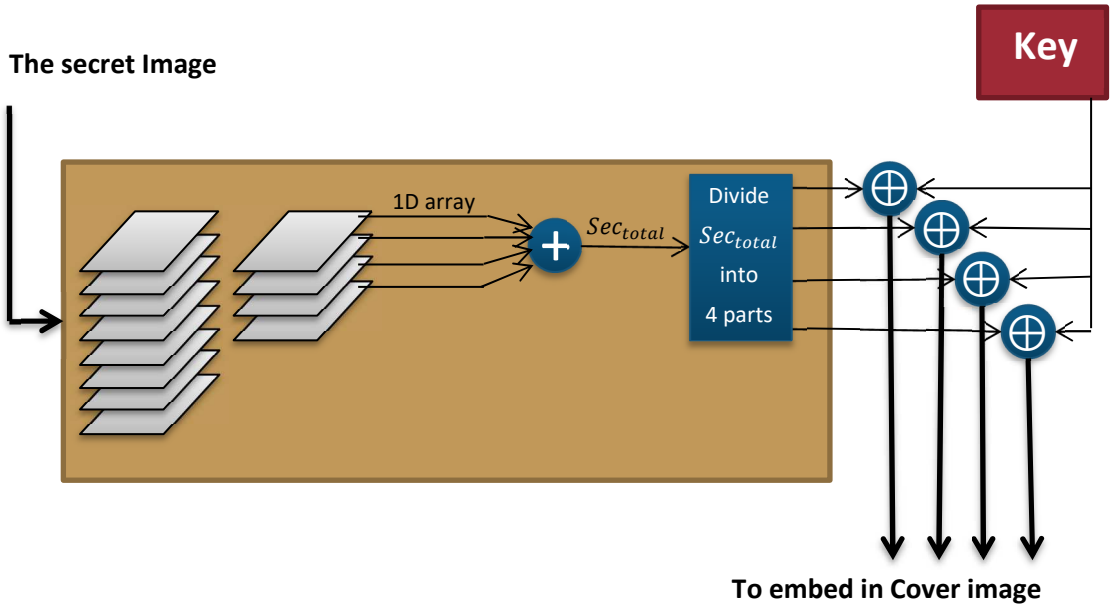


Figure (3.7) Encryption of Secret image using SPK.

2. AES encryption

In this method, encryption is applied on the secret image after taking the 4 upper bit-planes that selected from bit-plane slicing process. The 2D is encrypted before it converted to 1D array. The key used is 128 bits and number of rounds is 10.

Figure (3.8) illustrates the encryption process.

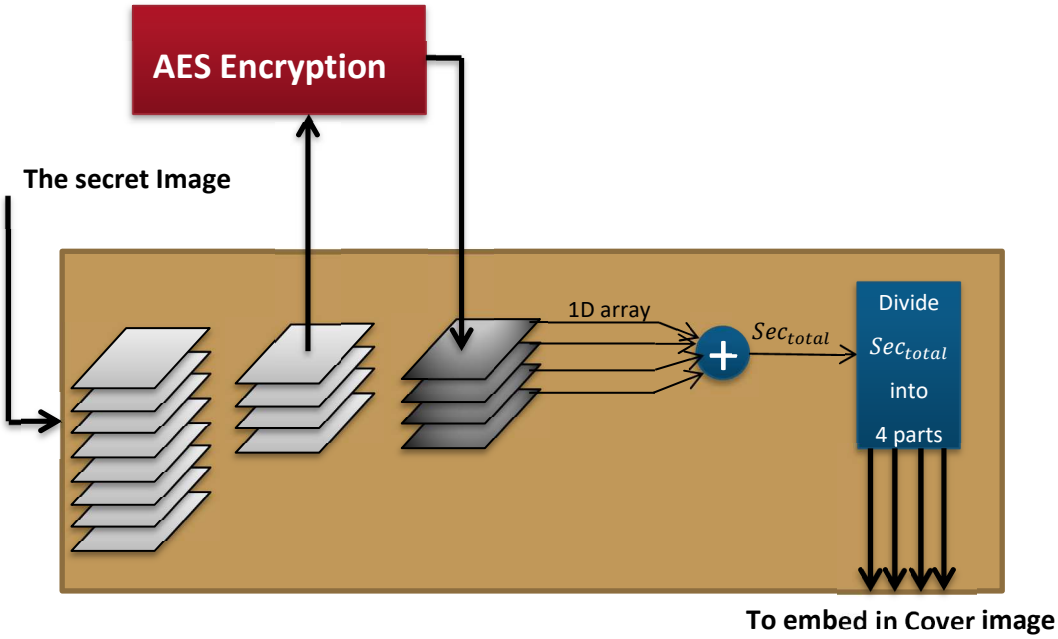


Figure (3.8) The encryption of secret image using AES algorithm.

The s-box of the AES is generated as follows :

1) Initiate the s-box with byte values in row-by-row ascending order. Thus, the value of the byte at row a, column b is {ab}.
2) Convert each byte value in the initiated s-box to its multiplicative inverse in the finite field GF(2^8), the value {00} is converted to itself.
3) Assume that each byte in the s-box consists of 8 bits named ($b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$). Apply the affine transform to each bit in the s-box bytes as follows:

$$\acute{b}_i = b_i \oplus b_{(i+4)mod8} \oplus b_{(i+5)mod8} \oplus b_{(i+6)mod8} \oplus b_{(i+7)mod8} \oplus v_i \quad \ldots(3.20)$$

Where v_i is the i^{th} bit of byte v with the value equal to {63}, where, ($v_7 v_6 v_5 v_4 v_3 v_2 v_1 v_0$) = (01100011). The prime (´) denotes that the variable is updatable by the value on the right.

This transformation can be depicted in matrix form as follows:

$$\begin{bmatrix} \acute{b}_0 \\ \acute{b}_1 \\ \acute{b}_2 \\ \acute{b}_3 \\ \acute{b}_4 \\ \acute{b}_5 \\ \acute{b}_6 \\ \acute{b}_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad \ldots(3.21)$$

Each element in the product matrix is elements of one row bitwise XORed with elements of one column. Moreover, the final operation (addition) in equation (3.21), is a bitwise XOR, the inverse s-box is obtained by taking the inverse of equation (3.21), starting by taking the multiplicative inverse in GF(2^8), then applying affine transformation.

Figure (3.9) shows the secret image and the encrypted secret image:

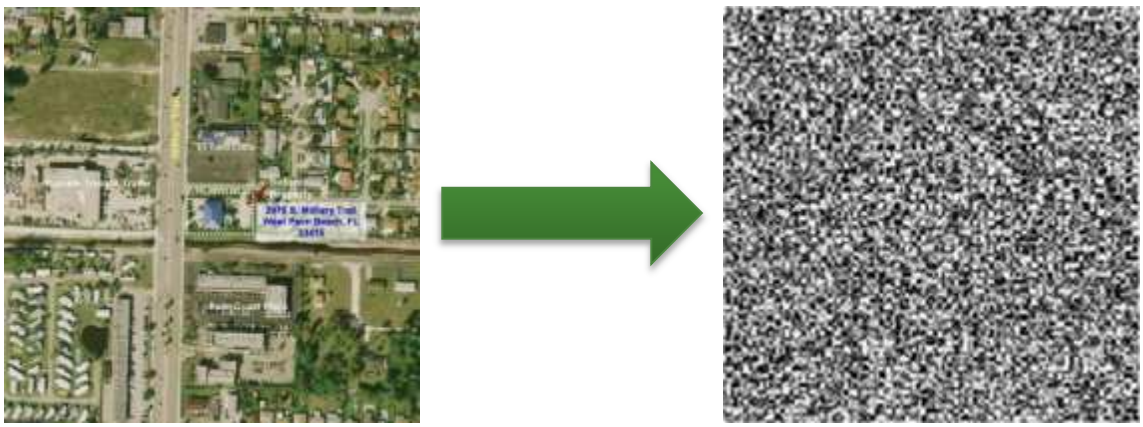


Figure (3.9) The secret image and the encrypted secret image.

3.2.4 Proposed embedding algorithm

In this algorithm, a secret image will be hidden in a cover image using LSB substitution. Embedding operation is variable. The number of LSBs used to embed could be varied from 1 bit to 8 bits. These numbers are used to add more security to the system because the receiver cannot extract the secret image without knowing the number of bits of each channel used for embedding. This embedding sequence is chosen by the sender. An example of the embedding sequence is in the Figure (3.10)

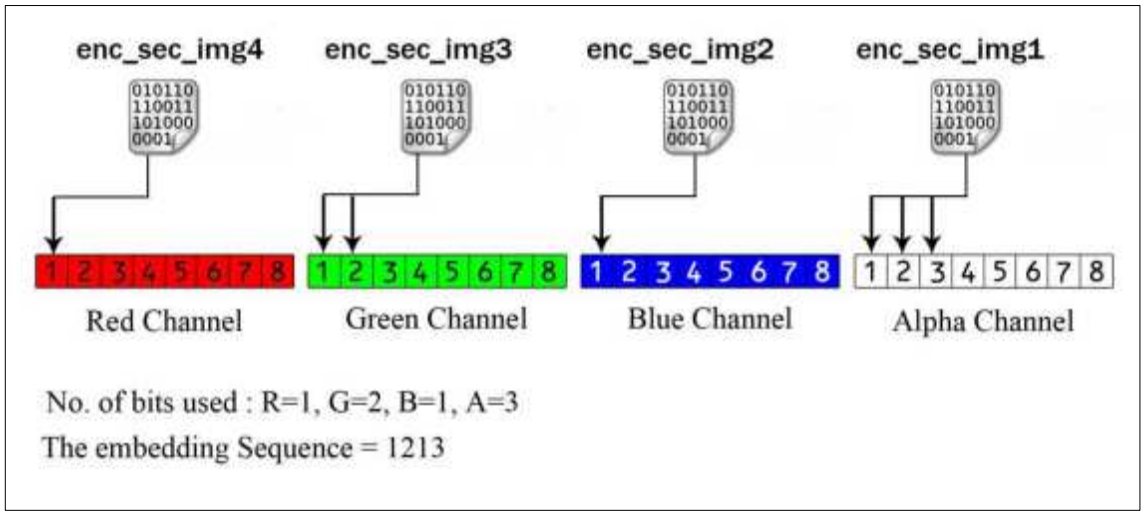


Figure (3.10) Example of embedding sequence.

The steps for these algorithms are:

Method 1 : using SPK algorithm as an encryption method (SPK model):

1- The secret image is decomposed using Bit-plane slicing to 4 bit-planes if it is a grayscale image, and 12 bit-planes if it is a color image. Then convert the selected bit-planes into 1D arrays.
2- Encrypt the 4 1D bit-streams using a private key.
3- Extract Red, Green and Blue planes form cover image, and define the Alpha channel.
4- Embed enc_sec_img1 into Alpha channel.
   Suppose the number of LSB bits can be used for embedding is N

If ($1 \leq N \leq 8$)

(Embed N bits of enc_sec_img1 into every pixel of Alpha channel until the message is not finished)

End

5- Embed enc_sec_img2 into Blue channel.

If ($1 \leq N \leq 8$)

(Embed N bits of enc_sec_img2 into every pixel of Blue plane until the message is not finished)

End

6- Embed enc_sec_img3 into Green channel.

If ($1 \leq N \leq 8$)

(Embed N bits of enc_sec_img3 into every pixel of Green plane until the message is not finished)

End

7- Embed enc_sec_img4 into Red channel.

If ($1 \leq N \leq 8$)

(Embed N bits of enc_sec_img4 into every pixel of Red plane until the message is not finished)

End

The embedding process

Input: the cover image + the secret image

Output: the stego image

Step 1 Decompose the secret image into 4 1D bit-streams, and encrypt these bit-streams with a private key

Step 2 Prepare the cover image and extract the 4 channels Red, Blue, Green and Alpha.

Step 3 Embed the 1st bit-stream into Alpha channel using variable number of Alpha plane bits from 1 bit to 8 bits

Step 4 Embed the 2nd bit-stream into Blue channel using variable

number of Blue plane bits from 1 bit to 8 bits

Step 5 Embed the 3rd bit-stream into Green channel using variable number of Green plane bits from 1 bit to 8 bits

Step 6 Embed the 4th bit-stream into Red channel using variable number of Red plane bits from 1 bit to 8 bits.

Step 7 End

The flow chart of hiding process is shown in Figure (3.11)

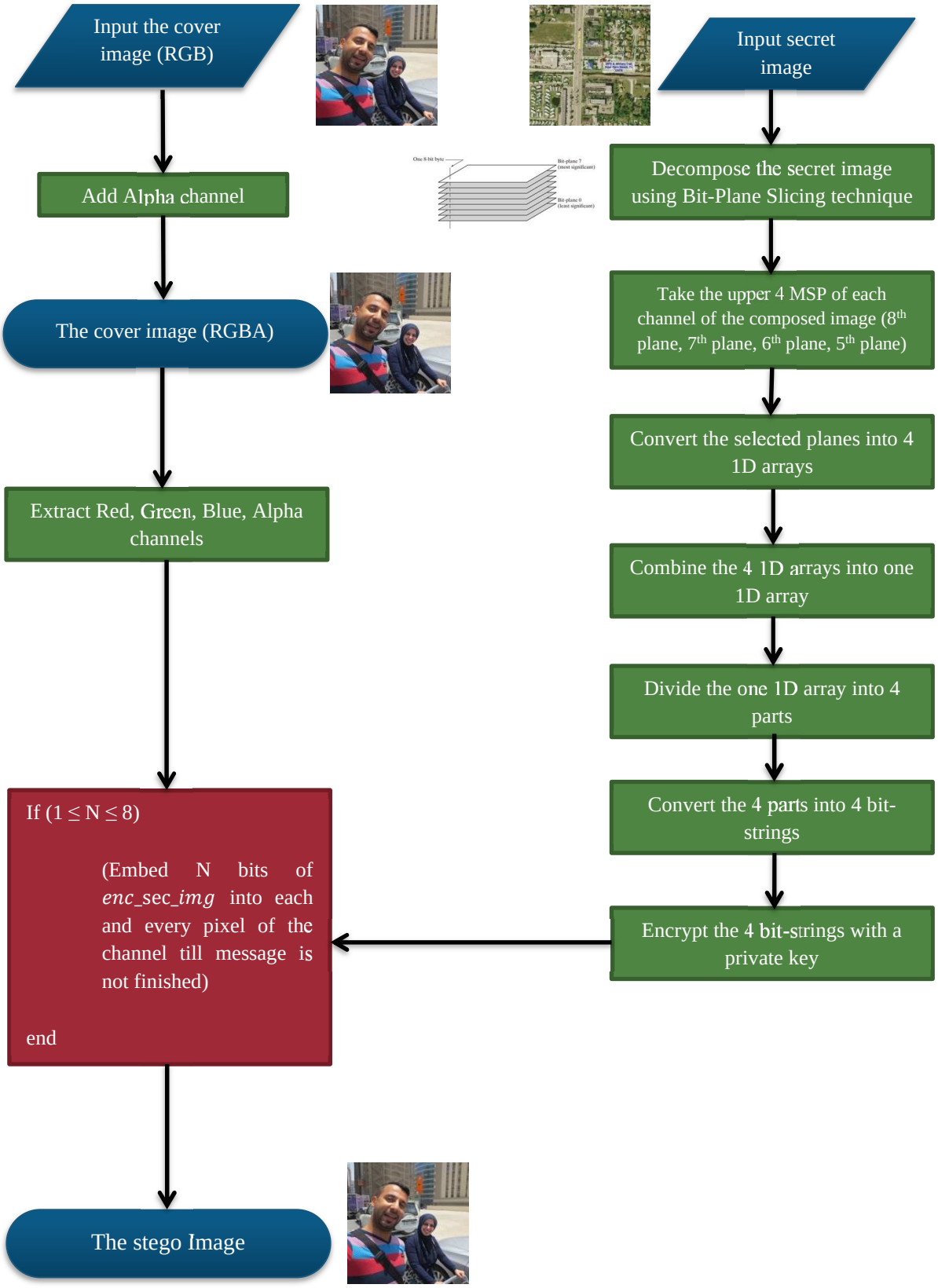


Figure (3.11) Flow chart of hiding process for SPK model.

Method 2 : using AES algorithm as an encryption method (AES model) :

1- The secret image is decomposed using Bit-plane slicing to 4 bit-planes if it is a grayscale image, and 12 bit-planes if it is a color image.
2- Encrypt the selected bit-planes directly using AES algorithm.
3- Convert the encrypted bit-planes into 4 1D arrays.
4- Extract Red, Green and Blue planes form cover image, and define the Alpha channel.
5- Embed sec_{str1} into Alpha channel.
   Suppose the number of LSB bits can be used for embedding is N
   If ($1 \leq N \leq 8$)
   (Embed N bits of sec_{str1} into every pixel of Alpha channel until the message is not finished)
   End
6- Embed sec_{str2} into Blue channel.
   If ($1 \leq N \leq 8$)
   (Embed N bits of sec_{str2} into every pixel of Blue plane until the message is not finished)
   End
7- Embed sec_{str3} into Green channel.
   If ($1 \leq N \leq 8$)
   (Embed N bits of sec_{str3} into every pixel of Green plane until the message is not finished)
   End
8- Embed sec_{str4} into Red channel.
   If ($1 \leq N \leq 8$)
   (Embed N bits of sec_{str4} into every pixel of Red plane untill the message is not finished)
   End

The embedding process

Input: the cover image + the secret image

Output: the stego image

Step 1 Decompose the secret image into 4 bit-planes for each channel, and encrypt these bit-planes using AES algorithm, then transform the encrypted bit-planes into 4 binary streams.

Step 2 Prepare the cover image and extract the 4 channels Red, Blue, Green, and Alpha.

Step 3 Embed the 1st bit-stream into Alpha channel using variable number of Alpha plane bits from 1 bit to 8 bits

Step 4 Embed the 2nd bit-stream into Blue channel using variable number of Blue plane bits from 1 bit to 8 bits

Step 5 Embed the 3rd bit-stream into Green channel using variable number of Green plane bits from 1 bit to 8 bits

Step 6 Embed the 4th bit-stream into Red channel using variable number of Red plane bits from 1 bit to 8 bits

Step 7 End

The flow chart of hiding process is shown in Figure (3.12)

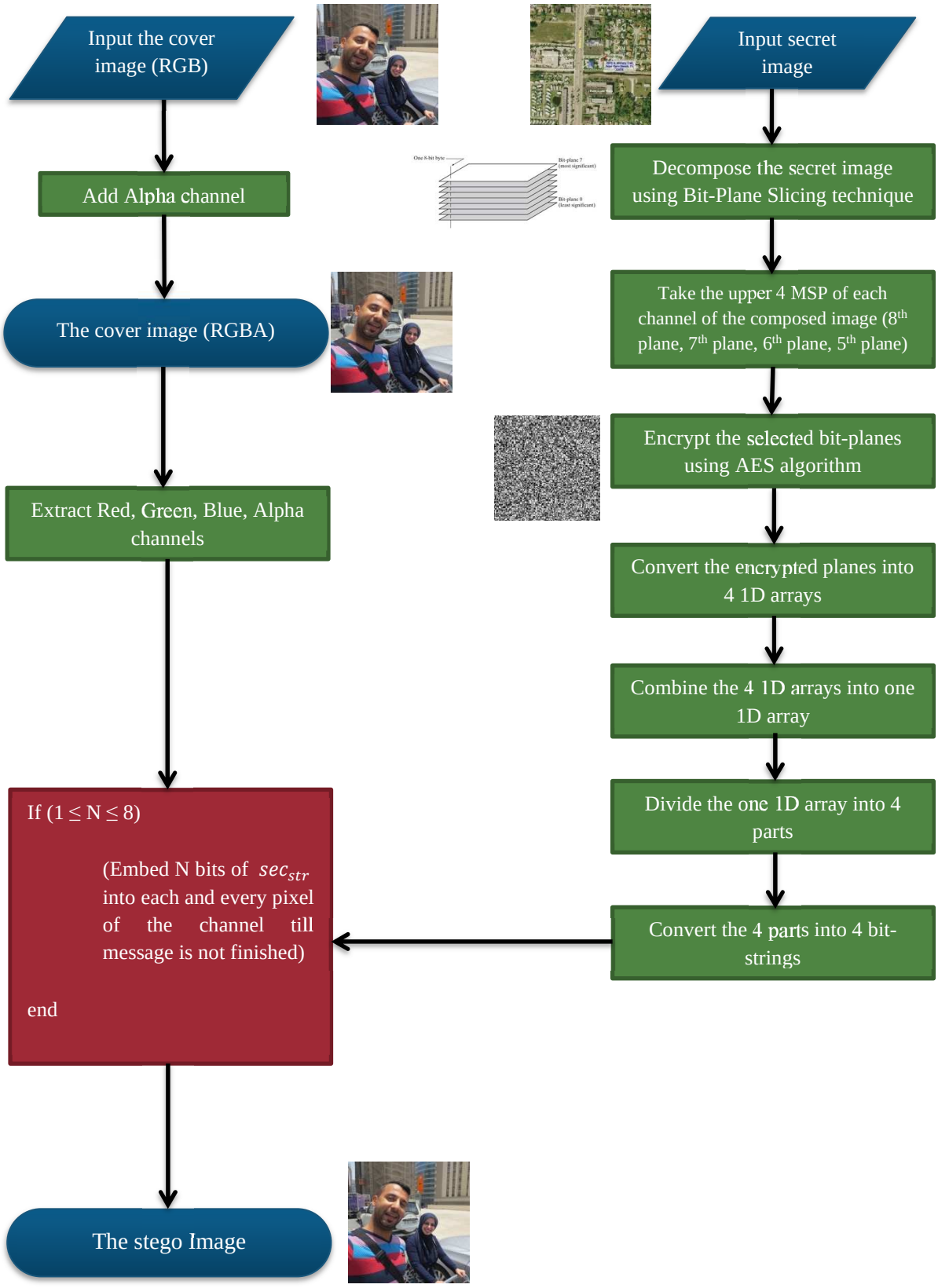


Figure (3.12) Flow chart of hiding process for AES model.

3.3 The Receiver Side

To extract the secret information, the receiver will not need the original cover to be compared with the stego image. The receiver must know the secret key, and the number of bits used in each channel to carry the secret information (the embedding Sequence). There must be a secure channel to transmit the key and the embedding sequence. The key is a symmetric key, which means the encryption and decryption process use the same key.

The proposed system of the receiver side is shown in the Figure (3.13), and (3.14):

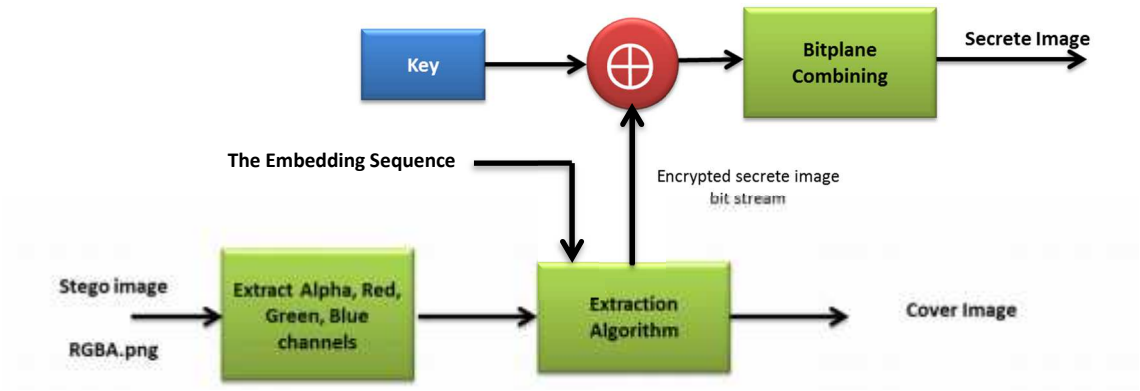


Figure (3.13) Main block diagram of the receiver side using SPK model.

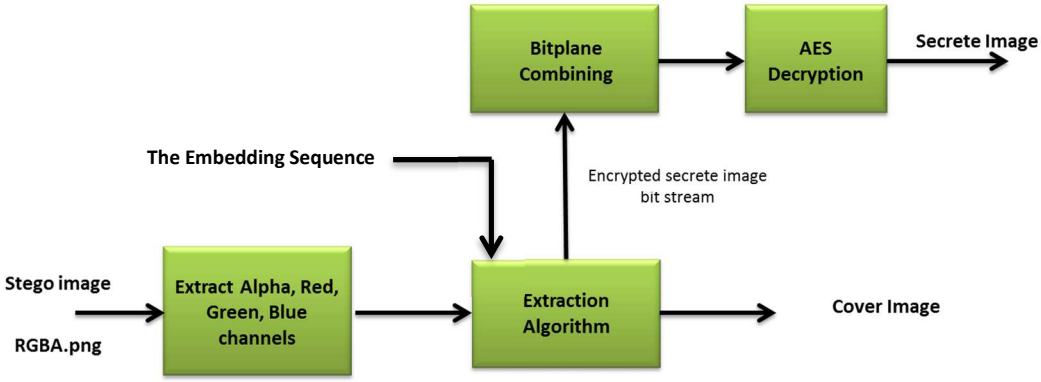


Figure (3.14) Main block diagram of the receiver side using AES model.

The extracting algorithm is the inverse of the embedding algorithms, as shown below:

A) Extraction algorithm using SPK decryption :

1. Extract Alpha channel and Red, Green and Blue plane from RGBA stego image.
2. Use the embedding sequence to extract the encrypted secret bit strings ($enc\_\text{sec}\ \_img$) from each plane of the image. The original cover image is produced in this stage.
3. Use the secret key to decrypt the secret strings by XOR secret key with encrypted bits.

$$\begin{aligned} \text{sec}_{str1} &= \{enc\_\text{sec}\ \_img1 \oplus \text{sec}\ \_key\} \\ \text{sec}_{str2} &= \{enc\_\text{sec}\_img2 \oplus \text{sec}\ \_key\} \\ \text{sec}_{str3} &= \{enc\_\text{sec}\_img3 \oplus \text{sec}\ \_key\} \\ \text{sec}_{str4} &= \{enc\_\text{sec}\_img4 \oplus \text{sec}\ \_key\} \end{aligned} \quad \ldots(3.22)$$

4. Combine all these bit-planes into one image to find the recovered secret image Sec_im using following formula.

$$Sec\_im = (\text{sec}_{str1} * 16) + (\text{sec}_{str2} * 32) + (\text{sec}_{str3} * 64) + (\text{sec}_{str4} * 128) \quad \ldots(3.23)$$

The flow chart of the extraction process is shown in Figure (3.15)

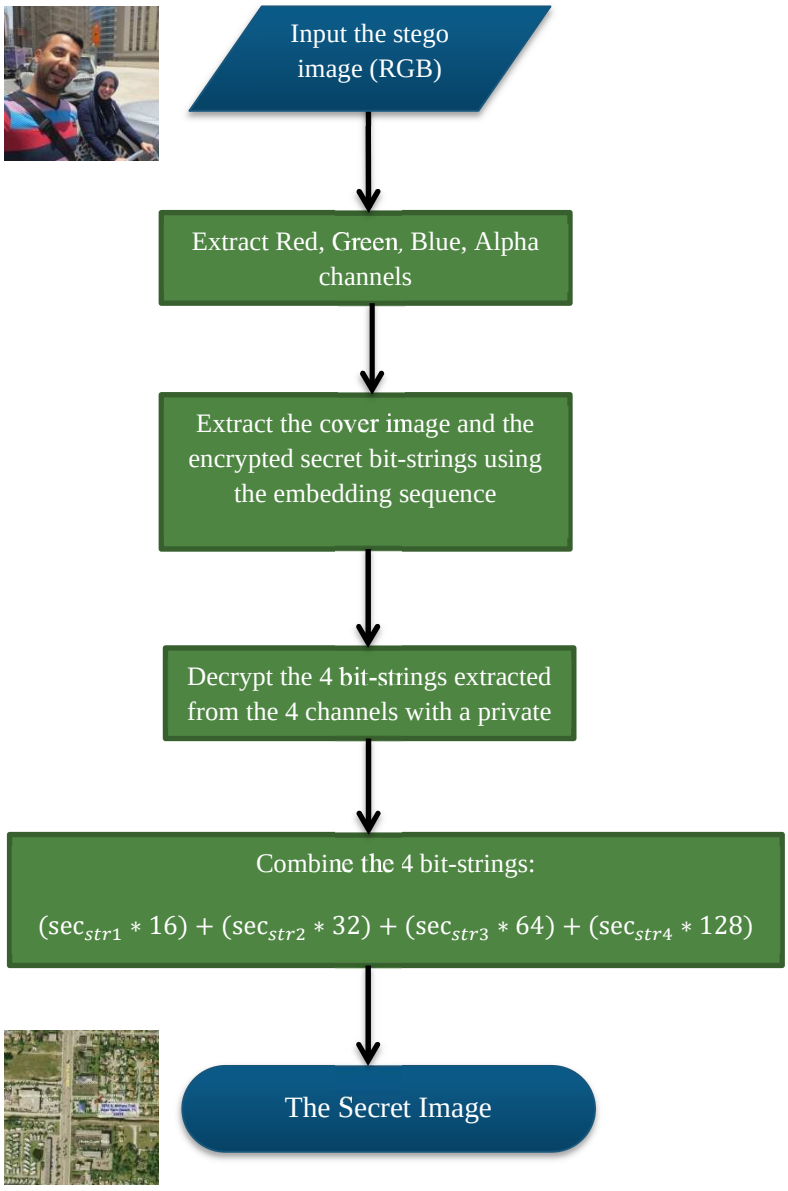


Figure (3.15) Flow chart of the extraction process using SPK model.

B) Extraction algorithm using AES decryption :

1. Extract Alpha channel and Red, Green and Blue plane from RGBA stego image.
2. Use the embedding sequence to extract the bit strings from each plane of the image. The original cover image is produced in this stage.
3. Combine all these bit-planes into one image to find the recovered encrypted secret image *enc_Sec_im* using equation (3.23).

4. Decrypt the encrypted secret image (*enc_Sec_im*) using AES decryption process to recover the secret image.

The flow chart of the extraction process is shown in Figure (3.16)

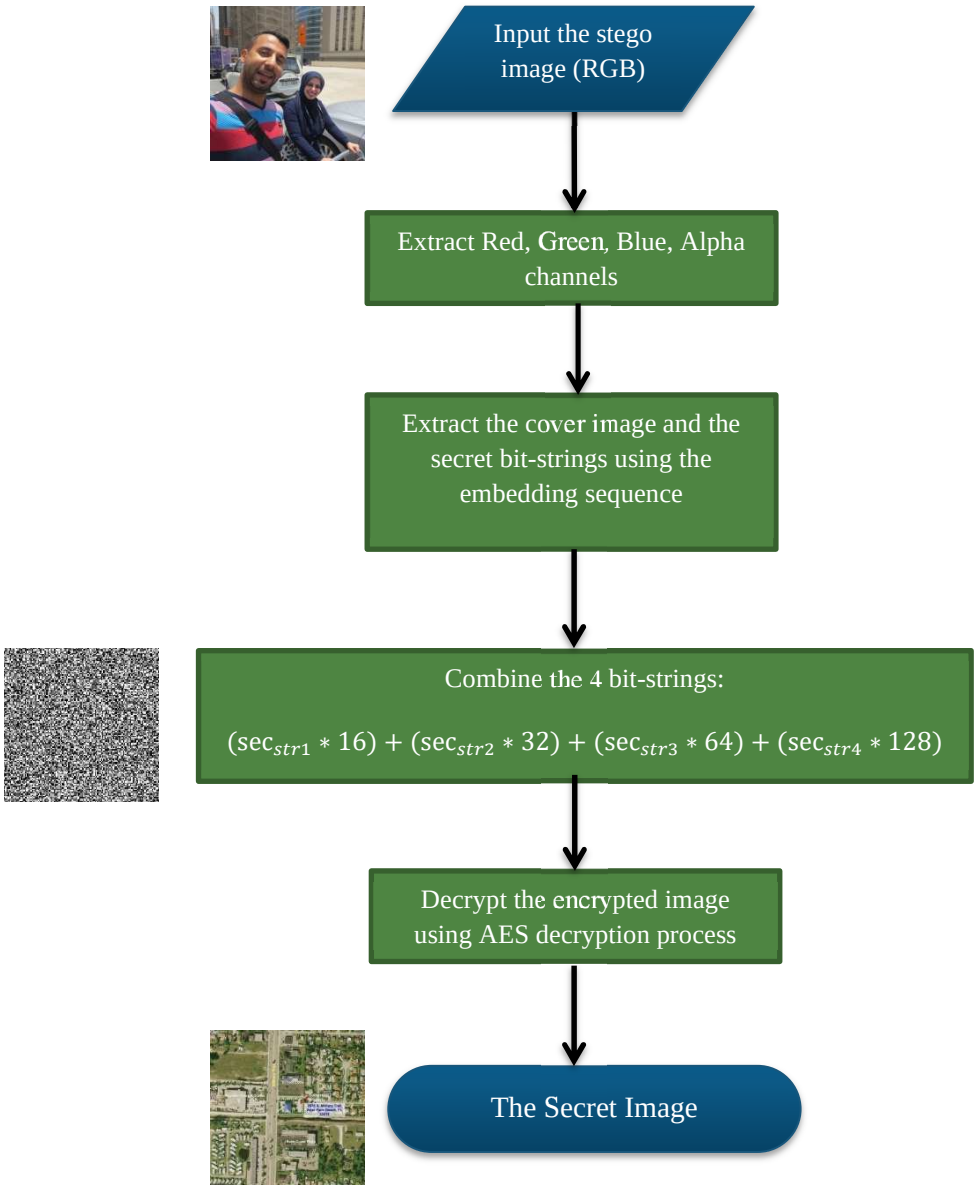


Figure (3.16) Flow chart of the extraction process using AES model.

Chapter Four

Results and Discussion

4.1 Introduction

In this chapter, simulation results are given to demonstrate the performance of the suggested algorithms. The main idea of these algorithms is based on embedding the secret data after processing it by Bit-plane slicing and encrypting it by an encryption algorithm in RGBA cover image after adding the fourth channel (Alpha channel) to the original RGB cover image. The proposed technique increases the embedding capacity and decreases the stego image distortion by hiding the secret information in least significant bits of color channels of the image. Beside that, an encryption algorithm and a variable embedding sequence are used to increase the security, because when a strong algorithm is used, the only way to break the system is to obtain the key and the way that bits embedded in the cover image.

Also after obtaining the results, four tests are applied to measure the quality of stego image. The tests have been performed on a personal computer of 2.20 GHz CPU (CORE i7), and the proposed systems are implemented by MATLAB* (R2011a).

*MATLAB: is a numerical computing environment and fourth-generation programming language. Developed by Math Works, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran. (Wikipedia)

4.2 Measures of Quality of the Stego Images

There are many tests that can be used to measure the quality of the image. The following four tests are used:-

4.2.1 Peak-Signal-to-Noise-Ratio

According to the human visual system (HVS), some degree of distortion between the original image and the modified one is accepted. Here the Peak-Signal-to-Noise-Ratio (PSNR) is employed to test the performance of the method and usually measured in db. To compute the peak signal to noise ratio, then [50]:-

$$PSNR = 10log_{10}\frac{(L-1)^2}{\frac{1}{W\times H}\sum_{i=1}^{W}\sum_{j=1}^{H}[St(i,j)-C(i,j)]^2} \quad \ldots(4.1)$$

Where:

H: height of the two images (because the two images must be of the same size).

W: width of the two images.

i and j : row and column numbers.

L: is the number of the gray scale levels in the two images.

$C(i,j)$: The cover image.

$St(i,j)$: The stego image.

Typical PSNR values range between 20 and 40 dB, the PSNR value of identical images is infinite or undefined [51].

4.2.2 Mean Square Error

Mean Square Error (MSE) shows the mean square error between cover image C and stego image S [51].

$$MSE = \frac{1}{W*H}\sum_{i=0}^{W-1}\sum_{j=0}^{H-1}[St(i\ ,\ j) - C(i\ ,\ j)]^2 \qquad \ldots(4.2)$$

Where:

H: height of the two images (because the two images must be of the same size).

W: width of the two images.

i and j : numbers of row and column.

$C(i,j)$: The cover image.

$St(i,j)$: The stego image.

4.2.3 Normalized Cross-Correlation

In image-processing applications, the brightness of the original image and the modified image can vary according to lighting and exposure conditions; here the two images can be first normalized. That is, the normalizes cross-correlation (NCC) of a cover image C(i,j) with a stego image S(i,j) is given by [52]:

$$NCC = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n}[St(i,j)*C(i,j)]}{\sum_{i=0}^{m}\sum_{j=0}^{n}[C(i,j)*C(i,j)]} \qquad \ldots(4.3)$$

Normalized cross-correlation is one of the techniques applied for images matching, a process used for finding incidences of a pattern or an element within an image.

4.2.4 Average Difference

The Average Difference (AD) of an image is given by [53]:

$$AD = \frac{\sum M*N[1(m,n)-2(m,n)]}{M*N} \qquad \ldots(4.4)$$

Large value of AD indicates that the image has poor quality.

4.3 The Results

For test the proposed method, a personal image was used as an original cover image with size of 512*512, bit depth of 24, and color system of RGB. Different images with different color spaces (grayscale and color images) and different sizes are chosen as secret images, and different amount of bits is embedded in each channel of the RGBA cover image.

Also two different types of encryption were used in this system. First, the simple private key (SPK) encryption, and second, the AES algorithm.

4.3.1 Results of SPK model

1. Grayscale secret images

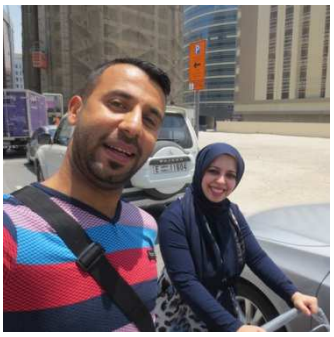
An image of (S400 rocket system) chosen as a secret image with gray- level, 256*256 size, and 8 bit depth. This test is given a code name of SPK-G256. The results are shown in the Table (4.1):

Table (4.1) The results of embedding (256*256) gray-level secret image into (512*512) RGB cover image - SPK model.

No. of bits used	PSNR	MSE	NCC	AD	Bits per channel used			
					R	G	B	A
4	44.296	1.19667	0.998176	0.224758	1	1	1	1
5	42.8214	2.27554	0.997124	0.357586	1	1	1	2
6	42.0624	2.33537	0.997098	0.361763	1	1	2	2
7	41.8718	2.32717	0.997531	0.314182	1	2	2	2
8	41.4298	2.29889	0.997331	0.339325	2	2	2	2
9	38.9966	5.83959	0.995917	0.513168	2	2	2	3
10	38.3686	5.74263	0.996051	0.497177	2	2	3	3

The above results represent the effect of embedding capacity of 25% and a different embedding sequence tested indicates the number of bits used in each pixel for embedding.

Figure (4.1) shows the original image, secret image, stego-image :



The cover image.jpg

The secret image (256x256)

The Stego image.png

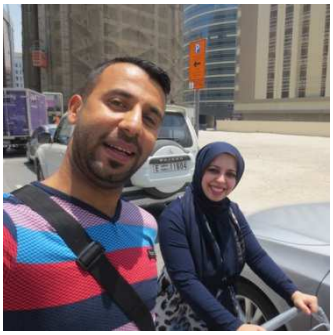
Figure (4.1) Original image, secret image, stego-image, and Stego image of SPK-G256.

The second test is SPK-G256-2: an image of (Military Cavalry) was chosen as a secret image with gray- level, 256*512 size, and 8 bit depth. The results, which represent the embedding capacity of 50% are shown in the Table (4.2):

Table (4.2) The results of embedding (256*512) gray-level secret image into (512x512) RGB cover image – SPK model.

No. of bits used	PSNR	MSE	NCC	AD	Bits per channel used			
					R	G	B	A
4	44.2688	1.25441	0.998173	0.225071	1	1	1	1
5	42.7947	2.23184	0.997292	0.335991	1	1	1	2
6	42.3101	2.45509	0.997128	0.351263	1	1	2	2
7	41.6418	2.32717	0.997586	0.414182	1	2	2	2
8	41.2208	2.45489	0.997121	0.439325	2	2	2	2
9	38.5876	5.71145	0.995317	0.611147	2	2	2	3
10	38.3339	5.60996	0.995922	0.510933	2	2	3	3

Figure (4.2) shows the original image, secret image, and stego-image obtained from embedding process :



The cover image.jpg

The secret image (256x512)

The Stego image.png

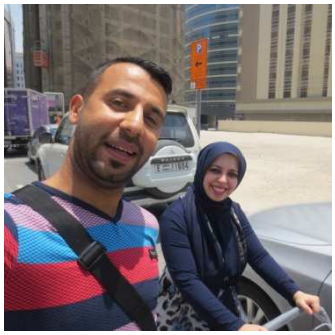
Figure (4.2) Original image, secret image, stego-image of SPK-G256-2.

In the third test SPK-G512, an image of a military site, gray-scale image with the same size of cover image (512*512) is chosen to test the system at the capacity of 100%. Table (4.3) shows the results of using a secret image with the same size of the cover image.

Table (4.3) The results of embedding (512*512) gray-level secret image into (512*512) RGB cover image – SPK model.

No. of bits used	PSNR	MSE	NCC	AD	Bits per channel used			
					R	G	B	A
4	38.1995	4.48968	0.992183	0.850788	1	1	1	1
5	36.5257	9.35152	0.98726	1.4116	1	1	1	2
6	35.8066	9.42559	0.987377	1.39049	1	1	2	2
7	35.346	12.1527	0.987593	1.39125	1	2	2	2
8	35.0255	11.1411	0.9873	1.41145	2	2	2	2
9	33.0036	23.8395	0.982912	2.10169	2	2	2	3
10	32.1526	23.8505	0.983247	2.06615	2	2	3	3

Figure (4.3) shows the original image, secret image, and stego-image generated by the embedding process :



The cover image.jpg

The secret image (512x512)

The stego image.png

Figure (4.3) Original image, secret image, stego-image of SPK-G512.

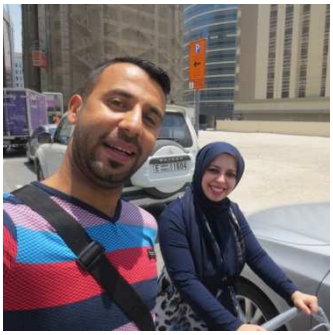
2. Color secret images

For this test SPK-R256, An image of a secret weapon is used as a secret image with size of 256*256, bit-depth of 24 with color space of RGB. Table (4.4) shows the results obtained from embedding the secret image into the cover image.

Table (4.4) The results of embedding (256*256) color secret image into (512*512) RGB cover image – SPK model.

No. of bits used	PSNR	MSE	NCC	AD	Bits per channel used			
					R	G	B	A
4	44.3035	1.03381	0.998166	0.225544	1	1	1	1
5	42.9859	1.97912	0.99726	0.341068	1	1	1	2
6	42.4456	1.99928	0.997337	0.331875	1	1	2	2
7	41.8316	2.79623	0.996897	0.398254	1	2	2	2
8	41.7393	2.33826	0.997176	0.360535	2	2	2	2
9	39.5335	5.7229	0.995589	0.554817	2	2	2	3
10	38.5757	5.81397	0.995624	0.551746	2	2	3	3

Figure (4.4) shows the original image, secret image, and stego-image :



The cover image.jpg

The secret image
(256x256)

The Stego image.png

Figure (4.4) Original image, secret image, stego-image of SPK-R256.

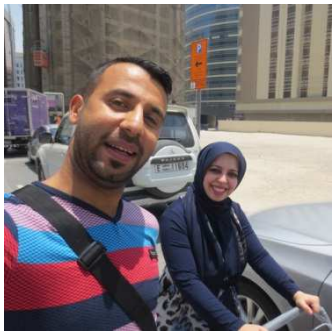
To test the system in the capacity of 50%, in the test SPK-R256-2 a color secret image is used of 256*512 size.

Table (4.5) shows the results obtained from this test.

Table (4.5) The results of embedding (256*512) color secret image into (512*512) RGB cover image – SPK model.

No. of bits used	PSNR	MSE	NCC	AD	Bits per channel used			
					R	G	B	A
4	44.3353	1.25648	0.998181	0.225066	1	1	1	1
5	42.6461	2.24865	0.997851	0.334551	1	1	1	2
6	42.2846	2.46001	0.997114	0.350982	1	1	2	2
7	41.5374	2.35541	0.997422	0.414182	1	2	2	2
8	41.1049	2.45151	0.997121	0.439325	2	2	2	2
9	38.3472	5.71866	0.995517	0.624344	2	2	2	3
10	38.2775	5.73477	0.995878	0.516289	2	2	3	3

Figure (4.5) shows the original image, secret image, and the stego-image obtained from embedding process :



The cover image.jpg

The secret image
(256x512)

The stego image.png

Figure (4.5) Original image, secret image, stego-image of SPK-R256-2.

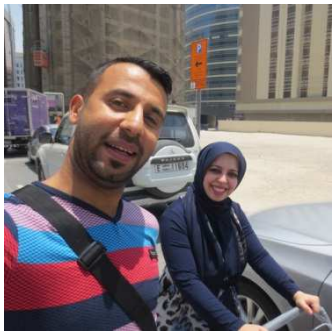
For testing the system in the capacity of 100%, in the test SPK-R512 a color secret image is used of 512*512 size (the same size of the cover image).

Table (4.6) shows the results obtained from this test.

Table (4.6) The results of embedding (512*512) color secret image into (512*512) RGB cover image – SPK model.

No. of bits used	PSNR	MSE	NCC	AD	Bits per channel used			
					R	G	B	A
4	38.1553	4.73063	0.992192	0.850731	1	1	1	1
5	36.4582	9.74084	0.98704	1.43308	1	1	1	2
6	35.7322	9.94222	0.98712	1.41737	1	1	2	2
7	35.2427	12.5415	0.987048	1.44532	1	2	2	2
8	34.933	11.6965	0.986863	1.45187	2	2	2	2
9	32.9779	24.2012	0.98271	2.12202	2	2	2	3
10	32.1398	24.2739	0.98308	2.08162	2	2	3	3

Figure (4.6) shows the original image, secret image, and the stego-image obtained from the embedding process :



The cover image.jpg

The secret image (512x512)

The stego image.png

Figure (4.6) Original image, secret image, stego-image of SPK-R512.

4.3.2 Results of AES model

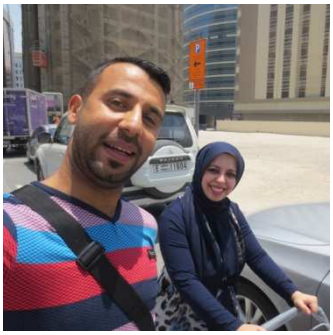
1. Grayscale secret images

In the test given a code name of AES-G256, an image of (S400 rocket system) was chosen as a secret image with gray- level, 256*256 size, and 8 bit depth. The results are shown in the Table (4.7) :

Table (4.7) The results of embedding (256*256) gray-level secret image into (512*512) RGB cover image – AES model.

No. of bits used	PSNR	MSE	NCC	AD	Bits per channel used			
					R	G	B	A
4	44.0745	1.19667	0.998176	0.224758	1	1	1	1
5	42.8201	2.27554	0.997124	0.357586	1	1	1	2
6	42.8214	2.33537	0.997098	0.361763	1	1	2	2
7	42.8248	2.32717	0.997531	0.314182	1	2	2	2
8	41.3613	2.29889	0.997331	0.339325	2	2	2	2
9	38.7408	5.83959	0.995917	0.513168	2	2	2	3
10	38.7436	5.74263	0.996051	0.497177	2	2	3	3

Figure (4.7) shows the original image, secret image, and stego-image :



The cover image.jpg

The secret image
(256x256)

The Stego image.png

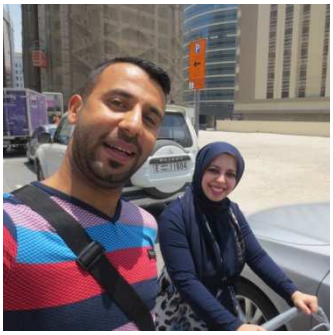
Figure (4.7) Original image, secret image, stego-image of AES-G256.

The second test AES-G256-2 is a secret image with gray- level, 256*512 size, and 8 bit depth. The results are shown in the Table (4.8) :

Table (4.8) The results of embedding (256*512) gray-level secret image into (512*512) RGB cover image – AES model.

No. of bits used	PSNR	MSE	NCC	AD	Bits per channel used			
					R	G	B	A
4	44.1656	1.25441	0.998173	0.225071	1	1	1	1
5	42.806	2.24565	0.997297	0.336273	1	1	1	2
6	42.7864	2.33814	0.997051	0.371763	1	1	2	2
7	42.7831	2.32995	0.997517	0.394195	1	2	2	2
8	41.2154	2.31249	0.997323	0.439145	2	2	2	2
9	39.0408	5.84714	0.995908	0.513168	2	2	2	3
10	39.1159	5.62587	0.995911	0.510933	2	2	3	3

Figure (4.8) shows the original image, secret image, the stego-image obtained from embedding process :



The cover image.jpg

The secret image
(256x512)

The Stego image.png

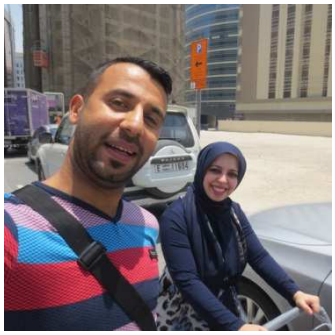
Figure (4.8) Original image, secret image, stego-image AES-G256-2.

In the test AES-G512, an image of a military site, gray-scale image with the same size of cover image (512*512) is chosen to test the system at the capacity of 100%. Table (4.9) shows the results of using a secret image with the same size of the cover image.

Table (4.9) The results of embedding (512*512) gray-level secret image into (512*512) RGB cover image – AES model**.**

No. of bits used	PSNR	MSE	NCC	AD	Bits per channel used			
					R	G	B	A
4	38.4088	4.48968	0.992183	0.850788	1	1	1	1
5	36.2904	9.35152	0.98726	1.4116	1	1	1	2
6	36.2925	9.42559	0.987377	1.39049	1	1	2	2
7	36.2951	12.1527	0.987593	1.39125	1	2	2	2
8	35.2007	11.1411	0.9873	1.41145	2	2	2	2
9	32.6629	23.8395	0.982912	2.10169	2	2	2	3
10	32.6664	23.8505	0.983247	2.06615	2	2	3	3

Figure (4.9) shows the original image, secret image, stego-image.



The cover image.jpg

The secret image (512x512)

The stego image.png

Figure (4.9) Original image, secret image, stego-image AES-G512.

2. Color secret images

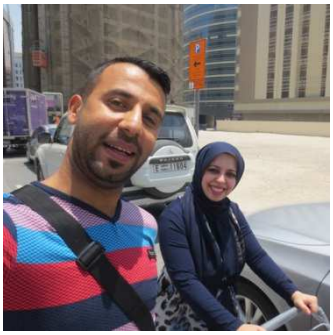
In this test which was given a code name of AES-R256, an image of a secret weapon is used as a secret image with size of 256*256, bit-depth of 24 with color space of RGB.

Table (4.10) shows the results obtained from embedding the secret image into the cover image.

Table (4.10) The results of embedding (256*256) color secret image into (512*512) RGB cover image – AES model.

No. of bits used	PSNR	MSE	NCC	AD	Bits per channel used			
					R	G	B	A
4	44.1877	1.03381	0.998166	0.225544	1	1	1	1
5	42.9572	1.97912	0.99726	0.341068	1	1	1	2
6	42.9612	1.99928	0.997337	0.331875	1	1	2	2
7	42.9828	2.79623	0.996897	0.398254	1	2	2	2
8	42.6136	2.33826	0.997176	0.360535	2	2	2	2
9	40.6552	5.7229	0.995589	0.554817	2	2	2	3
10	40.6573	5.81397	0.995624	0.551746	2	2	3	3

Figure (4.10) shows the original image, secret image, and stego-image.



The cover image.jpg

The secret image
(256x256)

The Stego image.png

Figure (4.10) Original image, secret image, stego-image of AES-R256.

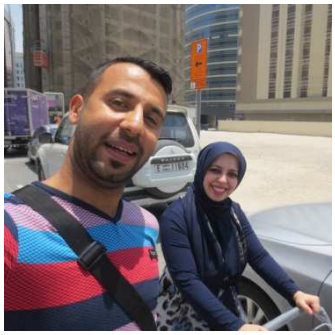
For testing the system in the capacity of 50%, in the test of AES-R256-2, a color secret image is used of 256*512 size.

Table (4.11) shows the results obtained from this test.

Table (4.11) The results of embedding (256*512) color secret image into (512*512) RGB cover image – AES model.

No. of bits used	PSNR	MSE	NCC	AD	Bits per channel used			
					R	G	B	A
4	44.2411	1.25759	0.998171	0.225098	1	1	1	1
5	42.7876	2.26884	0.997292	0.335991	1	1	1	2
6	42.7915	2.33801	0.997065	0.371785	1	1	2	2
7	42.9114	2.32956	0.997532	0.394236	1	2	2	2
8	41.2045	2.31231	0.997347	0.439178	2	2	2	2
9	39.2361	5.84702	0.995926	0.513199	2	2	2	3
10	39.0789	5.73477	0.995878	0.516324	2	2	3	3

Figure (4.11) shows the original image, secret image, and the stego-image:



The cover image.jpg

The secret image
(256x512)

The stego image.png

Figure (4.11) Original image, secret image, stego-image AES-R256-2.

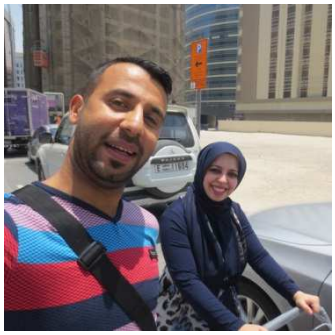
For testing the system in the capacity of 100%, in the test of AES-R512, a color secret image is used of 512*512 size (the same size of the cover image).

Table (4.12) shows the results obtained from this test.

Table (4.12) The results of embedding (512*512) color secret image into (512*512) RGB cover image – AES model.

No. of bits used	PSNR	MSE	NCC	AD	Bits per channel used			
					R	G	B	A
4	38.2262	4.73063	0.992192	0.850731	1	1	1	1
5	36.2246	9.74084	0.98704	1.43308	1	1	1	2
6	36.3384	9.94222	0.98712	1.41737	1	1	2	2
7	36.5561	12.5415	0.987048	1.44532	1	2	2	2
8	35.1617	11.6965	0.986863	1.45187	2	2	2	2
9	32.7261	24.2012	0.98271	2.12202	2	2	2	3
10	32.8414	24.2739	0.98308	2.08162	2	2	3	3

Figure (4.12) shows the original image, secret image, and the stego-image obtained from the embedding process :



The cover image.jpg

The secret image (512x512)

The stego image.png

Figure (4.12) Original image, secret image, stego-image of AES-R512.

4.4 Discussion of the results

According to readings above, the test results can be discussed from the viewpoint of steganography three key requirements (capacity, invisibility, and security).

4.4.1 The capacity of the proposed system

The capacity is the amount of the data in a cover image that can be modified without relapsing the integrity of the cover image. The embedding operation of the steganography system needs to maintain the statistical and perceptual quality of the cover image. Capacity is represented by the maximum amount of bits can be embedded in the cover image without degrade the stego image quality.

The amount of the hidden data relative to the size of the cover image is known as rate of embedding or capacity [54]. The Rational Embedding Capacity (REC) of the proposed system is calculated by the following formula:

$$REC = \frac{S(i,j)\times p}{C(i,j)\times N} \times 100\% \quad \ldots(4.5)$$

Where:

$S(i,j)$: The number of pixels of secret image

$C(i,j)$: The number of pixels of cover image

p : No. of bits in each of secret image pixels

N: No. of cover bits used for embedding

To test the performance of the proposed system, a comparison between the proposed method and ref. [10, 16, 55, and 56] in term of maximum embedded bits and from the results below, it can be seen that the capacity of the proposed system has majorly improved.

For this purpose, a cover image is selected to be the (Lena) image (512×512) see Figure (4.13). Ref. [10] is a steganography based on spatial domain and LSB substitution on PNG cover image using Shamir method for secret sharing to embed in alpha channel, ref. [16] used LSB substitution and RGBA color space for cover image, ref. [55] used the Reflected Binary Gray Code (RBGC), in the wavelet domain, and ref. [56] used BPCS in the frequency domain. Figure (4.14) illustrates the comparison among these techniques.

Figure (4.13) The cover image (Lena).

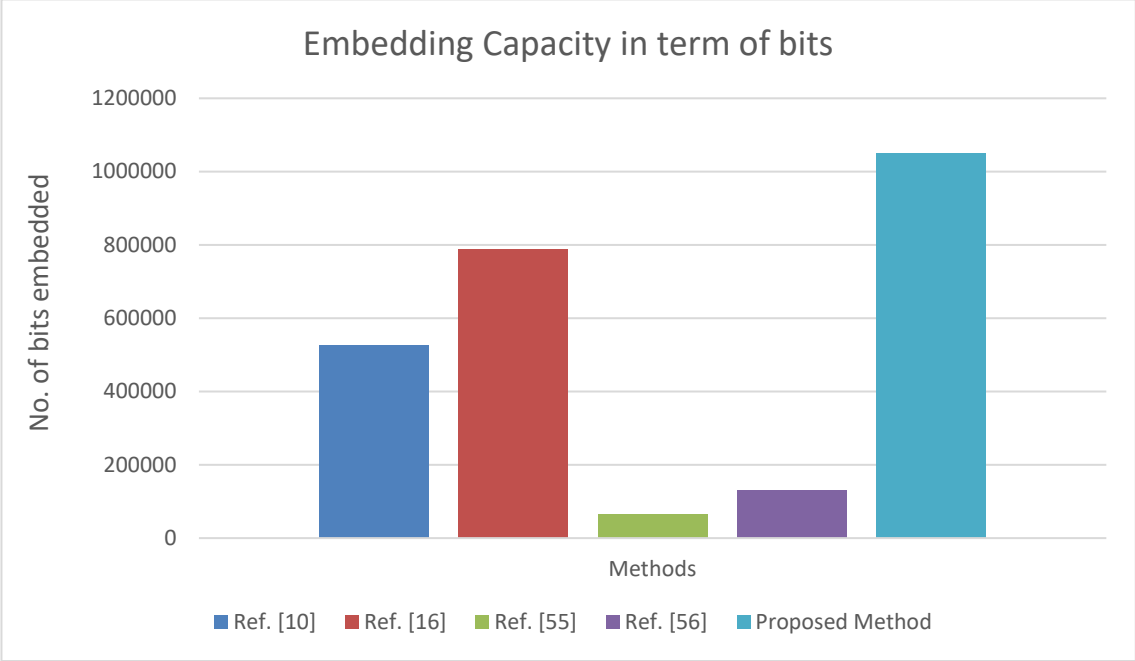


Figure (4.14) Comparison of capacity between the proposed method and other methods**.**

The embedding capacity in term of percentage is compared with the percentage capacity reached by ref. [16, 55, 56, and 57]. Ref. [57] used chaotic map and Contourlet transform for embedding. The comparison is listed in table (4.13):

Table (4.13) comparison of capacity percentage between the proposed method and other methods.

The method	Ref. [16]	Ref. [55]	Ref. [56]	Ref. [57]	The proposed method
Capacity %	100	25	52.89	50	100

The capacity of 100% is easy to define by (the ability of system to embed a secret data of the same size of the cover image). According to this definition, a test designed to represent a real 100% capacity of this system, using the same image for cover and secret image.

While the two images (cover and secret) are identical, the successful embedding and extracting with low distortion means the system has 100% capacity.

Table (4.14) shows the results of objective test and Figure (4.15) shows the results of subjective tests

Table (4.14) The results of embedding (512*512) color secret image the same image.

No. of bits used	PSNR	MSE	NCC	AD
4	38.466	4.96884	0.992155	0.850075

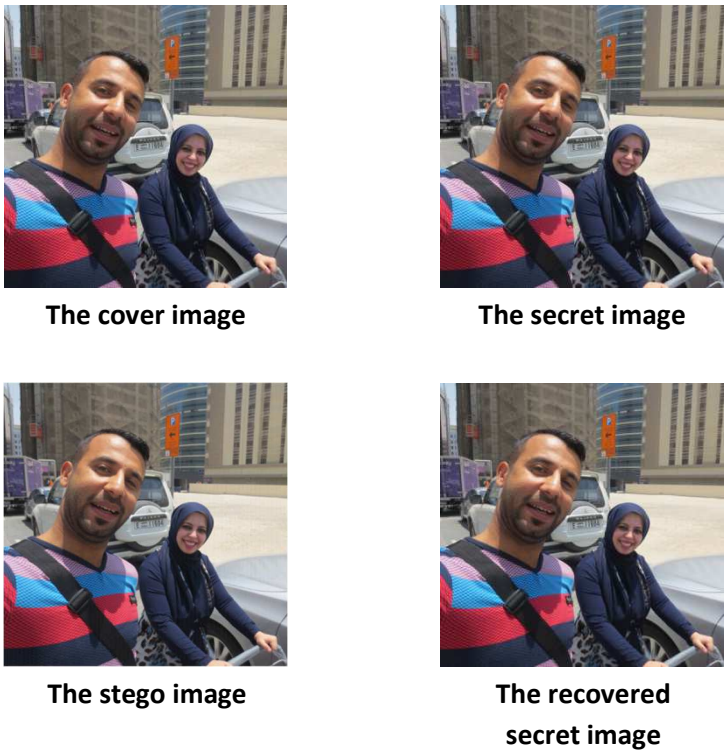


Figure (4.15) The cover image, the secret image, the stego image, and the recovered secret image.

The comparison above is done using only 4 bits from cover image to embed, while using more cover image bits increases the capacity, where :

$$Capacity = C(i,j) \times N \quad \ldots(4.6)$$

Where:

$C(i,j)$: The number of pixels of cover image.

N: the number of bits of the cover image used to embed.

4.4.2 The invisibility of the proposed system

In this method, the message is hidden in the least significant bits of image pixels. Changing the LSB of the pixels does not introduce much difference in the image. The secret data hidden in the LSBs of the cover image channels (Red, Green, Blue, Alpha).

The use of Alpha channel gives the advantages of increasing the cover capacity as well as acts as a transparent mask that can handle a part of secret data with very high efficiency and very low distortion of the stego image. The tests above done using the 4 channels of the cover image with various channels LSB changing to use the maximum capacity that the cover can handle, although the increasing of LSBs used increases the distortion of the stego image as a trade-off.

The subjective test of the stego image is very important test to show the strength of the algorithm, while visual attacks benefit from the ability of human eyes to distinguish between noise and visual patterns. Hence, the presence of the secret information must be invisible to the human eyes first.

The results show that in the capacity of 100%, the distortion of the stego image becomes noticeable to the human eyes by increasing the number of bits of the cover image used to embed. Figure (4.16) illustrate the stego image by using different amount of cover bits to embed.

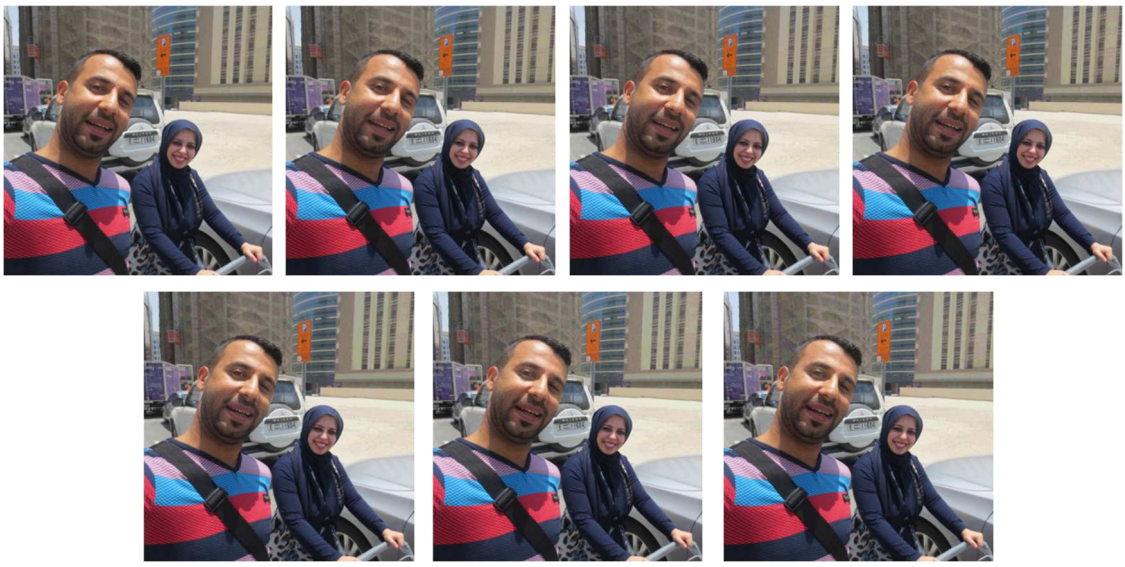


Figure (4.16) The stego images obtained from embedding process with changing the number of bits used to embed.

The techniques that can be used to measure the undetectability or imperceptibility of steganographic systems are varied from one system to another depending on the type of cover media used for data hiding. For instance, image quality represents a sign for the undetectability of steganography system based upon image, while file size may imply to the existence of hidden data in a text based steganography and thus lead to the detection of hidden information.

Two types of perceptibility can be evaluated, these are fidelity and quality. For Steganography system based upon image, the fidelity is defined as " the perceptual similarity between the original cover image and the stego image " [58].

However, attackers and most likely recipients, have no access to the original cover image. In addition, steganography system should avoid stirring the attention of any unauthorized person in the process of secret communication and therefore, the stego image must have of a very high quality. Therefore, the image quality is the focus of interest of most of the steganography techniques so as to avoid raising suspicions and thus avoid hidden data detection.

However, although the PSNR and MSE are by definition fidelity metrics, they are also a widespread considered quality metric. Therefore, a high quality image necessitates a high PSNR value and thus, the cover image and stego image are broadly similar and cannot differentiate between them. Accordingly, “Fidelity” is defined as" the perceptual quality of stego files and therefore PSNR and MSE describe how imperceptible the secret message is "[58]. Hence, the higher the quality of stego images means the higher the imperceptibility of the steganography system. Therefore, testing the quality of stego image is a significant measure to evaluate the efficiency of the steganography technique [59].

To evaluate the performance of the proposed technique in term of invisibility, a comparison between the proposed method (with the two encryption methods) and ref. [11, 16, 57, 60, 61] has been shown in figure (4.17). For this purpose, a cover image is selected to be the (Lena) image (512×512) and hiding capacity is 25%. Ref. [13] is a steganography based on integer wavelet domain, ref. [60] used BPCS to palette-based image, and ref. [61] used the Intermediate Significant Bit Planes.

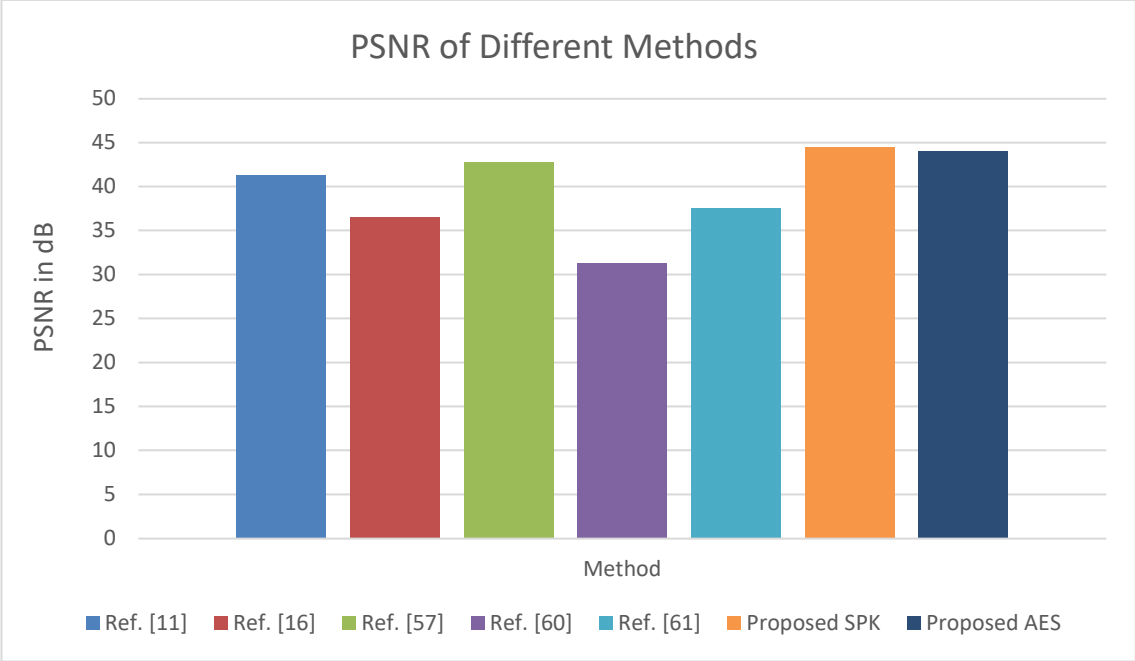


Figure (4.17) The comparison of PSNR between the proposed system and other methods.

The above tests shows there is slight difference between the statistical results of the two algorithms (SPK and AES), this difference occurs because in AES algorithm there is a 128 bit key added to the secret information and embedded with the secret data, also the difference in processing the two algorithms make this minor difference happened.

4.4.3 The security of the proposed system

The embedding algorithm is considered a secure if the embedded data cannot be reveled after detection by an attack based on a full knowledge of the embedding algorithm and the knowledge of at least one stego image.

According to that, the security of the proposed system is the security of the encryption algorithm used.

Two algorithms used for encryption, one is SPK and the other is AES. The results show that there is a little difference between the two algorithms in term of (invisibility), while there is no difference in term of capacity.

Although the security of AES is much higher than SPK, but time consuming of SPK is much lower than AES.

Table (4.15) summarizes the effects of the two algorithms on the proposed system performance.

Table (4.15) The differences between SPK and AES effects on the proposed system.

The Term	SPK	AES
Invisibilty	Higher	Lower
Capacity	The same	The same
Security	Very low	Very high
Time consuming	Lower	Higher

AES rather than SPK provide a high degree of the security to the system where SPK is only 2 bits key length, while AES is 128 bit length.

In fact there are no possible attack on AES better than brute-force attack. Assuming a computer that tries all possible keys at the rate of one billion keys/second. In this case, the attacker will need about 10 000 000 000 000 000 000 000 (10 billion trillions) years to try all possible keys for the version AES-128 [62].

The way that the AES encrypt the secret image also gives a degree of invisibility and robustness beside the high degree of security, where the extracted encrypted image (in the case of detection and successful extraction) is looking like a random image (or insignificant data) while there is no sign that the extracted image is actually a secret encrypted image. The attacker cannot distinguish between the encrypted image and any randomized pixel's values image. Figure (4.18) shows an encrypted secret image which is it not a thing only a random pixels and cannot say it is a secret image or just a random piece of the attacked image (stego image).

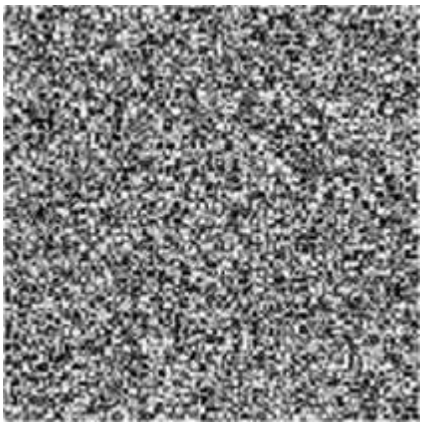


Figure (4.18) Encrypted secret image.

Chapter Five

Conclusions and Suggestions for Future Work

5.1 Conclusions

The following remarks can be concluded from this work:-

1. From the results obtained from the cover image and different type of the secret message, the stego image is obtained with very close properties to the original cover image and the correlation is very close to one so it is so difficult to distinguish between them. These are considered as good results which show the robustness of the proposed algorithm to be achieved for data hiding on image.
2. Alpha channel position in the pixel with the pattern R-G-B-A makes it the Least Significant Byte, this situation makes it very suitable for embedding, that is, Alpha channel is a transparent channel and acts as a hidden layer of the pixel, so it can handle 3 bits without any distortion in the image.
3. The use of the Alpha channel alone gives very good results in undetectability but not in capacity, where the capacity of the cover image will be no more than 25%. Using the other channels Red, green, and Blue in addition to the Alpha channel increases the capacity of the image while maintaining a good invisibility.
4. The use of Bit-Plane slicing shrinking the size of the data to embed, that is, decreases the size of the secrete image and this increases the capacity of the system and the ability to embed secrete data equal or bigger than the carrier itself.

5. The use of an encryption algorithm and the capability to control the number of cover image bits used in embedded increases the security of the system and adds the security factor as a third advantage of the system beside the high capacity and the undetectability.
6. Using two encryption methods in this system, the simplest method and the most complex method, is to show the flexibility of the proposed system and adaptation of the algorithm with the very wide range of encryption algorithms starting from the simplest one to the most complex one. The proposed system can act as a block cipher or a stream cipher depending on the encryption method without affecting the capacity or the invisibility of the system.
7. The statistical test results show that there are little differences in the term of invisibility between SPK and AES, while the subjective test shows that there is no difference between the stego image obtained from SPK model and the one obtained from AES model according to human eye observation. The capacity is the same between SPK model and AES model.
8. AES is superior to SPK in the security level and the size of the key, but it is more complex and consuming more time than SPK. However, the security level of an encryption system is more desirable than its simplicity or its time saving. Although there are some cases the use of SPK become more efficient as in the case of embedding binary images or text.

5.2 Suggestions for Future Work

Suggestions for future research can be summarized by some points:

1. The implementation of this method is done in spatial domain. Embedding technique can be done in frequency domain using one of the transformation techniques.

2. The Alpha channel added to the cover image is a non-pre-multiplied Alpha, in future; a pre-multiplied Alpha can be used to increase robustness with a suitable algorithm for embedding and extraction.
3. The algorithm can be developed to embed (image, text, voice) in the same time using each channel to each type of secrete information and the fourth channel carrying the embedding order or information about the used pixels of the cover image.

References

[1] S. A. Laskar , and K. Hemachandran, " **Secure Data Transmission Using Steganography And Encryption Technique** ", International Journal on Cryptography and Information Security (IJCIS), Vol.2, No.3, pp.161-172, September, 2012.

[2] E. Kawaguchi, and R. O. Eason, " **Principles and Applications of BPCS Steganography** ", Proceedings of SPIE International Society for Optical Engineering, Vol.3528, pp.464-473. January 1999.

[3] J. Krinn, " **Introduction to steganography** ", Global Information Assurance Certification, SANS Institute, 2000.

[4] T. Morkel, " **Image Steganography Applications for Secure Communication** ", M.Sc. Thesis, Faculty of Engineering, Built Environment and Information Technology, University of Pretoria, Pretoria, May 2012.

[5] P. Goel, " **Data Hiding in Digital Images : A Steganographic Paradigm** ", M.Sc. Thesis, Department of Computer Science & Engineering Indian Institute of Technology–Kharagpur, May 2008.

[6] N. F. Johnson, and S. Jajodia, " **Exploring Steganography: Seeing the Unseen** ", IEEE Computer, Vol.31, Issue 2, pp.26-34, February 1998.

[7] A Tiwari, S. R. Yadav, and N. K. Mittal, " **A Review on Different Image Steganography Techniques** ", International Journal of Engineering and Innovative Technology (IJEIT) Vol.3, Issue 7, pp.121-124, January 2014.

[8] A. Rocha, and S. Goldenstein, " **Steganography and Steganalysis in Digital Multimedia : Hype or Hallelujah ?** ", Institute of Computing, University of Campinas (Unicamp), RITA, Vol.15, Issue 1, pp.83-110, 2008.

[9] R. English, " **Comparison of High Capacity Steganography Techniques** ", IEEE, International Conference of Soft Computing and Pattern Recognition, pp.448-453, December 2010.

[10] C. W. Lee, and W. H. Tsai, " **A New Steganographic Method Based on Information Sharing via PNG Images** ", IEEE, pp.807-811, 2010.

[11] S. A. Parah, and J. A. Sheikh, and G. M. Bhat, " **Data Hiding in Intermediate Significant Bit Planes, A High Capacity Blind Steganographic Technique** ", IEEE International Conference on Emerging Trends in Science, Engineering and Technology, pp.192-197, 2012.

[12] P. R. Rudramath, and M. R. Madki, " **High Capacity Data Embedding Technique Using Improved BPCS Steganography** ", International Journal of Scientific and Research Publications, India, Vol.2, Issue 7, pp.1-4, July 2012.

[13] C. F. Lee, and Y. L. Huang," **An efficient image interpolation increasing payload in reversible data hiding** ", Expert Systems with Applications, No.39, Elsevier, pp.6712-6719, 2012.

[14] A. Nichal, and S. Deshpande, " **A High Capacity Data Hiding Method for JPEG2000 Compression System** ", International Journal of Engineering Research and Applications (IJERA), Vol.2, Issue 4, pp.751-755, June-July 2012.

[15] N. Batra, and P. Kaushik, " **Implementation of Modified 16×16 Quantization Table Steganography on Colour Images** ", International Journal of Advanced Research in Computer Science and Software Engineering, Vol.2, Issue 10, pp.244-250, October 2012.

[16] S. Sharma, and U. Kumari, " **A High Capacity Data-Hiding Technique Using Steganography** ", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Vol.2, Issue 3, pp.288-292, May – June 2013.

[17] H. F. Ahmed, and U. Rizwan , " **Embedding Multiple Images in an Image Using Bit Plane Slicing** ", International Journal of Advanced Research in Computer Science and Software Engineering, Vol.3, Issue 1, pp.327-335, January 2013.

[18] W. W. Zin, " **Message Embedding In PNG File Using LSB Steganographic Technique** ", International Journal of Science and Research (IJSR), Vol.2, Issue 1, pp.227-230, January 2013.

[19] S. I. Sowan, " **Steganography For Embedding Data In Digital Image** ", M.Sc. Thesis, University Putra Malaysia, March 2003.

[20] H. Wang, and S. Wang, " **Cyber Warfare: Steganography Vs. Steganalysis** ", Communications of the ACM, Vol.47, No.10, pp.76-82, 2004.

[21] M. S. Hassanein, " **Secure Digital Documents Using Steganography and QR Code** ", Ph.D. Thesis, Department of Computer Science, Brunel University, November 2014.

[22] S. Malik, and W. Mitra, " **Hiding Information- A Survey** ", Journal of Information Sciences and Computing Technologies (JISCT), Vol.3, No.3, pp.232-240, May 2015.

[23] A. Al-Mohammad, " **Steganography-Based Secret and Reliable Communications : Improving Steganographic Capacity and Imperceptibility** ", Ph.D. Thesis, Brunel University, August, 2010.

[24] C. L. Liu, and S. R. Liao " **High-Performance JPEG Steganography Using Complementary Embedding Strategy** ", Pattern Recognition Journal, Vol.41, Elsevier, pp.2945-2955, 2008.

[25] N. Provos, and P. Honeyman, " **Steganography Hide and Seek : An Introduction to Steganography** ", IEEE Security & Privacy, pp.32-44, May-June 2003.

[26] Z. N. Abdulhameed, " **High Capacity Steganography Based on Chaos and Contourlet Transform for Hiding Multimedia Data** ", M.Sc. Thesis, Electrical Engineering Department, College of Engineering, AL-Mustansiriya University, January 2014.

[27] K. Rabah, " **Steganography The Art of Hiding Data** ", Information Technology Journal, Vol.3, Issue 3, pp.245-269, 2004.

[28] L. Y. Por, T. F. Ang, and B. Delina, " **WhiteSteg: A New Scheme in Information Hiding Using Text Steganography** ", WSEAS Transactions on Computers, Vol.7, Issue 6, pp.735-745, June 2008.

[29] S. K. Bandyopadhyay, et al., " A **Tutorial Review on Steganography** ", Proceedings of the International Conference on Contemporary Computing, pp.105-114, 2008.

[30] D. Artz, " **Digital Steganography: Hiding Data within Data** ", IEEE Internet Computing Journal, Vol.5, No.3, pp.75-80, May-June 2001.

[31] T. G. Handel, and M. T. Sandford, " **Hiding Data in the OSI Network Model** ", Lecture Notes in Computer Science, Vol.1174, pp.23-38, 1996.

[32] P. C. Mandal, " **Modern Steganographic Technique: A Survey** ", International Journal of Computer Science & Engineering Technology (IJCSET), Vol.3, No.9, pp.444-448, September 2012.

[33] S. Ertürk, " **Digital Image Processing** ", February 2003 Edition, National Instruments Corporation, February 2003.

[34] J. D. Foley, et al., " **Introduction to Computer Graphics** ", Addison-Wesley Publishers, 1994.

[35] S. Feruza, and T. H. Kim, " **Review on YCrCb Color Space Optimization, TV Images Compression, Algorithm of Signals**

Sources Isolation and Optical Fiber ", International Journal of Smart Home, Vol.3, No.4, pp.43-62, October 2009.

[36] J. Fridrich, " **Steganography in Digital Media: Principles, Algorithms, and Applications** ", Cambridge University Press, 2010.

[37] T. Porter, and T. Duff, " **Compositing Digital Images** ", Computer Graphics, Vol.18, No.3, pp.253-259, July 1984.

[38] J. F. Blinn, " **Jim Blinn's Corner: Compositing Part 2: Practice** ", IEEE Computer Graphics & Applications, pp.82-87, November 1994.

[39] A. R. Smith, " **Image Compositing Fundamentals** ", Microsoft Technical Memo 4, v4.15, August 1995.

[40] M. Adler, et al., " **Portable Network Graphics (PNG): Functional specification** ", 1st Edition, W3C Recommendation, October 1996. Read online on : http://www.w3.org/TR/REC-png-961001

[41] R. C. Gonzalez, and R. E. Woods, " **Digital Image Processing** ", 2nd Edition, Prentice Hall Inc., 2008.

[42] D. Selent, " **Advanced Encryption Standard** ", Rivier Academic Journal, Vol.6, No.2, pp.1-14, Fall 2010.

[43] C. Kaufman, R. Perlman, and M. Speciner, " **Network Security: Private Communication in a Public World** ", 2nd Edition, Prentice Hall PTR, 2002.

[44] M. Pitchaiah, P. Daniel, and Praveen, " **Implementation of Advanced Encryption Standard Algorithm** ", International Journal of Scientific & Engineering Research Vol.3, Issue 3, March 2012.

[45] C. Paar, and J. Pelzl, " **Understanding Cryptography : A textbook For Students and Practitioners** ", Springer, 2010.

[46] J. Daemen, and V. Rijmen, " **AES Proposal: Rijndael** ", AES submission document on Rijndael, September 1999.

[47] W. Stallings, " **Cryptography and Network Security Principles and Practice** ", 5th Edition, Pearson Education Inc., 2006.

[48] K. Kamalam, and S. Saranya, " **An Effective Method in Steganography to Improve Protection Using Advanced Encryption Standard Algorithm** ", International Journal of Engineering Trends and Technology (IJETT), Vol.18, No.4, pp.175-180, December 2014.

[49] **Advanced Encryption Standard (AES)**, Federal Information Processing Standards Publication 197, November, 2001.

[50] M. Mishra, A. R. Routray, and S. Kumar," **High Security Image Steganography with Modified Arnold's Cat Map** ", International Journal of Computer Applications, Vol.37, No.9, pp.16-20, January 2012.

[51] A. J. Sadiq, " **Comparison Steganography in Spatial Domain of Image** ", Journal of Baghdad College of Economic Sciences, No.29, pp.379-391, 2012.

[52] T. Petricek, and T. Svoboda, " **Matching by Normalized Cross-Correlation Reimplementation, Comparison to Invariant Features** ", Research Reports of CMP, Czech Technical University in Prague, No. 9, September 2010

[53] D. Nagamalai, E. Renault, and M. Dhanuskodi, " **Advances in Digital Image Processing and Information Technology** ", First International Conference on Digital Image Processing and Pattern Recognition, Springer, 2011.

[54] S. Venkatraman, A. Abraham, and M. Paprzycki, " **Significance of Steganography on Data Security** ", International Conference on Information Technology : Coding and Computing, ITCC 2004.

[55] A. Cheddad, " **Steganoflage : A New Image Steganography Algorithm** ", PhD Thesis, School of Computing & Intelligent Systems Faculty of Computing & Engineering, University of Ulster, September 2009.

[56] S. Khaire, and S. L. Nalbalwar, " **Review: Steganography – Bit Plane Complexity Segmentation (BPCS) Technique** ", International Journal of Engineering Science and Technology, Vol.2, Issue 9, pp.4860-4868, 2010.

[57] Z. N. Abdulhameed, and M. K. Mahmood, " **High Capacity Steganography based on Chaos and Contourlet Transform for Hiding Multimedia Data** ", International Journal of Electronics and Communication Engineering & Technology (IJECET), Vol.5, Issue 1, pp.26-42, January 2014.

[58] I. J. Cox, et.al, " **Digital Watermarking and Steganography** ", 2nd Edition, Morgan Kaufmann Publishers, Elsevier, 2008.

[59] N. Wu, and M. S. Hwang, " **Data Hiding: Current Status and Key Issues** ", International Journal of Network Security, Vol.4, No.1, pp.1-9, January 2007.

[60] M. Niimi, et al., " **High Capacity and Secure Digital Steganography to Palette-Based Images** ", IEEE ICIP, pp.917-920, 2002.

[61] K. B. Raja, et.al, " **Robust Image Adaptive Steganography using Integer Wavelets** ", IEEE Image Processing, 2011.

[62] A. W. Naji, et al., " **Novel Framework for Hidden Data in the Image Page within Executable File Using Computation Between Advance Encryption Standard and Distortion Techniques** ", International Journal of Computer Science and Information Security (IJCSIS), Vol.3, No.1, August 2009.

List of Publications

1. N. S. Alseelawi, T. Z. Ismaiel, and F. A. Sabir, " **High Capacity Steganography Method Based upon RGBA Image** ", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol.4, Issue 6, pp.119-125, June 2015.
2. N. S. Al-Seelawi, T. Z. Ismaeel, Y. E. Majeed, " **New High Capacity Secure Steganography Technique** ", International Journal of Research in Computer and Communication Technology (IJRCCT), Vol.5, Issue 1, January 2016.

الخلاصة

في السنوات الماضية كان موضوع اخفاء المعلومات فعال حيث ظهرت العديد من الخوارزميات للعمل على تطوير تقنيات فعالة في مجال إخفاء المعلومات. و إخفاء المعلومات هو العلم الذي يتعامل مع إخفاء البيانات السرية في بعض الوسائط والناقل قد يكون صورة أو صوت أو نص أو فيديو. وسوف يكون التعامل في هذا العمل مع إخفاء المعلومات داخل الصورة.

هذا العمل يقدم تقنيات اخفاء الصور مبنية على آلية استبدال البت الأقل اهمية (LSB), الرسالة المراد اخفاؤها هي عبارة عن صورة رقمية (صورة ذات تدرجات رمادية او صورة ملونة). في هذا العمل, قناة ألفا اضيفت الى الصورة الحاملة (الصورة الغطاء) ذات النظام اللوني (RGB) لزيادة عمق البت في الصورة الغطاء والتي ستصبح ذات نظام لوني RGBA. تم تطبيق تقنية تقطيع شرائح البتات (BPS) على الصورة السرية لضغطها ولتقليل حجم البيانات السرية المراد اخفاؤها.

لزيادة الأمان, تم تطبيق نموذجين للتشفير على النظام كلاً على حدة, نموذج المفتاح الخا البسيط (SPK) ونموذج معيار التشفير المتقدم (AES) حيث SPK يمثل طريقة تشفير استخدام عملية XOR سيطة, ينما AES يمثل معيار التشفير المستخدم اليوم.

النتائج ودراسات المقارنة اظهرت فعالية التقنية المقترحة في توليد الصور المضمنة (stego). الرسالة السرية يمكن لها ان تصل الى نفس حجم الصورة الغطاء وتكون الصورة المضمنة قريبة للصورة الغطاء وذلك لأن الترا ط (Correlation) قريب جدا للواحد والـ PSNR لغاية 38.1995 dB للصورة السرية ذات التدرج الرمادي و 38.1553 للصورة السرية الملونة في نموذج الـ SPK, ينما تصل الـ PSNR لغاية 38.4088 للصور الرمادية و 38.2262 للصور الملونة في نموذج الـ AES.

 الإضافة لذلك, تم عمل مقارنة ين العديد من طرق الاخفاء المبنية على التحويل الموجي والمبنية على التحويل المكاني, حيث ان خوارزمية الاخفاء المقترحة توفر سعة تضمين وجودة الصورة المضمنة مع قيمة PSNR اعلى من الطرق التي تمت المقارنة معها. لقد تم انجاز هذا العمل استخدام لغة الماتلاب لبناء جميع البرامج في هذا البحث.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بغداد
كلية الهندسة

نظام اخفاء للصور رباعية القنوات مبني على تقنية معيار التشفير المتقدم

رسالة
مقدمة إلى كلية الهندسة في جامعة بغداد
كجزء من متطلبات نيل درجة ماجستير علوم
في هندسة الإلكترونيك والإتصالات / هندسة الحاسبات

من قبل

نوار سعد ارحيم

بإشراف

د . فراس علي صابر

رمضان 1436

تموز 2015