Design and Implementation of Dual Image based Reversible Data Hiding Techniques

Thesis submitted to the

VIDYASAGAR UNIVERSITY

for award of the degree of

DOCTOR OF PHILOSOPHY

(Science)

In

COMPUTER SCIENCE

BY

BISWAPATI JANA

Department of Computer Science

Vidyasagar University

Midnapore-721102, INDIA

June, 2016

CERTIFICATE

This is to certify that the thesis entitled **"Design and Implementation of Dual Image based Reversible Data Hiding Techniques"** being submitted by **Sri Biswapati Jana** for the award of degree of **DOCTOR OF PHILOSOPHY** to the VIDYASAGAR UNIVERSITY is a record of bonafide research work carried out by him under our guidance and supervision. **Sri Biswapati Jana** has worked in the department of **Computer Science, Vidyasagar University** to the regulations of this University.

In our opinion, this thesis is of the standard required for the award of degree of **DOCTOR OF PHILOSOPHY**.

The results, embodied in this thesis, have not been submitted to any University or Institution for the award of any degree or diploma.

Dr. Shyamal Kumar Mondal
Associate Professor
Department of Applied Mathematics with Oceanology and Computer Programming
Vidyasagar University
Midnapore 721 102, INDIA

Place : Midnapore
Date:

Prof. Debasis Giri
Professor
Department of Computer Science and Engineering
Haldia Institute of Technology
Haldia 721 657, INDIA

Place : Haldia
Date:

Dedicated to my family

ACKNOWLEDGEMENT

I take this opportunity of expressing my deep sense of gratitude, regards and appreciation to my supervisors Professor Debasis Giri, Department of Computer Science and Engineering, Haldia Institute of Technology, Haldia, Purba Medinipur, and Dr. Shyamal Kumar Mondal, Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Paschim Medinipur for proposing topics of the present Ph.D. thesis and for introducing me to the magnificent field of research in Computer Science. I am deeply indebted to them for their constant encouragement, constructive criticism, wise and valuable suggestions received all through the period of this investigation.

I am grateful to Dr. Shyamal Kumar Mondal, Convenor, Ph.D. Committee, Department of Computer Science, Vidyasagar University for his support and encouragement.

I convey my heartfelt thanks to Professor J. K. Mondal, Department of Computer Science and Engineering, Kalyani University, Nadia due to his constant support and encouragement throughout the period of investigation. Also, I convey my heartfelt thanks to Professor M. M. Pal, Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Professor P. C. Jana and Professor S. Saha, Department of Physics and Technophysis, Vidyasagar University, Dr. B. Das, Assistant Professor, Department of Mathematics, S. K. B. University, Purulia and faculty members of the Department of Computer Science, Vidyasagar University for their encourage and help in different ways.

I would like to express my gratitude to co-researchers Mr. S. Singha, Mr. P. Choudhury, Mr. P. Pal, Mr. M. Hossion, for their constant help, encouragement and helpful suggestions. I acknowledge my debt to my friend, Dr. A. K. Mishra, Department of Sanskrit, Vidyasagar University for always inspiring and encouraging me and also for his continuous support.

I take this privilege to convey my regards to Professor Ranjan Chakrabarti, Honorable Vice-Chancellor and Dr. Jayanta Kishore Nandi, Registrar, for providing me all facilities needed to carry out my research work.

I must acknowledge my debt for those who have constantly inspired, encouraged and helped me. It is almost impossible for me to express any formal gratitude to my mother, Late. Sarajini Jana without which the execution of this work would not have been possible.

At length, I ardently acknowledge my debt to my wife, Mrs. Sharmistha Jana, who is the source of inspiration for both my research and my life and without whose unstinting efforts this thesis could not have been given a due shape. I also acknowledge my debt to family members for his patience and continuous support. Throughout the research work, I also supported by the sweetness of my beloved daughters Miss. Basundhara Jana and Miss. Mrittika Jana.

Vidyasagar University
Paschim Medinipur
Date: ***(Biswapati Jana)***

LIST OF PUBLICATIONS

PUBLISHED

1. High payload reversible data hiding scheme using weighted matrix, (2016, March), **Optik International Journal for Light and Electron Optics**, 127(6), 3347-3358, Elsevier. **Impact Factor: 0.677**.
   Indexing: Compendex, Engineering Index, INSPEC, Science Citation Index (SCI), Scisearch, Technology and Applied Sciences, Scopus, Engineering Information Compendex, Google Scholar,

2. Dual-Image Based Reversible Data Hiding Scheme Using Pixel Value Difference Expansion,(2016, July), **International Journal of Network Security**, 18(4), 633-643. **Impact Factor: 1.3921**.
   Indexing: EI-Compendex, Summon by Serial Solutions, SCImago, Scopus and SciVerse, Data Base systems and Logic Programming (DBLP), EBSCO, Directory of Open Access Journals (DOAJ) and Google Scholar.

3. An Efficient Data Hiding Scheme Using Hamming Error Correcting Code, (2015, September), Published in the proceedings of the Sixth International Conference on Computer and Communication Technology, (ICCCT-2015), ACM digital library, ACM New York, NY, USA 2015, pp. 360-365.
   Indexing: ACM Digital Library, Google Scholar.

4. Dual Image Based Reversible Data Hiding Scheme Using Three Pixel Value Difference (TPVD), (2016, February), Published in the proceedings of the Third International Conference on Information System Design and Intelligent Application (INDIA 2016), Advances in Intelligent Systems and Computing, Springer, Vol. 434, pp. 403-412.
   Indexing: ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springer link.

5. Weighted Matrix Based Reversible Data Hiding Scheme Using Image Interpolation, (2015, December), Published in the proceedings of the International Conference on Computational Intelligence in Data Mining (ICCIDM-2015), Advances in Intelligent Systems and Computing, Springer India, Vol. 411, pp. 239-248.
   Indexing: ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springer link.

6. An Efficient Weight Matrix Based Reversible Data Hiding Scheme, (2015, December), Published in the proceedings of the International Conference on Computers and Management (ICCM-2015), Jaipur, Rajasthan, December 16-17, 2015.

7. Reversible Data Hiding Through Hamming Code Using Dual Image,(2015, October), Published in the proceedings of the International Congress on Information and Communication Technology (ICICT - 2015), Advances in Intelligent Systems and Computing, Vol. 439, pp. 495-504.
   Indexing: ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springer link.

8. Dual Image based Reversible Data Hiding Scheme Using Pixel Value Difference With Exploiting Modification Direction, (2016, February), Published in the proceedings of the First International Conference on Intelligent Computing and Communication ($ICIC^2$), Advances in Intelligent Systems and Computing (AISC), Springer.
   Indexing: ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springer link.

COMMUNICATED

9. Partial Reversible Data Hiding Scheme Using (7,4) Hamming Code, **Multimedia Tools and Application**, Springer, **Impact Factor 1.346**, (Revised version submitted).

10. Dual Image Based Reversible Data Hiding Scheme Using (7,4) Hamming Code, **Multimedia Tools and Application**, Springer, **Impact Factor 1.346**, (Revised version submitted).

ABSTRACT

Communication through data hiding is an important and demanding issues for many applications. The important parameters to measure the performance of data hiding schemes are imperceptibility, data hiding capacity and robustness which are inversely proportional to each others. So, there is a challenge to design some innovative data hiding techniques and solved while maintaining the tradeoff among these three parameters. After extraction the secret message from the stego media, the recovery of original image is also demanding issues in several human centric application areas. Many data hiding techniques are developed since last decades which have either limited embedding capacity and/or lower visual quality.

In the light of this discussion, some innovative secured data hiding schemes have been proposed to maintain a perfect balance of these important components of data hiding schemes that is payload, imperceptibility and robustness. Some new data hiding methods are designed and solved using Hamming code, Pixel Value Difference (PVD), Difference Expansion (DE), Exploiting Modification Direction (EMD) and Weighted Matrix based techniques.

In this thesis, two partial reversible data hiding schemes have been designed through error creation using Hamming code. Reversibility has been achieved using dual image but data hiding capacity is limited. To improve the embedding capacity, three new reversible data hiding techniques are designed and solved using PVD, DE and EMD methods. The maximum embedding capacity of these suggested methods is 2.15 bits per pixel (bpp) with moderate visual quality.

Again to improve the data hiding capacity while maintaining good visual quality, three more new data hiding techniques are introduced and solved using weighted ma-

trix. In these schemes, data hiding capacity has been achieved finally at 3.46 bpp with visual quality measured by Peak Signal to Noise Ratio (PSNR) is 35.39 dB. Dual image and image interpolation techniques accelerate the data hiding capacity, visual quality and security of proposed data hiding schemes. To enhance the security of these schemes, shared secret key has been introduced. All these schemes are compared with the state-of-the-art methods and observed a considerable improvement in terms of visual quality as well as capacity. Development of some new innovative data hiding methods are not enough, but their security analysis is paramount important. So, these suggested methods are analyzed through some standard steganalysis and tested under some known steganographic attacks. We observed that all these proposed schemes are robust against several steganographic attacks.

Contents

List of Abbreviations

Acronyms	*Descriptions*
BFA	Brute Force Attack
BPP	Bits Per Pixel
CC	Correlation Coefficient
CRS	Capacity Reversible Steganography
DDHHC	Dispersed Data Hiding using Hamming Code
DE	Difference Expansion
DH	Data Hiding
DHHC	Data Hiding using Hamming Code
DI	Dual Image
DM	Difference Matrix
DRDHHC	Dual Image based Reversible Data Hiding using Hamming Code
DRDHWM	Dual Image based Reversible Data Hiding using Weighted Matrix
EDRDHHC	Enhanced Dual Image based Reversible Data Hiding using Hamming Code
EMD	Exploiting Modification Direction
EPRDHHC	Enhanced Partial Reversible Data Hiding using Hamming Code
HA	Histogram Attack
HECC	Hamming Error Correcting Code
HC	Hamming Code
IDRDHWM	Interpolated Dual Image based Reversible Data Hiding using Weighted Matrix
INP	Interpolating with Neighbor Pixel
LM	Location Map
IRDHWM	Interpolated Image based Reversible Data Hiding using Weighted Matrix
LSB	Least Significant Bit

Acronyms	*Descriptions*
LSB-M	Least Significant Bit - Matching
LSB-M-R	Least Significant Bit - Matching - Revisited
LSB-R	Least Significant Bit - Replacement
MF-PVD	Modulus Function - Pixel Value Difference
MM	Magic Matrix
MPSNR	Modified Peak Signal to Noise Ratio
MSE	Mean Square Error
NI	New Image Matrix
NMI	Neighbor Mean Interpolation
NNI	Nearest Neighbor Interpolation
NPEU	Non Pivot Embedding Unit
PEU	Pivot Embedding Unit
PPM	Pixel Pair Matching
PVD	Pixel Value Difference
PVDEMD	Pixel Value Difference with Exploiting Modification Direction
PVDDE	Pixel Value Difference with Difference Expansion
PVD-PRT	Pixel Value Difference - Patched Reference Table
PVDE	Pixel Value Different Expansion
PRDHHC	Partial Reversible Data Hiding using Hamming Code
PSNR	Peak Signal to Noise Ration
RDE	Reduced Difference Expansion
RDH	Reversible Data Hiding
RDHHC	Reversible Data Hiding using Hamming Code
RE	Relative Entropy
ROI	Region of Interest
RS	Regular Singular
R	Reference Table
TPVDDE	Three Pixel Value Difference with Difference Expansion
SD	Standard Deviation
SI	Secret Image
WM	Weighted Matrix

List of Common Symbols

The following symbols are used in the thesis:

Cov	Co-variance
δ	Shared Secret Key
gcd	Greatest Common Divisor
ξ	Shared Secret Key
κ	Shared Secret Position
lb	Lower Boundary of the Sub - Range of Range Table (R)
mod	Modulus Operation
ω	Data Embedding Position
$\otimes$	Entry - wise Multiplication Operation
$\oplus$	Bitwise Exclusive OR
ub	Upper Boundary of the Sub - Range of Range Table (R)
ρ	Correlation Coefficient
σ	Standard Deviation
wb	Width of the Sub - Range of Range Table (R)

List of Notations

The following notations are used in the thesis:

B	Block Size
C	Cover Image
C_p	Preprocessed Cover Image
D	Secret Data
D_{len}	Length of Secret Data
F_i	Image Matrix
h	Histogram of Cover Image
$h^{'}$	Histogram of Stego Image
I	Original Gray Scale Image
k	Seed Value
K	Key Matrix
N_B	Number of Blocks
p	Payload
P_n	New Pixel
P_o	Original Pixel
pv	Positional Value
R	Range Table
R_M	Number of Regular Group with Mask M
R_{-M}	Number of Regular Group with Mask - M
S	Stego Image
SA	Stego Auxiliary
$SA^{'}$	Stego Auxiliary after Data Embedding for First Stage

SA''	Stego Auxiliary after Data Embedding for Second Stage
SM	Stego Major
SM'	Stego Major after Data Embedding for First Stage
SM''	Stego Major after Data Embedding for Second Stage
S_M	Number of Singular Group with Mask M
S_{-M}	Number of Singular Group with Mask - M
SD	Standard Deviation
t	Number of bits to be embedded within a pixel pair
T	Threshold
val	Value of the sum of entry-wise multiplication
W	Weighted Matrix

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Introduction

Data hiding is the art and science of data smuggling that communicates information by concealing secret message through innocuous cover media such as images, audio signals, videos, documents and so on. Various kinds of multimedia objects can be used as cover media to hide the existence of secret information from an eavesdropper, but digital images are the most commonly used media because they are ubiquitous and moreover, images speak more than words. Due to the higher degree of distortion tolerance with a larger hiding capability, digital images are being used as cover media in data hiding applications for the past few decades.

Now-a-days, data hiding provide secured and private communication that becomes the essential requirement of various types of applications. Data hiding plays an important role in multimedia security. It is useful in various purposes such as copyright protection, covert communication, content authentication, forensic tracking, tamper detection and many other human centered approaches. It consists of several branches such as Steganography, Watermarking, Secret Sharing, Visual Cryptography etc. Steganography and Watermarking are two main research areas in data hiding. These two approaches conceal secret information within cover media by changing some of its attributes, but they have still some properties distinguishable from each other. In Steganography, embed messages are hard to reveal by an adversary, but in Watermarking it may not always be true. The main intention is to concentrate on precluding the adversary from moving out the content of the confidential messages by applying a variety of distortion techniques. Some data hiding schemes proposed in this research work are classified in the category of Steganography through gray scale digital image as per their degree of redundancy. The objective of Steganography are quite different from Cryptography. The cryptographic schemes scramble secret messages so that if intercepted, messages cannot be understood but Steganography camouflages the messages to hide its existence and makes it seem almost invisible. An encrypted message may draw suspicion while an invisible message will not.

Depending on the manner of data embedding, current data hiding algorithms can be grouped into three domains: spatial, frequency and compress domains. Each domain has its own advantage and disadvantage with regard to hiding capacity, execution time and storage space. Whereas, algorithms in spatial domain embed secret messages by directly manipulating the im-

age pixel values. However, algorithms in the frequency domain first transform the input image into frequency coefficients. Then the secret message is embedded by coefficient modifications. Algorithms in the compress domain adopt the image representation by a series of compress code as their embedding media. Data embedding is accomplished by modifying the compress code. Transform domain methods are more robust compared to spatial domain methods.

Sometimes, after extracting secret data from the cover media, recovery of cover image is essential in some applications such as remote sensing, military application, medical image sharing, multimedia archive management etc. According to whether the cover image pixels can be recovered or not after data extraction, current data hiding schemes are classified into two categories: reversible and irreversible. The scheme of reversible data hiding usually exploits the techniques of histogram shifting, prediction error, and difference expansion etc. On the other hand, irreversible data hiding schemes such as data hiding using Pixel Value Difference (PVD), Exploits Modification Direction (EMD), Weighted Matrix (WM) often have greater data hiding capacities, but the modification caused by data embedding are not invertible. In addition to this, dual image based data hiding techniques are often being used recently. During data embedding, dual image based techniques can generate two similar copies stego-image from the cover image to increase data embedding capacity and enhance security. It is hard for an adversary to extract the hidden content without simultaneous two stego-images. This concept is talked about as a particular case of secret sharing.

However, communication through data hiding usually puts stress on simply finding the presence of a secret message. Thus, the imperceptibility becomes the most significant place for the data hiding schemes. For sophisticated data hiding strategies, it has been proven in practice that one efficient style of increasing security is to reduce the number of changes that is inserted into the cover media. A high embedding efficiency becomes the principal aim to accomplish for the current data hiding schemes by substituting the payload. The goal of data hiding is to ensure embedded data extraction and original cover image reconstruction. The performance of a reversible data hiding schemes are evaluated by three aspects: embedding capacity (payload), visual quality (measured by PSNR) and computational complexity. For a desired capacity, one expects to minimize the distortion and meanwhile keep computational complexity as low as

possible. To get high capacity, repeated embedding process may applied, leading to rapid decrease of visual quality and increase of computational complexity. The key factors of secured hidden data communications are high security, high embedding capacity and good imperceptibility. Each of these requirements occupies each corner of a triangle in a data hiding system and there is always a trade-off between these contradictory requirements.

Imperceptibility: The first and foremost requirement of any data hiding algorithm is the imperceptibility. The embedded secret data within cover image should not cause any degradation in visual quality. The secret message should remain invisible, it should not be detectable to the human eyes and there should not be any visual distortion within stego-image so that it remains unsusceptible and unsafe. Higher the stego-image quality, more invisible the hidden message which can be measured through PSNR. A higher PSNR value means a lower degree of distortion.

Payload: The amount of inserted information within stego-image is considered as payload. The payload should be higher as much as possible with an acceptable resultant stego quality. It is measured by some absolute value or relative measurement (bits per pixel) or data embedding rate. The importance of data hiding schemes are based on the tradeoff between payload or data hiding capacity and stego-image quality. So, a scheme does have its contribution to the field of research if it increases the payload while maintaining an acceptable quality of stego-image or improves image quality while keeping the hiding capacity at the same level or better.

Robustness: Robustness is the level of difficulty required by an eavesdropper to decide whether an image contains hidden message(s) or not. An effective data hiding scheme would be the one where an image can sustain under steganographic attack that may prove inconclusive. Statistical analysis is the practice of detecting hidden information through applying statistical tests on stego image.

Stanley [56] suggests that another important property of data hiding is speed or time complexity where information should be embedded as quickly as possible. However, it is not feasible that any data hiding algorithm should sacrifice above mentioned criteria to embed information in a timely manner.

In recent years, the demand of efficient secured high capacity data communication through data hiding is increasing. To accomplish good quality stego with high payload and robustness, is a challenging problem to the researchers. After extracting the confidential message from stego media, the demand of image reversibility without any distortion goes high. In this light, it is necessary to investigate reversible data hiding approaches which enhance security and improve embedding capacity. However, data hiding is a double-edged sword since terrorist and illegal organizations may use it to undermine social stability, endanger public safety and engage in criminal activities. Thus security analysis (steganalysis) plays an important role as counter process of data hiding. Fridrich et al. [15] suggest that the power to discover secret information in stego images is associated with the data length. This means that a short message hiding within a big size carrier will result in a little amount of distortion and hence this is practically hard to distinguish any hidden content within stego media. It is obvious that every data hiding scheme may cause undesirable artifacts in the resulted stego image which is used as a tool to detect and estimate the length of secret message through security analysis. The steganalysis falls into two broad categories: specific or targeted and universal. Specific steganalysis can reveal the secret message, but it is hard to know which data hiding methods were used to generate stego images. While the latter, also called blind steganalysis, is more attractive in practical application, because it can detect the secret message independent of the data hiding algorithms. Blind security analysis is a critical task than targeted analysis because the analyst does not know how secret message can be embedded. In this case, the analyst develop an algorithm for checking marks of tampering found within the suspected media which contains secret messages. Fridrich et al. [16] developed an authentic and exact method called Regular Singular (RS) analysis for detecting the Least Significant Bit (LSB) embedding within the image.

In this thesis, some new innovative reversible data hiding techniques have been designed using dual image and implemented. Design of any security scheme is not enough, but their security guarantee is of paramount impermanence. If the detection of secret information within a media is made by an eavesdropper then the data hiding scheme will fail.

1.2 Literature Review

Brief review of some existing data hiding schemes have been described below:

1.2.1 Brief review of data hiding through Hamming code

Hamming [21] devised a sophisticated pattern of parity checking code that could correct single error along with the detection of double errors. Crandall [12] first pointed out that embedding efficiency could be improved by coding methods and suggested the matrix coding. Westfeld [67] introduced data hiding techniques through matrix encoding using Hamming code. Tseng et al. [64] proposed data hiding scheme by taking into consideration the quality of image after data hiding. It ensured that any bit that is modified in the host image is adjacent to another bit which has a value equal to the former's new value. Willems and Dijk [68] suggested that the embedding code based on the ternary Hamming code and ternary Golay code is optimum in a sense that they achieve the smallest possible distortion. Then Fridrich and Soukal [19] presented a data hiding scheme using matrix embedding that is efficient for embedding messages. This scheme is based on random linear code of small dimension which provides good embedding efficiency, where the relative payload is above 0.9 bits per pixel (bpp). A data hiding scheme suggested by Zhang et al. [73], which improves the embedding efficiency of binary covering function that employed the capacity more efficiently by extending the block of binary cover code. This method performs equally with ternary code without binary-ternary conversion of the message. Again Fridrich et al. [20] observed that the quality which determines the embedding efficiency is not the covering radius but the average distance to code. For the linear code, the highest embedding efficiency is not necessarily achieved using code with the smallest covering radius. Chang et al. [9] proposed a data hiding method using (7, 4) Hamming code. This scheme embeds a section of seven bits within a set of seven original image pixels at a time. They achieve embedding payload 0.99 (bpp) where average PSNR equals to 50 (dB). Kim et al. [28] developed Data Hiding using Hamming Code (DHHC) to hide secret messages within halftone image. Here, they used codeword to generate a syndrome value. Then using Exclusive-OR operation they embed four bits secret data within the codeword of four bits. Ma et al. [49] suggested an improvement of Kim et al.'s scheme by altering pixel pair which reduces data embedding capacity by half. Recently, a Dispersed Data Hiding scheme through Hamming Code (DDHHC) has been designed by Lien et al. [41] using space filling curve decomposition.

In this scheme, average PSNR is 44.31 (dB), when $4,096$ bits are embedded. Using Hamming code, Kim et al. gained good quality image and their modified PSNR (MPSNR) and payload are 48.20 (dB) and 0.86 (bpp) respectively. Lien et al. achieved 29.66 (dB) for embedding 65,536 bits. Recently, Cao et al. [4] developed high payload Hamming code based data hiding schemes with embedding rate up to 3 (bpp) with PSNR 51 (dB). High payload steganographic scheme also has been developed recently by Bai and Chang [2] for compressed images. Their payload is 2 (bpp) but PSNR is below 30 (dB).

In data hiding schemes, achievement of reversibility and enhancement of security while maintaining good visual quality through Hamming code is still an important issue. So far in the literature, it is found that no such scheme exists, where reversibility has been achieved through Hamming code. The use of shared secret key in data hiding through Hamming code is also rarely available. In the present research, dual image based reversible data hiding schemes using (7, 4) Hamming code has been proposed.

1.2.2 Brief review of data hiding through Pixel Value Difference (PVD), Difference Expansion (DE) and Exploiting Modification Direction (EMD)

A simple data hiding scheme is the Least Significant Bit - Replacement (LSB-R) has been introduced by Turner [60]. The LSB-R scheme is unbalanced because even valued pixel will never be decremented and odd valued pixel will never be incremented. This asymmetry is easily detected by some detectors [16]. To overcome this problem, Sharp [53] proposed LSB matching (LSB-M) scheme which does not replace LSB but randomly either increments or decrements *one* in LSB of cover image when no match is found with secret data bit. Embedded message within the scheme is also detected by the detector suggested by Ker [27]. To enhance the LSB-M scheme, Mielikainen [50] proposed the LSB matching revisited (LSB-M-R) where payload was same as LSB-M but changes are fewer, which guarantees good quality stegos. Zang and Wang [72] claimed that the modification direction of Mielikainen's scheme is not exploited fully; that is why they developed a data hiding scheme by Exploiting Modification Direction (EMD) which achieves maximum data hiding capacity through one bit per pixel (bpp).

A novel data hiding scheme has been introduced by Wu and Tsai [69] using Pixel Value Difference (PVD). The PVD scheme calculates the difference between two adjacent pixels of cover image and the number of data bits are to be embedded depending on the absolute difference value and a predefined reference table. Data bits are embedded by modifying these two pixel values. Zhang et al. [74] have shown that the scheme proposed by Wu and Tsai [69] is vulnerable to steganalysis based on histogram of pixel value difference. It can provide an estimate of the embedded data length due to its abnormal behavior. They suggested a pseudo random dithering approach which removes the undesirable steps existing in the PVD histogram of the stego image which preserve invisibility of large embedding capacity. Wang et al. [66] followed the idea of PVD and presented a data embedding method using PVD and modulus. It uses the same technique that was used in Wu and Tsai [69] to decide the number of bits to be concealed into a given pixel pair and then the remainder of these two pixels are calculated. Data is then embedded by modifying the remainder values. Compared to Wu and Tsai's method, Modulus Function -Pixel Value Difference (MF-PVD) reaches a higher payload with good image quality. A loss-less data hiding scheme was designed by Lin and Hsueh [43] which embeds secret message into a cover image using the two differences - between the first and second pixel as well as between the second and third pixel in a three pixel block. The average payload and pure payload capacities are 1.39 and 1.32 (bpp) respectively for PSNR greater than 30 (dB). Chang et al. [7] proposed three PVD (TPVD) to provide large embedding capacity and reduce the distortion by optimal approach of choosing the address point and adaption. They achieved smaller than 38 (dB) PSNR with 1.5 (bpp). PVD scheme developed by Wang et al. [66] had abnormal increase and fluctuation of PVD histogram which may reveal the existence of a hidden message that has been solved by Joo et al. [25]. They used some adjusting process which helps to remove fluctuation around the border of sub-range and achieve high capacity with good imperceptibility. After embedding around 52,275 bytes data they achieve 48.9 (dB) PSNR. Their scheme is also secure against various attacks like RS analysis, steganalysis for LSB matching and PVD histogram based attack. In 2010, Luo et al. [47] proposed a new data hiding scheme based on edge adaption which can take the embedding region corresponding to the length of secret data and the difference between consecutive pixel in the cover image. But their PVD scheme was not good for adaptive embedding. It may lead to possible attack by counting the difference of adjacent pixels in both vertical and horizontal direction that can be exploited by Li et al. [40].

A new data hiding approach was designed by Yang et al. [70] in which two pairs of pixel in a block are processed at the same time. The exploited edge area is more efficient to increase embedding capacity but the quality has been slightly dropped. In 2012, Zaker and Hamzeh [71] observed that the histogram of difference value of stego image under the TPVD is vulnerable to a particular statistical analysis. So they introduced a new steganalytic measure named *Growing Anomalies* that has a linear relationship with secret messages. This proposed steganalyzer can classify with test image as stego or cover with 97% accuracy when they contain more than 10% secret data. A histogram modification scheme for loss-less data hiding has been suggested by Tsai et al. [63] that can calculate the difference between each processing pixel and its neighbor and then use these differences to construct the histogram while the secret message is also being embedded into the pixel located at the peak value based on a histogram shifting scheme in gray scale image. The data capacity for one peak value of each histogram can achieve 44,168 bits on average PSNR value around 50 (dB) but for two peak value data hiding capacity can achieve 61,885 bits on average PSNR values around 47 (dB).

Hong [23] presented a new strategy using the idea of PVD and a patched reference table (PVD-PRT) to provide a better image quality and extendable embedding capacity. In addition, Hong and Chen [24] developed a steganography method based on pixel pair matching (PPM). This method utilized the values of pixel pairs as reference coordinates. To hide the message bits, this method first search for a coordinate in the neighborhood set of this pixel pair based on the message bits. Then, it replaces the pixel pair with the selected coordinate to embed the message bits.

Chen [11] proposed the PVD based method to embed unequal amount of secret information using pixel complexity. In this approach, secret information was embedded in an embedding cell of size (2×2), which was composed of randomized embedding units to reduce the falling of boundary program and eliminate sequential embedding. Each embedding cell has two embedding units Pivot Embedding Unit (PEU) and Non Pivot Embedding Unit (NPEU). The difference value of the pair pixel in PEU is calculated to determine the complexity of the pair and to determine the amounts of secret bits to be embedded. More bits will be embedded in the

complex area and less in the smooth area. This scheme achieve 47.3 (dB) PSNR when embedded with 54,384 bytes secret data.

Recently, reversible data hiding has attracted much attention to the researchers. Reversible Data Hiding (RDH) is a technique to embed a piece of information into a cover media to generate the stego-media, from which the original cover media can be exactly recovered after extracting the embedded messages. RDH, introduced by Barton [3] which compresses some alternate overlapping bits and add bit stream first then embed them into data block. Fridrich et al. [17] suggested a high capacity data hiding method that embed some message into a cluster of bits. Tian [58] designed a data hiding scheme using difference expansion technique to hide the secret message within a pair of pixel. Alattar [1] modified Tian's method and used the distance difference between four pixels. Lee et al. [39] utilized the histogram of the difference of pixel values to hide the secret data within cover media for improving the visual quality. Being reversible, the original and the embedded data can be completely restored. A RDH using histogram shifting has been proposed by Ni et al. [51]. After that Lin et al. [42] and Tsai et al. [61] suggested to improve RDH scheme through multilevel histogram shifting. Thodi et al. [59] presented RDH scheme that combine histogram shifting and difference expansion.

Chang et al. [5] offered dual image based data hiding technique using EMD method. They first established a mod function of a (256×256) magic matrix. Then convert the secret data bits into numeral system of base-5. Two bits secret data are embedded within a pixel pair of each image at a time. Lee et al. [35] introduced a loss-less data hiding technique that utilizes centralized difference expansion to hide more secret data into smoother areas of cover image. Later, Lee et al. [36] embedded secret message using the center point direction of pixels to get the stego-pixels. To protect the deterioration of the image quality, Lou et al. [44] proposed Reduced Difference Expansion (RDE) method. Lou's scheme is not only reversible but also meets low computational cost with high capacity data embedding scheme. Lee and Huang [34] developed a dual-image based RDH method. In their scheme, the average embedding rate is 1.07 (bpp). Qin et al. [52] presented a dual image based data hiding scheme using EMD. A LSB matching data hiding technique has been designed by Lu et al. [45]. The stego images are obtained through the mod function. To achieve the reversibility in data hiding, the LSBs are

checked via an averaging procedure then modification has been performed using a rule table. Chang et al. [5] embedded secret message bits by the mod function to accomplish a higher data hiding capability of 1.00 (bpp), but the visual quality of image was substandard to the method proposed by Lee et al. [39]. Zhang and Wang [73] suggested EMD method, which takes n pixels as embedding bits, and embed digits in $(2n + 1)$ base number system. Kieu and Chang [31] presented a new extraction function by modifying the extraction function proposed by Zhang and Wang's scheme. To solve the irreversibility of the EMD method in Zhang and Wang's scheme, they suggested a novel data hiding strategy based on EMD with reversibility by using two steganographic images, which can achieve satisfactory performances of the data embedding capacity and the quality. Shen and Huang [54] developed a data hiding scheme using PVD and improved EMD but the scheme was not reversible. Qin et al. [52] presented only EMD as a reversible data hiding scheme. In 2016, Lee et al. [37] developed an efficient reversible data hiding with reduplicated exploiting modification direction using image interpolation. Kuo et al. [32] presented a high capacity data hiding scheme using multi-bit encoding function. The embedding capacity of Kuo et al.'s scheme is 4.5 (bpp) but the image quality is nearer to 30 (dB).

Thus, designing an innovative scheme is still an important issue which could maintain good quality image and increase data embedding capacity through dual-image. In this thesis, some data hiding schemes have been proposed based on PVD, DE and EMD using dual-image which achieve good visual qualities and high embedding capacity.

1.2.3 Brief review of data hiding through Weighted Matrix

A. Westfeld [67] introduced F5 algorithm in which matrix based data embedding occurs using binary Hamming code. They embed k-bits secret message by modifying one bit of $2^k - 1$ least significant bits in the host data. The embedding efficiency increases with the increase of k, while the payload decreases contrarily. In order to increase the embedding efficiency and payload simultaneously, an extended F5 algorithm was developed by Fan et al. [13]. They come up with a brand new idea to realize this aim through adding n-layer extension into previous technique and modifying the form of original hash function. Jung and Yoo [26] suggested a new data hiding method using image interpolation through Neighbor Mean Interpolation (NMI). Lee and Huang [38] proposed improve image interpolation technique by Interpolating with Neighbor-

ing Pixels (INP). After that, Tang et al. [57] designed high capacity RDH through multi-layer embedding (CRS) with payload 1.79 (bpp) and PSNR is nearer to 33.85 (dB). In 2016, Tsai et al. [62] proposed an adjustable interpolation-based data hiding scheme based on LSB substitution and histogram shifting. This is two-stage data hiding scheme based on interpolation, LSB substitution, and histogram shifting.

A good data embedding method using a key matrix K and a weighted matrix W has been proposed by Tseng et al. [64] for binary image, that can concealed only two bits in a (3×3) pixel block. A better data hiding scheme through weighted matrix has been presented by Fan et al. [14] for gray scale image that can concealed only four secret data bits within a (3×3) block. Both these matrix based data hiding schemes one can perform only one modular sum of entry-wise-multiplication with weighted matrix W and a (3×3) pixel block. Achieving high capacity with reversibility in data hiding through weighted matrix while maintaining good visual quality is still an important research issue. RDH becomes a very important and challenging task in hidden data communication especially in medical and military applications for ownership identification, authentication and copy right protection.

In the literature, no researcher has considered reversibility with high embedding capacity using weighted matrix. In this thesis, some new weighted matrix based data hiding schemes have been formulated and solved using dual image and image interpolation.

1.3 Problem Domain

Data hiding is the technique of secured concealed communication which carry private data via some multimedia object so that the representation of private message will not draw any attention from the eavesdroppers while they are being moved through an open public channel. There is a high risk of disclosing while they are being transferred through unsecured public channel. Therefore achieving safe secure communication is one of the important objectives of current research. The story of prisoner's problem was presented by Simmons in 1983 [55] in which the merits and capabilities were explained when the public channel is unsecured.

If the probability of modification within the cover image is less, the security of the data hiding method may increase. A possible way to enhance data hiding security is to increase the embedding efficiency [*number of embedding bits per one embedding change*]. Matrix encoding is one of the popular technique of data hiding which can be used to increase the embedding efficiency. The concept was first proposed by Crandall [12] and implemented by Westfeld [67]. The basic idea is to divide coefficients into groups and use Hamming error correcting codes to limit the changes in each group. A(d, n, k) code can be used to embed k bits into n coefficients by making changes at most d coefficients. The data hiding through Hamming code recently proposed by Chang et al. [9], Kim et al. [28], Ma et al. [49], Kim and Yang [30] and Lien et al. [41]. Chang et al. [9] presented data hiding scheme on (7,4) Hamming code which is not reversible scheme and its visual quality is nearer to 50 (dB). Kim et al. [28] used halftone image to hide secret data using Hamming code where payload and the visual quality is limited. Ma et al. [49] and Lien et al. [41] also used halftone image for data hiding where visual quality is poor. Kim and Yang's [30] data hiding scheme is not reversible. All these developed schemes do not consider any shared secret key to enhance the security. They do not consider reversibility in their developed schemes which is one of the important issues in current research on data hiding.

In Wu and Tsai's [69] PVD based data hiding scheme, the quality as well as the capacity is limited and the scheme is not reversible. The data hiding capacity of Wang et al.'s [66] scheme is same as Wu and Tsai's scheme although the quality is a bit improved due to modulus function but the scheme does not achieve reversibility. Joo and Lee [25] proposed data hiding scheme to enhance the security by preventing abnormal increase of histogram values by a novel adjust-

ing process but the scheme can not recover original image successfully. Chen [11] proposed data embedding technique by pixel pair matching (PPM) to embed more information and to improve image quality but the scheme is also not reversible. All these schemes do not consider any shared secret key to enhance the security in data hiding. Data hiding using DE and EMD method is also paid more attention in the current research. Lu et al. [45] developed dual image based data hiding scheme with payload only one (bpp) but no shared secret key has been considered to enhance the security. Qin et al. [52] design a hybrid reversible data hiding scheme by combining PVD, DE and EMD with payload 1.16 (bpp) but did not considered any shared secret key in their approach. Data hiding in a special domain is not as much secure as other domains because data hiding is carried out in LSB. So, use of shared secret key is very important issue in data hiding for authentication, copyright control and privacy protection. It is possible to enhance security without compromising quality and embedding capacity.

Some reversible data hiding schemes have been proposed by Chang et al. [5], [10], using dual image but their embedding capacity falls short to the demand for today's digital world. The data hiding capacity is nearer to one (bpp). So there is a scope to improve the embedding capacity in dual image based data hiding schemes. Lee et al. [36], [34] also developed dual image based data hiding scheme with poor data hiding capacity. In the literature, none have attempted to achieve reversibility through PVD based data hiding scheme. To achieve a good quality image with low modification, in low cost is a challenge in designing a new reversible data hiding scheme using dual image.

In PVD, DE and EMD based data hiding schemes, overflow and underflow may occur frequently during data embedding. This may effect to measure the performance of data hiding. This is also a challenging job to design a new RDH scheme to control overflow and underflow situation without disturbing quality, security and capacity.

Tseng et al. [64] suggested a secure scheme that uses binary image as cover media and can conceal only two bits secret data within a (3×3) pixel block. After that Fan et al. [13] proposed an improved efficient data hiding scheme which can hide only four bits secret data within a (3×3) pixel block. To increase the data hiding capacity Jung and Yoo [26] first advised

data hiding scheme using image interpolation then Lee and Huang [38] proposed more eminent data hiding scheme through image interpolation using multi-layer embedding. Tang et al. [57] observed that the average payload 1.79 (bpp) with PSNR 33.85 (dB) when using image interpolation with multi-layered data hiding scheme. In the literature, a single weighted matrix has been used for data embedding in Tseng et al. [64] and Fan et al.'s [13] scheme. There is an opportunity to enhance security through modification of weighted matrix for every new block using shared secret key.

In Tseng et al. [64] and Fan et al.'s [13] scheme only one entry-wise-multiplication has been performed to embed only few bits secret data in a single block. So, there is a possibility to improve data hiding capacity by performing repeated entry-wise-multiplication operation using image interpolation and dual image through weighted matrix. In this literature, no researcher has exploited reversibility in data hiding through weighted matrix. Dual image provides security in a data hiding scheme because without simultaneous dual image, it is hard for eavesdroppers to retrieve secret data from stego images. This is a special case of secret sharing.

1.4 Motivations and Objectives of the Thesis

The main objective of this thesis is to design some new secured high capacity reversible data hiding schemes. Now, the key issues on the data hiding schemes are embedding capacity, the perceived quality, reversibility and security.

(i) So far, the data hiding through Hamming code, PVD and Weighted matrix which are not reversible, has a limited embedding capacity. Therefore, the motivation is to increase data hiding capacity and achieve reversibility using said techniques.

(ii) In the literature, it is seen that to send the secret message to the receiver, it is necessary to send the length of secret data through Hamming code based data hiding schemes. This has motivated us to develop some new schemes in which the length of secret message is not required.

(iii) So far, few data hiding techniques have been developed using dual image and image interpolation techniques which have a limited data embedding capacity with moderate magnitude of visual quality. From studying such types of techniques, we are motivated to investigate data hiding schemes using dual image and image interpolation in such a way that its capacity and quality have been improved.

(iv) There are many research works in which secret message has been communicated innocently through steganography without having any shared secret key. But, in real world it is seen that these techniques are less secure. This forces us to develop some innovative data hiding schemes to enhance the security of message using shared secret keys.

(v) In existing literature related with weighted matrix based data hiding schemes, it is observed that there is a limitation of data hiding capacity which is less. Again it is also seen that existing methods are not reversible. But, in present day there are many application areas in which reversibility is very essential. So noticing this, we are motivated to formulate some schemes through which these two drawbacks can be overcome.

The objectives of this thesis have been described elaborately as follows:

(i) **Designing some high payload data hiding schemes:**

In the literature, there are some techniques using Hamming code, PVD and Weighted matrix which have certain data hiding capacity, but from our experience in applications of data hiding schemes in some real life problems, it is seen that it is not sufficient for data hiding capacity. So, for this purpose, the objective of this thesis is to design some techniques to increase the payload using Hamming code, PVD and Weighted matrix which are discussed in Chapters 3, 4, and 5 respectively.

(ii) **Use of shared secret key in data hiding schemes:**

From the literature survey on data hiding schemes, it is seen that till now there exists some security loop hole for sending message from sender to receiver. So, our objective is to develop some schemes using a shared secret key in such a way that data hiding schemes will be more strengthened than previous ones. For this purpose in Chapters 3, 4 and 5, some schemes have been developed using Hamming code, PVD-DE, PVD-EMD, TPVD-DE and Weighted matrix incorporating shared secret keys in sequel.

(iii) **Introducing reversibility in data hiding:**

Though there exists many research work on data hiding schemes, till now no one has developed a scheme using Hamming code, PVD or Weighted matrix to achieve the reversibility. So, here, our objective is to develop some algorithms to achieve the reversibility through Hamming code, PVD and Weighted matrix to developed data hiding schemes which is explained in Chapters 3, 4 and 5 respectively.

(iv) **Conservation of perceptibility**

We know that in steganography, perceptibility is the main requirement in any data hiding algorithm. So, the first and foremost objective is to maintain perceptibility in all our proposed schemes.

1.5 Organization of the Thesis

In this thesis, some new reversible data hiding techniques are designed and solved. The thesis is divided into seven chapters.

Chapter 1

(Introduction)

This chapter contains an introduction giving an overview of the development on data hiding schemes. Brief review of data hiding, Problem domain, Motivations, Objectives and Organization of the thesis are included in this section.

Chapter 2

(Data Hiding Methodologies)

In this chapter, data hiding methodologies have been described that are used to solve different types of data hiding problems. In the development of the data hiding schemes in this thesis, following data hiding methods have been used.

(i) Hamming Code

(ii) Pixel Value Difference (PVD)

(iii) Difference Expansion (DE)

(iv) Exploiting Modification Direction (EMD)

(v) Weighted Matrix based Data Hiding

(vi) Image Interpolation

(vii) Dual Image based Data Hiding Methods

We have then discussed Steganalysis and Steganographic Attacks.

Chapter 3

(Reversible Data Hiding using Hamming Code)

3.1: Partial Reversible Data Hiding using (7,4) Hamming Code (PRDHHC)

Secure data communication through Hamming code based data hiding without knowing the length of secret message is a challenging problem. A data hiding scheme using Hamming code with shared secret position is developed and solved. In this method, the original cover image is partitioned into (7×7) pixel block then collect LSB of each pixel. Now, adjust redundant bits using odd parity. The bit at the shared secret position is complemented and secret data bit is embedded through error creation. For the next row, the shared secret position is updated by the data embedding position of the previous row. The process is repeated to embed all secret message bits within cover image. If a row contains all 1s or 0s, then secret data bit is embedded at the first position. At the receiver end, bit at the shared secret position is complemented first and then secret data bit is retrieved by applying Hamming error correcting code. The extraction process will be continued until error is found at the secret position. In this scheme, Hamming adjusted cover image is recovered by complement bits at both the secret position and data embedding position but original cover image could not be recovered. It is observed that PSNR of PRDHHC scheme is nearer to 58 (dB) which is more than other existing schemes but the payload is only 0.142 (bpp). This is not reversible scheme.

3.2: Dual Image based Reversible Data Hiding using Hamming Code (DRDHHC)

To overcome the irreversibility of previous approach, dual image has been proposed. In this scheme, two copies of LSBs are collected and redundant bits at positions 1, 2 and 4 of first copy are adjusted by odd parity using bit positions 3, 5, 6 and 7; and the redundant bit at positions 3, 5, 6, and 7 of second copy are adjusted by odd parity using bits at positions 1, 2 and 4. After successfully embedding the secret data bits using previous technique, two stego pixel blocks are distributed between dual stego images depending on shared secret key. The secret data bits are successfully recovered at the receiver end from dual images by the help of shared secret position and Hamming error correcting code. After extracting the secret message from dual

stego images, bits from 3, 5, 6 and 7 positions from first stego image and bits from 1, 2 and 4 positions of second one are combined and rearranged to recover original cover image. The average PSNR of this proposed RDH scheme is greater than 53 (dB) and the maximum payload is 0.142 (bpp).

3.3: Enhanced Partial Reversible Data Hiding using Hamming Code (EPRDHHC)

Data embedding concept of PRDHHC is used in three LSB layers (LSB, LSB+1 and LSB+2) of cover image to enhance the payload. In this approach, PSNR is nearer to 52 (dB) and payload is 0.426 (bpp). The main drawback of this approach is that, it can not recover original cover image successfully after extraction the secret data.

3.4: Enhanced Dual Image based Reversible Data Hiding using Hamming Code (EDRDHHC)

To achieve reversibility, the techniques of DRDHHC and EPRDHHC have been combined to embed secret data. Dual image concept has been taken from DRDHHC and three LSB layers (LSB, LSB+1, LSB+2) concept has been taken from EPRDHHC. This is an RDH scheme in which the average PSNR is greater than 38 (dB) and the payload is 0.426 (bpp).

All the experimental results are presented graphically and numerically. The results are compared with existing schemes. The results of different steganalysis (RS analysis, Statistical analysis) and steganographic attacks (Histogram attack and Brute force attack) are presented. There is a scope to improve the data hiding capacity while maintaining good visual quality through other data hiding approaches discussed in next chapter.

Key features of the schemes in Chapter 3:

(i) Achieving reversibility with good visual quality is the main key feature of these proposed Hamming code based data hiding schemes.

(ii) Any arbitrary length of secret message can be communicated through these data hiding

schemes.

(iii) Shared secret position has been used to enhance security. It has been updated for new block using $\kappa_{i+1} = (\kappa_i \times \omega) \mod 7 + 1$, where i= 1, 2, 3, $\ldots, N_B$. N_B represents the number of block, κ_0 is the shared secret position and ω is the data embedding position.

(iv) In the dual image based schemes, both shared secret position κ_0 and shared secret key ξ have been used. The stego image blocks are distributed between dual stego images depending on the bit pattern of secret key ξ.

Chapter 4

(Reversible Data Hiding using PVD, DE and EMD)

4.1: Dual Image based RDH using PVD with DE (PVDDE)

To enhance the embedding capacity while preserving good visual quality, a dual-image based RDH scheme using PVD with DE (PVDDE) has been proposed. Here, a secret message is partitioned into n bits, where $(n-1)$ bits are embedded using PVD and one bit is embedded through DE and generate two sets of pixel pair. These two sets of pixel pair are distributed within dual images depending on the bit pattern of a shared secret key. At the receiver end, extraction of the hidden message is performed through either PVD or DE that also depends on the same secret key. Here, overflow and underflow situation has been controlled which may occurs at the data embedding stage. The payload is 1.25 (bpp) and PSNR is greater than 37 (dB) in this approach.

4.2: Dual Image based RDH using PVD with EMD (PVDEMD)

To increase the payload, another dual-image based RDH scheme using PVD with EMD has been proposed. First, enlarge the original image using image interpolation technique then embed secret data bits within pixel pair using PVD. Here, four data bits are embedded through PVD method and two data bits are embedded using EMD method. After embedding two sets of stego pixel pairs have been generated. After that stego pixel pairs are distributed between dual image based on the bit pattern of a shared secret key. At the receiver end, the stego pixel pairs are distinguished using secret key. Then corresponding PVD or EMD methods are used to extract hidden message and recover original image. In this approach, the PSNR is 40.43 (dB) and payload is 1.75 (bpp).

4.3: Dual Image based RDH using Three PVD (TPVD) with DE (TPVDDE)

Further, RDH method using Three Pixel Value Difference (TPVD) with DE (TPVDDE) has been proposed. The embedding capacity of this method is 2.15 bpp which is higher than other existing schemes but the PSNR is less than 30 (dB).

All the experimental results are numerically and graphically illustrated. The results of these new three different approaches are compared with existing schemes. The effect of different steganalysis (RS analysis, Relative entropy and Statistical analysis) and steganographic attacks (Histogram attack Brute force attack) are demonstrated.

Key features of the schemes in Chapter 4:

(i) Data embedding using PVD method was not reversible. Reversibility has been achieved through proposed data hiding schemes using PVD, DE and EMD methods.

(ii) Data embedding capacity has been increased in PVD based data hiding methods using dual image and image interpolation.

(iii) Shared secret key has been used to distribute stego pixel pairs among dual images to enhance security.

(iv) Overflow and underflow situations have been controlled which may appears during data embedding.

Chapter 5

(Reversible Data Hiding using Weighted Matrix)

5.1: Dual Image based RDH using Weighted Matrix (DRDHWM)

Weighted matrix based RDH using dual image has been introduced. First, partition the cover image into (3×3) pixel block. Then perform modular sum of entry-wise-multiplication between image block and a predefined weighted matrix. After that calculate the difference between value of modular sum and selected data unit. To embed these secret data, increase or decrease the pixel value that depend on the sign of the calculated difference value. The pixel has been selected

depending on the position of the element of weighted matrix. Now, store the difference value within stego pixel by adding with original pixel value. The process is repeated nine times to embed thirty six bits secret data within the selected block. For each next i-th block (i = 1, 2, ...), update weighted matrix W_{i+1} as $W_{i+1} = (W_i \times \kappa - 1) \mod 9$, where $gcd\ (\kappa,\ 9) = 1$ and κ is shared secret key. Finally, the original and stego pixels are distributed between dual images depending on the bit pattern of a shared secret key. At the receiver end, secret message has been extracted successfully using predefined weighted matrix and the shared secret key. The original image has been recovered without any distortion from dual stego images because the original pixels are kept unaltered within stego images during data embedding which ensure reversibility. In this scheme, payload is 1.98 (bpp) and PSNR is greater than 39 (dB).

5.2: Interpolated Image based RDH using Weighted Matrix (IRDHWM)

To increase the payload, a high capacity secure RDH scheme has been proposed. First, enlarge the size of original image into double through image interpolation. Then partition the original image into (3×3) pixel block and interpolated image into (5×5) pixel block. Perform modular sum of entry-wise-multiplication of image block with predefined weighted matrix. Calculate the difference value between modular sum of entry-wise-multiplication and selected data unit. In each operation, the data embedding position is identified and stored at three least significant bits of the interleaved pixel of interpolated image. Embed secret data by increasing or decreasing the original pixel value by one. Twelve multiplication operations have been performed to embed forty-eight bits secret data within a (5×5) pixel block of interpolated image. For next each i-th block, ($i = 1,\ 2, \ldots$), update weighted matrix W_{i+1} as $W_{i+1} = (W_i \times \kappa - 1) \mod 9$, where $gcd\ (\kappa, 9) = 1$ and κ is shared secret key. The data hiding capacity of this approach is 2.96 (bpp) and PSNR is 37.37 (dB).

5.3: Interpolated Dual Image based Reversible Data Hiding using Weighted Matrix (IDRDHWM)

Finally, a very high capacity RDH scheme has been proposed through weighted matrix using interpolated dual image. The data embedding procedure has been done in two stages. In the first

stage, embed thirty six bits secret data within a block of dual image through repeated embedding process of nine times using weighted matrix. In the next stage, repeat embedding process twenty four times to embed ninety six bits secret data within each block of interpolated dual image. After hiding one hundred and thirty-two bits secret data within one block of dual image update the weighted matrix. For i-th block ($i = 1, 2, \ldots$), the weighted matrix W_{i+1} can be updated as $W_{i+1} = (W_i \times \kappa - 1) \mod 9$, where $gcd\,(\kappa, 9) = 1$ and κ is a shared secret key. In the extraction process, the positional values are extracted from interpolated dual stego images and the secret data is recovered by performing modular sum of entry-wise-multiplication between weighted matrix and original pixel block. Again rearrange the pixel using shared secret key from dual image and perform the same extraction operation thirty-three times and extract one hundred thirty-two bits secret data from a pixel block. The scheme provides average embedding payload 3.46 (bpp) with PSNR greater than 35 (dB).

All the experimental results of the proposed methods are numerically and graphically illustrated. The results of these different approaches are compared with existing methods. Steganalysis and steganographic attacks on stego images are performed which are also illustrated numerically.

Key features of the schemes in Chapter 5:

(i) Achieve high payload with good visual quality in weighted matrix based data hiding schemes.

(ii) Achieve reversibility in weighted matrix based data hiding schemes.

(iii) Update weighted matrix $W_{i+1} = (W_i \times \kappa - 1) \mod 9$, where $gcd\,(\kappa, 9) = 1$ and $i = 1, 2, 3, ..., N_B$, N_B represents the number of blocks in the cover image for each new block to enhance security.

Chapter 6

(Analysis and Discussions)

In this chapter, proposed data hiding schemes are analyzed. The comparisons of suggested schemes with respect to data embedding capacity and visual quality are presented here. The

results of steganalysis and various steganographic attacks are presented.

Chapter 7

(Conclusion and Future Research Work)

At the end, some limitations and the scope of future research works have been discussed.

Chapter 2

Data Hiding Methodologies

2.1 Introduction

Data hiding is the technique to communication secret information innocently by embedding these within a cover media. It is an important issue in the field of multimedia security. Many data hiding schemes have been developed in last few decades by the researchers. The aim of this research work is to improve data hiding capacity while maintaining good visual quality and enhance security. In this section, we have discussed different data hiding approaches using Hamming code, Pixel Value Difference (PVD), Difference Expansion (DE), Exploiting Modification Direction (EMD), Weighted Matrix (WM), Image Interpolation and Dual Image which have been used in this research work. We then compared different existing schemes in terms of data hiding capacity and visual quality of stego images. Finally, different approaches of steganalysis and steganographic attacks are explained.

2.2 Hamming code

Richard Hamming [21] formulated a sophisticated pattern of parity checking code called Hamming code. It is linear code for error correction that can be used to detect and correct single bit error. The length of linear code n with k dimension are represented as $[n, k]$ codes. If c is a $[n, k]$ linear code, the dual to it is termed as $[n, n-k]$ linear code. If H is a checker matrix for c then the matrix H will be $(n-k) \times k$ and the row of which are orthogonal to c and $\{x \mid H\ x^T\ =\ 0\}\ =\ c$ and

$$(m_1, m_2, \ldots, m_k)^T = H.(LSB(x_1), (LSB(x_2), \ldots, LSB(x_n))^T \tag{2.1}$$

Any secret message of k bits, say $(m_1, m_2, \ldots, m_k)$ can be embedded in the LSB of n pixel, say $(x_1, x_2, \ldots, x_n)$ by at most G changes. Here, G is the largest number of possible changes and G_a is the average number of changes. The embedding efficiency can be measured by (k/G_a) and embedding rate will be (k/n). The position of erroneous bits must be determined to correct the error. For n-bit code $log_2(n)$ bits are required. The Table 2.1 shows the parity check for the matrix of $(7, 4)$ Hamming code.

The Hamming code is used by odd parity to allow the identification of a single bit error shown in Table 2.2. Creation of the codeword as follows:

(i) Parity bit positions are marked which are power of two such as $2^0,\ 2^1,\ 2^2, \ldots$.

Table 2.1: Parity check matrix for $(7, 4)$ Hamming code

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(ii) All remaining positions are allowed to embed secret data bits such as 3, 5, 6, ...

(iii) The sequence of bits for every parity or redundant bits are computed as follows:

Redundant bit r_1: One bit position is checked and skipped alternatively which are $(1, 3, 5, 7, 9, 11, \dots)$

Redundant bit r_2: Two bit positions are checked and skipped alternatively which are $(2, 3, 6, 7, 10, 11, \dots)$

Redundant bit r_4: Four bit positions are checked and skipped alternatively which are $(4, 5, 6, 7, \dots)$

Redundant bit r_8: Eight bit positions are checked and skipped alternatively which are $(8 - 15, 24 - 31, 40 - 47, \dots)$

(iv) Adjust parity bit through odd parity.

More detail are found in the book "The theory of error correcting codes" by F. J. MacWilliams, N. J. A. Sloane [48]. Recently data hiding through Hamming code has become a very interesting

Table 2.2: Redundant bits adjustment using odd parity for error detection and correction

Highlighted bits with power of 2	01	02	03	04	05	06	07	08	09	10	11
Insert data	1	1	1	1	0	0	0	0	0	0	1
Highlight the check bit	1	1	1	1	0	0	0	0	0	0	1
Odd parity of 2^0	1		1		0		0		0		1
Odd parity of 2^1		1	1			0	0			0	1
Odd parity of 2^2				1	0	0	0				
Odd parity of 2^3								0	0	0	1

field in current research which is used in security and online application. Data can be embedded within image which contains ownership identification, authentication and copy right protection.

In this section, Kim et al.'s [28] data hiding scheme has been discussed first and then Lien et al.'s [41] data hiding scheme is explained using Hamming code for halftone image.

2.2.1 Kim et al.'s scheme

Kim et al. [28] proposed Data Hiding using Hamming Code (DHHC) which employed $(15, 11)$ Hamming code to hide secret data into a halftone image. The cover image is divided into (4×4) block. Fifteen bits codeword is used for a block and syndrome (S_1) is calculated from codeword using $S_1 = H \times (c)^t$, where H is the parity checker matrix and c is a 7-bits sequence binary number known as codeword. The non-zero syndrome value denotes the bit error position. The flipping bit of the position value in a codeword will be the correct bits within codeword. Then four bits secret message EXclusive-OR-ed with the codeword. The halftone image of size $(n \times n)$ which compose n pixels is divided into continuous blocks of size (4×4). They used code length $n = 2^{r-1}$ and the number of bits that is encoded in each codeword is $k = (n - r)$, where r is a non-negative integer. They considered minimum Hamming distance $d = 3$ so that one error can be corrected and two errors can be detected. For example, let message $m = 101$, code word $c = 1101001$. Then calculate the syndrome $H \times c^t$ =(000). To hide the secret message, an Exclusive- OR $(\oplus)$ is computed and we get $w = H \times c^t \oplus m$. If w is zero, then no need to complement, otherwise, find the w^{th} column of c and complement the w^{th} pixel bit. In DHHC, the MPSNR (Modified Peak Signal to Noise Ratio) of lena image is 32.03 (dB) when $16,384$ bits are embedded and 44.71 (dB) when embedded with $4,096$ bits.

2.2.2 Lien et al.'s scheme

Lien et al. [41] suggested Dispersed Data Hiding using Hamming Code (DDHHC) in which an image is divided into sixteen sub images and then using space filling curve, they put the index of each sub images. Those pixels correspond to the same index in each sub images gathered as a block of sixteen pixels. To embed $(4 \times m)$ bits of secret messages, they randomly select m blocks from the image and then these blocks are sorted by the index number. After that 4 bits secret messages are embedded within 15 bits codeword randomly. To read embedded data, stego image is partitioned into blocks using space filling curve partition. Then secret data is extracted from 15 bits codeword of each block. In DDHHC, the MPSNR is 44 (dB) after embedding $4,096$ bits secret data within cover image.

The main drawback of these Hamming code based data hiding schemes are irreversibility and these techniques are designed for halftone image only. There is scope to develop Hamming code based reversible data hiding schemes for gray scale images using secret key which enhances security.

2.3 Pixel Value Difference (PVD)

Pixel Value Difference (PVD) is a data hiding method where difference of two consecutive pixels have been taken to embed confidential information. If the difference is high it means pixels belong to the edge area of the image, where large quantity of data bits are possible to embed. If the difference of two adjacent pixels is small that means the pixel pair belongs to the smooth area of the image where less amount of secret data bits are possible to embed. In PVD method, a specific range table (R) has been used to calculate the quantity of data bits which are possible to embed within pixel pair. The sub-range of the range table is always power of 2. The difference value is mapped into the R. The number of secret message bits are to be embedded depending on the sub-range of R.

PVD based data hiding method is introduced by Wu and Tsai (2003) [69]. Then few PVD based data hiding schemes are developed by the researchers. Some of them are discussed here. Wang et al. (2008) [66], Joo et al. (2010) [25] and Chen's (2014) [11] PVD based data hiding schemes are described below.

2.3.1 Wu and Tsai's scheme

Wu and Tsai [69] introduced a novel data embedding method called PVD, where the difference of two adjacent pixels in the cover image is used for data hiding. The quantity of data bits to be concealed within the pair of pixels are determined by their absolute difference and a predefined range table (R) shown in Fig. 2.1. Consider a cover image C $(M \times N)$ and divide it into non overlapping blocks B_i, $\{B_i | i = 1, 2, \ldots, \lfloor (M \times N/2) \rfloor\}$ in raster scan order, where each block contains two consecutive pixels. Consider the consecutive pixel pair (x_i, x_{i+1}) and

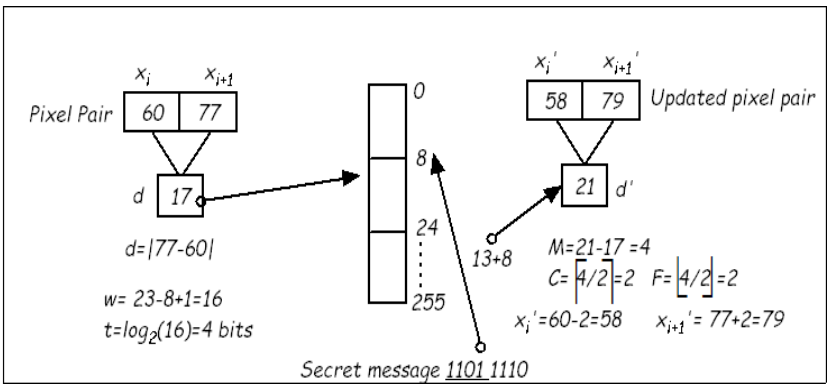


Figure 2.1: Wu and Tsai's PVD scheme with example

the difference d_i is calculated using two pixels x_i and x_{i+1} as

$$d_i = |x_i - x_{i+1}| \tag{2.2}$$

The absolute difference $|d_i|$ belongs to the range between 0 to 255. If the difference $|d_i|$ is close to 0 it means pixels are taken from smooth area of the image and if it is close to 255 it means pixels are taken from the edge area of the image. In this approach, a range table R has been proposed with n contiguous sub-range R_m, $\{R_m | m = 1, 2, \ldots, n\}$. Each sub-range R_m has a lower and a upper bound, namely lb and ub respectively. So, $R_m \in [lb, ub]$. The width wb of each sub-range R_m is obtained by

$$wb = ub - lb + 1 \tag{2.3}$$

The number of confidential message bits (t) are to be concealed that will be determined by the sub-range of R, where the difference $|d_i|$ is mapped. Now calculate (t) as

$$t = \lfloor \log_2(wb) \rfloor \tag{2.4}$$

Then select t bits from secret message D and convert it into decimal value v. To embed v unit secret data, compute new difference $|d_i^{'}|$ using

$$d_i^{'} = v + lb \tag{2.5}$$

Again calculate $d_i^{''} = d_i^{'} - d_i$, then modify the pixel values x_i and x_{i+1} to get $x_i^{'}$ and $x_{i+1}^{'}$ using following equation.

$$(x_i^{'}, x_{i+1}^{'}) = \begin{cases} x_i + \lceil d_i^{''}/2 \rceil, x_{i+1} - \lfloor d_i^{''}/2 \rfloor, & \text{if } x_i \geq x_{i+1} \text{ and } d_i^{'} > d_i \\ x_i - \lfloor d_i^{''}/2 \rfloor, x_{i+1} + \lceil d_i^{''}/2 \rceil, & \text{if } x_i < x_{i+1} \text{ and } d_i^{'} > d_i \\ x_i - \lceil d_i^{''}/2 \rceil, x_{i+1} + \lfloor d_i^{''}/2 \rfloor, & \text{if } x_i \geq x_{i+1} \text{ and } d_i^{'} \leq d_i \\ x_i + \lceil d_i^{''}/2 \rceil, x_{i+1} - \lfloor d_i^{''}/2 \rfloor, & \text{if } x_i < x_{i+1} \text{ and } d_i^{'} \leq d_i, \end{cases} \tag{2.6}$$

Finally the stego image S is generated using modified pixels $x_i^{'}$ and $x_{i+1}^{'}$.

The extraction process of Wu and Tsai's data hiding scheme is described here. First select two consecutive pixels from stego image S in raster scan order. Then calculate difference $d_i^{'}$, where $d_i^{'} = |x_i - x_{i+1}|$. To extract secret data v, subtract lb from $d_i^{'}$ that is $v = d_i^{'} - lb$, where lb is the lower bound of the sub-range of the range table (R). Then convert v into binary form and get the hidden information.

The visual quality of the stego image of this approach is evaluated by Peak Signal to Noise Ration (PSNR) which is 40.3 (dB) after embedding 52,204 bytes within cover image. The major drawback of Wu and Tsai's scheme is that they could not suggest any solution to overcome the overflow and underflow problem if and when they occur. They simply leave those blocks where $x_i^{'}$ and $x_{i+1}^{'}$ fall in the boundary of the range [0, 255]. In this situation, the pixel value $x_i^{'}$ and $x_{i+1}^{'}$ are remain same as the original pixel value. To solve this problem, Wang et al. [66] suggested a new data hiding method using PVD which is better than Wu and Tsai's scheme with respect to image quality with same payload and to overcoming the underflow and overflow situations.

2.3.2 Wang et al.'s scheme

Wang et al. [66] compute the remainder of two consecutive pixels instead of their difference using the modulus function. The data hiding procedure of Wang et al. [66] is described below:

Input: Cover image C $(M \times N)$, Range table R, Secret data D

Output: Stego image S

Step 1: The number of bits (t) which are to be embedded within cover image are calculated according to Wu and Tsai's scheme.

Step 2: Calculate the remainder $B_{rem}(i)$ of each block B_i using the following equation

$$B_{rem}(i) = (x_i + x_{i+1}) \bmod 2^t \tag{2.7}$$

Step 3: Calculate $d^{'}$ and $d_1^{'}$ using the following equation

$$\begin{cases} d^{'} = |B_{rem}(i) - v| \\ d_1^{'} = (2^t - |B_{rem}(i) - v|) \end{cases} \tag{2.8}$$

Where v is the decimal value of t bits secret data. After embedding secret data the new pixels $x_i^{'}$ and $x_{i+1}^{'}$ can be obtained as follows:

rule 1: $(x_i^{'}, x_{i+1}^{'}) = (x_i - \lceil d^{'}/2 \rceil, x_{i+1} - \lfloor d^{'}/2 \rfloor$

if $B_{rem}(i) > v$ and $d^{'} \leq (2^t)/2$ and $x_i \geq x_{i+1}$

rule 2: $(x_i^{'}, x_{i+1}^{'}) = (x_i - \lfloor d^{'}/2 \rfloor, x_{i+1} - \lceil d^{'}/2 \rceil$

if $B_{rem}(i) > v$ and $d^{'} \leq (2^t)/2$ and $x_i < x_{i+1}$

rule 3: $(x_i^{'}, x_{i+1}^{'}) = (x_i + \lfloor d_1^{'}/2 \rfloor, x_{i+1} + \lceil d_1^{'}/2 \rceil$

if $B_{rem}(i) > v$ and $d^{'} > (2^t)/2$ and $x_i \geq x_{i+1}$

rule 4: $(x_i^{'}, x_{i+1}^{'}) = (x_i + \lceil d_1^{'}/2 \rceil, x_{i+1} + \lfloor d_1^{'}/2 \rfloor$

if $B_{rem}(i) > v$ and $d^{'} > (2^t)/2$ and $x_i < x_{i+1}$

rule 5: $(x_i^{'}, x_{i+1}^{'}) = (x_i + \lfloor d^{'}/2 \rfloor, x_{i+1} + \lceil d^{'}/2 \rceil$

if $B_{rem}(i) \leq v$ and $d^{'} \leq (2^t)/2$ and $x_i \geq x_{i+1}$

rule 6: $(x_i^{'}, x_{i+1}^{'}) = (x_i + \lceil d^{'}/2 \rceil, x_{i+1} + \lfloor d^{'}/2 \rfloor$

if $B_{rem}(i) \leq v$ and $d^{'} \leq (2^t)/2$ and $x_i < x_{i+1}$

rule 7: $(x_i^{'}, x_{i+1}^{'}) = (x_i + \lceil d_1^{'}/2 \rceil, x_{i+1} + \lfloor d_1^{'}/2 \rfloor$

if $B_{rem}(i) \leq v$ and $d^{'} > (2^t)/2$ and $x_i \geq x_{i+1}$

rule 8: $(x_i^{'}, x_{i+1}^{'}) = (x_i - \lfloor d_1^{'}/2 \rfloor, x_{i+1} - \lceil d_1^{'}/2 \rceil$

if $B_{rem}(i) \leq v$ and $d^{'} > (2^t)/2$ and $x_i < x_{i+1}$

After completion of **Step 3** if the pixel values $x_i^{'}$ and $x_{i+1}^{'}$ exceed the gray scale range [0, 255] then goto **Step 4**, otherwise goto **Step 5**.

Step 4: To over come the overflow and underflow situation the modified pixel $x_i^{''}$ and $x_{i+1}^{''}$ will be calculated as follows

$$(x_i^{''}, x_{i+1}^{''}) = \begin{cases} (x_i^{'} + (2^t/2), x_{i+1}^{'} + (2^t/2)), & \text{if } x_i^{'} < 0 \text{ or } x_{i+1}^{'} < 0 \\ (x_i^{'} - (2^t/2), x_{i+1}^{'} - (2^t/2)), & \text{if } x_i^{'} > 255 \text{ or } x_{i+1}^{'} > 255 \\ (0, x_i^{'} + x_{i+1}^{'}), & \text{if } x_i^{'} < 0 \text{ and } x_{i+1}^{'} \geq 0 \\ (255, x_i^{'} + (x_{i+1}^{'} + 255)), & \text{if } x_i^{'} > 255 \text{ and } x_{i+1}^{'} \geq 0 \\ (x_i^{'} + (x_{i+1}^{'} - 255), 255), & \text{if } x_i^{'} \geq 0 \text{ and } x_{i+1}^{'} > 255 \end{cases} \quad (2.9)$$

Step 5: Select the next block for data embedding.

Step 6: Repeat **Step 2** to **Step 5** until all data is embedded.

Step 7: End.

The extraction procedure of Wang et al.'s scheme is described below:

Input: Stego image S ($M \times N$), Range table R.

Output: Secret data D.

Step 1: Take two pixels and perform d_i, wb and t using equation (2.2), (2.3) and (2.4) respectively.

Step 2: Calculate the remainder $B_{rem}(i)$ of each block B_i using the following equation

$$B_{rem}(i) = (x_i^{'} + x_{i+1}^{'}) \bmod 2^t \tag{2.10}$$

then convert $B_{rem}(i)$ into its binary form of t bits. Concatenate the secret data bits to get secret data D.

Step 3: Select the next block for data extraction and goto **Step 1**.

Step 4: Continue **Step 1** to **Step 3** to extract entire secret data.

Step 5: End.

Example 2.3.1 *Consider two pixels* $x_i = 50$ *and* $x_{i+1} = 56$ *and the secret message* $D = 100$. *First calculate the difference* $d = |50 - 56| = 6$. *Here* d *belongs to sub-range* $[0, 7]$ *of the range table* R. *Then compute* $w = (7 - 0 + 1) = 8$. *The number of bits* $t = \lfloor \log(8) \rfloor = 3$, *that is* 3 *bits data is embedded within this pair. Extract* 3 *bits data from* D *that is* 100 *and its decimal value* $v = 4$. *Next calculate* $B_{rem}(i) = (50 + 56) \bmod 2^3 = 2$. *Therefore,* $d^{'} = |2 - 4| = 2$ *and* $d_1^{'} = (8 - |2 - 4|) = 6$. *According to* ***rule 6*** *of* ***Step 3***, *the new pixel values* $x_i^{'} = (50 + \lceil 2/2 \rceil) = 51$ *and* $x_{i+1}^{'} = (56 + \lfloor 2/2 \rfloor) = 57$.

At the time of data extraction, first calculate the difference $d^{'} = |51 - 57| = 6$. *Here* $d^{'}$ *falls to sub range* $[0, 7]$ *of the range table* R. *Then compute* $w = (7 - 0 + 1) = 8$. *The number of bits* $t = \lfloor \log(8) \rfloor = 3$, *that is* 3 *bits are extracted from this pair. Next calculate* $B_{rem}(i) = (51 + 57) \bmod 2^3 = 4$. *Then convert* $B_{rem}(i)$ *into binary form that is* 100. *So, the secret message* D *is* 100. ■

After embedding 52,275 bytes of secret data through Wang et al.'s scheme, PSNR is 42.6 (dB). Although, Wang et al.'s method generates better quality of stego image, but using the histogram analysis one can easily detect the existence of secret message within the stego image. To solve this problem, Joo et al. [25] proposed a data hiding method through improved modulus function.

2.3.3 Joo et al.'s scheme

The data hiding method of Joo et al.'s [25] scheme is different for odd blocks and even blocks. The embedding procedure of this method is described below:

Input: Cover image C ($M \times N$), Secret message D, Range table R.

Output: Stego image S ($M \times N$).

Step 1: Consider a cover image C of size ($M \times N$). At first the C is partitioned into disjoined blocks B_i, $\{B_i | i = 1, 2, \ldots, \lfloor (M \times N/2) \rfloor\}$ in raster scan order where each block contains two consecutive pixels. Let two pixels are x_i and x_{i+1}. Then determine the even blocks and odd blocks as follows:

$$\begin{cases} \text{if } (i \bmod 2) = 1, & \text{then } odd \text{ block} \\ \text{if } (i \bmod 2) = 0, & \text{then } even \text{ block} \end{cases} \tag{2.11}$$

Step 2: The difference value d_i is calculated between two pixels x_i and x_{i+1} as $d_i = |x_i - x_{i+1}|$. Here, $t = \lfloor \log w \rfloor$, where t is the number of embedded bits and $wb = ub - lb + 1$.

Step 3: Take t bits from data D and determine its decimal form as v. Then calculate $B_{rem}(i)$ using the following equation

$$B_{rem}(i) = (x_i - x_{i+1}) \bmod 2^t \tag{2.12}$$

Now, calculate the parameter Z using the following equation

$$Z = \begin{cases} v - B_{rem}(i), & \text{if } |v - B_{rem}(i)| \leq 2^{t-1} \\ v - B_{rem}(i) - 2^{t-1}, & \text{if } |v - B_{rem}(i)| > 2^{t-1} \\ v - B_{rem}(i) + 2^{t-1}, & \text{if } |v - B_{rem}(i)| < 2^{t-1} \end{cases} \tag{2.13}$$

Step 4: New pixel values $x_i^{'}$ and $x_{i+1}^{'}$ are computed based on the even/odd blocks using the following equation

$$(x_i^{'}, x_{i+1}^{'}) = \begin{cases} x_i + p_1, & \text{if } B_i \text{ block is } odd \\ x_i + p_2, & \text{if } B_i \text{ block is } even\ , \end{cases} \tag{2.14}$$

where, $p_1 = \lceil Z/2 \rceil$ and $p_2 = \lfloor Z/2 \rfloor$.

Step 5: Compute $d_i^{'} = |x_i - x_{i+1}|$. If $d_i^{'} \neq d_i$, re-adjust the pixel values $x_i^{'}$ and $x_{i+1}^{'}$ to $x_i^{''}$ and $x_{i+1}^{''}$ respectively. Then calculate $d_i^{''} = |x_i^{''} - x_{i+1}^{''}|$ so that $d_i^{'} = d_i$.

Step 6: End.

The extraction procedure of Joo et al.'s method is described below:

Input: Stego image S ($M \times N$), Range table R.

Output: Secret message D.

Step 1: Partitioned the stego image S $(M \times N)$ into non overlapping blocks B_i, $\{B_i | i = 1, 2, \ldots, \lfloor (M \times N)/2 \rfloor\}$ in raster scan order where each block contain two consecutive pixels $x_i^{'}$ and $x_{i+1}^{'}$.

Step 2: The difference $d_i^{'}$ of two pixels $x_i^{'}$ and $x_{i+1}^{'}$ is calculated as $d_i^{'} = |x_i^{'} - x_{i+1}^{'}|$. Now calculate $t = \lfloor \log(w) \rfloor$, where t is the number of bits embedded within pixel pair and $w = ub - lb + 1$ where $d_i^{'}$ mapped to range table R.

Step 3: Calculate the remainder $B_{rem}(i)$ of each block B_i using the following equation

$$B_{rem}(i) = (x_i^{'} + x_{i+1}^{'}) \bmod 2^t \tag{2.15}$$

then convert $B_{rem}(i)$ into its binary form of t bits and get data D.

Step4: End.

Lower bound (lb)	0	8	16	32	64	128
Upper bound (ub)	7	15	31	63	127	255
Number of bits to be Embedded	3	3	4	5	6	7

Figure 2.2: Range table of Wu and Tsai's PVD scheme

Example 2.3.2 *Consider two pixels* $x_i = 40$ *and* $x_{i+1} = 45$ *and the secret message* $D = 110$. *Assume that this block is odd block. First calculate the difference* $d = |40 - 45| = 5$. *Here* d *belongs to the sub-range* $[0, 7]$ *of the range table* R *shown in Fig.* 2.2. *Then compute* $w = (7-0+1) = 8$. *The number of bits* $t = \lfloor \log_2(8) \rfloor = 3$, *that is* 3 *bits are embedded into this pair. Extract* 3 *bits data from* D *that is* 110 *and convert it into decimal value* $v = 6$. *Now, calculate* $B_{rem}(i) = (40 + 45) \bmod 2^3 = 5$. *Therefore,* $Z = |6 - 5| = 1$, *since* $|6 - 5| \leq 2^{3-1}$ *then* $p_1 = \lceil 1/2 \rceil = 1$ *and* $p_2 = \lfloor 1/2 \rfloor = 0$. *This block is odd so new pixel value are* $x_i^{'} = (40 + 1) = 41$

and $x'_{i+1} = (45 + 0) = 45$.

To recover the secret information first calculate the difference $d' = |41 - 45| = 4$. *Here* d' *mapped to sub range* $[0, 7]$ *of the range table* R *shown in Fig.* 2.2. *Then compute* $w = (7 - 0 + 1) = 8$. *The number of bits* $t = \lfloor \log(8) \rfloor = 3$, *that is* 3 *bits secret data are extracted from this pair. Next calculate* $B_{rem}(i) = (41 + 45) \bmod 2^3 = 6$. *Then convert* $B_{rem}(i)$ *into binary form that is* 110. *So, the secret message* D *is* 110. ■

After embedding 52,275 bytes secret data using this scheme the PSNR is 41.9 (dB). There is a scope to improve the visual quality of the stego image which is proposed by Chen [11].

2.3.4 Chen's scheme

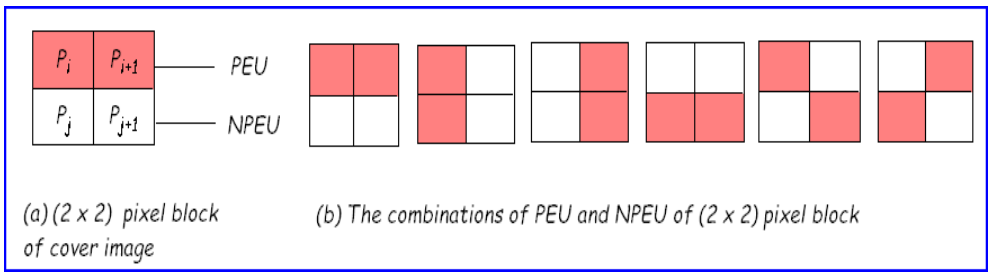


Figure 2.3: All combinations of PEU and NPEU in Chen's scheme for (2×2) block

To improve the quality of stego image, Chen [11] proposed a new PVD based data hiding method using Pixel Pair Matching (PPM) technique. The cover image C $(M \times N)$ is divided into (2×2) non overlapping blocks B_i, $\{B_i | i = 1, 2, \ldots, \lfloor (M \times N)/2 \rfloor\}$. In each block two pixels p_i and p_{i+1} are for pivot embedding unit (PEU) and other pixels p_j and p_{j+1} are for non-pivot embedding unit (NPEU). The Fig. 2.3 shows all possible combination of PEU and NPEU of (2×2) blocks. The Fig. 2.4 shows a numerical example of Chen's data hiding scheme. The difference d_i is calculated between the pixel pair of PEU to determine the number bits to be embedded within the selected pair as follows.

$$d_i = |p_i - p_{i+1}| \tag{2.16}$$

number of data bits that are to be embedded within the NPEU is same as the PEU. They proposed two reference tables RT_m and RT_n which can be generated through m-ary and n-ary number respectively, where m and n are two integers where $m < n$. A predefined threshold T

is used to decide which table is used during data embedding in PEU and NPEU.

$$R = \begin{cases} RT_m\ , & \text{if } d_i \leq T \\ RT_n\ , & \text{otherwise} \end{cases} \tag{2.17}$$

The embedding procedure is described as follows:

Input: Cover image $C\ (M \times N)$, Secret data D, Random seed k, Threshold T, Reference table RT_m and RT_n.

Output: Stego image $S\ (M \times N)$.

Step 1: Partitioned $C\ (M \times N)$ into (2×2) disjoined block B_i, $\{B_i | i = 1, 2, \ldots, \lfloor (M \times N)/2 \rfloor\}$ in raster scan order.

Step 2: PEU and NPEU can be determined through random seed k. Calculate the difference d_i using equation (2.16). Select d_i bits from the secret data D and convert it into decimal value v which is embedded through the reference table RT_m and RT_n that are decided by equation (2.17).

Step 3: Search the reference table of co-ordinate $(x,\ y)$, where $x = p_i$ and $y = p_{i+1}$ for matching the value of v and get new co-ordinate $(x^{'},\ y^{'})$ which must be nearest to $(x,\ y)$.

Step 4: If $d_i^{'} = d_i$, then $p_i^{'} = x^{'}$ and $p_{i+1}^{'} = y^{'}$, where $d_i^{'} = |x^{'} - y^{'}|$. Otherwise, search the reference table of co-ordinate $(x,\ y)$ until $d_i^{'} = d_i$.

Step 5: After data embedding in PEU, next embed data in NPEU. The pixel value of the NPEU are p_j and p_{j+1}. Select next d_i bits from secret message D and convert into decimal value v_i. Then perform **Step 3** by replacing p_i with p_j and p_{i+1} with p_{j+1}. The new pixel pair is $p_j^{'} = x^{'}$ and $p_{j+1}^{'} = y^{'}$.

Step 6: Select the next block for data embedding.

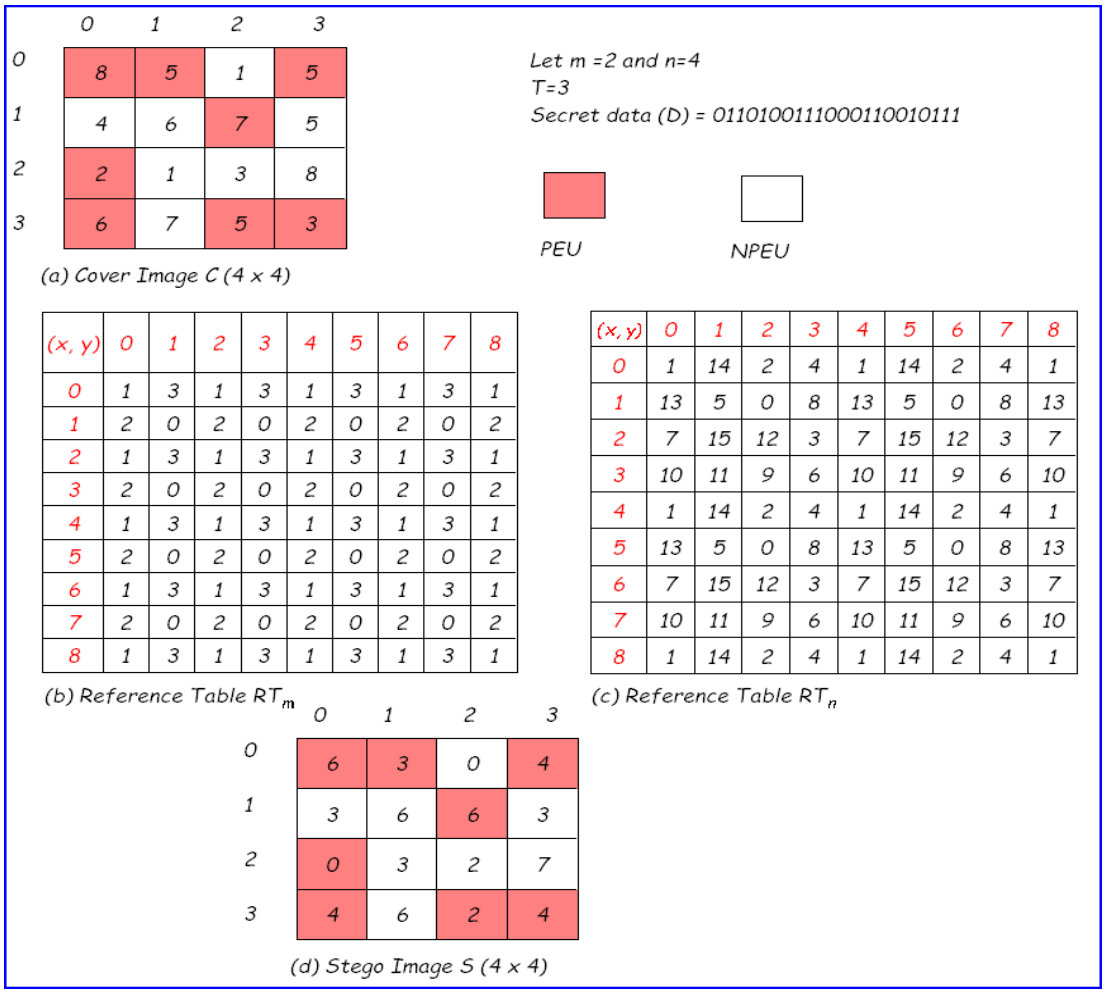
Step 7: Repeat **Step 2** to **Step 6** until all secret data bits are embedded.

Step 8: End.

The extraction procedure is described as follows:

Input: Stego image $S\ (M \times N)$, random seed k, threshold T, reference table RT_m and RT_n.

Output: Secret message D.



	0	1	2	3
0	8	5	1	5
1	4	6	7	5
2	2	1	3	8
3	6	7	5	3

(x, y)	0	1	2	3	4	5	6	7	8
0	1	3	1	3	1	3	1	3	1
1	2	0	2	0	2	0	2	0	2
2	1	3	1	3	1	3	1	3	1
3	2	0	2	0	2	0	2	0	2
4	1	3	1	3	1	3	1	3	1
5	2	0	2	0	2	0	2	0	2
6	1	3	1	3	1	3	1	3	1
7	2	0	2	0	2	0	2	0	2
8	1	3	1	3	1	3	1	3	1

(x, y)	0	1	2	3	4	5	6	7	8
0	1	14	2	4	1	14	2	4	1
1	13	5	0	8	13	5	0	8	13
2	7	15	12	3	7	15	12	3	7
3	10	11	9	6	10	11	9	6	10
4	1	14	2	4	1	14	2	4	1
5	13	5	0	8	13	5	0	8	13
6	7	15	12	3	7	15	12	3	7
7	10	11	9	6	10	11	9	6	10
8	1	14	2	4	1	14	2	4	1

	0	1	2	3
0	6	3	0	4
1	3	6	6	3
2	0	3	2	7
3	4	6	2	4

Figure 2.4: Numerical example of Chen's data hiding scheme

Step 1: Divide the stego image S into (2×2) non overlapping block B_i, $\{B_i | i = 1, 2, \ldots, \lfloor (M \times N)/2 \rfloor\}$ in raster scan order.

Step 2: PEU and NPEU are determined by the random seed k. Then calculate the difference $d_i^{'}$ using following equation

$$d_i^{'} = |p_i^{'} - p_{i+1}^{'}| \tag{2.18}$$

Retrieve data w from PEU using reference table RT_m and RT_n depending on equation (2.17). Then convert w into its binary form of $d_i^{'}$ bits. In the similar way, data can be retrieved from NPEU.

Step 3: Select the next block for data extraction.

Step 4: Continue **Step 2** and **Step 3** to extract entire data.

Step 5: End.

After embedding 54,383 bytes secret data in this scheme, the PSNR is 47.3 (dB).

2.3.5 Comparison of existing schemes

Table 2.3: Comparison of some existing PVD based data hiding methods

	Wu and Tsai [69]		Zhang and Wang [74]		Wang et al. [66]		Joo et al. [25]		Chen [11]	
Image	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
Lena	50,894	41.5	50,023	44.3	50,894	43.4	50,894	43.4	52,418	47.5
Baboon	57,028	37.0	53,568	40.4	57,043	40.2	57,043	39.2	57,474	47.1
Boat	52,320	39.6	50,926	42.4	52,490	41.1	52,490	41.0	53,718	47.4
House	52,418	40.0	51,003	42.7	52,572	42.4	52,572	41.5	54,913	47.2
Peppers	50,657	41.5	49,968	43.9	50,885	43.4	50,815	42.5	54,608	47.3
Average	52,204	40.3	50,803	43.1	52,275	42.6	52,275	41.9	54,383	47.3

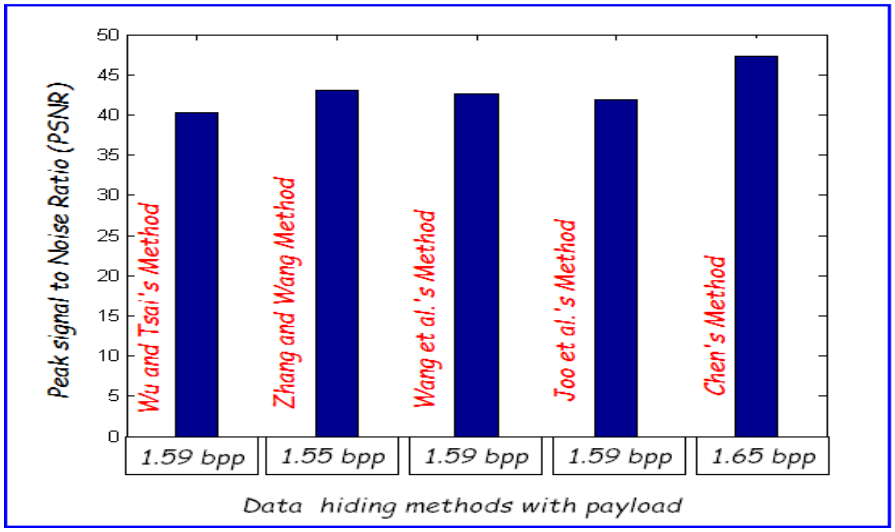


Figure 2.5: Comparison graph of existing PVD based data hiding methods

Table 2.3 shows the comparison of some existing PVD based data hiding methods. It is observed that average PSNR and capacity of Wu and Tsai's [69] method are 40.3 (dB) and 52,204 bytes respectively. In Zhang and Wang's [66] method, the PSNR is 43.1 (dB) which is 2.8 (dB) higher than Wu and Tsai's method but the capacity is 50,803 bytes which is 1401 bytes lower. In Wang et al.'s method average capacity is 52,275 bytes which is higher than Wu and Tsai's method and Zhang and Wang's method and the average PSNR is 42.6 (dB) which is higher than Wu and Tsai's method but lower than Zhang and Wang's method. Joo et al.'s method has same embedding capacity with Wang et al.'s method but the average PSNR is

lower. In Chen's scheme the PSNR is 47.3 (dB) and capacity is 54,383 bytes which is higher than all other PVD based data hiding methods. Fig. 2.5 shows the comparison graph among existing PVD based data hiding methods. But all these PVD based data hiding schemes are not reversible data hiding schemes. There is a scope to develop reversible data hiding scheme using PVD with higher quality and capacity.

2.4 Difference Expansion (DE)

Difference Expansion (DE) is one of the reversible data hiding schemes proposed by J. Tian [58]. This is a simple but efficient method. In this method, the difference between two neighboring pixels is expanded for data hiding. Although the average of two pixels remain same after data embedding the image quality is not enhanced due to the expansion of difference. In 2009, Lou et al. [44] proposed a multi-layer data hiding scheme based on DE to reduce the difference. Here, Tian's and Lou et al.'s DE based data hiding schemes are described.

2.4.1 J. Tian's scheme

Difference Expansion (DE) procedure for gray-scale images has been proposed by J. Tian [58]. In this method, a cover image is divided into non-overlapping blocks, which contains two consecutive pixels. At first, compute the difference between the pixel pair. Then the secret bit is embedded in the Least Significant Bit (LSB) of the difference value. That means each pair can hide only one bit secret data. In this method, data can't be embedded into those blocks where underflow or overflow problem may occur. For this reason location map is used to represent those blocks. The embedding procedure is described as follows:

Input: Cover image $C(M \times N)$, Secret data D, Location map LM.

Output: Stego image $S(M \times$N).

Step 1: Cover image C is divided into several blocks C_b, $\{C_b | b = 1, 2, \ldots, \lfloor (M \times N/2) \rfloor\}$ in raster scan order.

Step 2: Assume that the pixel pair is x_i and x_{i+1}. Then the difference d_i and average a_i is

calculated using following equation

$$\begin{cases} d_i = |x_i - x_{i+1}| \\ a_i = \lfloor \frac{x_i + x_{i+1}}{2} \rfloor \end{cases} \tag{2.19}$$

Step 3: Select one bit data D_i from D. Using the following equation that bit is embedded and the new difference $d_i^{'}$ is calculated as follows:

$$d_i^{'} = 2 \times d_i + D_i \tag{2.20}$$

Step 4: Then the modified pixel pair $x_i^{'}$ and $x_{i+1}^{'}$ are calculated as follows.

$$(x_i^{'}, x_{i+1}^{'}) = \begin{cases} (a_i + \lfloor \frac{d_i^{'}+1}{2} \rfloor, a_i - \lfloor \frac{d_i^{'}}{2} \rfloor), & \text{if } x_i \geq x_{i+1} \\ (a_i - \lfloor \frac{d_i^{'}}{2} \rfloor, a_i + \lfloor \frac{d_i^{'}+1}{2} \rfloor), & \text{if } x_i < x_{i+1} \end{cases} \tag{2.21}$$

Step 5: Select the next block for data embedding.

Step 6: **Step 2** to **Step 5** are repeated until all data is embedded.

Step 7: End.

The extraction procedure is as follows:

Input: Stego image S $(M \times N)$, Location map LM.

Output: Cover image C $(M \times N)$, Secret message D.

Step 1: Stego image S is divided into several blocks $S_b^{'}$, $\{S_b^{'} | b = 1, 2, \ldots, \lfloor (M \times N/2) \rfloor\}$in raster scan order.

Step 2: Then the difference $d_i^{'}$ and average $a_i^{'}$ are calculated using following equation

$$\begin{cases} d_i^{'} = |x_i^{'} - x_{i+1}^{'}| \\ a_i^{'} = \lfloor (x_i^{'} + x_{i+1}^{'})/2 \rfloor \end{cases} \tag{2.22}$$

Step 3: Extract secret data bit D_i from $d_i^{'}$ using $D_i = d_i^{'} \mod 2$.

Step 4: Calculate the original difference d_i using $d_i = \lfloor \frac{d_i^{'}}{2} \rfloor$.

Step 5: The original pixel pair is x_i and x_{i+1} is calculated using the following equation

$$(x_i, x_{i+1}) = \begin{cases} (a_i + \lfloor \frac{d_i+1}{2} \rfloor, a_i - \lfloor \frac{d_i}{2} \rfloor), & \text{if } x_i^{'} \geq x_{i+1}^{'} \\ (a_i - \lfloor \frac{d_i}{2} \rfloor, a_i + \lfloor \frac{d_i+1}{2} \rfloor), & \text{if } x_i^{'} < x_{i+1}^{'} \end{cases} \tag{2.23}$$

Step 6: Select the next block for data extraction.

Step 7: Repeat **Step 2** to **Step 6** until all the data is extracted.

Step 8: End.

Example 2.4.1 *Consider pixel pair* $(x_i, x_{i+1}) = (120, 127)$ *and secret message* D *is* 101. *The difference* $d = |120 - 127| = 7$ *and average* $a = \lfloor \frac{120+127}{2} \rfloor = 123$. *Select one bit data* D_i *from* D *that is* 1. *Now,* $d^{'} = (7 \times 2) + 1 = 15$. *Since* $120 < 127$, *then* $x_i^{'} = 123 - \lfloor \frac{15}{2} \rfloor = 116$ *and* $x_{i+1}^{'} = 123 + \lfloor \frac{15+1}{2} \rfloor = 131$.
In the data recovery phase, first calculate the difference $d^{'}$ *as* $d^{'} = |116 - 131| = 15$ *and average* $a^{'} = \lfloor \frac{116+131}{2} \rfloor = 123$. *Then extract one bit data* D_i *from* $d^{'}$ *as* $D_i = 15 \mod 2 = 1$
So, $d = \lfloor \frac{15}{2} \rfloor = 7$. *Since* $116 < 131$, *so* $x_i = 123 - \lfloor \frac{7}{2} \rfloor = 120$ *and* $x_{i+1} = 123 + \lfloor \frac{8}{2} \rfloor = 127$. ■

To overcome the underflow or overflow problem, $d_i^{'}$ must satisfy the following equation.

$$\begin{cases} |d_i^{'}| \leq 2 \times (255 - a), & \text{if } 128 \leq a \leq 255 \\ |d_i^{'}| \leq 2 \times a + 1), & \text{if } 0 \leq a \leq 127 \end{cases} \tag{2.24}$$

If $d_i^{'}$ satisfies the above equation, then d_i is called expandable otherwise d_i is called un-expandable. So, the number of embedded bits are same as the number of expandable difference. In this method, data is embedded in one layer. There is a scope to embed secret data in multi-layer but the quality of the stego image will be affected. To develop multi-layer data embedding scheme, Lou et al. [44] proposed a Reduce Difference Expansion (RDE) method.

2.4.2 Lou et al.'s scheme

To reduce the difference expansion and improve the visual quality, Lou et al. [44] proposed a scheme which uses a logarithm transformation function before expanding the difference. In RDE method, data bits are embedded into multi-layer which increase the hiding capacity. In the first layer, secret data bits are embedded horizontally in raster scan order. After that secret data bits are embedded vertically in raster scan order into second layer. Same procedure is applied until the last layer is processed.

The difference d_i and average a_i are calculated using the equation (2.19). Then apply the logarithm function to reduce the difference as follows.

$$d_i^{'} = \begin{cases} d_i, & \text{if } d_i < 2 \\ d_i - 2^{\lfloor \log_2 d_i \rfloor - 1}, & \text{if } d_i \geq 2 \end{cases} \tag{2.25}$$

Here a Location Map (LM) has been used to get the original difference. When the difference value is less than 2, the pixel values remain same and LM is set to 0. When the difference value is greater than or equals to 2, the pixel values are changed and LM is set to 1. So, the size of LM is same as the pixel pair. The LM will be send to the receiver during last layer embedding. Finally, the value of LM is set using the following equation.

$$LM = \begin{cases} 0, & \text{if } d_i^{'} = d_i \\ 1, & \text{if } d_i^{'} \neq d_i \end{cases} \tag{2.26}$$

Then embed one bit secret data D_i taken from secret data D as follows.

$$d_i^{''} = 2 \times d_i^{'} + D_i \tag{2.27}$$

Now, the modified pixel pair $x_i^{'}$ and $x_{i+1}^{'}$ are as follows.

$$(x_i^{'}, x_{i+1}^{'}) = \begin{cases} (a_i + \lfloor \frac{d_i^{''}+1}{2} \rfloor, a_i - \lfloor \frac{d_i^{''}}{2} \rfloor), & \text{if } x_i \geq x_{i+1} \\ (a_i - \lfloor \frac{d_i^{''}}{2} \rfloor, a_i + \lfloor \frac{d_i^{''}+1}{2} \rfloor), & \text{if } x_i < x_{i+1} \end{cases} \tag{2.28}$$

Now, the extraction procedure is describe below:

First calculate the difference $d_i^{''}$ and $a_i^{'}$ using the following equation.

$$\begin{cases} d_i^{''} = |x_i^{'} - x_{i+1}^{'}| \\ a_i^{'} = \lfloor (x_i^{'} + x_{i+1}^{'})/2 \rfloor \end{cases} \tag{2.29}$$

Then extract secret data bit D_i from $d_i^{''}$ using $D_i = d_i^{''} \mod 2$ and calculate the difference $d_i^{'}$ as $d_i^{'} = \lfloor \frac{d_i^{''}}{2} \rfloor$.

The original difference d_i is calculated using the following equation.

$$d_i = \begin{cases} d_i^{'} + 2^{\lfloor \log_2 d_i^{'} \rfloor - 1}, & \text{if } LM = 1 \\ d_i^{'} + 2^{\lfloor \log_2 d_i^{'} \rfloor}, & \text{if } LM = 0 \end{cases} \tag{2.30}$$

So, the original pixel pair x_i and x_{i+1} are calculated using

$$(x_i, x_{i+1}) = \begin{cases} (a_i + \lfloor \frac{d_i+1}{2} \rfloor, a_i - \lfloor \frac{d_i}{2} \rfloor), & \text{if } x_i^{'} \geq x_{i+1}^{'} \\ (a_i - \lfloor \frac{d_i}{2} \rfloor, a_i + \lfloor \frac{d_i+1}{2} \rfloor), & \text{if } x_i^{'} < x_{i+1}^{'} \end{cases} \tag{2.31}$$

Example 2.4.2 *Let us consider pixel pair* $x_i = 120$ *and* $x_{i+1} = 127$ *and secret message* D *is* 101. *The difference* $d = |120 - 127| = 7$ *and average* $a = \lfloor \frac{120+127}{2} \rfloor = 123$. $d_i^{'} = 7 - 2^{\lfloor \log_2 d_i \rfloor} - 1 = 7 - 2 = 5$ *and* LM *is set to* 1. *Select one bit data* D_i *from* D *that is* 1. *So,* $d^{''} = 5 \times 2 + 1 = 11$. *Since* $120 < 127$, $x_i^{'} = 123 - \lfloor \frac{11}{2} \rfloor = 118$ *and* $x_{i+1}^{'} = 123 + \lfloor \frac{11+1}{2} \rfloor = 129$.

In the data recovery phase, first calculate the difference $d^{''}$ *using* $d^{''} = |118 - 129| = 11$ *and average* $a^{'} = \lfloor \frac{118+129}{2} \rfloor = 123$. *Then extract one bit data* D_i *from* $d^{''}$ *as* $D_i = 11 \mod 2 = 1$. *So,* $d^{'} = \lfloor \frac{11}{2} \rfloor = 5$. *Since* LM *is* 1, $d = 5 + 2^{\lfloor \log_2 5 \rfloor - 1} = 5 + 2 = 7$. *Since* $118 < 129$, $x_i = 123 - \lfloor \frac{7}{2} \rfloor = 120$ *and* $x_{i+1} = 123 + \lfloor \frac{8}{2} \rfloor = 127$. ■

2.5 Exploiting Modification Direction (EMD)

Exploiting Modification Direction (EMD) is a data hiding method where a group of n pixels are used as an embedding unit and embed secret digit in $(2n + 1)$ notational system where $n \geq 2$. In 2006, Zhang and Wang [72] proposed EMD method where one pixel is increased or decreased by 1 during data embedding. In this method, a function $f()$ is used for embedding and extracting the secret message. For $n = 2$, there are four modification direction. In 2011, Kieu and Chang [31] proposed fully exploiting modification direction method by improving the $f()$ function. This method achieves high embedding capacity and good quality stego image. In 2014, Qin et al. [52] proposed a reversible EMD method using two steganographic images. This method also achieve high embedding capacity and good quality stego images. Here, three EMD based data hiding schemes proposed by Zhang and Wang, Kieu and Chang and Qin et al.'s are discussed.

2.5.1 Zhang and Wang's scheme

Zhang and Wang's [72] proposed a data hiding scheme based on Exploiting Modification Direction (EMD). Consider a cover image C of size ($M \times N$), where M is the height and N is the width of the image. First the pixel values of the cover image are randomly permuted using a secret key. Then the pixels are divided into several blocks of n pixels ($x_1, x_2, \ldots, x_n$), where $n \geq 2$. In this method, n cover pixels can hide secret digits in a $(2n + 1)$-ary notational system. Before embedding, the secret message is divided into several parts of k bits using the following

equation.

$$k = \lfloor m.\log_2(2n+1) \rfloor \tag{2.32}$$

where m is the decimal value of k bits in a $(2n+1)$-ary notational system. There are $2n$ possible ways of modification which may happen for each block of n pixels. During data embedding only one pixel of each block is incremented or decremented by 1.

For data embedding, first calculate the function $f()$ using following equation.

$$f(x_1, x_2, \ldots, x_n) = (x_1 \times 1 + x_2 \times 2 + \ldots x_n \times n) \bmod (2n+1) \tag{2.33}$$

When the secret digits $d = f$, the pixel value remain the same which means no modification is required. When $d \neq f$, then calculate s as $s = (d - f) \mod (2n+1)$. If $s < n$, then the pixel value of x_s is increased by 1, otherwise, the pixel value of x_{2n+1-s} is decreased by 1. In the extraction process, the receiver can easily extract secret digit by calculating the function $f()$ from stego pixel block.

Example 2.5.1 *Assume two pixels x_1 and x_2 are 113 and 120 respectively for $n = 2$. Secret message $D = (1011)_2$. Let $m = 1$. So, $k = \lfloor 1.\log_2(2 \times 2 + 1) \rfloor = 2$. Select two bits from secret message that is $(10)_2$ and convert it into 5-ary notational number that is $d = (2)_5$. Now, calculate f value using equation (2.33) as $f(113, 120) = (113 \times 1 + 120 \times 2) \mod 5 = 3$. Since $2 \neq 3$, calculate s as $s = (2-3) \bmod 5 = 4$. Here, $4 \geq 2$, so the pixel value of $x_{5-4} = x_1$ that is 113 is decreased by 1 and we get 112. So, new pixel values are $x_1' = 112$ and $x_2' = 120$. To extract data, calculate f value using equation (2.33). So, $f(112, 120) = (112 \times 1 + 120 \times 2) \mod 5 = 2$. Then convert f value into 2 bits binary form that is 10.* ■

The PSNR of Zhang and Wang's method is more than 52 (dB) and the embedding capacity is $(\log_2(2n+1))/n$ (bpp). If $n = 2$, then embedding capacity is 1 (bpp). To improve the data embedding capacity, Kieu and Chang [31] proposed a new EMD method which is described below.

2.5.2 Kieu and Chang's scheme

The cover image C is partitioned into non overlapping blocks B_i, $\{B_i | i = 1, 2, \ldots \lfloor (M \times N/2) \rfloor\}$ using a Pseudo Random Number Generator (PRNG) with a seed value k. Each block

contains two consecutive pixels say x_i and x_{i+1}. Depending on a threshold $T \geq 2$, compute the number of secret bits t which can be embedded in each block. Then calculate the modified $f()$ function using the following equation.

$$f(x_i, x_{i+1}) = (x_i \times (T-1) + x_{i+1} \times T) \bmod T^2 \tag{2.34}$$

The above function generates a mapping matrix H of size $(M \times N)$. The element of x_i^{th} row and x_{i+1}^{th} column in H matrix is $f(x_i, x_{i+1})$, that means $H(x_i, x_{i+1})$ = $f(x_i, x_{i+1})$. Fig. 2.6 shows the H matrix where $T = 2$. They used a searching element r, where $r = \lfloor T/2 \rfloor$. r is used to find out the searching area in H matrix for data embedding. The searching area is always $(2 \times r + 1) \times (2 \times r + 1)$ square matrix in H matrix where the element $H(x_i, x_{i+1})$ is located at the center. The square matrix is defined as

$$W_{(2\times r+1)\times(2\times r+1)}(T, (x_i, x_{i+1}), r) = \{H(x_i - r + u, x_{i+1} - r + v) \,|0 \leq u \leq 2 \times r,$$
$$0 \leq v \leq 2 \times r, u \neq v\} \tag{2.35}$$

In some cases, two or more elements are found from searching square matrix that is $H(x_a, y_a)$, $H(x_b, y_b)$ and $H(x_c, y_c)$. In that situation, the minimum value is taken to improve the stego image quality using the following formula.

$$P_{min} = min_{j=a,b,c}\{|x_i - x_j| + |x_{i+1} - y_j|\} \tag{2.36}$$

Finally, the stego pixel pair is $(y_i, y_{i+1}) = (x_j, y_j)$.
Now the data embedding procedure is described below:

Input: Cover image C ($M \times N$), Secret message D, Threshold T $(2 \leq T \leq 23)$, Seed k.

Output: Stego image S ($M \times N$)

Step 1: Cover image C is divided into blocks of two consecutive pixels using seed k.

Step 2: Generate matrix H of size ($M \times N$) using the equation (2.35).

Step 3: Calculate the number of bits t and searching element r as follows.

$$\begin{cases} t = \lfloor \log_2 T^2 \rfloor \\ r = \lfloor T/2 \rfloor \end{cases} \tag{2.37}$$

Step 4: Select the first block.

Step 5: Select t bits from secret message D and convert it into decimal form v.

Step 6: If $v = H(x_i, x_{i+1})$, then stego pixel pair is same as the original pixel pair that means $(y_i, y_{i+1}) = (x_i, x_{i+1})$. Otherwise, search the square matrix to find the element $H(g, h) = v$. Then the stego pixel pair is $(y_i, y_{i+1}) = (g, h)$.

Step 7: Select the next block for data embedding.

Step 8: **Step 5** to **Step 7** are repeated until all data is embedded.

Step 9: End.

The extraction procedure is as follows:

Input: Stego image S ($M \times N$), Threshold T, Seed k.

Output: Secret message D

Step 1: Stego image S is divided into blocks of two consecutive pixels using seed k.

Step 2: Calculate the number of bits t which has been extracted from each pair.

$$t = \lfloor \log_2 T^2 \rfloor \tag{2.38}$$

Step 3: Select first block.

Step 4: Extract secret data v by calculating $f()$ function using following equation.

$$f(y_i, y_{i+1}) = (y_i \times (T-1) + y_{i+1} \times T) \bmod T^2 \tag{2.39}$$

Step 5: Convert v into its binary form of t bits.

Step 6: Select the next block for data extraction.

Step 7: **Step 4** to **Step 6** are repeated until all data is extracted.

Step 8: End.

i, j	0	1	2	3	4	...	...	253	254	255
0	0	2	0	2	0	...	...	2	0	2
1	1	3	0	3	1	...	...	3	1	3
2	2	0	2	0	2	...	...	0	2	0
3	3	1	3	1	3	...	...	1	3	1
4	0	2	0	2	0	...	...	...	...	..
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
253	1	3	1	3	1	...	...	3	1	3
254	2	0	2	0	2	...	...	0	2	0
255	3	1	3	1	3	...	...	3	1	3

Figure 2.6: Mapping matrix for T = 2 of Kieu and Chang's data hiding scheme

Example 2.5.2 *Let the pixel pair* $(x_i, x_{i+1}) = (2, 3)$*, Threshold value* $T = 2$ *and secret message* $D = 1110$*. Calculate the number of bits* t *which are to be embedded. So,* $t = \lfloor \log_2 2^2 \rfloor = 2$ *and* $r = \lfloor 2/2 \rfloor = 1$*. Select two bits from* D *that is* $(11)_2$ *and convert it into decimal form* v *that means* $(3)_{10}$*. Now, embed* v *within the pixel pair (*$x_i, x_{i+1}) = (2, 3)$*. The Fig.* 2.6 *shows that* $H[2, 3] = 0$ *that is* $H[2, 3] \neq v$*. So search the square matrix* $W_{3\times3}(2, (2, 3), 1)$ *to get an element equal to* v*. There are three elements found such as* $H[1, 3]$*,* $H[3, 2]$ *and* $H[3, 4]$*. According to equation* (2.36)*,* $P_{min} = min((|2 - 1| + |3 - 3|), (|2 - 3| + |3 - 2|), (|2 - 3| + |3 - 4|)) = min(1, 2, 2) = 1$*. So,* $H[1, 3]$ *is taken and the stego pixel pair becomes* (y_i, y_{i+1}) = $(1, 3)$*.*
The recovery process is stated below:
Consider the stego pixel pair is $(y_i, y_{i+1}) = (1, 3)$ *and* $T = 2$*. Calculate the number of bits* t *which can be extracted. So,* $t = \lfloor \log_2 2^2 \rfloor = 2$*. To extract secret data calculate* $f()$ *value using equation* (2.39)*. So,* $f(1, 3) = (1 \times (2 - 1) + 3 \times 2) \mod 4 = 3$*. Then convert* $f()$ *value into* 2 *bits binary form that is* 11*.* ■

The embedding capacity of Kieu and Chang's method is $\lfloor \log_2 T_2 \rfloor / 2$ (bpp). If $T = 2$ then embedding capacity is 1 (bpp) and PSNR is 52.39 (dB).

Although the above two methods give better image quality and higher embedding capacity, those methods are not reversible that means the cover image can't be recovered after data extraction. To solve this irreversibility, Qin et al. [52] proposed reversible EMD method using two steganographic images. In the next section, Qin et al.'s scheme is illustrated.

2.5.3 Qin et al.'s scheme

Consider cover image C ($M \times N$) for data embedding and generate two steganographic images $S1(M \times N)$ and $S2(M \times N)$ after data embedding which are visually similar and are same in size as the cover image. During data embedding, all pixel values of the first stego image increase or decrease by 1 like Zhang and Wang's method. Depending on the cover image and the first stego image, all pixel values of the second stego image are modified by less than or equal to $(2n + 1)$, where n is the number of pixel. On the receiver side, the secret message can be extracted from two stego images and the cover image can be easily recovered from these images using some specific rule. First the cover image C is preprocessed called C_p to overcome overflow or underflow problem. So, the pixel values of the cover image C belongs to $[0, 2n]$ or $[255 - 2n, 255]$ are modified as $[2n + 1]$ or $[255 - 2n - 1]$ respectively. Store the original pixel value and the co-ordinate of modified pixel to achieve reversibility. An arithmetic coding is used to compress this information to generate extra information. These extra information are embedded with the secret message. Now the embedding procedure is described in two phases.

Input: Preprocessed cover image $C_p(M \times N)$, Secret data D_1 and D_2 .

Output: Stego image S1 ($M \times N$) and S2 ($M \times N$).

Phase 1: Embed secret message D_1 to generate the stego image S1.

Step 1: At first, the preprocessed cover image C_p is divided into non overlapping blocks B_j, $\{B_j | j = 1, 2, \ldots, \lfloor (M \times N/2) \rfloor\}$ in raster scan order where each block contains two consecutive pixels. Let two pixels are x_i and x_{i+1}.

Step 2: Select the first block.

Step 3: Calculate the $f()$ value using following equation

$$f(x_i, x_{i+1}) = (x_i \times 1 + x_{i+1} \times 2) \mod (2n + 1) \tag{2.40}$$

Step 4: Embed the secret digit $d_1 \in D_1$ within the pixel pair. If d_1 is equal to f then the pixel pair remain same. So, $S1_i = x_i$ and $S1_{i+1} = x_{i+1}$. If $d_1 \neq f$ then calculate s as s = $(d_1 - f) \bmod (2n + 1)$. If $s < n$, then the pixel values of $S1_i$ and $S1_{i+1}$ are calculated using the following equation

$$\begin{cases} S1_{i+s-1} = x_{i+s-1} + 1 \\ S1_{i+n-s} = x_{i+n-s} \end{cases} \tag{2.41}$$

else

$$\begin{cases} S1_{i+2n-s} = x_{i+2n-s} - 1 \\ S1_{i+s-n-1} = x_{i+s-n-1} \end{cases} \tag{2.42}$$

Step 5: Select the next block for data embedding.

Step 6: Repeat **Step 3** to **Step 5** until all data bits are embedded.

Step 7: End.

Phase 2: Embed secret data D_2 to generate the stego image S2.

Step 1: There are three cases arises between (x_i, x_{i+1}) and $(S2_i, S2_{i+1})$. Depending on these cases the secret message d_2 is embedded. Three cases are mentioned below.

$$\begin{cases} \text{Case } 1 : x_i = S2_i \text{ and } x_{i+1} = S2_{i+1} \\ \text{Case } 2 : x_i = S2_i \text{ and } x_{i+1} \neq S2_{i+1} \\ \text{Case } 3 : x_i \neq S2_i \text{ and } x_{i+1} = S2_{i+1} \end{cases} \tag{2.43}$$

If Case 1 is satisfied then calculate the $f()$ value using equation (2.40). If $d_2 \in D_2$ is equal to $f()$ then the pixel pair remain same that is $S2_i = x_i$ and $S2_{i+1} = x_{i+1}$. If $d_2 \neq f()$ then calculate s as s = ($d_2 - f$) $\bmod (2n+1)$. If $s < n$ then the pixel values of $S2_i$ and $S2_{i+1}$ are calculated using the following equation.

$$\begin{cases} S2_{i+s-1} = x_{i+s-1} + 1 \\ S2_{i+n-s} = x_{i+n-s} \end{cases} \tag{2.44}$$

else

$$\begin{cases} S2_{i+2n-s} = x_{i+2n-s} + 1 \\ S2_{i+s-n-1} = x_{i+s-n-1} \end{cases} \tag{2.45}$$

If Case 2 is satisfied, then find the value of p such that $f()$ value is equal to d_2.

$$d_2 = f[x_i, x_{i+1} - p \times sign(S1_{i+1} - x_{i+1})] \tag{2.46}$$

where, p is an integer, $p \in \{1, 2, \ldots, 5\}$ and $sign()$ return 1 or -1 depending on the value of $(S1_{i+1} - x_{i+1})$. Then the pixel pair $(S2_i, S2_{i+1})$ is computed as

$$\begin{cases} S2_i = x_i \\ S2_{i+1} = [x_{i+1} - p \times sign(S1_{i+1} - x_{i+1})] \end{cases} \tag{2.47}$$

If Case 3 is satisfied, then find the value of p such that $f()$ value is equal to d_2.

$$d_2 = f[x_i - p \times sign(S1_i - x_i), x_{i+1}] \tag{2.48}$$

Then the pixel pair $(S2_i, S2_{i+1})$ is computed as

$$\begin{cases} S2_i = [x_i - p \times sign(S1_i - x_i)] \\ S2_{i+1} = x_{i+1} \end{cases} \tag{2.49}$$

Step 2: Repeat **Step 1** until all data is embedded.

Step 3: End.

The data and cover image recovery process is as follows:

Input: Stego image $S1$ $(M \times N)$ and $S2$ $(M \times N)$.

Output: Preprocessed cover image C_p $(M \times N)$, Secret data D_1 and D_2.

Step 1: Divide the stego images $S1$ and $S2$ into blocks $B1_j$ and $B2_j$, $\{j = 1, 2, \ldots, \lfloor (M \times N/2) \rfloor\}$ which contain two consecutive pixels.

Step 2: Secret data D_1 and D_2 are extracted from all blocks of the stego $S1$ and $S2$ by calculating the $f()$ value.

Step 3: Select the blocks $B1_1$ and $B2_1$ from $S1$ and $S2$ respectively.

Step 4: Calculate t using following equation.

$$t = |(S1_i - S2_i) + (S1_{i+1} - S2_{i+1})| \tag{2.50}$$

If $t = 0$ or 1 then the original pixel pair $(x_i, x_{i+1}) = (S1_i, S1_{i+1})$.

If $t > 1$ then there are two cases that may arise.

Case 1: If $S1_i \neq S2_i$ and $S1_{i+1} = S2_{i+1}$ then the pixel pair (x_i, x_{i+1}) is computed using the following equation

$$\begin{cases} x_i = S1_i - sign(S1_i - S2_i) \\ x_{i+1} = S1_{i+1} \end{cases} \tag{2.51}$$

Case 2: If $S1_i = S2_i$ and $S1_{i+1} \neq S2_{i+1}$ then the pixel pair (x_i, x_{i+1}) is computed using the following equation.

$$\begin{cases} x_i = S1_i \\ x_{i+1} = S1_{i+1} - sign(S1_{i+1} - S2_{i+1}) \end{cases} \tag{2.52}$$

Step 5: Select the next block from $S1$ and $S2$.

Step 6: Repeat **Step 4** to **Step 5** until cover image C_p is recovered.

Step 7: End.

Then preprocessed image C_p needs to be post-processed to generate the original cover image C with the help of some extra information which has been embedded with the secret data.

Example 2.5.3 *Let the pixel pair* $(x_i, x_{i+1}) = (10, 12)$ *and secret data* $D_1 = (11)_2$ *and* $D_2 = (01)_2$. *So,* $d_1 = (3)_5$ *and* $d_2 = (1)_5$.

Phase 1: *First calculate* f *value using equation* (2.40). *So,* $f(10, 12) = (10 \times 1 + 12 \times 2) \mod (2 \times 2 + 1) = 4$. *Since* $m_1 = 3 \neq 4$, *calculate* s *as* $s = (3 - 4) \mod 5 = 4$. *Here* $s > n$ *that is* $4 > 2$, *so according to the equation* (2.42), $S1_{i+4-4} = x_{i+4-4} - 1$ *that is* $S1_i = x_i - 1 = 10 - 1 = 9$ *and* $S1_{i+4-2-1} = x_{i+4-2-1}$ *that is* $S1_{i+1} = x_{i+1} = 12$.

Phase 2: *After completion of phase* 1 *data embedding, data is embedded through phase* 2. *Here the* $x_i = 10$, $x_{i+1} = 12$ *and* $S1_i = 9$, $S1_{i+1} = 12$. *Since* $10 \neq 9$ *and* $12 = 12$, *then case 3 is satisfied. Calculate* p *such that* $f()$ *value is equal to* d_2. $d_2 = f[10 - p \times sign(9 - 10), 12] = f(10 + p, 12) = f(10 + 2, 12) = (12 \times 1 + 12 \times 2) \mod 5 = 1$. *So,* $p = 2$. *Now according to the equation* (2.49), $S2_i = [10 - 2 \times sign(9 - 10)] = 12$ *and* $S2_{i+1} = 12$.

The recovery process is as follows:

The pixel pair $(S1_i, S1_{i+1}) = (9, 12)$ *and* $(S2_i, S2_{i+1}) = (12, 12)$. *Now, calculate* t *using equation* (2.50). *So,* $t = |(9 - 12) + (12 - 12)| = 3$. *Here* $3 > 1$, $9 \neq 12$ *and* $12 = 12$. *So, case* 1 *is satisfied. According to the equation* (2.51), *the pixel pair* $x_i = 9 - sign(9 - 12) = 9 - (-1) = 10$ *and* $x_{i+1} = 12$ *has been extracted.* ■

The PSNR of two stego images S1 and S2 with respect to original image are 52 (dB) and 41 (dB) respectively and the embedding capacity is 1.16 (bpp). To improve the embedding capacity, Shen and Huang [54] proposed a new data hiding method using PVD and EMD but this technique is not reversible.

2.5.4 Shen and Huang's scheme

The cover image C ($M \times N$) is divided into non overlapping blocks B_i, $\{B_i | i = 1, 2, \ldots \lfloor (M \times N/2) \rfloor\}$ in raster scan order where the block contains two consecutive pixels. Consider two

pixels are x_i and x_{i+1}. The difference between two pixels x_i and x_{i+1} is calculated using the equation.

$$d_i = |x_i - x_{i+1}| \tag{2.53}$$

In this approach, a range table R has been proposed with n contiguous sub-range R_m, $\{R_m|m = 1, 2, \ldots, n\}$. Each sub-range R_m has a lower and a upper bound, namely lb and ub respectively. So, $R_m \in [lb, ub]$. The width w_b is obtained depending on the range table R_m where the difference d_i is mapped, using the equation.

$$w_b = ub - lb + 1 \tag{2.54}$$

Then calculate parameter T using following equation.

$$T_i = \lfloor \log_2(w_b) \rfloor \tag{2.55}$$

The number of bits to be embedded, denoted by t is calculated using the equation

$$t_i = \lfloor \log_2(T_i^2) \rfloor \tag{2.56}$$

The embedding procedure is describe below:

Input: Cover image C $(M \times N)$, Secret data D, Range table R.

Output: Stego image S $(M \times N)$.

Step 1: Divide the cover image C into blocks which contain two consecutive pixels.

Step 2: Select the first block.

Step 3: Then compute the difference d_i and width w_b using equation (2.53) and equation (2.54) respectively.

Step 4: Calculate T_i and t_i using equation (2.55) and equation (2.56) respectively.

Step 5: Select t_i bits from secret data D and convert into its decimal value v_i of T_i^2-ary notational system.

Step 6: Now calculate $f()$ value using the following equation.

$$f(x_i, x_{i+1}) = (x_i \times (T - 1) + x_{i+1} \times T) \bmod T^2 \tag{2.57}$$

Step 7: If $v_i = f(x_i, x_{i+1})$, then the pixel value remains the same that means no modification is needed. So, $(x_i^{'}, x_{i+1}^{'}) = (x_i, x_{i+1})$. else if $v_i > f(x_i, x_{i+1})$, then the modified pixel pair is obtained as

$$\begin{cases} x_i^{'} = x_i - [(v_i - f(x_i, x_{i+1})) \bmod T_i] \\ x_{i+1}^{'} = x_{i+1} + \lfloor \frac{v_i - f(x_i, x_{i+1})}{T_i} \rfloor + [(v_i - f(x_i, x_{i+1})) \bmod T_i] \end{cases} \tag{2.58}$$

else $v_i < f(x_i, x_{i+1})$, then the modified pixel pair is obtained as

$$\begin{cases} x_i^{'} = x_i - [(f(x_i, x_{i+1}) - v_i) \bmod T_i] \\ x_{i+1}^{'} = x_{i+1} - \lfloor \frac{f(x_i, x_{i+1}) - v_i}{T_i} \rfloor - [(f(x_i, x_{i+1}) - v_i) \bmod T_i] \end{cases} \tag{2.59}$$

Step 8: Select the next block.

Step 9: Repeat **Step 3** to **Step 8** until all data is embedded.

Step 10: End.

Select pixel pairs which are defined below when d_i and $d_i^{'}$ does not belongs to the same sub-range of the range table R, where $d_i^{'} = |x_i^{'} - x_{i+1}^{'}|$. Now, the pixel pairs will be $\{(x_i^{'} - 3T_i, x_{i+1}^{'} - 3), (x_i^{'} - 2T_i, x_{i+1}^{'} - 2), (x_i^{'} - T_i, x_{i+1}^{'} - 1), (x_i^{'} + T_i, x_{i+1}^{'} + 1), (x_i^{'} + 2T_i, x_{i+1}^{'} + 2), (x_i^{'} + 3T_i, x_{i+1}^{'} + 3)\}$ So, the selected pixel pair is called $(x_i^{''}, x_{i+1}^{''})$ such that d_i and $d_i^{'}$ belong to the same sub-range. Overflow or underflow problem may occur when the pixel pair $(x_i^{''}, x_{i+1}^{''})$ does not belongs to $[0, 255]$. To solve this problem, determine the new pixel pair (x_i^*, x_{i+1}^*) which is closest to (x_i, x_{i+1}). This can be computed using the following optimization function.

Minimize: $(x_i - x)^2 + (x_{i+1} - y)^2$;

Subject to: $f(x, y) = v_i; Div(d_i) = Div(d_i^*)$

where $d_i = |x_i - x_{i+1}|$, $d_i^* = |x - y|$ and $Div(d_i)$ is the division where d_i belongs to.

The recovery process is described below:

Lower bound (lb)	0	8	16	32	64	128
Upper bound (ub)	7	15	31	63	127	255
Number of bits to be Embedded	3	3	4	5	6	7

Figure 2.7: Range table of Shen and Huang's data hiding scheme

Input: Stego image S $(M \times N)$, Range table R.

Output: Secret data D.

Step 1: Divide the stego image S into blocks which contain two consecutive pixels.

Step 2: Select first block. Assume that the stego pixel pair is $(x_i^{'}, x_{i+1}^{'})$.

Step 3: Then compute the difference $d_i^{'}$ as $d_i^{'} = |x_i^{'} - x_{i+1}^{'}|$.

Step 4: Calculate w_b of sub range R_m where the difference $d_i^{'}$ is belongs.

Step 5: Calculate T_i and t_i using equation (2.55) and equation (2.56) respectively.

Step 6: Extract secret message v_i by calculating f value using the following equation.

$$v_i = f(x_i^i, x_{i+1}^{'}) = (x_i^{'} \times (T-1) + x_{i+1}^{'} \times T) \bmod T^2 \tag{2.60}$$

Step 7: Convert v_i into its binary form of t_i bits.

Step 8: Select the next block.

Step 9: Repeat **Step 3** to **Step 8** until all data is extracted.

Example 2.5.4 *Consider two pixels* $x_i = 122$ *and* $x_{i+1} = 135$ *and the secret data* $D = (110)_2$. *First calculate the difference* $d = |122 - 135| = 13$. *Here* d *belongs to the sub range* $[8, 15]$ *of the range table* R *shown in Fig.* 2.7. *Then compute* $w = (15 - 8 + 1) = 8$. *Calculate* T *as* $T = \lfloor \log_2(8) \rfloor = 3$. *The number of bits* $t = \lfloor \log_2 3^2 \rfloor = 3$, *that means* 3 *bits are embedded within this pair of pixel. Extract* 3 *bits data from* D *that is* $(110)_2$ *and its decimal value* $v = (6)_{10}$. *Then calculate* f *value using equation* (2.57). *So,* $f(122, 135) = (122 \times (3-1) + 135 \times 3) \bmod 3^2 = 1$. *Since* $6 > 1$, *according to equation* (2.58) $x_i^{'} = 122 - [(6-1) \bmod 3] = 120$ *and* $x_{i+1}^{'} = 135 + \lfloor \frac{6-1}{3} \rfloor + [(6-1) \bmod 3] = 135 + 1 + 2 = 138$.

Here, $d = 13$ *and* $d^{'} = |120 - 138| = 18$ *does not belong to the same sub-range. Choose any one pixel pair out of six such that* d *and* $d^{'}$ *belong to the same sub-range that is* $[8, 15]$. *So, the pixel pair is* $(120 + 2 \times 3, 138 + 2) = (126, 140)$.

On the receiver side first calculate $d^{'} = |126 - 140| = 14$. *So,* $d^{'}$ *belongs to the sub range* $[8, 15]$ *of the range table* R. *Then compute* $w = (15 - 8 + 1) = 8$. *Calculate* T *as* $T = \lfloor \log_2(8) \rfloor = 3$. *The number of bits* $t = \lfloor \log_2 3^2 \rfloor = 3$. *To extract* v *calculate* f *function using equation* (2.60)

as $v = f(126, 140) = (126 \times (3-1) + 140 \times 3) \mod 3^2 = 6$. *Convert* v *into binary form of* 3 *bits that is* $(110)_2$. ■

After embedding maximum amount of secret data within cover image the payload is around 1.56 (bpp) and the PSNR of stego image of size (512×512) is around 41.67 (dB).

Table 2.4: Comparison of existing EMD based data hiding methods

	Zhang and Wang [72]		Kieu and Chang [31]		Qin et al. [52]		Shen and Huang [54]	
Image	bpp	PSNR	bpp	PSNR	bpp	PSNR	bpp	PSNR
Lena	1	52.09	1.5	49.88	1.16	46.72	1.53	42.46
Baboon	1	52.10	1.5	49.89	1.16	46.73	1.69	38.88
F16	1	52.12	1.5	49.89	1.16	46.72	1.59	41.89
Barbara	1	52.11	1.5	49.89	1.16	46.73	1.62	40.15
Boat	1	52.10	1.5	49.90	1.16	46.73	1.55	41.60
Goldhill	1	52.11	1.5	49.89	1.16	46.72	1.54	41.40
Peppers	1	52.11	1.5	49.89	1.16	46.38	1.52	43.49
Bridge	1	52.04	1.5	49.86	1.16	46.50	1.58	41.72
Average	1	52.1	1.5	49.89	1.16	46.72	1.56	41.67

2.5.5 Comparison of existing schemes

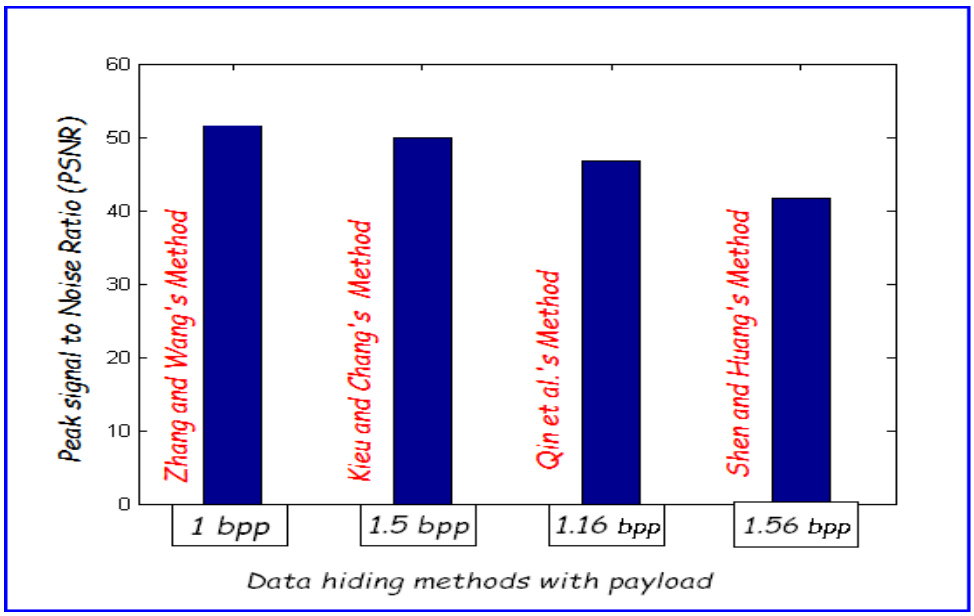


Figure 2.8: Comparison graph of existing EMD based data hiding methods

Table 2.4 shows the comparison of existing EMD methods. It observed that the capacity of Zhang and Wang's method is 1 (bpp) and the average PSNR is 52.1 (dB) when $n = 2$. In Kieu

and Chang's method, the capacity is 1.5 (bpp) and PSNR is 49.89 (dB) when $s = 3$. Qin et al.'s method has lower PSNR 46.72 (dB) than Kieu and Chang's method when capacity is 1.16 (bpp). Shen and Huang's method has highest capacity 1.56 (bpp) than other three data hiding methods and average PSNR is 41.67 (dB). Fig. 2.8 shows the comparison graph of existing EMD based data hiding methods.

2.6 Weighted Matrix based data hiding

Westfeld [67] introduced matrix based data embedding technique using Hamming code. It embed r-bits secret data through modification one bit of $(2^r - 1)$ least significant bit in the host image. The embedding efficiency increases with the increase of r while the payload decreases contrarily. In order to increase the embedding efficiency and payload simultaneously, an extended F5 algorithm is proposed by Fan et al. [13]. They come up with a brand new idea to realize the aim through adding n-layer extension and modifying the form of original hash function. A secure data hiding scheme for binary image using a key matrix K and a weighted matrix W has been proposed by Tseng et al. [64] which can hide only 2 bits secret data within a (3×3) pixel block. Fan et al. [14] proposed an improved efficient data hiding scheme using weighted matrix for gray scale image which can hide four bits secret data within a (3×3) block. Both the aforesaid matrix based data hiding schemes, only one modular sum of entry-wise-multiplication operation ($\otimes$) between weighted matrix W and (3×3) pixel block of the original image has been performed. Using only one embedding operation r-bits secret data can be embedded within a block by modifying only one bit. High data embedding capacity and reversibility is still an important research issues in data hiding through weighted matrix. Here Tseng's and Fan's weighted matrix based data hiding schemes are discussed.

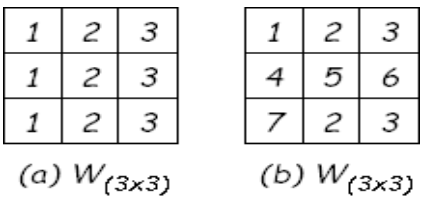


Figure 2.9: Example of Weighted Matrix for (a) r = 2 and (b) r = 3

2.6.1 Tseng et al.'s scheme

Tseng et al. [64] proposed a data hiding scheme using binary image (BI). They partitioned BI into non-overlapping blocks (B_i) of size ($m \times n$). Number of bits (r) to be embedded within each block are calculated using following equation.

$$2^r - 1 \leq (m \times n) \tag{2.61}$$

A binary key matrix (K) and integer weighted matrix (W) of the same size as B_i are to be shared by sender to receiver before data communication.

The criterion of preferring W is that each element of matrix is arbitrarily allotted a value from the combination $(0, 1, 2, \ldots, 2^{r-1} + 1)$ and each element appears at least once in W, where r denotes the number of secret bits which will be embedded within each block of cover image B_i. Fig. 2.9 shows an example of weighted matrix W. The main working formula of Tseng's scheme is:

$$SUM((B_i \oplus K) \otimes W) \bmod 2^r = (b_1 b_2 \ldots b_r)_2 \tag{2.62}$$

where $\oplus$ denotes bitwise exclusive-OR operation between image block and key matrix, and $\otimes$ denotes entry-wise-multiplication operation between result of $\oplus$ operation and weighted matrix. The SUM function return the summation value of all the elements of matrix $((B_i \oplus K) \otimes W)_{m \times n}$. $(b_1 b_2 \ldots b_r)_2$ are the result of $\bmod$ value between summation result and 2^r. The main goal is to modify B_i so that the modulo result $(b_1 b_2 \ldots b_r)_2$ will be a secret r bits data. If modulo result already gives the value equals to secret data then no modification will be required in B_i. The numerical example of Tseng's data embedding and data extraction procedure is shown in Fig. 2.10. According to the principal, the result of $SUM((B_i \oplus K) \otimes W) \bmod 2^r$ equals to data bits. In Fig. 2.10 it is shown that the modulo result is $(4)_2$ which is not equal to data $(5)_2$. So, modification of $B_i \; to \; B_i'$ is required during data embedding. The modification will be done at the d-th location of B_i that has been calculated using equation (2.63). Here, $d = 1$, so complement the pixel value at location $B_i[x, y],$ if $W[x, y] = d$ (here $W[1, 1] = 1$, so $B_i[1, 1]$ is flip to 0 and gets $B_i'[1, 1] = 0$).

$$d = (b_1 b_2 \ldots b_r)_2 \; - \; SUM[(B_i \oplus K) \otimes W] \bmod 2^r \tag{2.63}$$

Since sender modify each block of image matrix so that each block has to follow the main principle of data hiding using weighted matrix stated in equation (2.63). At the time of extraction,

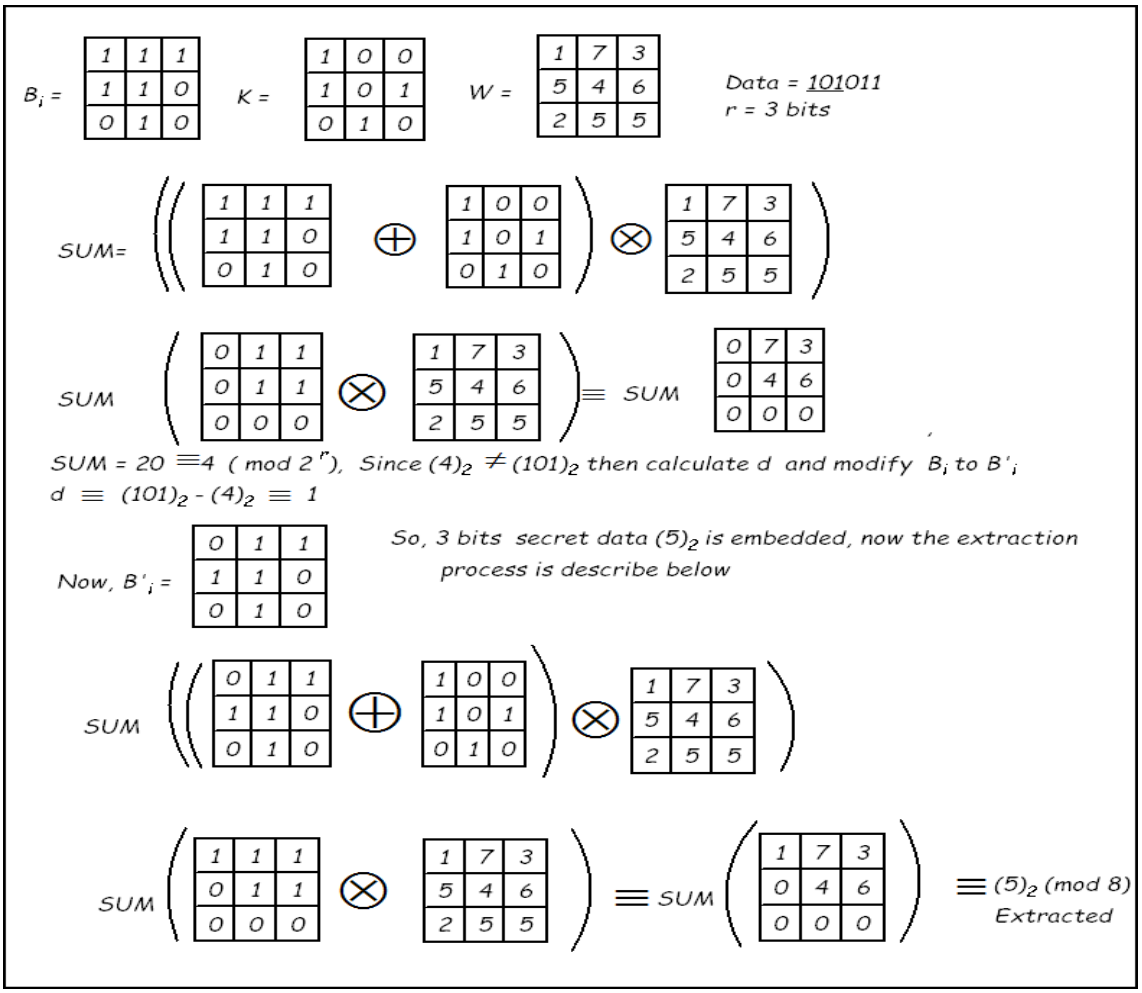


Figure 2.10: Numerical example of Tseng's data hiding scheme

receiver calculate $SUM[(B_i \oplus K) \otimes W] \bmod 2^r$ to get the hidden message because each block of B_i' satisfies the principal. So, receiver will collect the secret data from the modulo result.

2.6.2 Fan et al.'s scheme

Fan et al. [14] suggested a weighted matrix based data hiding scheme using gray scale image. An $(m \times n)$ integer weighted matrix W will be shared by sender and receiver before data communication. The criterion of preferring W is that each element of matrix is arbitrarily allotted a value from the combination $(0, 1, 2, \ldots, 2^{r-1} + 1)$ and each element appears at least once in W, where r denotes the number of secret bits those will be embedded into each block of cover image B_i. Next, it will embed r data bits, say $b_1 b_2 \ldots b_r$ into image block B_i using the following equation

$$d = (b_1 b_2 \ldots b_r)_2 - SUM(B_i \otimes W) \bmod 2^r, \tag{2.64}$$

where $\otimes$ denotes entry-wise-multiplication operator and $i = 1, 2, \ldots, N_B$, where N_B is the number of blocks. The function $SUM(.)$ represents the modular summation of all the entries of matrix $(B_i \otimes W)$. If d is equal to zero modulo 2^r then B_i is intact; otherwise, modify B_i to B_i' to satisfy the following equation

$$SUM(B_i' \otimes W) = b_1 b_2 ... b_r (\mathrm{mod}\ 2^r) \tag{2.65}$$

The receiver can derive $b_1 b_2 \ldots b_r$ by computing $SUM(B_i' \oplus W)\ (mod\, 2^r)$. The scheme in-

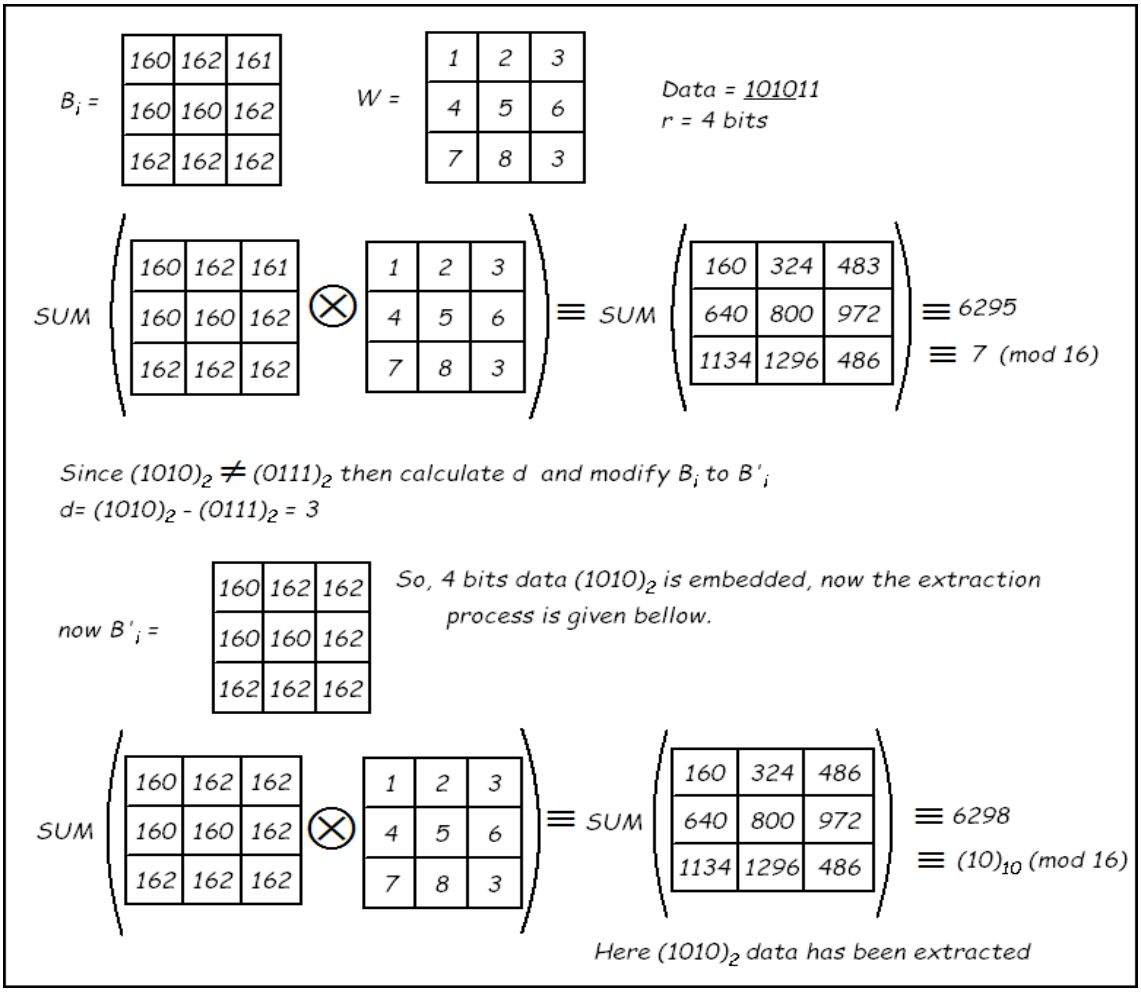


Figure 2.11: Numerical example of Fan et al.'s data hiding scheme

creases the data embedding capacity in a block of size ($m \times n$) to $\lfloor \log_2(2 \times m \times n) \rfloor$.

Example 2.6.1 *The Fig 2.11 shows embedding and extraction process through a numerical illustration.* B_i *is the image block, which is performed sum of entry-wise-multiplication with the weighted matrix (*W*) and we get the SUM which is* 6295*. Observe that* $(6295 \mod 16) \equiv (7)_{10}$ *which is not equals to* 4 *bits secret data that is* $(1010)_2$*, so need to compute* d *value using*

equation (2.64). *Calculate* $d = (10 - 7)_{10} = (3)_{10}$. *Now* B_i *is modified to* B_i' *by changing the pixel value* $B_i(1,3) = 161$ *to* $B_i'(1,3) = 162$. *The location* $(1,3)$ *is chosen because of* $W(1,3) = d$. B_i' *is now stego pixel block.*
During extraction we perform only $\otimes$ *operation between* B_i' *and* W *then find that the operation satisfies the principal of data embedding as mentioned in equation* (2.64). *So using modulo operation the data* $(10)_{10}$ *has been extracted that is shown in Fig. 2.11.* ■

The key matrix (K) and weighted matrix (W) are used in Tseng's scheme but in Li Fan's scheme they only use weighted matrix (W) (which may be considered as shared secret key). Another difference is that, in Tseng's scheme, $\oplus$ and $\otimes$ operation are used but in case of Li Fan's work, only $\otimes$ operation is used.

2.7 Image Interpolation

Image interpolation is one of the simplest computational techniques for image enlargement. This method normally generates high resolution image from its present resolution. The image will be enlarged in size by adding extra pixels which is known as interpolated pixels. The interpolated pixel is estimated by the help of present neighbor pixel values. At the time of data hiding only interpolated or additional pixels are modified but non interpolated or original pixel values are not effected. This technique is reversible and hidden data can be extracted successfully. Interpolation follows the technique of Nearest Neighbor Interpolation (NNI) where interpolated pixels are estimated by the nearest pixel value. In 2009, Jung and Yoo [26] proposed a method of image interpolation named Neighbor Mean Interpolation (NMI), where additional pixels are adjusted by adjacent pixel. In 2012, Lee and Huang [38] proposed image Interpolation by Neighboring Pixel (INP) scheme where neighbor pixel values are calculated to set the additional pixel values. In 2014, Tang et al. [57] proposed Capacity Reversible Steganography (CRS) that calculate additional pixel values with the help of a proposed formula. The interpolation technique is widely used in medical image processing, remote sensing and digital photo film scanning method. Here Jung and Yoo's, Lee and Huang's and Tang et al's. scheme has been explained.

2.7.1 Neighbor Mean Interpolation (NMI)

NMI calculates the mean of neighboring pixels and inserted into a newly allocated pixel using image interpolation shown in Fig. 2.12. On the pixel $I(i,j)$, the pixel $C(i,j)$ is calculated using

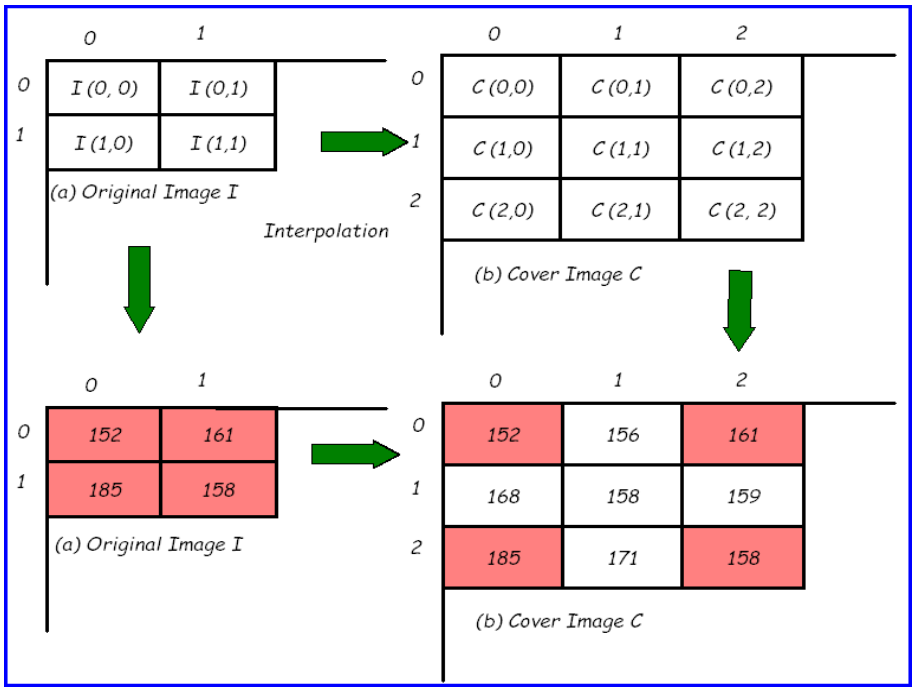


Figure 2.12: Neighbor Mean Interpolation (NMI) scheme with an example

equation (2.66), where $0 \leq j \leq i$ and m and $n, = 0, 1, \ldots, 127.$ and k represents coefficient value of scaling-up, here it is defined as 2.

$$C(i,j) = \begin{cases} I(i,j), & \text{if } i = k.m, j = k.n \\ (I(i,j-1) + I(i,j+1))/k, & \text{if } i = k.m, j = k.n+1 \\ (I(i-1,j) + I(i,j))/k, & \text{if } i = k.m+1, j = k.n \\ (I(i-1,j-1) + C(i-1,j) & \\ +C(i,j-1))/(k+1), & \text{otherwise} \end{cases} \tag{2.66}$$

Using NMI method, calculate a difference value using four non overlapping consecutive pixel value using

$$d = C(k.x + \beta, k.y + \delta) - C(k.x, k.y), \tag{2.67}$$

where $0 \leq x, y \leq 127$ and β, δ is 0 or 1 and k is a scaling factor. Then the number of bits which is possible to embed is calculated by

$$n = \lfloor \log_2 |d| \rfloor. \tag{2.68}$$

To embed $n-$bit the new pixel $C^{'}(i,j)$ is computed using

$$C'(i,j) = C(i,j) + b, \tag{2.69}$$

where b is the integer value of n bit secret data. To extract the hidden message equation (2.70) has been used, where $0 \leq j \leq i$ and x, and $y, = 0, 1, \ldots, 127.$ and k represents coefficient value of scaling-up factor, here it defined 2.

$$b = \begin{cases} C'(i,j) - (C'(i,j) + C'(i,j))/k, & \text{if } i = k.x, j = k.y \\ C'(i,j) - (C'(i,j) + C'(i,j+1))/k, & \text{if } i = k.x, j = k.y+1 \\ C'(i,j) - (C'(i,j) + C'(i+1,j))/k, & \text{if } i = k.x+1, j = k.y \\ C'(i,j) - (k.C'(i,j) + C'(i,j+2)/k & \\ +C'(i+2,j)/k)/(k+1), & \text{otherwise} \end{cases} \tag{2.70}$$

The PSNR in NMI approach is higher than 35 (dB) with high volume data embedded that is 4,25,199 bits.

2.7.2 Interpolating with Neighboring Pixel (INP)

In 2012, Lee and Hung [38] proposed reversible data hiding using Interpolation by Neighboring Pixels (INP). In this method, using equation (2.71) generate the scaling-up image as proposed by Jung and Yoo [26].

$$C(i,j) = \begin{cases} I(i,j), & \text{if } i = k.m, j = k.n \\ (I(i,j-1) + (I(i,j-1) + I(i,j+1))/k)/k, & \text{if } i = k.m, j = k.n+1 \\ (I(i-1,j) + (I(i-1,j) + I(i,j))/k)/k, & \text{if } i = k.m+1, j = k.n \\ (I(i-1,j-1) + (I(i-1,j-1) + C(i-1,j) & \\ +C(i,j-1))/(k+1))/k, & \text{otherwise} \end{cases} \tag{2.71}$$

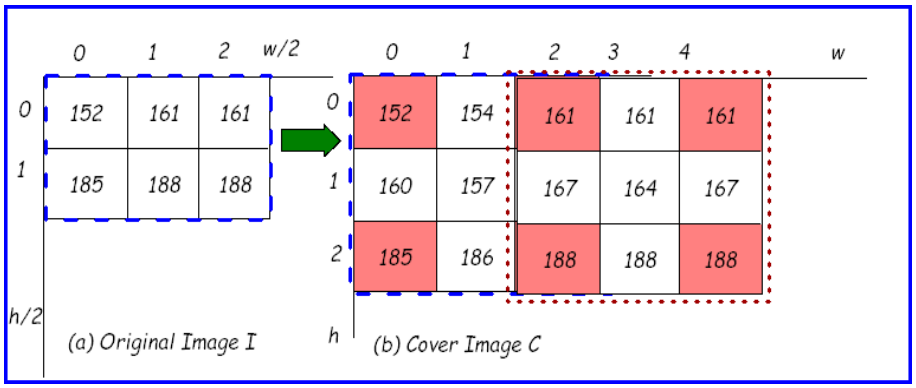


Figure 2.13: Interpolating with Neighboring Pixel (INP) with an example

In INP approach, they calculate the difference value between a pivot and its neighboring non-pivot pixel. The greater the difference, greater the number of secret bits embedded. They use

(3×3) overlapping block to enhance the embedding capacity shown in Fig. 2.13. In INP approach, they achieve payload between 1.29 to 2.27 (bpp) where PSNR varies 20.49 to 22.45 (dB) for different (512×512) image [38].

2.7.3 High Capacity Reversible Steganography (CRS)

The CRS algorithm works based on the equation (2.72), which takes some advantages of the difference of similar properties neighboring pixels which are shown in Fig. 2.14. The CRS is applied to produce a $(m \times n)$ cover image C from $(m/2 \times n/2)$ sized original image.

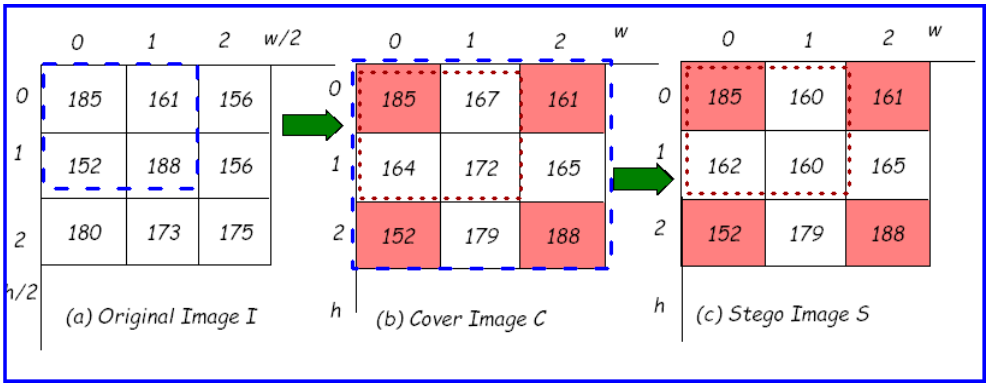


Figure 2.14: Capacity Reversible Steganography (CRS) with an example

$$\begin{cases} I_{min} = \min\{C(i,j), C(i+2,j), C(i,j+2), C(i+2,j+2)\} \\ I_{max} = \max\{C(i,j), C(i+2,j), C(i,j+2), C(i+2,j+2)\} \\ AD = \frac{3\,X\,I_{min}+I_{max}}{4} \\ C(i,j) = I(i,j) \\ C(i,j+1) = \frac{AD+(C(i,j)+C(i,j+2))/2}{2} \\ C(i+1,j) = \frac{AD+(C(i,j)+C(i+2,j))/2}{2} \\ C(i+1,j+1) = \frac{(C(i,j)+C(i+1,j)+C(i,j+1)}{3}, \end{cases} \tag{2.72}$$

where $i = 2 \times m, j = 2 \times n$; m and $n = 0, 1, 2, \ldots, k$. Then CRS calculate the difference value d_1, d_2 and d_3 pixels $C(i, j+1), C(i+1, j)$ and $C(i+1, j+1)$ respectively. When the differences are obtained then calculate the length of secret bit using

$$n_k = \lfloor \log_2 d_k \rfloor, k = 1, 2, 3 \tag{2.73}$$

Finally, stego image S can be obtained using

$$\begin{cases} S(i,j) = C(i,j) \\ S(i,j+1) = C(i,j+1) - b_1 \\ S(i+1,j) = C(i+1,j) - b_2 \\ S(i+1,j+1) = C(i+1,j+1) - b_3 \end{cases} \tag{2.74}$$

where $i = 2 \times m, j = 2 \times n$; m and $n = 0, 1, 2, \ldots, 127$. The extraction process is the reverse of embedding process. Table 2.5 shows the comparison of existing image interpolation methods. In CRS method they achieved average PSNR 33.85 (dB) with average payload 1.79 (bpp).

Table 2.5: Comparison of existing image interpolation methods

	NMI [26]		INP [38]		CRS [57]	
Image	PSNR (dB)	Capacity (bpp)	PSNR (dB)	Capacity (bpp)	PSNR (dB)	Capacity (bpp)
Image 01	32.82	1.12	32.96	1.81	32.35	2.20
Image 02	33.10	0.96	33.60	1.45	33.05	1.82
Image 03	31.65	1.28	32.24	1.97	32.42	2.41
Image 04	31.56	1.34	32.27	2.04	32.38	2.20
Image 05	32.15	0.75	32.54	1.79	32.57	2.25
Image 06	29.70	1.91	30.22	2.13	30.66	2.02
Image 07	33.68	1.40	34.33	1.39	34.73	1.79
Image 08	33.70	0.94	34.31	1.42	34.65	1.57
Image 09	33.80	0.82	34.41	1.26	34.75	1.72
Image 10	33.35	1.63	30.73	2.03	31.08	2.05

2.8 Dual Image based data hiding

Data can be hidden not only in a single image but also in a dual image. Dual-image techniques have often been used recently. Dual image based data hiding technique can copy a cover image into two similar stego images to enhance the overall embedding capacity. Furthermore, these schemes can increase the security in data hiding schemes. Without two stego images being simultaneously obtained, it is impossible for illegal persons to extract the complete secret message. The concept of dual image based data hiding scheme can be treated as a special case of secret sharing. In this section, we have discussed some recently developed dual-image based data hiding schemes.

2.8.1 Chang et al.'s scheme

In 2007 Chang et al. [5] introduced dual image based data hiding technique. In this method, they considered 4 bits secret data and converted it into two digits w_1 and w_2 in a base-5 notational system. Then a pixel pair (x, y) has been considered from the cover image and embed two digits within that pair. Two sets of stego pixel pairs $(x^{'}, y^{'})$ and $(x^{''}, y^{''})$ are generated after modifying

the pixel pair using the magic matrix (MM). The element of MM belongs to [0, 4]. Two sets of diagonal lines DL_1 and DL_2 are obtained through the MM. Each diagonal line has five candidate elements and they intersect each others. The diagonal lines are computed using

$$\begin{cases} DL_1 = \{MM(x+2, y-2), MM(x+1, y-1), MM(x, y), \\ \qquad MM(x-1, y+1), MM(x-2, y+2)\} \\ DL_2 = \{MM(x+2, y+2), MM(x+1, y+1), MM(x, y), \\ \qquad MM(x-1, y-1), MM(x-2, y-2)\} \end{cases} \tag{2.75}$$

If the pixel values x or y is less than 2 or greater than 253 then pixel pair can not be used for data embedding. In that situation, the stego pixel pair $(x^{'}, y^{'}) = (x, y)$ and $(x^{''}, y^{''}) = (x, y)$. Embedding secret data w_1 using DL_1, means w_1 is mapped with the candidate element of DL_1. These mapping pixels are the stego pixel pair $(x^{'}, y^{'})$. Embedding secret data w_2 using DL_2, means w_2 is mapped with the candidate element of DL_2. This mapping pixels are the stego pixel pair $(x^{''}, y^{''})$.

At the receiver end, the magic matrix MM is known. So, the receiver can easily extract the secret message from the stego pixel pairs using the magic matrix MM and recover the original image using two sets of diagonal lines DT_1 and DT_2 which are calculated as

$$\begin{cases} DT_1 = \{MM(x^{'}+2, y^{'}-2), MM(x^{'}+1, y^{'}-1), \\ \qquad MM(x^{'}, y^{'}), MM(x^{'}-1, y^{'}+1), MM(x^{'}-2, y^{'}+2)\} \\ DT_2 = \{MM(x^{''}+2, y^{''}+2), MM(x^{''}+1, y^{''}+1), \\ \qquad MM(x^{''}, y^{''}), MM(x^{''}-1, y^{''}-1), MM(x^{''}-2, y^{''}-2)\} \end{cases} \tag{2.76}$$

The original pixel pair (x, y) has been regarded as the pixel pair of the candidate element where DT_1 and DT_2 intersect. In this method, the embedding capacity is 1 (bpp) and PSNR is 45.11 (dB) for Stego 1 and 45.12 (dB) for Stego 2 image.

2.8.2 Lee and Huang's scheme

In 2013, Lee and Huang [34] proposed a RDH scheme. based on orientation combination techniques. Consider 5 bits from the secret message and get its decimal value. If the decimal value belongs to range [16, 24] then that 5 bits secret data is possible to embed. During data embedding, first convert 5 bits data into two digits w_1 and w_2 in a base-5 notational system. Otherwise,

4 bits data are to be embedded. So, convert 4 bits data into two digits w_1 and w_2 in a base-5 notational system. Then select a pixel pair (p, q) from the cover image and generate major pixel pair (p_m, q_m) and auxiliary pixel pair (p_a, q_a) where $p_m = p_a = p$ and $q_m = q_a = q$. Now, w_1 is embedded within pixel pair (p_m, q_m) using a specified rule base and generate major stego pixel pair $(p_m^{'}, q_m^{'})$. Similarly, w_2 is embedded within pixel pair (p_a, q_a) using a specified rule base and generate auxiliary stego pixel pair $(p_a^{'}, q_a^{'})$. After embedding secret data, pixel values increase or decrease by at most 1. Then using a secret key bit stream distribute the pixel pair among dual image. To enhance security, a key stream $k = k_1, k_2 \ldots$, where $k_i \in \{0, 1\}$. If k_i=1, then $(p_m^{'}, q_m^{'})$ is stored in the first stego image and $(p_a^{'}, q_a^{'})$ is stored in the second stego image. If k_i = 0,then $(p_m^{'}, q_m^{'})$ is stored in the second stego image and $(p_a^{'}, q_a^{'})$ is stored in the first stego image.

At the receiver end, the receiver can fetch the pixel pair using key bits stream. Then retrieve secret message bits using another specified rule. In this method, the embedding capacity is 1.07 (bpp) and PSNR is 49.76 (dB) for Stego 1 and 49.56 (dB) for Stego 2.

2.8.3 Chang et al.'s scheme

In 2013, Chang et al. [10] proposed a reversible data hiding scheme where data bits are embedded based on the Magic matrix. Magic matrix MM is computed using.

$$F(x, y) = (x + 3 \times y) \bmod 9 \tag{2.77}$$

where x and y are two pixels and $F(x, y)$ is referred as secret message. All elements of magic matrix belongs to [0, 8]. Take a pixel x from cover image and copy it into y. Then fetch n bits secret data from secret message and convert it into decimal form dec, where $n = 4$. Then check the decimal value dec is equal to 8 or not. If $dec = 8$, then embed n bits data otherwise embed $(n - 1)$ bit data. Then check the element of the magic matrix of the corresponding pixel pair. If it is equal to secret data then the stego pixel pair is same as the original pixel pair. Otherwise replace the current pixel pair (x, y) by a pixel pair whose element of the magic matrix is equal to secret data. The stego pixel pair $(x^{'}, y^{'})$ is produced by changing at most k in current pixel pair (x, y), where $k \in [0, 4]$. The pixel $x^{'}$ is stored into the first stego image and the pixel $y^{'}$ is stored into the first stego image. If the pixel x is increased by k then the pixel y is decreased by

k. On the receiver side, receiver can extract the secret message using the following equation

$$dec = (x^{'} + 3 \times y^{'}) \bmod 9 \tag{2.78}$$

Then we check the decimal value which is greater than 7 or not. If it is greater than 7 then convert it into n bits binary form otherwise covert into $(n-1)$ bits binary form. The original pixel is recovered by computing the average of two stego pixels. In this method, the embedding capacity is 1.53 (bpp) and PSNR is 39.89 (dB) for Stego 1 and 39.89 (dB) for Stego 2.

2.8.4 Lu et al.'s scheme

In 2015, Lu et al. [45] proposed a data hiding technique based on central folding strategy. First select n bits data from secret message and convert it into decimal form dec. To avoid image distortion dec is reduced by central folding strategy and generate $dec^{'}$ as $dec^{'} = dec - 2^{n-1}$. Where dec is a folded secret symbol and n is the number of bits. When $dec < 2^{n-1}$, dec was represented as a negative number. When $dec = 2^{n-1}$, it was represented as 0. When $dec > 2^{n-1}$ it was expressed as a positive number. Now, take a pixel x from the original image and embed secret digit $dec^{'}$. So, the stego pixel pair $x^{'}$ and $x^{''}$ can be calculated using the following formula

$$\begin{cases} x^{'} = x + \lfloor \frac{dec^{'}}{2} \rfloor \\ x^{''} = x + \lceil \frac{dec^{'}}{2} \rceil \end{cases} \tag{2.79}$$

The pixel $x^{'}$ is stored into stego image S_1 and the pixel $x^{''}$ is stored into stego image S_2. Secret data can not be embedded into those pixels which do not belong to the range $[2^{n-1}, 256 - 2^{n-1}]$. During data extraction, first we check whether stego pixels $x^{'}$ and $x^{''}$ are equal or not. If they are equal and do not belong to range $[2^{n-1}, 256 - 2^{n-1}]$ then there is no secret message. Otherwise secret message is extracted from stego pixels. First calculate the difference $dec^{'}$ and dec using the following equation.

$$\begin{cases} dec^{'} = x^{'} - x^{''} \\ dec = dec^{'} + 2^{k-1} \end{cases} \tag{2.80}$$

The original pixel is recovered by computing the average of two stego pixels using the formula.

$$x = \lceil \frac{x^{'} + x^{''}}{2} \rceil \tag{2.81}$$

In this method, the embedding capacity is 1 (bpp) and PSNR is 49.89 (dB) for Stego 1 and 52.90 (dB) for Stego 2 image when $k = 2$.

2.8.5 Comparison of existing schemes

Table 2.6: Comparisons of existing dual image based data hiding methods

Methods	Measures	Images					
		Lena	Peppers	Boat	Goldhill	Zelda	Baboon
Chang et al. [5]	PSNR(1)	45.12	45.14	45.12	45.13	45.13	45.11
	PSNR(2)	45.13	45.15	45.13	45.14	45.11	45.13
	PSNR(Avg.)	45.13	45.15	45.13	45.14	45.12	45.12
	Capacity(bpp)	1	0.99	1	1	0.99	0.99
Lee and Huang [34]	PSNR(1)	49.76	49.75	49.76	49.77	49.77	49.77
	PSNR(2)	49.56	49.56	49.57	49.57	49.58	49.56
	PSNR(Avg.)	49.66	49.66	49.67	49.67	49.68	49.77
	Capacity(bpp)	1.07	1.07	1.07	1.07	1.07	1.07
Chang et al. [10]	PSNR(1)	39.89	39.94	39.89	39.9	39.89	39.91
	PSNR(2)	39.89	39.94	39.89	39.9	39.89	39.91
	PSNR(Avg.)	39.89	39.94	39.89	39.9	39.89	39.91
	Capacity(bpp)	1.53	1.52	1.53	1.53	1.53	1.53
Qin et al. [52]	PSNR(1)	52.11	51.25	51.11	52.11	52.06	52.04
	PSNR(2)	41.34	41.52	41.57	41.34	41.57	41.56
	PSNR(Avg.)	46.72	46.39	46.84	46.72	46.82	46.80
	Capacity(bpp)	1.16	1.16	1.16	1.16	1.16	1.16
Lu et al. [45]	PSNR(1)	49.89	49.89	49.89	49.90	49.89	49.89
	PSNR(2)	52.90	52.92	52.90	52.90	52.88	52.87
	PSNR(Avg.)	51.40	51.41	51.40	51.40	51.39	51.38
	Capacity(bpp)	1	0.99	1	1	0.99	0.99

Table 2.6 shows the comparison of existing dual image based data hiding methods. We observed that Chang et al.'s method [5] and Lu et al.'s method [45] has same payload 1 (bpp) but the average PSNR of Chang et al. method [5] is lower than Lu et al.'s method [5]. The average PSNR of Lee and Huang method [34] is 49.66 (dB) and payload is 1.07 (bpp), which is more than Chang et al.'s method [5]. In Chang et al.'s method [10] the payload is 1.53 (bpp) which is greater than Lee and Huang's method [34] but the PSNR is 39.90 (dB) which is lower than Lee and Huang's method [34]. Qin et al.'s method [52] has higher PSNR 46.72 (dB) than Chang et al.'s method [10] but the payload is 1.16 (bpp) which is lower than Chang et al.'s method [10]. Fig 2.15 shows the comparison graph among existing dual image based reversible data hiding methods.

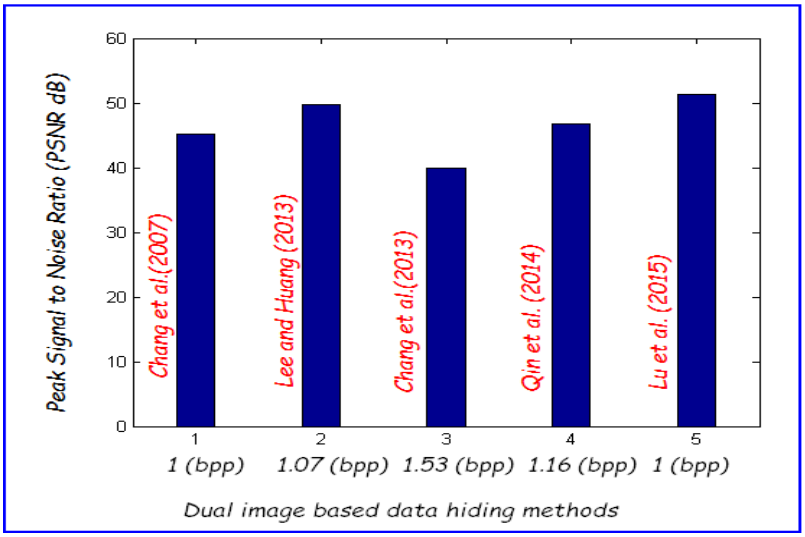


Figure 2.15: Comparison graph of existing dual image based data hiding schemes

2.9 Steganalysis and Steganographic Attacks

The art of discovering a secret message within a suspected image is called steganalysis. The goal of steganalysis is to gather enough evidence about the presence of embedded message and to break the security of its carrier. The importance of steganalytic techniques that can reliably detect the presence of hidden information in images is increasing. Most data hiding schemes leave some clues to detect the hidden message from stego media, those are not perfectly secure. The steganalyst try to find out the presence of message within stego media in various ways. The way is divided into two categories-Targeted and Blind. Some of targeted steganalysis are Visual attack, Statistical attack and Structural attack etc. and one of the important blind steganalysis is RS analysis [16].

2.9.1 RS Analysis

The RS analysis was proposed by the J. Fridrich [18]. It is the first quantitative analysis of LSB steganography. In RS analysis method, first the stego image is divided into disjoint groups G of n adjacent pixels $(x_1, \ldots, x_n)$. Each pixel value is in a set P that is $P = \{0, 1, \ldots, 255\}$. Here each group consists of 4 consecutive pixels in a row. A discrimination function f that returns a real number $f(x_1, \ldots, x_n) \in R$ to each pixel group $G = (x_1, \ldots, x_n)$. The main goal is the use the discrimination function to identify the **Smoothness** or **Regularity** of each group of pixels

G. The discrimination function f is defined as follows

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \tag{2.82}$$

An invertible function F which operates on P is called **flipping**. Flipping consists of two-cycles which permutes the pixels value. So, $F^2 = Identity\ or\ F(F(x)) = x$ for all x belongs to P. Flipping the LSB of each pixel value and the corresponding permutation $F1$ is $0 \leftrightarrow 1, 2 \leftrightarrow 3, \ldots, 254 \leftrightarrow 255$. Another function, named shift LSB flipping and treated as F_{-1}. So, the permutation F_{-1}:$(-1) \leftrightarrow 0, 1 \leftrightarrow 2, \ldots, 255 \leftrightarrow 256$. In other words, F_{-1} flipping can be defined as below

$$F_{-1}(x) = F_1(x + 1) - 1,\ for\ \ all\ \ x\ . \tag{2.83}$$

There are three types of groups *Regular* groups, *Singular* groups and *Unusable* groups which are defined depending on the discrimination function f and the flipping operation F. Depending on the condition groups are defined below.

$$\begin{cases} G \in \textit{Regular} & \text{if } f(F(G)) > f(G) \\ G \in \textit{Singular} & \text{if } f(F(G)) < f(G) \\ G \in \textit{Unusable} & \text{if } f(F(G)) = f(G) \end{cases} \tag{2.84}$$

where, $F(G) = (F(x_1), \ldots, F(x_n))$. The flipping operation will be executed with the help of a mask value M which is a n tuples with values -1, 0, and 1. The flipped group $F_M(G)$ is defined as $F_M(1)(x1), F_M(2)(x2), \ldots, F_M(n)(xn)$. The RS analysis based on analyzing how the number of regular and singular groups changes with the increased message length embedded in the LSB plane. Then calculate the value of RS analysis using the following equation.

$$(|R_M - R_{-M}| + |S_M - S_{-M}|)/(R_M + S_M) \tag{2.85}$$

where, R_M and R_{-M} is the total number of regular group with mask M and $-M$ respectively. S_M and S_{-M} are the total number of singular group with mask M and $-M$ respectively. If value of $R_M \simeq R_{-M}$ and $S_M \simeq S_{-M}$, then the method is secure. Also, when the value of RS analysis is close to 0 that means the method is secure.

2.9.2 Relative Entropy

To measure the security of a data hiding method one can calculate relative entropy (the differences) between the probability distribution of the original image and the stego image which

has been calculated by following equation (2.86). Let p_m and q_n be probability measures for original image I and stego image S respectively. The relative entropy distance $D(S||I)$ (also known as Kullback-Leibler distance) is defined as follows

$$D(S||I) = \sum q_n(x) log \frac{q_n(x)}{p_m(x)}, \tag{2.86}$$

when relative entropy between two probability distribution functions is zero then the system is perfectly secure. D$(S||I)$ is a nonnegative continuous function and equals to zero if and only if p_m and q_n coincide. Thus D$(S||I)$ can be naturally viewed as a distance between the measures p_m and q_n. Relative entropy of the probability distribution of the original image I and the Stego image S vary depending upon number of bits of secret message. When the number of characters in the secret message is increasing, the relative entropy is increasing.

2.9.3 Statistical Analysis

When two variables are considered in a bivariate data, say these two variables to be correlated if the change in the value of one is related to the change in the value of other. Correlation may be of two types:

(i) If the increase in the value of one variable brings on average the increase in the value of the other variable, then these two variables are said to be positively correlated.

(ii) If the increase in the value of one variable brings on average the decrease in the value of the other variable, then these two variables are said to be negatively correlated.

In a scatter diagram, in most of the cases, it is noticed that the points plotted in a diagram are more or less concentrated in the neighborhood of a curve which is called regression curve. In the case of simple regression that is when the two regression curve are linear then their degree of collinearity is measured by a quantity known as correlation coefficient (CC) and it is generally denoted by ρ_{xy}.

Let $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ be a set of n pairs of observations in a bivariate distribution having two variables x and y. Then the correlation coefficient ρ_{xy} between the two variables x and y is defined as follow.

$$\rho_{xy} = \frac{Cov(x, y)}{\sigma_x, \sigma_y}, \tag{2.87}$$

where, $Cov(x,y) = \frac{1}{n}.\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$ is called the co-variance between (x,y) and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Here σ_x and σ_y are the standard deviation (SD) of x_i and y_i, $i = 1, 2, \ldots, n$. $\sigma_x = \sqrt{\frac{1}{n}\sum x_i^2 - \bar{x}^2}$ and $\sigma_y = \sqrt{\frac{1}{n}\sum y_i^2 - \bar{y}^2}$.

2.9.4 Histogram Attack

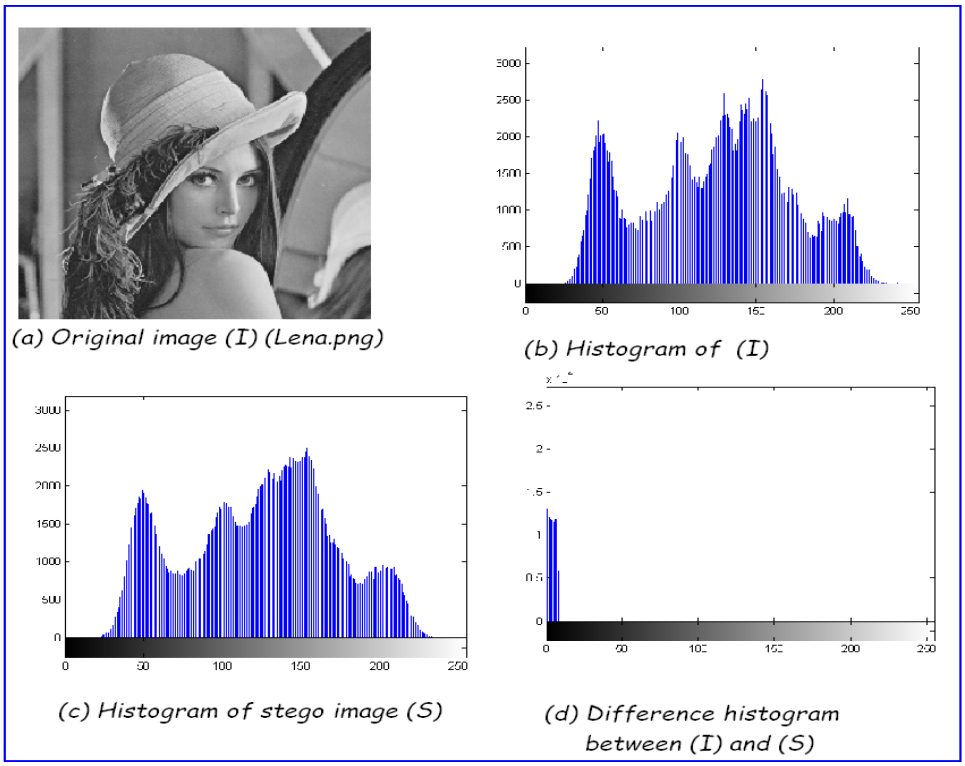


Figure 2.16: Example of Histogram Attack

A new steganalytic attack has been proposed by Jeremiah J. Harmsena [22] in 2003. The attack based on the fact that noise adding in the spatial domain corresponds to low-pass filtering of the histogram. Histogram of cover image is represented as h whereas histogram of stego image is represented as $h^{'}$. The change of histogram can be measured by

$$D_h = \sum_{m=1}^{255} |h_m^{'} - h_m| \tag{2.88}$$

For example, Fig. 2.16 describes the histogram of the cover image and stego image and their difference histograms. The stego image is produced from cover image employing the maximum data hiding capacity.

2.9.5 Brute Force Attack

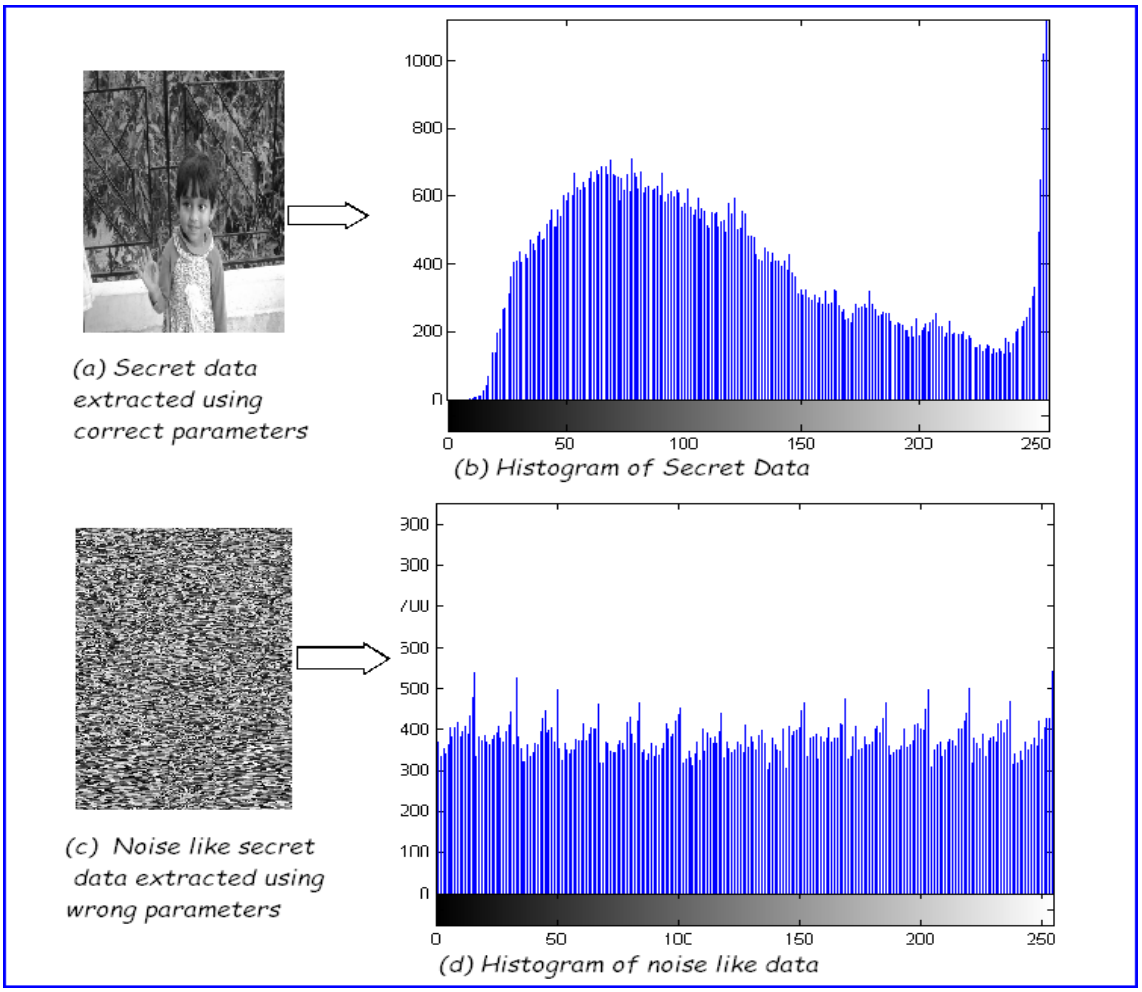


Figure 2.17: Example of Brute Force Attack

To retrieve the secret data from stego image adversaries may guess the required parameters on trial and error basis. The scheme will be secure if it prevents retrieval by possible malicious attacks. For example, consider the Fig. 2.17 which shows the example of getting noise like secret data when applied wrong parameters to revel the hidden message. The scheme should be designed in such a manner that if the malicious attacker holds the original image and stego image and is fully aware of the proposed scheme, the hidden message still cannot be correctly revealed without knowing the required parameters.

Chapter 3

Reversible Data Hiding (RDH) using Hamming code

3.1 Introduction

Communication through data hiding is useful in various applications such as copyright protection, covert communication, content authentication, forensic tracking, tamper detection, ownership identification and many other human centered approaches. Reversible data hiding is attracting much attention from the researcher due to its progressive applications in different areas like health care, military communication and law enforcement. In recent years, the demand of efficient, secure and reversible hidden data communication is increasing. In the literature, it is found that no such schemes exist, where reversibility has been considered in data hiding through Hamming code.

In this chapter, secret message bits are embedded within LSB of the cover image through error creation caused by tamper in any position except secret position and error position has been detected at the receiver end with the help of Hamming code. Here, two different Hamming code based data hiding approaches are proposed. The first approach is irreversible data hiding scheme through Hamming code using single image and other one is reversible data hiding scheme through Hamming code using dual image. In single image approach, one shared secret position has been used during data embedding and extraction. Where as in dual image approach, one shared secret position and one shared secret key have been used for data embedding and data extraction.

In the first approach, secret message bits are embedded within LSB layer of cover image through error creation. Errors are created in two different bit positions of a row in LSB block where one is at the secret position and other one at a suitable location for error creation. The suitable location is the position of a row in LSB layer which is opposite of the data bit. That means if the data bit is 1 then the position of a row in LSB layer which contain 0 is the suitable location or vice versa. The receiver can successfully retrieve the secret message using shared secret position, but it is hard for an adversary to retrieve the secret message without secret key because errors have been created in two different places in a row of LSB block which is also hard to detect through Hamming code. To enhance embedding capacity, three LSB layers (LSB, LSB+1 and LSB+2) have been used to embed secret message bits within cover image. The receiver can detect and correct the error with the help of Hamming error correcting code. This

is a partial reversible data hiding scheme using Hamming code (PRDHHC) where receiver can recover Hamming adjusted cover image but cannot recover original cover image.

In the second approach, reversible data hiding schemes using Hamming code (RDHHC) have been proposed using dual image. In this approach, one shared secret position and one shared secret key has been used. The shared secret position is used to create error in a row of LSB layer and shared secret key is used to distribute stego block between two stego images. To achieve reversibility through dual image, two indexes (Index 1 and Index 2) have been introduced to choose redundant bits from each row of LSB layers. The RDHHC provides secure hidden data communication using Hamming code where receiver extracts secret data and recovers original image successfully. To enhance the data hiding capacity, three LSB layers (LSB, LSB+1 and LSB+2) have been used. All these proposed methods can create errors in two different locations in every row of LSB layer and the receiver can detect and correct the error at data embedding position with the help of Hamming error correcting code. Dual image is used to recover the original cover image successfully after extracting secret messages.

Four data hiding schemes have been proposed in this chapter. Two partial reversible data hiding schemes have been proposed using single image where LSB layer and three LSB layers (LSB, LSB+1, LSB+2) are used for data hiding. The payload of these two schemes are 0.142 (bpp) and 0.426 (bpp) for LSB and three LSB layers respectively. But these single image based schemes are not reversible. To achieve reversibility, another two methods have been proposed using dual image where two indexes (Index 1 and Index 2) are introduced to choose redundant bits from each image. Both one LSB layer and three LSB layers are also used to hide data through error creation. The data hiding capacity is same as that of the previous schemes. The advantage of dual image based data hiding is that without having two stego images simultaneously it will be hard to the adversary to retrieve secret data. This is a special case of secret sharing.

3.2 Partial Reversible Data Hiding using (7,4) Hamming Code (PRDHHC)[1]

In this section, a Partial Reversible Data Hiding scheme using (7,4) Hamming Code (PRDHHC) with secret position (κ) has been proposed. In this scheme, the original cover image (I) has been partitioned into (7×7) pixel block and LSB has been collected. Then redundant bits (1^{st}, 2^{nd} and 4^{th} positions) of each row has been adjusted using odd parity. Repeat this process for all (7×7) blocks of cover image and generate parity adjusted cover image (C). Now calculate the secret position $\kappa = (\delta \mod 7) + 1$, where δ is a shared secret key. The bit at the secret position is complemented first then find a suitable location to insert data bit through error creation at any bit positions except the secret position. For the next row, the κ is updated by the data embedding position (ω) of the previous row. Repeat this process to embed secret message bits within the selected block. For each new block, the κ is updated by $\kappa_{i+1} = (\kappa_i \times \delta \times \omega) \mod 7 + 1$, where i = 0, 1, 2, 3, ..., N_B, and N_B is the number of blocks. Continue this process to embed all secret data bits within the cover image and produce stego image.

At the receiver end, the LSBs of (7×7) block of stego image has been considered and complement the bit at the secret position (κ) which is calculated using $\kappa = (\delta \mod 7) + 1$, where δ is a shared secret key. Then apply Hamming error correcting code to detect the error position for secret data bit. After finding the error, fetch the data bit from the error location (ω) and this (ω) has been considered as the secret position (κ) for next row. Before proceeding to the next row, complement the bit at the error location (ω) to recover Hamming adjusted cover image. The extraction process will be stopped when receiver found the error at the κ position only. The proposed PRDHHC scheme extracts hidden message and recovers Hamming adjusted cover image successfully by complement bits at both the κ and ω positions but can not recover original cover image, that is to say, the scheme is partially reversible. As per Kirchhoff's principle, everyone knows the algorithm but the secrecy depends on the key. In this scheme, a shared secret key (δ) is applied to enhance security through Hamming code based data hiding scheme.

[1]Minor Review Submitted in **Multimedia Tools and Application**, Springer, Impact Factor 1.346, with title *Partial Reversible Data Hiding Scheme using (7,4) Hamming Code*

An adversary can find the secret position by trial and error method without knowing secret key δ, because the secret position values lie within 1 to 7 only. But in every block, a new secret position has been assigned. As a result, for a (512×512) image the total number of blocks are $\lfloor \frac{512}{7} \rfloor \times \lfloor \frac{512}{7} \rfloor = 73 \times 73 = 5329$. So, number of trials required to find secret positions for every block are $5329 \times 7 = 37,303$.

Without secret position it is hard to retrieve hidden message by applying Hamming error correcting code because it cannot detect multiple errors in a single row. Here, we assume that one can use more than one secret positions in a single row of each block and in every row the secret position (κ) is updated by applying predefined formula $\kappa_{i+1} = ((\kappa_i \times \delta) \mod 7) + 1$, where δ is a shared secret key. The possible number of secret positions are given below:

Corollary 3.2.1 *In each block, the number of secret positions are* $7!$*, when more than one secret positions are considered in a single row.* □

Corollary 3.2.2 *The secret positions has been updated for every row of each block using* $\kappa_{i+1} = ((\kappa_i \times \delta) \mod 7) + 1$*, where* δ *is the shared secret key. The number of blocks in a* $(m \times n)$ *image are* $m \times (n/7)$*.* □

Corollary 3.2.3 *Total number of secret positions are* $m \times (n/7) \times 7!$*, For example, In a* (512×512) *image the number of possible secret positions are* $(512 \times 512/7) \times 7! = 37376 \times 5040 = 188,375,040$*.* □

3.2.1 Data Embedding Process

The schematic diagram of proposed PRDHHC data embedding scheme is shown in Fig. 3.1. The original image is shown in Fig. $3.1(a)$. The corresponding gray value of (7×7) pixel block and their LSBs are shown in Fig. $3.1(b)$ and $3.1(c)$ respectively. The redundant bits r_1, r_2 and r_3 are adjusted using odd parity which is shown in the Fig. $3.1(d)$. During data embedding, a shared secret key δ is used to calculate secret position κ using $\kappa = (\delta \mod 7) + 1$. The data bit embedding and secret position updating process are shown in Fig. $3.1(f)$. Here, bit in orange color represents the bit modification for secret position and bit in green color represents the bit modification for secret message. $(1/0)$ represents the previous/changed bit in that position shown in Fig. 3.1. After that a final LSB matrix of the stego image is produced which is shown

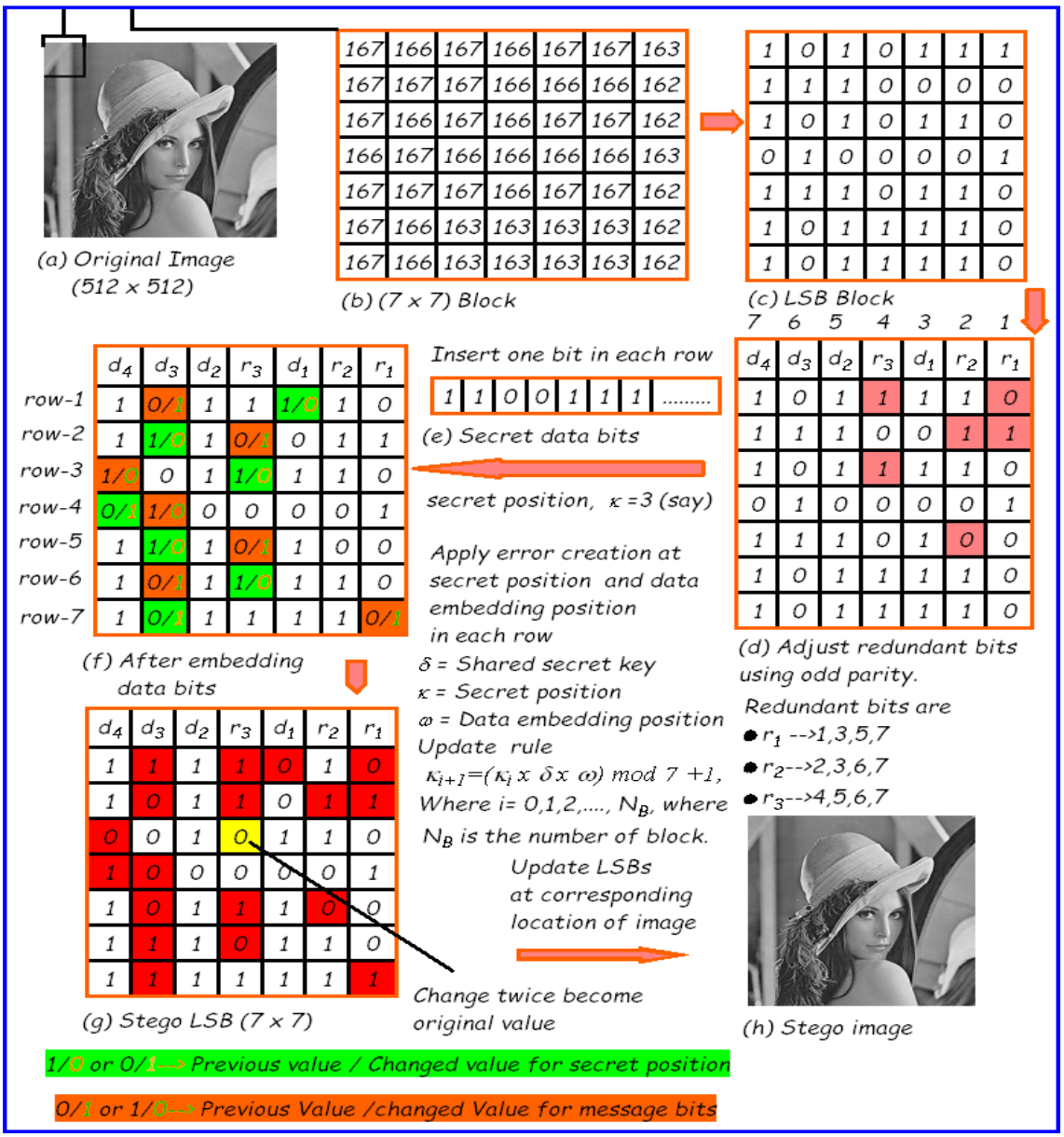


Figure 3.1: The schematic diagram of data embedding process in PRDHHC

in Fig. $3.1(g)$. The LSB matrix is finally updated into stego image. The following algorithm (Algorithm 1) describes the data embedding process.

3.2.2 Data Extraction Process

At the receiver end, the stego image has been partitioned into blocks of size (7×7) pixels and then collect LSBs. The corresponding stego image, their pixels of first block and corresponding LSB matrix are shown in Fig. $3.2(a)$, $3.2(b)$ and $3.2(c)$ respectively. Here, consider that the secret position (κ) is 3 which is calculated using shared secret key δ. The extraction process

Input: Original image I, Secret message bits D and a shared secret key δ, secret position κ is calculated using (κ) = $(\delta \mod 7) + 1$;

Output: Stego image S ;

Step 1: Partition the original image I into (7×7) pixel block and repeat **Step 2** for each block;

Step 2: $count = 1$;

for i = *1 to 7* **do**

    Apply odd parity to adjust redundant bits (r_1, r_2 and r_3) for i^{th} row of LSB matrix R_l;

end

Step 3:

for i = *1 to 7* **do**

    (a) Complement the κ^{th} position of i^{th} row of R_l ;

    (b) Embed the D_{count} in i^{th} row of R_l through error creation

    if D_{count} == 0 **then**

        Create an error by complement at any suitable position (ω) of R_l which contain 1 except κ position;

    end

    {*** When suitable position is not available ***}

    if (($D_{count} == 0$) & i^{th} *row of* $(R_l == 0)$) **then**

        Set $\omega \leftarrow 1$;

    end

    if $D_{count} == 1$ **then**

        Create an error by complement at any suitable position (ω) of R_l which contain 0 except κ position;

    end

    {*** When suitable position is not available ***}

    if (($D_{count} == 1$) & i^{th} *row of* $(R_l == 1)$) **then**

        Set $\omega \leftarrow 7$;

    end

    (c) κ is updated by ω for data embedding in next row of R_l;

    (d) $count = count + 1$;

end

Step 4: If all data bits are embedded then goto **Step 5**, otherwise, select new block $R_l = R_{l+1}$ and goto **Step 3** using $\kappa_{i+1} = (\kappa_i \times \delta \times \omega) \mod 7 + 1$, where i = 0, 1, 2, 3, ..., N_B, and N_B is the number of blocks;

Step 5: Update image matrix R according to modified R_l matrix to produce the stego image;

Step 6: End;

Algorithm 1: Data embedding process of PRDHHC

from the first row of the first block and extracted secret message bits are shown in Fig. $3.2(d)$. The Hamming adjusted recovered LSBs of cover image is shown in Fig. $3.2(e)$. Final recovered Hamming adjusted cover image is shown in the Fig. $3.2(f)$. The extraction process will be stopped when the receiver finds an error at the κ location. It is possible when no data bits are embedded at the sender site and receiver try to find out the error after complementing the bit at the κ location. Naturally, error has been found at the κ location only. After extracting the secret data bits, receiver complements both position bits, one for secret position and another for the data embedding position, which generates hamming adjusted cover image. The bits changed using odd parity are not fully recovered at the receiver end, that is to say, this scheme is partially

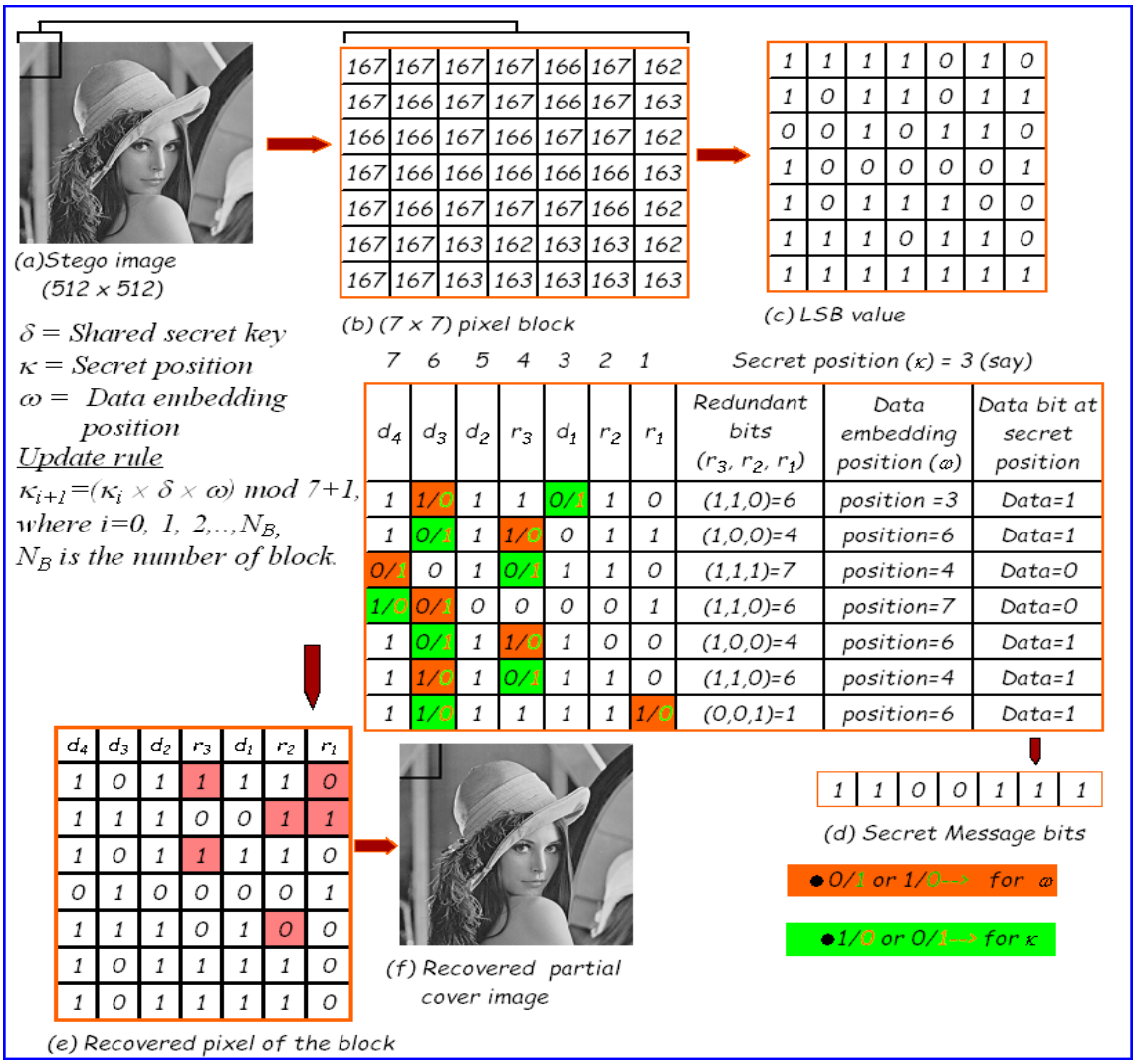


Figure 3.2: The schematic diagram of data extraction process in PRDHHC

reversible. The message extraction process is described using Algorithm 2.

Time Complexity: The time complexity of this proposed data embedding and extraction algorithm are calculated here. During adjustment of three redundant bits using odd parity within a selected block it is required to traverse thrice in a row which takes (3×7) times. For a block it takes $(3 \times 7 \times 7)$ and for a $(m \times n)$ image it takes $(3 \times m/7 \times n/7)$ times. So, the required time complexity of this algorithm is $\mathcal{O}(mn)$ as stated **Step 2** in Algorithm 1. Now, to embed secret data bits in LSB again it visit each 7 bits in a row twice, one for finding secret position and another for finding secret message bits position. So, it takes (2×7) for each row and $(2 \times 7 \times 7)$ for a block. Hence, the required time complexity for $(m \times n)$ image is $(2 \times m/7 \times n/7) = \mathcal{O}(mn)$. The total time complexity during data embedding is equal to $2 \times \mathcal{O}(mn) \equiv \mathcal{O}(mn)$.

Input: Stego image S, shared secret key δ.

Output: The secret message bits D, Hamming adjusted cover image C

Step 1: Divide the stego image S into (7×7) blocks and convert each block into LSB matrix S_l, where $l = 1, 2, ..., (m/7 \times n/7)$, Calculate $\kappa = (\delta \mod 7) + 1$

Step 2:

for *i=1 to 7* **do**

(a) Complement the bit at the κ^{th} position of i^{th} row in S_l matrix ;

(b) Find error position (ω) using odd parity, $r_{(1\times3)} \leftarrow S_l$ where r is the redundant value calculated by odd parity;

(c) $\omega \leftarrow decimal(r_{(1\times3)})$, where ω indicate data embedding position;

if *(*$(\omega \neq 0)$ & $(\omega \neq \kappa)$*)* **then**

$D_i \leftarrow S_l(\omega)$;

end

{*** For no error found***}

if $((\omega == 0)\&(S_l == 0))$ **then**

$D_i \leftarrow S_l(1); \omega \leftarrow 1;$

end

{*** For no error found***}

if $((\omega == 0)\&(S_l == 1))$ **then**

$D_i \leftarrow S_l(7); \omega \leftarrow 7;$

end

{*** For stop execution ***}

if $(\omega == \kappa)$ **then**

goto **Step 4**;

end

{*** When no data bits are embedded through error creation, So** $(\omega == \kappa)$ *****}

(c) Complement bit at the error location;

(d) κ is updated by error position (ω);

end

Step 3: Select $S_l = S_{l+1}$ goto **Step 2** using new secret position $\kappa_{j+1} = (\kappa_j \times \delta \times \omega) \mod 7 + 1$, where j = 0, 1, 2, 3, …, N_B and N_B is the number of blocks.;

Step 4: End;

Algorithm 2: Data extraction process of PRDHHC

The time complexity to calculate the syndrome value in Hamming +1 and in Hamming +3 schemes is $\mathcal{O}(mn)$. But to detect and correct the errors using Hamming error correcting code, it takes $\mathcal{O}(mn\log(mn))$. So, the total time complexity in Hamming +1 and in Hamming +3 schemes is $\mathcal{O}(mn)$ + $\mathcal{O}(mn\log(mn))$ that is $\mathcal{O}(mn\log(mn))$. For the image of the same size of row and column, that is $m = n$, the complexity will be $\mathcal{O}(n^2\log(n))$ which is same with our approach. At the time of extraction, we simple find the location of secret position and then apply Hamming error correcting code to find the data embedding position. To find the secret position in a row of a selected block, it required only seven comparisons and for a $(m \times n)$

image block it tooks $(m/7 \times n/7) = \mathcal{O}(mn)$. Then to find the error using Hamming error correction code, it has the time complexity $\mathcal{O}(mn \log(mn))$. So, the total time complexity is $\mathcal{O}(mn)$ + $\mathcal{O}(mn \log(mn))$ to extract the hidden data and recover the partial cover image.

3.2.3 Experimental Results and Comparisons

The gray scale image of size (512×512) pixel has been used as cover image in this experiment which is shown in Fig 3.3. Distortion of stego image from cover image can be evaluated by

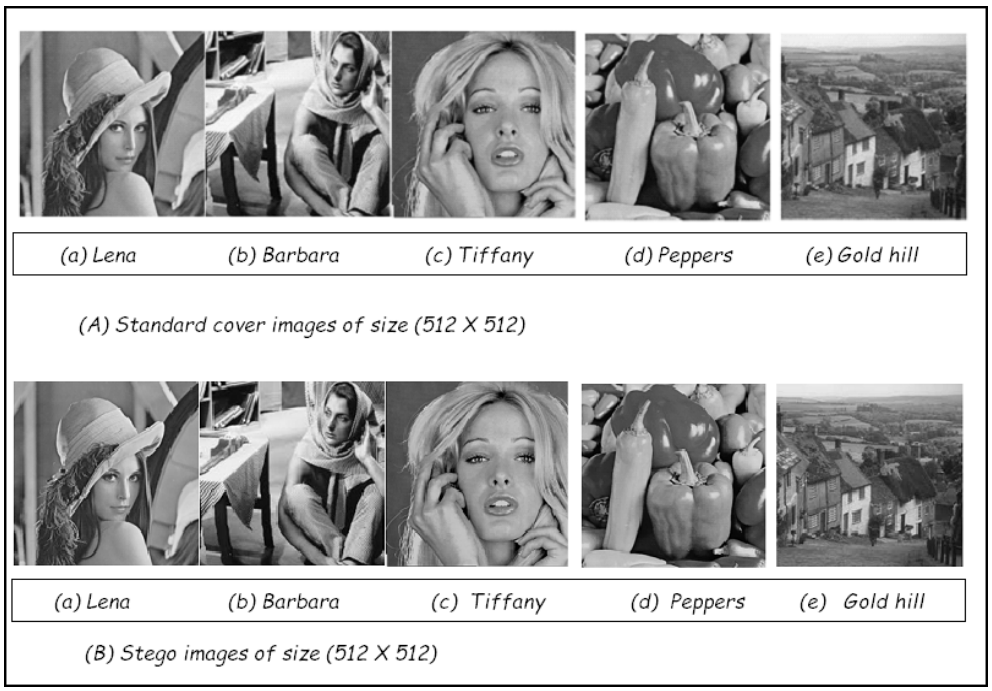


Figure 3.3: Standard cover images and generated stego images in PRDHHC

means of two parameters namely, Mean Square Error (MSE) and Peak Signal to Noise Ratio $(PSNR)$. The MSE is calculated using

$$MSE = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} (x(i,j) - y(i,j))^2}{m \times n}, \tag{3.1}$$

where m and n denote the total number of pixels in the horizontal and the vertical dimensions of the image. $x(i,j)$ represents the pixels in the original image and $y(i,j)$ represents the pixels of the stego image. The Peak Signal to Noise Ratio $(PSNR)$ is calculated by

$$PSNR = 10\, log_{10} \frac{255^2}{MSE} \tag{3.2}$$

Table 3.1: PSNR(dB) of stego image with data embedding capacity in PRDHHC

Cover image (I)	Secret data (bits)	PSNR (dB)	Avg. PSNR (dB)
Lena (512×512)	4,096	61.68	59.55
	16,384	59.09	
	37,376	57.89	
Barbara (512×512)	4,096	61.09	59.43
	16,384	59.25	
	37,376	57.97	
Tiffany (512×512)	4,096	61.81	59.53
	16,384	59.01	
	37,376	57.77	
Pepper (512×512)	4,096	61.39	59.29
	16,384	59.21	
	37,376	57.28	
Gold hill (512×512)	4,096	60.71	59.23
	16,384	59.16	
	37,376	57.83	

Table 3.2: PSNR(dB) of images before and after data embedding in PRDHHC

Cover image (I)	PSNR of (I & S)	PSNR of (I & C)	PSNR of (C & S)
Lena (512×512)	57.89	52.45	54.58
Barbara (512×512)	57.97	51.52	53.58
Tiffany (512×512)	57.77	52.48	54.57
Pepper (512×512)	57.28	51.97	53.43
Gold hill (512×512)	57.83	50.82	52.23

Higher the values of PSNR, better the image quality. The analysis in terms of PSNR between original image and stego image has given promising results which are shown in Table 3.1.

The PSNR of this scheme is nearer to 57 (dB) which is measured after embedding 37,376 bits within (512×512) images. Again, the quality of images are measured after redundant bits adjustment and data embedding. The cover image (C) is produced after redundant bits adjustment and stego image (S) is produced after secret bit embedding within the adjusted cover image. The original cover image (I), stego image (S) and redundant bits adjusted cover image (C) are considered to study the analysis which are shown in Table 3.2. In this experiment, it has been observed that PSNR of (C & S) is more than PSNR of (I & C), which implies that the quality is not degraded much when data bits are embedded during error creation with respect to the changes made during redundant bits adjustment. It is found that redundant bits adjustment

cost is more than the error creation for secret data bits. The {PSNR of (I & C) - PSNR of (I & S)} is the effect of redundant bits adjustment and {PSNR of (C & S)- PSNR of (I & S)} is for changes made due to data embedding. The probability of the recovered and unrecovered bits during data extraction has been explained below:

Consider the size of the block B is (7×7), redundant bits in each blocks are R_b (7×3) and data bits in each blocks are D_b (7×4). The maximum possible number of bits changes in a block during odd parity adjustment are R_b (7×3). The maximum possible number of bits changes in a block during error creation is C_b (7×2), because two errors are created in every row of the block, which are recovered during data extraction. So, the probability of unrecovered bits are $\frac{B-D_b}{B} = \frac{R_b}{B} = \frac{21}{49}$ =0.43 and the probability of recovered bits after data extraction is $\frac{28}{49}$=0.57. In Hamming +1, $k+1$ bits of message are embedded in 2^k pixels and in Hamming +3 schemes, $k+3$ bits of message are embedded in $2^k - 1$ pixels. For an example, 7 bits of message can be embedded within 64, 49, 15 pixels in Hamming +1, our approach and Hamming +3 respectively. So, our approach is better than Hamming +1 but worse than Hamming +3 in terms of data embedding rate.

The visual quality of this approach is compared with Westfeld's Matrix Coding [67], Zhang et al.'s Hamming+1 [73], Kim et al.'s Hamming+3 [29], Kim et al.'s DHHC scheme [28] and Lien et al.'s DDHHC scheme [41] which are shown in Table 3.3.

Table 3.3: Comparison of PRDHHC with other existing schemes in terms of PSNR (dB)

Cover image (I)	Matrix Coding	Hamming +1	Hamming +3	DHHC	DDHHC	PRDHHC
Lena (512×512)	56.05	52.43	48.22	31.95	37.96	57.89
Barbara (512×512)	54.65	48.60	48.22	31.95	37.90	57.97
Tiffany (512×512)	53.40	47.46	48.20	31.49	35.43	58.77
Pepper (512×512)	54.01	47.26	48.20	31.89	37.37	58.28
Gold hill (512×512)	57.02	53.73	48.21	31.94	37.97	57.83

In DHHC and DDHHC schemes they use MPSNR to calculate visual quality as they used halftone images in their experiment but gray scale images are used in our proposed scheme.

Again five gray scale images of size (512×512) has been used to compare visual quality with Westfeld's Matrix Coding (MC) [67], Zhang et al.'s Hamming+1 [73], Kim et al.'s Hamming

+3 [29] schemes, Nearest Code (NC) [8], and Cao et al.'s Scheme [4] shown in Table 3.4

Table 3.4: Comparison of PRDHHC with recently developed scheme in terms of PSNR (dB)

Images	MC	Hamming+1	NC	Hamming+3	Cao et al.'s Scheme	PRDHHC
Lena (512×512)	56.05	52.43	47.02	48.22	51.14	57.89
Barbara (512×512)	54.65	48.60	47.01	48.22	51.15	57.97
Tiffany (512×512)	53.40	47.46	47.03	48.20	51.15	58.77
Pepper (512×512)	54.01	47.26	47.02	48.20	51.14	58.28
Gold hill(512×512)	57.02	53.73	47.02	48.21	51.14	57.83

It is observed that PSNR of PRDHHC is more than other existing schemes which is nearer to 58 (dB). The PSNR is 1.84 (dB) more than Matrix Coding and 10 (dB) more that Hamming $+3$ and Nearest Coding. Also, PRDHHC has been compared with most recently developed Cao et al.'s scheme [4] published in 2016 and shown that the PRDHHC is 6 (dB) more than that. Hamming $+3$ is lower PSNR than Hamming $+1$ due to the higher embedding capacity which is shown in Table 3.4.

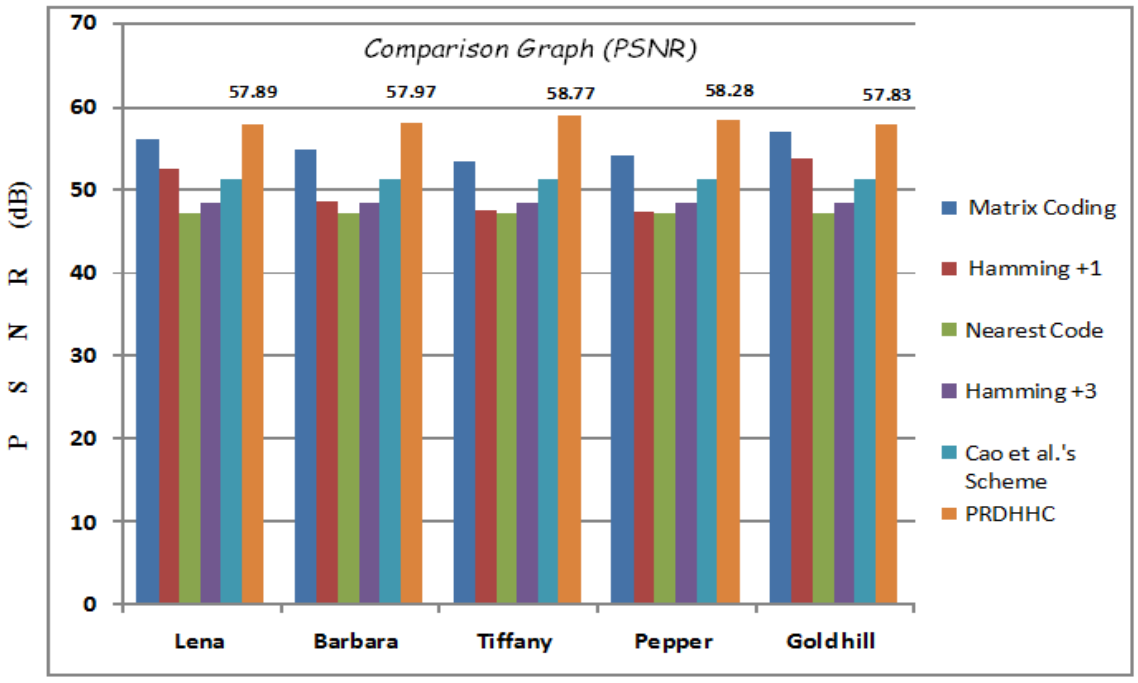


Figure 3.4: Comparison graph of PRDHHC with existing schemes in terms of PSNR (dB)

From the Fig. 3.4, we concluded that after embedding maximum number of bits within the cover image, the PSNR in PRDHHC scheme is more than other existing schemes. As a result,

visual quality of proposed PRDHHC is better than other existing schemes.

3.2.4 Steganalysis

Steganalysis is the art of discovering whether or not a secret message exists in a suspected image. Steganalysis does not however consider the successful extraction of the message. Here, we have analyzed our generated stego image using RS analysis, statistical analysis and compute the results of relative entropy.

3.2.4.1 RS Analysis

The results of RS analysis shown in Table 3.5. After embedding 37,376 bits within Lena (512×512) image it is observed that the values of R_M = 6304, R_{-M} = 5947, S_M = 4983, S_{-M} = 5029 and the corresponding RS value is 0.0357 which is nearly equal to zero. Thus rule $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ is satisfied for the stego image in this scheme. The experimental result of other tested images also satisfing the said rules for RS analysis. So, the proposed method is secure against RS attack.

Table 3.5: Experimental results of RS analysis in PRDHHC

Cover image (I)	Data (bits)	Stego image (S)				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Lena (512×512)	4096	7118	7107	3551	3594	0.0051
	16384	6768	6851	3944	3895	0.0123
	37376	6304	5947	4983	5029	0.0357
Barbara (512×512)	4,096	5627	5607	4067	4098	0.0011
	16,384	5563	5476	4291	4337	0.0135
	37,376	5636	5439	4717	4589	0.0314
Tiffany (512×512)	4,096	5897	5875	5076	5131	0.0070
	16,384	5893	5815	4960	5105	0.0205
	37,376	6018	5813	5107	5313	0.0369
Pepper (512×512)	4,096	6458	6451	6312	6213	0.0083
	16,384	6325	6236	6102	5824	0.0295
	37,376	6304	6076	6015	5814	0.0348
Gold hill (512×512)	4,096	5002	4996	5044	4965	0.0084
	16,384	5012	4876	5003	4856	0.0282
	37,376	5468	5252	5124	4954	0.0364

3.2.4.2 Relative Entropy

The results of relative entropy are shown in Table 3.6. After embedding 37,376 secret bits

Table 3.6: Experimental results of relative entropy in PRDHHC

Cover image (I)	Data (bits)	Entropy of (I)	Entropy of (S)	Relative entropy
Lena (512×512)	16,384	7.4451	7.4492	0.0041
	37,376	7.4451	7.4520	0.0069
Barbara (512×512)	16,384	7.0480	7.0520	0.0040
	37,376	7.0480	7.0580	0.0100
Tiffany (512×512)	16,384	7.2925	7.2974	0.0049
	37,376	7.2925	7.2996	0.0071
Pepper (512×512)	16,384	7.2767	7.2798	0.0031
	37,376	7.2767	7.2821	0.0054
Gold hill (512×512)	16,384	7.2367	7.2409	0.0042
	37,376	7.2367	7.2427	0.0060

within Lena image, the relative entropy is 0.0041 which is nearer to zero. It is also shown that when the number of bits in the secret message increases, the relative entropy also increases. The relative entropy nearer to zero implies the proposed scheme is secure.

3.2.4.3 Statistical Analysis

The proposed scheme is also assessed based on statistical distortion analysis by some image parameters like standard deviation (SD) (σ) and correlation coefficient (CC) (ρ) to check the impact on image after data embedding. The σ before and after data embedding and ρ of original and stego image are summarized in Table 3.7. Minimizing parameter difference is one of the

Table 3.7: Experimental results of SD (σ) and CC (ρ) in PRDHHC

Image	SD (σ)		CC (ρ)
	Cover image (I)	Stego image (S)	(I) versus (S)
Lena (512×512)	47.8385	47.4358	0.9864
Barbara (512×512)	38.3719	37.8500	0.9820
Tiffany (512×512)	61.5978	61.1221	0.9913
Pepper (512×512)	52.1356	51.2587	0.9908
Gold hill (512×512)	58.8723	57.2854	0.9867

primary aims in order to get rid of statistical attacks. From Table 3.7 it is seen that there is no substantial divergence between the standard deviation of the cover image and the stego image. The σ of original image is 47.8385 and the σ of stego image is 47.4358, and the difference is

0.4027 for lena image. The ρ of Lena image between the original cover image I and stego image S is 0.9864 which implies the change in the original image will result in a change in the same direction in the stego image. So it is hard to locate the embedding position in the stego image. This study shows that the magnitude of change in stego image based on image parameters is small from the original image. Since the image parameters have not changed much, the method offers a good concealment of data and reduces the chances of the secret data being detected. Thus, it indicates a perfectly secure steganographic system. But the scheme is not reversible.

3.3 Dual Image based Reversible Data Hiding using (7,4) Hamming code (DRDHHC) [2,3]

To achieve reversibility, a new dual image based data hiding scheme through (7,4) Hamming code (RDHHC) using shared secret key has been proposed. A block of seven pixels and their Least Significant Bits (LSBs) are collected from cover image and copied into two arrays then redundant bits are adjusted using odd parity such that any error creation is encountered at the sender end and recovered at the receiver end. Before data embedding, it is required to complement the bit at shared secret position. After that, secret message bit is embedded through error creation caused by tamper in any suitable position except secret position and that error is detected as well as corrected at the receiver end using Hamming error correcting code. One shared secret position κ and one shared secret key ξ helps to perform data embedding, data extraction operation. The secret data and original cover image are successfully recovered at the receiver end from dual stego image.

The schematic diagram of data embedding process is depicted in Fig. 3.5. During data embedding, a special situation may arise where no suitable location is available for data embedding. In that case, spacial location has been fixed depending on the value of secret data bit and the value of LSB bits in the array. The corresponding example is shown in Fig. 3.6. The

[2]Published in the proceedings of the International Congress on Information and Communication Technology (ICICT - 2015), Advances in Intelligent Systems and Computing, Springer, Vol. 439, pp.495-504, with title *Reversible Data Hiding through Hamming Code using Dual Image.*

[3]Minor Review Submitted in **Multimedia Tools and Application**, Springer, Impact Factor 1.346, with title *Dual Image based Reversible Data Hiding Scheme using (7,4) Hamming Code.*

diagram of secret data extraction and cover image reconstruction process is shown in Fig. 3.7. The corresponding pseudo code for data embedding and data extraction are given in Algorithm 3 and Algorithm 4 respectively.

3.3.1 Data Embedding Process

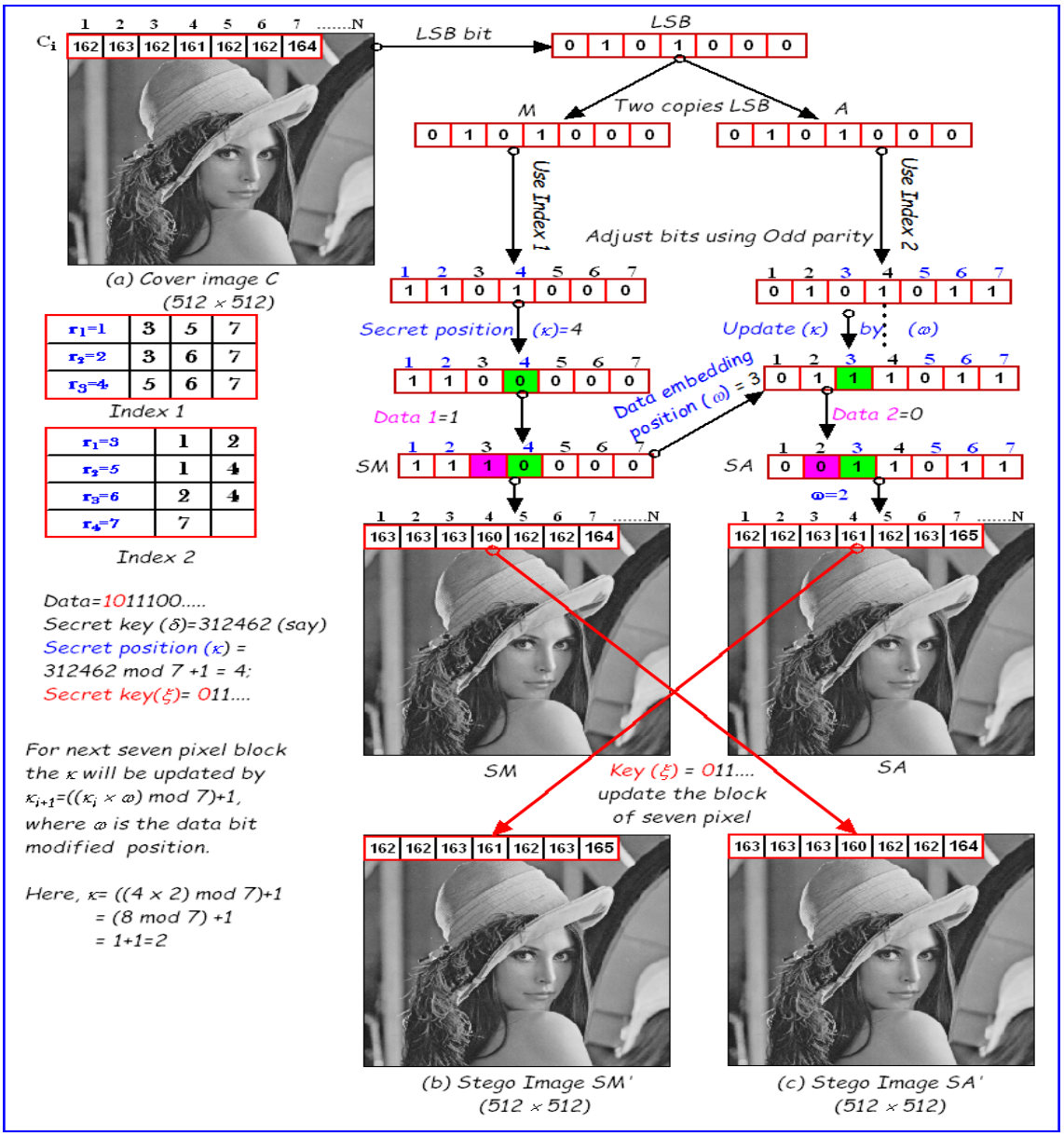


Figure 3.5: Schematic diagram of data embedding process in DRDHHC

First, we collect 7 consecutive pixels from the cover image (C). Then collect LSBs of those pixels and copy it into two arrays M and A. Then we adjust redundant bits of both the arrays separately using odd parity check. The redundant bits r_1, r_2 and r_3 of M array are adjusted

based on the number of 1s present in the bit positions which are mentioned in Index 1 of Fig. 3.5. For example, the r_1 bit is set to 1 if the number of 1 present in the bit positions at 3, 5 and 7 of M array are even. The redundant bits r_1, r_2, r_3 and r_4 of A array are also adjusted and updated in the bit positions at 3, 5, 6 and 7 of A array depending upon the number of 1 present in the bit positions mentioned in Index 2 of Fig. 3.5. Two shared secret keys δ and ξ are used to enhance security. The secret position κ is calculated using $\kappa = (\delta \mod 7) + 1$, where δ is a shared secret key. The κ is used during data embedding process and ξ is used during distribution of pixel block among dual image.

After adjustment of all redundant bits in both the arrays, complement the bit at κ-th position (here it is 4 in Fig. 3.5) of the M, then embed secret data bit (D) (here it is 1) by error creation in any suitable position except the secret position (κ). Here, any position from 3, 5, 6 and 7 will be a suitable position for data embedding using error creation because these positions contain zero and we try to embed one. Now, create error at any suitable position by complementing the value at that position (here it is 3 shown in Fig. 3.5). As a result, the data embedding position ω is updated by 3. If M contain all 0s and data bit is 1 then create error at any suitable position except κ position and set ω accordingly but if M contains all 0s and data bit is also 0 then it is not possible to create error by complementing because no suitable positions are available. In that case, the position 1 has been considered as the data embedding position and set ω = 1 without creating any error by complement. It is noted that all elements of the array M can not be 1 after parity bits adjustment using odd parity by Index 1 of Fig. 3.5, but there is a possibility to be all 0 after complement the bit at the 7-th position. The possible cases when no error creation is possible and condition for completion of execution are explained in Fig. 3.6.

Now, the data embedding position (ω) of M is used as secret position (κ) during data embedding in array A. Then we perform same data embedding procedure on A to embed next data bit by error creation. After creating error at secret position, create another error to embed data bit, and 2, 4, 6 and 7 bit positions are the suitable ones (here it is 2 in Fig. 3.5). If all elements of A are 1, and data bit is 0 then it is possible to create an error at any suitable position excluding κ-th position and then set ω accordingly. If all elements of A are 1 and data bits are also 1 then it is not possible to create an error by complement. In that case, the position 7 is the data embedding

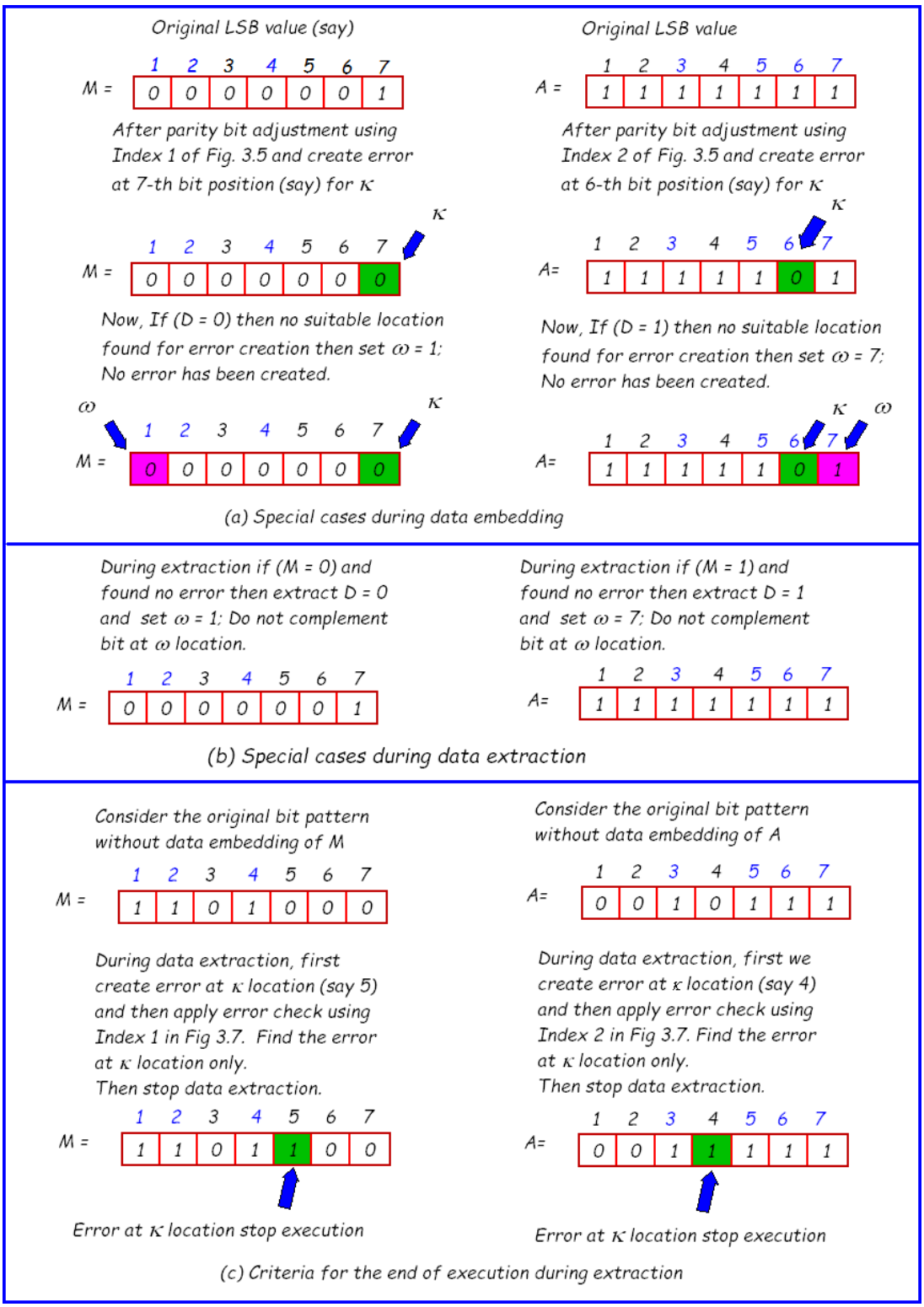


Figure 3.6: Special cases (a) Data embedding (b) Data extraction (c) Stop execution

position and set $\omega = 7$ without creating any error. It is noted that all elements of the array A can not be 0 after parity bit adjustment using odd parity through Index 2 in Fig. 3.5, but all bits

Input: Cover image C of size $(m \times n)$, Shared secret key δ and another secret key ξ, Secret data D.

Output: Two stego images namely Stego Major ($SM^{'}$) and Stego Auxiliary ($SA^{'}$) each of size $(m \times n)$.

Step 1: Calculate secret position κ = $(\delta \mod 7) + 1$, where δ is a shared secret key;

Step 2: $C^i_{(1\times7)} \leftarrow C_{(m\times n)}$, where $i = 1, 2, 3, \ldots, (m \times n/7)$; $p = i * 2$;

Step 3: $C^i_{LSB}(1 \times 7) \leftarrow$ LSB of $C^i_{(1\times7)}$;

Step 4: $M^i_{(1\times7)} \leftarrow$ modify redundant bit of C^i_{LSB} by odd parity using Index 1 of Fig 3.5;

Step 5: $A^i_{(1\times7)} \leftarrow$ modify redundant bit of C^i_{LSB} by odd parity using Index 2 of Fig 3.5;

Step 6: Complement $M^i_{(1\times7)}(\kappa)$, where κ is a shared secret value between 1 to 7;

Step 7: Create an error by complement for data bit at any suitable position (ω) of $M^i_{(1\times7)}$ except κ position;

Step 8: **if** $((D_{p-1} == 0)$ *&* $(M^i_{(1\times7)} == 0))$ **then** Set $\omega \leftarrow 1$;

{*** No suitable location found ***}

Step 9: κ is updated by ω for data embedding in $A^i_{(1\times7)}$

Step 10: Complement $A^i_{(1\times7)}(\kappa)$, where κ is a shared secret position between 1 to 7;

Step 11: Create an error by complement for data bit at any suitable position (ω) of $A^i_{(1\times7)}$ except κ position;

Step 12: **if** $((D_{p-1} == 1)$ *&* $(A^i_{(1\times7)} == 1))$ **then** Set $\omega \leftarrow 7$;

{*** No suitable location found ***}

Step 13: κ = $((\kappa \times \omega) \mod 7) + 1$;

Step 14:

if $C^i_{LSB}(x) == M^i(x)$ **then**
  $SM^i(x) \leftarrow C^i(x)$; where $x = 1, 2, \ldots, 7$
end

if $C^i_{LSB}(x)$*==1* **then**
  $SM^i(x) \leftarrow C^i(x) - 1$
else
  $SM^i(x) \leftarrow C^i(x) + 1$
end

Step 15:

if $C^i_{LSB}(x) == A^i(x)$ **then**
  $SA^i(x) \leftarrow C^i(x)$; where $x = 1, 2, \ldots, 7$
end

if $C^i_{LSB}(x)$*==1* **then**
  $SA^i(x) \leftarrow C^i(x) - 1$
else
  $SA^i(x) \leftarrow C^i(x) + 1$;
end

Step 16: Apply ξ and distribute each block within $SM^{'}$ and $SA^{'}$ as follows:

Step 17:

if $(\xi(y) = 0)$ **then**
  $SM'^i_{1\times7} \leftarrow SA^i_{1\times7}$ and $SA'^i_{1\times7} \leftarrow SM^i_{1\times7}$
else
  $SM'^i_{1\times7} \leftarrow SM^i_{1\times7}$ and $SA'^i_{1\times7} \leftarrow SA^i_{1\times7}$; where $y \leftarrow (remainder(i, length(\xi)) + 1)$
end

Step 18: Repeat **Step 1** to **Step 17** for next i^{th} value and continue until all data bits are embedded within $SM^{'}$ and $SA^{'}$;

Step 19: End

Algorithm 3: Data embedding process of DRDHHC

can be 1. Fig. 3.6 showns the possible special cases. After that, update κ for next block using $\kappa_{i+1} = ((\kappa_i \times \omega) \mod 7) + 1$, where κ_0 is the secret position and continue data embedding

procedure in both the arrays. The changing of secret key for each block increases the robustness of the scheme. Finally, we distribute the stego pixel block between two stego images $SM^{'}$ and $SA^{'}$ depending on the bit patterns of another secret key $\xi = (\xi_0\xi_2 \ldots \xi_l)_2$, where l represent the binary length of the key. If $\xi_i = 1$ then selected i-th pixel from M is stored the i-th position of $SM^{'}$ and i-th pixel from A is stored at i-th position of $SA^{'}$; otherwise, i-th pixel from M is stored at i-th position of $SA^{'}$ and i-th pixel from A is stored at i-th position of $SM^{'}$. As a result, two stego images $SM^{'}$ and $SA^{'}$ with same size and shape has been produced. The dual image scheme can increase the security of the secret message. Without two stego images, it is hard for an illegal person to extract the complete secret message. The concept of the dual image based data hiding scheme can be treated as a special case of secret sharing. The corresponding algorithm for data hiding is stated in Algorithm 3. The complexity of this proposed data embedding algorithm is $\mathcal{O}(mn)$ for the cover image of size $(m \times n)$.

3.3.2 Data Extraction Process

To extract the secret data, we first apply the secret key $\xi = (\xi_0\xi_2 \ldots \xi_l)_2$ to rearrange 7 pixels as a block. If $\xi_i = 1$ then selected block from $SM^{'}$ is stored in SM_i and block from $SA^{'}$ is in SA_i; otherwise, block from $SM^{'}$ is stored in SA_i and block from $SA^{'}$ is in SM_i. Now, we collect LSBs of seven consecutive pixels from both the stego images SM_i and SA_i. Then complement the bit at the shared secret position κ of the SM_i. The κ is calculated by $\kappa = (\delta \mod 7) + 1$, where δ is a shared secret key. After that, we apply Hamming error correcting code to find out the error position (ω). Next, we extract the secret data bit from the ω position and complement the bit at that position. The redundant bits of SM_i stego image are considered as per Index 1 in Fig. 3.7. The error position of SM_i is the data embedding position which is used as the secret position for SA_i image. Complement the bit at the secret position in SA_i then we find out the error using Hamming error correcting code. The redundant bits in Index 2 of Fig. 3.7 are considered for SA_i image.

Now, the secret position κ is updated for the next block using the formula $\kappa_{i+1} = ((\kappa_i \times \omega) \mod 7) + 1$, where ω is the data embedding position of SA_i (here it is 3 in Fig 3.7) and κ_0 is the shared secret position. After extracting secret message bits from dual stego images, we complement all the corresponding data embedding positions, which will produce Hamming

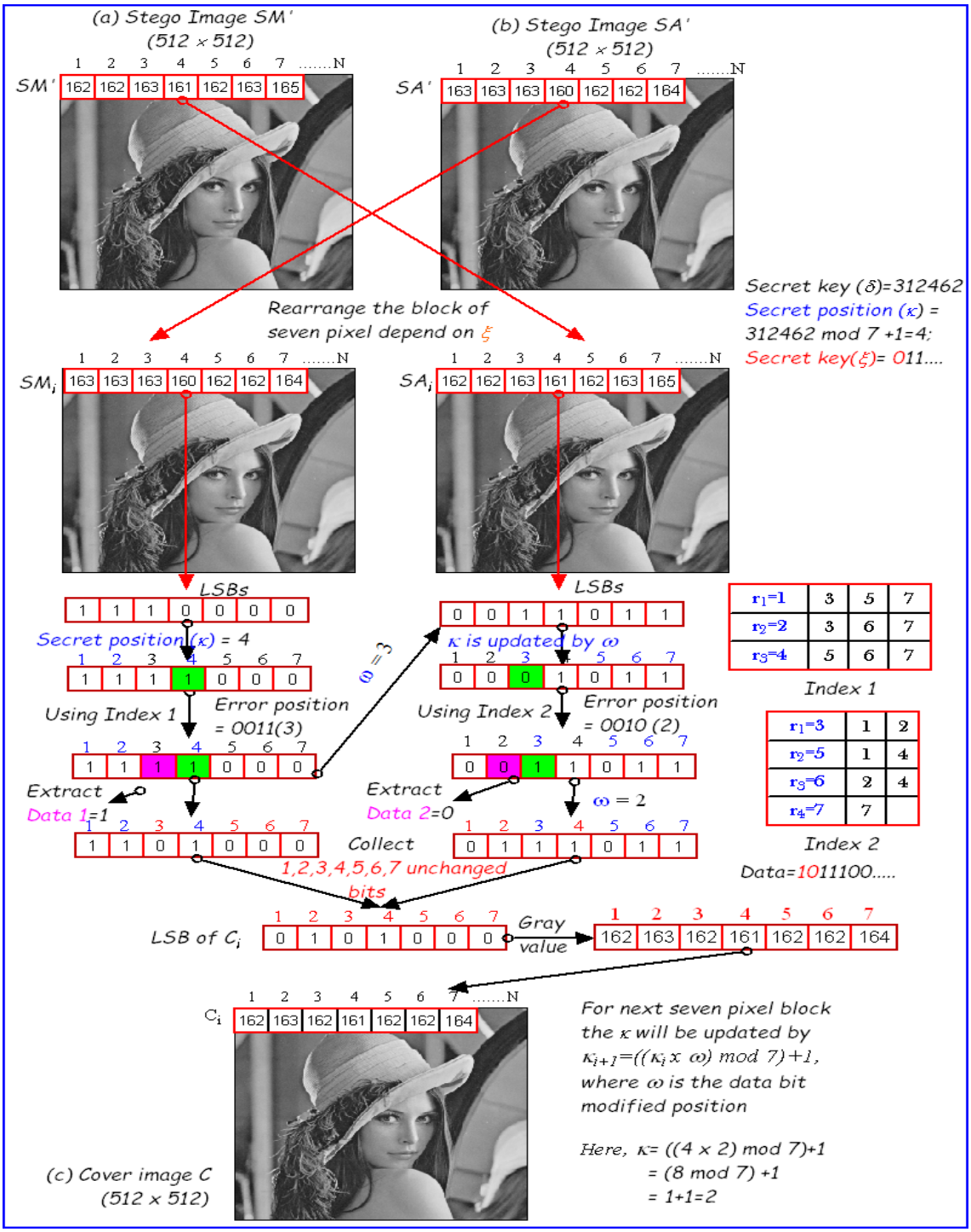


Figure 3.7: Schematic diagram of data extraction process in DRDHHC

adjusted cover image. There are a few particular special cases that may occurs when no error is found. In that situation, we will first check the LSB values and follow the condition given below. For all the cases, we need not to complement the bit at ω-th location after extraction.

If SM_i = 0 and find no error then extract $D = 0$ and set $\omega = 1$.

If SA_i = 1 and find no error then extract $D = 1$ and set $\omega = 7$.

Input: Two stego images namely Stego Major ($SM^{'}$) and Stego Auxiliary ($SA^{'}$), Shared secret key δ and another secret key ξ.

Output: Cover image C of size $(m \times n)$, Secret data D.

Step 1: Calculate secret position κ = $(\delta \mod 7) + 1$, where δ is a shared secret key.

Step 2:

if *($\xi(y)$ = 0)* **then**
$SM^i_{(1\times7)} \leftarrow SA'^i_{(1\times7)}$, $SA^i_{(1\times7)} \leftarrow SM'^i_{(1\times7)}$

else
$SM^i_{(1\times7)} \leftarrow SM'^i_{(1\times7)}$, $SA^i_{(1\times7)} \leftarrow SA'^i_{(1\times7)}$, where $y = (reminder(i, length(\xi)) + 1)$ and i = $1, 2, 3, \ldots, (m \times (n/7))$

end

Step 3: $M_{(m\times n)} \leftarrow LSB(SM_{(m\times n)})$, $A_{(m\times n)} \leftarrow LSB(SA_{(m\times n)})$;

Step 4: For each i, where i = 1,2,. . . , $(m \times (n/7))$;

Step 5: $M^i_{(1\times7)} \leftarrow M_{(m\times n)}$, $A^i_{(1\times7)} \leftarrow A_{(m\times n)}$;

Step 6: Complement $M^i_{(1\times7)}(\kappa)$, where κ is a shared secret value between 1 to 7;

Step 7: $r_{(1\times3)} \leftarrow M^i_{(1\times7)}$, where r is the redundant value calculated by odd parity using Index 1 of Fig. 3.7;

Step 8: $\omega \leftarrow decimal(r_{(1\times3)})$, where ω indicate the error position or data embedding position;

Step 9: **if** *($(\omega \neq 0)$ & $(\omega \neq \kappa)$)* **then** $D_{p-1} \leftarrow M^i_{(1\times7)}(\omega)$; $M^i_{(1\times7)}(\omega) \leftarrow$ Complement($M^i_{(1\times7)}(\omega)$);

Step 10: **if** *($(\omega == 0)\&(M^i_{(1\times7)} == 0)$)* **then** $D_{p-1} \leftarrow M^i_{(1\times7)}(1)$; $\omega \leftarrow 1$;

Step 11: **if** $(\omega == \kappa)$ **then** goto **Step 24**;

Step 12: κ is updated by ω for data extraction from $A^i_{(1\times7)}$;

Step 13: Complement $A^i_{(1\times7)}(\kappa)$, where κ is a shared secret value between 1 to 7;

Step 14: $r_{(1\times3)} \leftarrow A^i_{(1\times7)}$, where r is the redundant value calculated by odd parity using Index 2 of Fig. 3.7;

Step 15: $\omega \leftarrow decimal(r_{(1\times3)})$, where ω indicated the error position or data embedding position;

Step 16: **if** *($(\omega \neq 0)$ & $(\omega \neq \kappa)$)* **then** $D_p \leftarrow A^i_{(1\times7)}(\omega)$; $A^i_{(1\times7)}(\omega) \leftarrow$ Complement($A^i_{(1\times7)}(\omega)$);

Step 17: **if** *($(\omega == 0)\&(A^i_{(1\times7)} == 1)$)* **then** $D_p \leftarrow A^i_{(1\times7)}(7)$; $\omega \leftarrow 7$;

Step 18: **if** $(\omega == \kappa)$ **then** goto **Step 24**;

Step 19: **if** *LSB of* $(SM^i(x))$=$M^i(x)$ **then** $C'^i(x)$=$SM^i(x)$;

Step 20:

if $M^i(x)$=*1* **then**
$C'^i(x) = SM^i(x) + 1$;

else
$C'^i(x) = SM^i(x) - 1$; where $x = 3, 5, 6, 7$

end

Step 21: **if** *LSB of* $(SA^i(x))$=$A^i(x)$ **then** $C'^i(x)$=$SA^i(x)$;

Step 22:

if $A^i(x)$=*1* **then**
$C'^i(x) = SA^i(x) + 1$;

else
$C'^i(x) = SA^i(x) - 1$; where $x = 1, 2, 4$

end

Step 23: κ = $((\kappa \times \omega) \mod 7) + 1$, goto **Step 5** for next i^{th} value;

Step 24: Original cover image matrix $C'_{(m\times n)}$ and secret data D' has been produced;

Step 25: End

Algorithm 4: Data extraction process of DRDHHC

For all other cases, if error is found at the position of κ that means when ω is equal to κ then data extraction process will be stopped. This is the condition to find the end of secret message.

As a result, we can send any arbitrary length of secret data using this scheme. Finally, we can reconstruct the original cover image from both the Hamming adjusted dual stego image. We collect the bits from the bit positions at 3, 5, 6 and 7 of SM_i and bits from the bit positions at 1, 2 and 4 of SA_i. After that, we rearrange all the collected bits to construct the original cover image. The algorithm for data extraction and cover image reconstruction is described in Algorithm 4. The time complexity of data extraction algorithm is also $\mathcal{O}(mn)$ for cover image of size $(m \times n)$.

3.3.3 Experimental Results and Comparisons

The proposed reversible data hiding method is implemented through MATLAB Version 7.6.0.324 (R2008a). (512×512) gray scale images are used as original cover image and in this scheme which are shown in Fig. 3.8 and dual stego images are generated that are shown in Fig. 3.9.

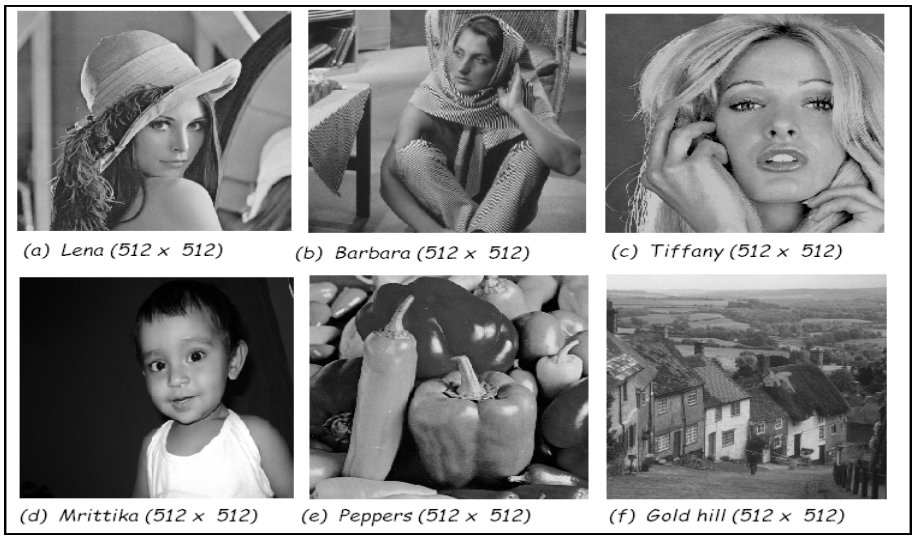


Figure 3.8: Standard cover images are used for experiment in DRDHHC

PSNR (I & $SM^{'}$) and PSNR (I & $SA^{'}$) represent the quality of the first and second stego images respectively while Avg. PSNR is the average quality of the stego images. From the Table 3.8 average PSNR of DRDHHC scheme is greater than 53 (dB) and the maximum embedding capacity is $2 \times (512 \times 512/7)$ bits = $74,898$ bits. The payload is measured by $p = \frac{\gamma}{(2 \times m \times n)}$ (bpp), where γ is the total embedding capacity of two stego images. The payload in DRDHHC scheme is 0.142 (bpp). The principle of hidden data communication is to keep message data as short as possible while using data hiding and kirchhoff's principle says that every one know the algorithm but the secrecy depends on key. So, to increase the security on data hiding two

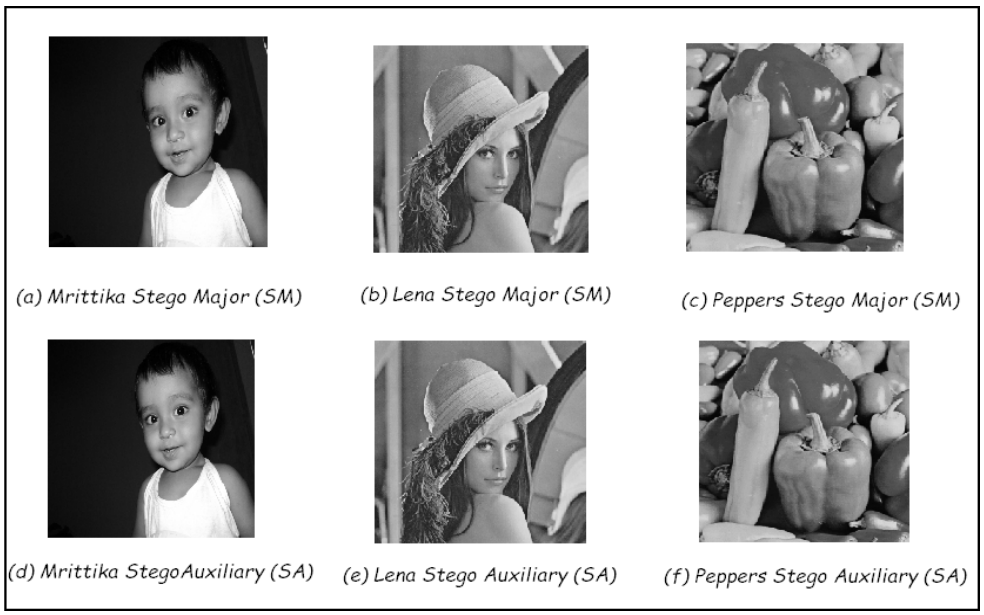


Figure 3.9: Dual stego images are produced after data embedding in DRDHHC

Table 3.8: PSNR (dB) of original image and dual stego images in DRDHHC

Original image I	Secret data (bits)	PSNR (I & $SM^{'}$)	PSNR (I & $SA^{'}$)	Avg. PSNR
Barbara (512×512)	18720	57.84	57.88	54.27
	37520	54.84	54.84	
	64800	52.47	52.47	
	74752	51.86	51.96	
Lena (512×512)	18720	57.55	57.64	53.96
	37520	54.27	54.34	
	64800	52.18	52.29	
	74752	51.85	51.56	
Peppers (512×512)	18720	57.47	57.50	53.94
	37520	54.22	54.20	
	64800	52.15	52.25	
	73728	51.84	51.94	
Mrittika (512×512)	18720	57.27	57.70	54.02
	37520	54.20	54.45	
	64800	52.63	52.74	
	73728	51.65	51.94	
Tiffany (512×512)	18720	57.56	57.59	53.63
	37520	54.29	54.28	
	64800	52.21	52.31	
	74752	51.88	51.97	
Goldhill (512×512)	18720	57.11	57.90	53.73
	37520	54.28	54.29	
	64800	52.17	52.26	
	74752	51.86	51.97	

different shared secret keys are used in this approach during data embedding and data extraction procedure.

The quality of stego image is measured using Peak Signal to Noise Ratio $(PSNR)$ between original cover image and stego image which are shown in Table 3.8 with different embedding capacity.

Table 3.9 shows that the PSNR of cover image versus both stego images before distribution of the pixel block and after distribution of the pixel block among dual image. It is observed that the PSNR of before pixel block distribution of cover image and SM is 53.79 (dB) and PSNR of cover image and SA is 50.23 (dB) and after pixel block distribution the PSNR of cover image and $SM^{'}$ is 51.85 (dB) that means PSNR is decrease by 1.94 (dB) than previous PSNR and PSNR of cover image and $SA^{'}$ is 51.56 (dB) that means PSNR is increases 1.33 (dB) for Lena image. So, after distribution the pixel block through ξ, overall PSNR is not differ much but increase the challenges to find the secret data. The comparison of DRDHHC scheme with other

Table 3.9: PSNR(dB) of before and after pixel distribution in DRDHHC

	Before distribution using ξ		After distribution using ξ	
Original Image I	PSNR (I & SM)	PSNR (I & SA)	PSNR (I & $SM^{'}$)	PSNR (I & $SA^{'}$)
Lena (512×512)	53.79	50.23	51.85	51.56
Barbara (512×512)	53.82	50.21	51.86	51.96
Tiffany (512×512)	53.80	50.22	51.88	51.97
Pepper (512×512)	53.79	50.23	51.84	51.94
Mrittika (512×512)	53.13	50.45	51.65	51.94
Gold hill (512×512)	53.76	50.23	51.86	51.97

existing methods those are using Hamming code based data hiding shown in Table 3.10.

Three sets of secret data bits $65,536$, $16,384$ and $4,096$ are used to compare with Kim et al.'s [28] (DHHC) and Lien et al.'s [41] (DDHHC) scheme. With the same embedding capacity with $16,384$ bits the MPSNR of DHHC is 24.06 (dB) less and DDHHC is 17.49 (dB) less than proposed DRDHHC. Again the proposed scheme is compared with Matrix Coding [67], Hamming $+1$ [73], Hamming $+3$ [29] and Coa et al.'s scheme [4] in terms of PSNR and it is nearer to the Hamming $+3$ scheme but less than Matrix Coding which is shown in Table 3.11. The proposed DRDHHC is a dual image based RDH scheme using (7,4) Hamming code.

Table 3.10: Comparison of DRDHHC with DHHC and DDHHC schemes

Embedded bits	65536 bits			16384 bits			4096 bits		
Scheme	DHHC	DDHHC	DRDHHC	DHHC	DDHHC	DRDHHC	DHHC	DDHHC	DRDHHC
Lena (512×512)	25.71	30.57	52.20	32.03	38.60	56.09	38.12	44.71	64.68
Barbara (512×512)	25.69	30.38	52.21	32.03	38.57	56.86	38.14	44.77	64.09
Tiffany (512×512)	24.71	27.15	52.11	31.72	36.24	56.73	38.04	42.91	63.81
Pepper (512×512)	25.60	29.90	52.12	31.98	38.02	56.16	38.10	44.19	64.39
Gold hill (512×512)	25.66	30.30	52.20	32.02	38.66	56.94	38.13	44.96	64.71

Table 3.11: Comparison of DRDHHC with existing schemes in terms of PSNR (dB)

Images	Matrix Coding	Hamming +1	Hamming +3	Coa et al.'s Scheme	DRDHHC
Lena (512×512)	56.05	52.43	53.95	51.14	53.96
Barbara (512×512)	54.65	48.60	53.93	51.15	54.27
Tiffany (512×512)	53.98	47.46	53.96	51.15	53.63
Pepper (512×512)	54.01	47.26	53.95	51.14	53.94
Gold hill (512×512)	57.02	53.73	53.95	51.14	53.73

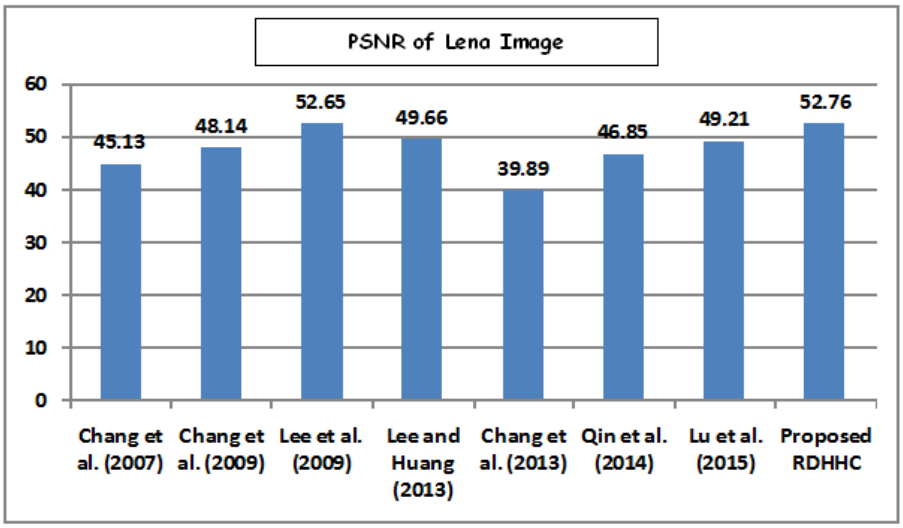


Figure 3.10: Comparison graph in terms of PSNR (dB) for Lena Image

So, again we have compared our proposed scheme with other existing dual image based RDH schemes proposed by Chang et al. (2007) [5], Chang et al. (2009) [6], Lee et al. (2009) [36], Lee and Huang (2013) [34], Chang et al. (2013) [10], Qin et al. (2014) [52] and Lu et al. (2015) [45]. The PSNR of DRDHHC is 7.63 (dB) greater than Chang et al.'s [5] scheme and nearer to Lee et al.'s [36] scheme. The quality of stego image of other existing schemes are lower than our proposed scheme. In this scheme, payload is lower than other dual image based

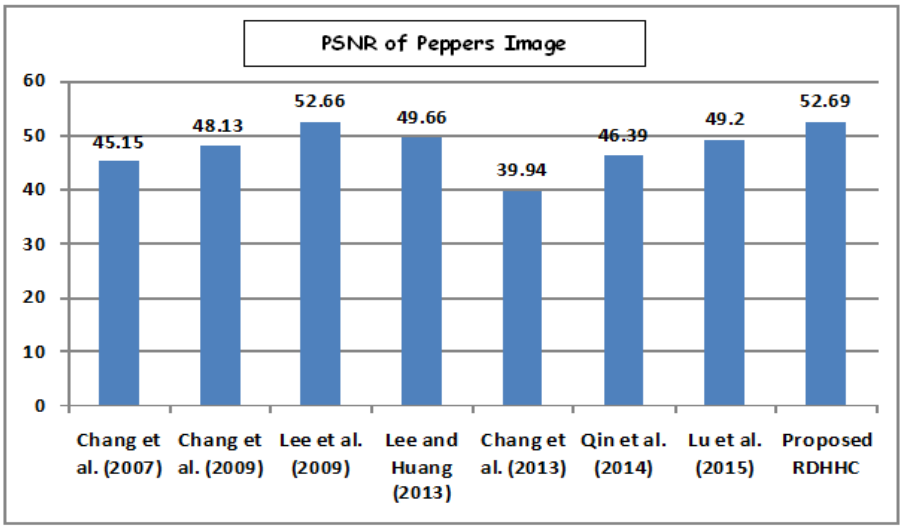


Figure 3.11: Comparison graph in terms of PSNR (dB) for Pepper Image

RDH schemes, but in terms of PSNR it is superior than other existing dual image based RDH schemes. Here, we have used shared secret position κ and shared secret key ξ to enhances the

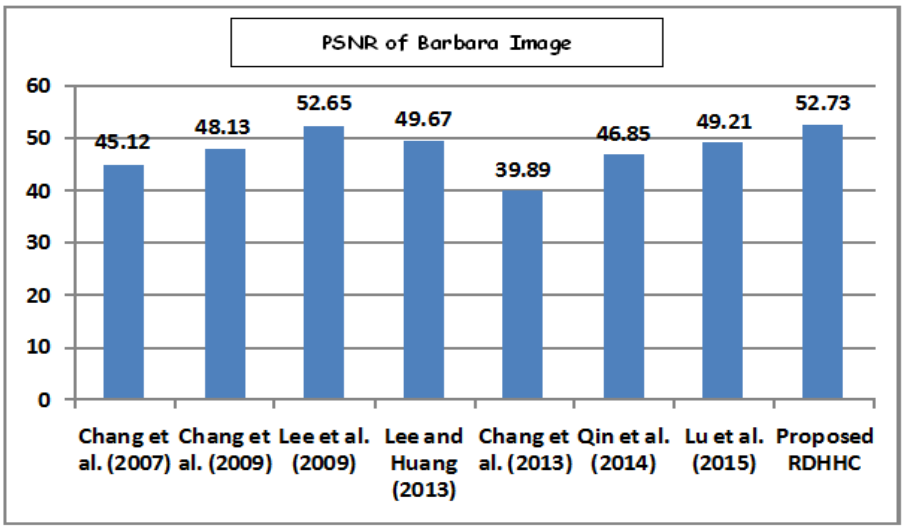


Figure 3.12: Comparison graph in terms of PSNR (dB) for Barbara Image

security in (7,4) Hamming code based data hiding scheme. It is hard to guess the length of ξ which is used to mix the pixel block among dual image. The Figs. 3.10, 3.11, 3.12 and 3.13 show the comparison graphs of Lena, Peppers, Barbara and Goldhill images with other existing dual image based RDH schemes.

From these comparison graphs, we conclude that PSNR of our proposed scheme is more than the other existing dual image based RDH schemes. As a result, visual quality is better in our

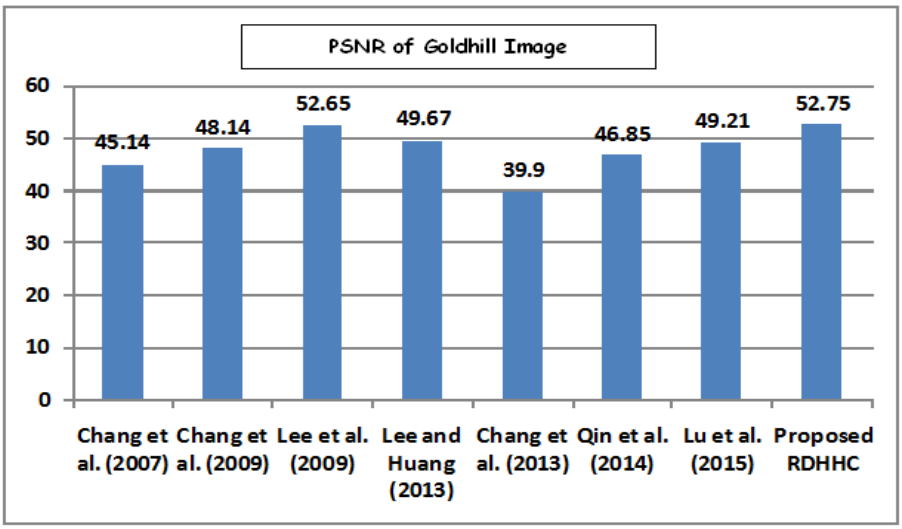


Figure 3.13: Comparison in terms of PSNR for Goldhill Image

proposed scheme than other exiting scheme shown in Table 3.12. In this scheme, embedding

Table 3.12: Comparison of DRDHHC with existing dual image based RDH methods

Method	Measure	Lena	Peppers	Barbara	Goldhill
Chang et al. (2007)	PSNR(1)	45.12	45.14	45.13	45.13
	PSNR(2)	45.13	45.15	45.11	45.14
	PSNR(Avg)	45.13	45.15	45.12	45.14
Chang et al. (2009)	PSNR(1)	48.13	48.11	48.14	48.13
	PSNR(2)	48.14	48.14	48.11	48.15
	PSNR(Avg)	48.14	48.13	48.13	48.14
Lee et al. (2009)	PSNR(1)	51.14	51.14	51.14	51.14
	PSNR(2)	54.16	54.17	54.16	54.16
	PSNR(Avg)	52.65	52.66	52.65	52.65
Lee and Huang (2013)	PSNR(1)	49.76	49.75	49.75	49.77
	PSNR(2)	49.56	49.56	49.58	49.57
	PSNR(Avg)	49.66	49.66	49.67	49.67
Chang et al. (2013)	PSNR(1)	39.89	39.94	39.89	39.9
	PSNR(2)	39.89	39.94	39.89	39.9
	PSNR(Avg)	39.89	39.94	39.89	39.90
Qin et al. (2014)	PSNR(1)	52.11	51.25	52.12	52.12
	PSNR(2)	41.58	41.52	41.58	41.58
	PSNR(Avg)	46.85	46.39	46.85	46.85
Lu et al. (2015)	PSNR(1)	49.20	49.19	49.22	49.23
	PSNR(2)	49.21	49.21	49.2	49.18
	PSNR(Avg)	49.21	49.20	49.21	49.21
Proposed DRDHHC	PSNR(1)	52.71	52.67	52.70	52.73
	PSNR(2)	52.81	52.72	52.76	52.78
	PSNR(Avg)	52.76	52.69	52.73	52.75

capacity is lower than other dual image based scheme but in terms of security it is better than other existing dual image based schemes, because two shared secret keys δ and ξ are used during data communication. The new shared secret position κ is calculated by $\kappa = (\delta \mod 7) + 1$, where maximum possible numbers of κ are $\lfloor(512 \times 512/7)\rfloor$. It is hard to guess the length of ξ which is used to shuffle the pixel block among dual image. The comparison graph among dual image based reversible data hiding is shown in Fig. 3.14. It is observed that PSNR of this proposed scheme is more than the other existing dual image based schemes. But the payload is 0.142 (bpp) only.

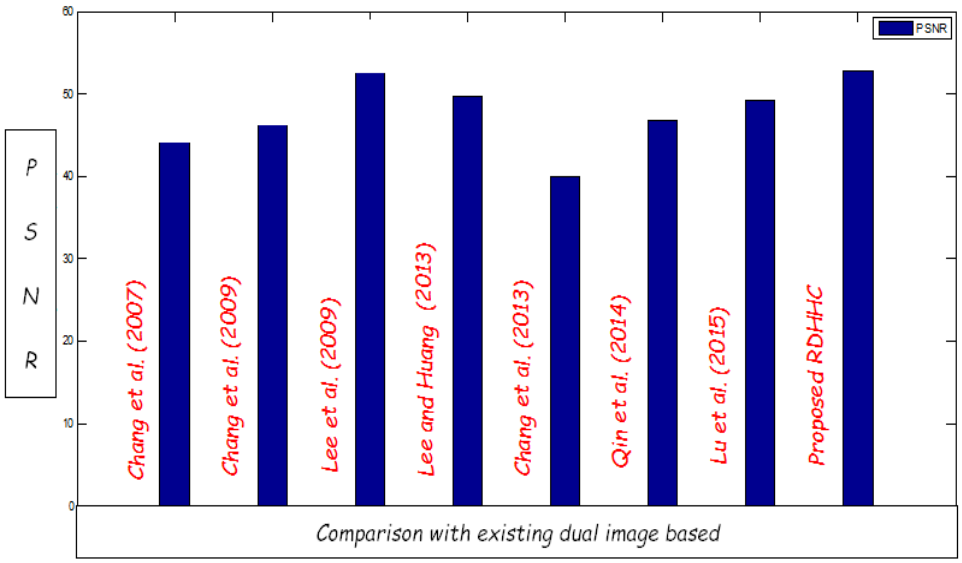


Figure 3.14: Comparison graph of DRDHHC with existing dual image based schemes

3.3.4 Steganalysis and Steganographic Attacks

Here, RS analysis and relative entropy are presented to check the vulnerability of the proposed scheme. Some attacks like histogram attack, statistical attacks and brute force attack are presented. It is observed that the scheme is robust against some steganographic attacks.

3.3.4.1 RS Analysis

The analysis of stego images using well known RS analysis are presented in this section. It is observed from Table 3.13 and 3.14 that the values of R_M and R_{-M}, S_M and S_{-M} are nearly equal for stego images SM' and SA'. Thus rules $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ are satisfied for

the stego image in this approach. So, the proposed method is secured against RS attack. In this experiment, the ratio of R and S lies between 0.0382 to 0.0578 for $SM^{'}$ and 0.0339 to 0.0428 for $SA^{'}$ of Lena image which is considered as original cover image. Other results are depicted on Table 3.13 and 3.14.

Table 3.13: RS analysis of DRDHHC scheme for stego image $SM^{'}$

Image	Data	$SM^{'}$				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Lena (512×512)	37720	6425	6201	5124	5342	0.0382
	64800	6341	6120	5014	5324	0.0467
	74752	6207	5835	4997	5273	0.0578
Barbara (512×512)	37720	5742	5423	4678	4789	0.0412
	64800	5602	5364	4598	4853	0.0483
	74752	5641	5287	4509	4709	0.0545
Tiffany (512×512)	37720	5941	5763	5496	5789	0.0411
	64800	5812	5698	5364	5603	0.0315
	74752	5844	5586	5256	5023	0.0442
Pepper (512×512)	37720	6745	6453	6012	5847	0.0358
	64800	6645	6354	5941	5641	0.0469
	74752	6524	6214	5842	5441	0.0574
Gold hill (512×512)	37720	6845	6425	5487	5641	0.0465
	64800	6654	6475	5341	4952	0.0473
	74752	6523	6143	5136	4862	0.0560

Table 3.14: RS analysis of DRDHHC scheme for stego image $SA^{'}$

Image	Data	$SA^{'}$				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Lena (512×512)	37720	6458	6274	5462	5231	0.0348
	64800	6345	6147	5341	5142	0.0339
	74752	6125	5846	5210	5003	0.0428
Barbara (512×512)	37720	5863	5642	5874	5684	0.0350
	64800	5687	5469	5684	5341	0.0493
	74752	5512	5263	5547	5224	0.0517
Tiffany (512×512)	37720	5987	5784	5487	5236	0.0395
	64800	5874	5569	5364	5166	0.0447
	74752	5741	5478	5123	4857	0.0486
Pepper (512×512)	37720	6687	6486	5874	5462	0.0488
	64800	6541	6347	5784	5364	0.0498
	74752	6452	6243	5674	5241	0.0529
Gold hill (512×512)	37720	6745	6542	5489	5294	0.0325
	64800	6348	6174	5364	5187	0.0299
	74752	6423	6128	5236	4962	0.0488

3.3.4.2 Relative Entropy

The relative entropy of original and stego images are shown in Table 3.15. It is shown that when number of bits in the secret message increases, the relative entropy in stego images also increases. The difference of relative entropy is nearer to zero, which implies the proposed scheme provides secure hidden communication. The relative entropy of dual stego images $SM^{'}$ and $SA^{'}$ are calculated which are shown in Table 3.15.

Table 3.15: Results of relative entropy for $SM^{'}$ and $SA^{'}$ in DRDHHC

Image	Data(bits)	Entropy of I	Entropy of $SM^{'}$	Entropy of $SA^{'}$	$D(SM^{'}\&I)$	$D(SA^{'}\&I)$
Lena (512×512)	37720	7.4451	7.4469	7.4467	0.0018	0.0016
	64800	7.4451	7.4512	7.4503	0.0061	0.0053
	74752	7.4451	7.4546	7.4526	0.0095	0.0075
Barbara (512×512)	37720	7.0480	7.0503	7.0520	0.0023	0.0040
	64800	7.0480	7.0525	7.0530	0.0045	0.0050
	74752	7.0480	7.0540	7.0551	0.0060	0.0071
Tiffany (512×512)	37720	7.2925	7.2969	7.2971	0.0044	0.0046
	64800	7.2925	7.2997	7.3004	0.0072	0.0079
	74752	7.2925	7.3001	7.3075	0.0076	0.0150
Pepper (512×512)	37720	7.2767	7.2797	7.2805	0.0030	0.0038
	64800	7.2767	7.2810	7.2870	0.0043	0.0103
	74752	7.2767	7.2910	7.2935	0.0143	0.0168
Gold hill (512×512)	37720	7.2367	7.2387	7.2398	0.0020	0.0031
	64800	7.2367	7.2464	7.2485	0.0097	0.0118
	74752	7.2367	7.2490	7.2499	0.0123	0.0132

3.3.4.3 Statistical Analysis

The output of the DRDHHC are assessed based on statistical distortion analysis by SD (σ) and CC (ρ) to check the impact on image after secret data embedding. The σ before and after data embedding and ρ of cover and stego images is summarized in Table 3.16. It is seen that there is no significant difference between the σ of the cover image and the stego image. Since the image parameters have not changed much, the method offers a good concealment of data and reduces the chances of secret data detection. Thus, it indicates a perfectly secure steganographic system.

3.3.4.4 Histogram Attack

Fig 3.15 depicted the histogram of the original cover image and stego images and their difference histogram. The stego image is produced from cover image employing maximum data

Table 3.16: Experimental results of SD (σ) and CC (ρ) in DRDHHC

Image	SD (σ)			CC (ρ)		
	Cover image (I)	Stego image (SM')	Stego image (SA')	I & SM'	I & SA'	SM' & SA'
Lena (512×512)	47.8385	47.4358	47.5321	0.9864	0.9835	0.9754
Barbara (512×512)	38.3719	38.4500	38.8541	0.9852	0.9820	0.9789
Tiffany (512×512)	61.5978	61.1221	61.6442	0.9913	0.9874	0.9652
Pepper (512×512)	52.1356	51.8987	52.2450	0.9908	0.9856	0.9687
Gold hill (512×512)	58.8723	57.2854	58.5423	0.9867	0.9825	0.9712

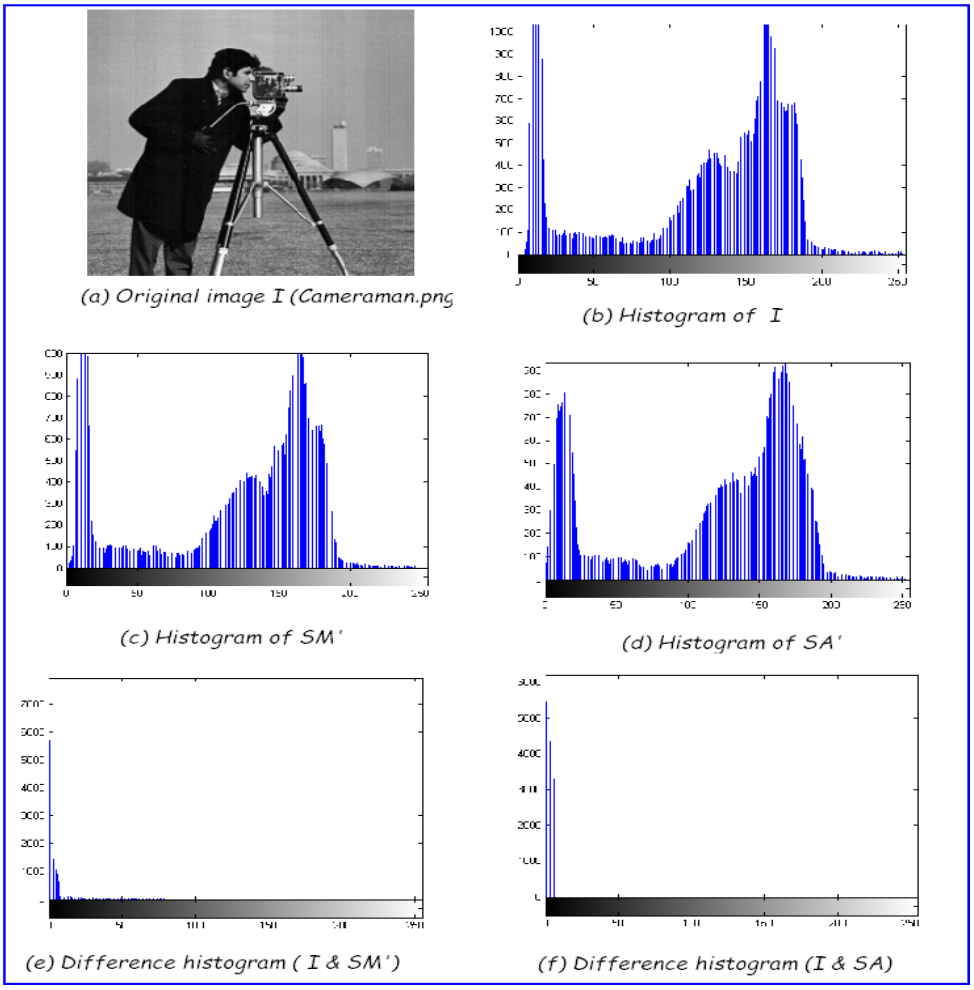


Figure 3.15: Histogram of original image, stego images and their difference

hiding capacity. It is observed that the shape of the histogram is preserved after embedding the secret data. The difference of the histogram is very small. It is observed that, bins close to zero are more in number and the bins which are away from zero are less in number. This confirms the quality of stego image. There is no step pattern observed which ensures that the proposed method is robust against histogram attacks.

3.3.4.5 Brute Force Attack

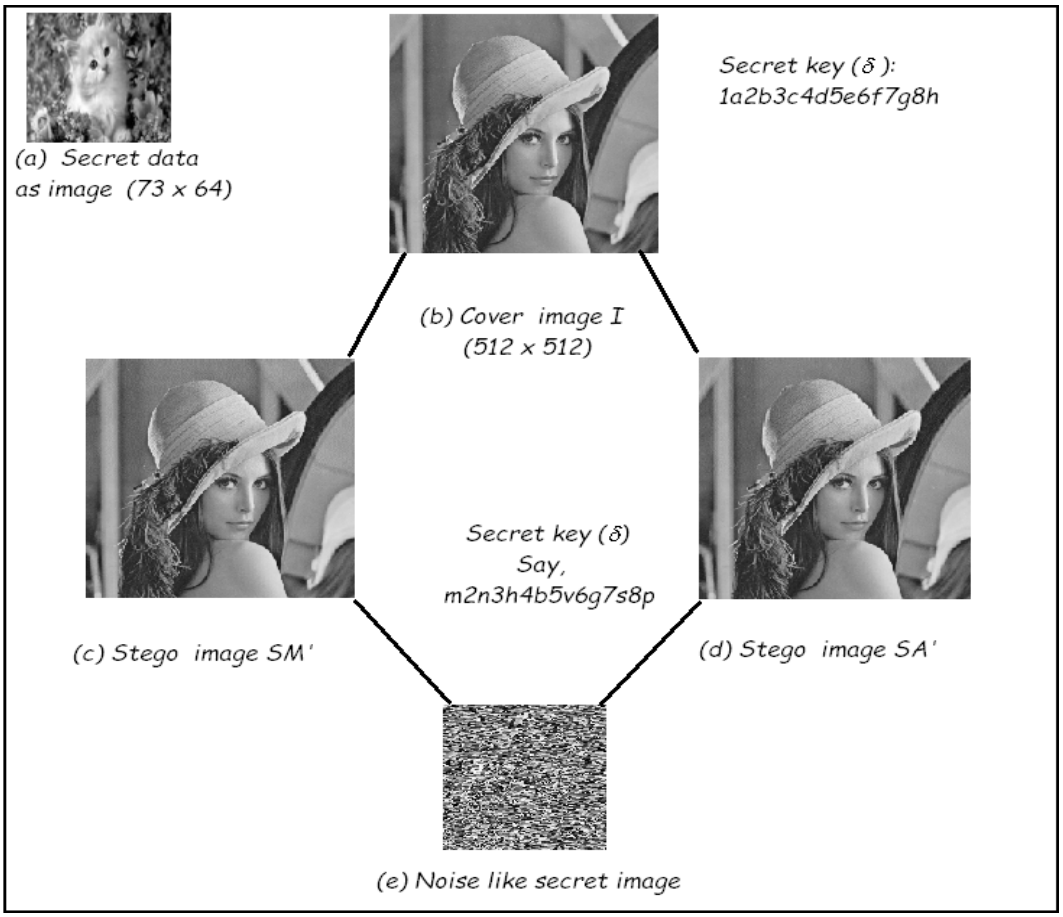


Figure 3.16: Result of Brute Force Attack in DRDHHC

Two shared secret keys δ and ξ are used during both embedding and extraction stage. The scheme is secured to prevent possible malicious attacks. The proposed scheme constructs two stego images which protect original information by hiding secret information in both images $SM^{'}$ and $SA^{'}$. The Fig. 3.16 shows the revelation example where both the keys are unknown. If the malicious attacker holds the original image and the dual images and is fully aware of the proposed scheme, the hidden message still cannot be correctly revealed without knowing the correct secret keys. For example, Fig. 3.16 shows two stego derived from lena images using secret keys which are different from that used to construct without knowing secret key. The result indicates that the attacker only acquires noise like images when applying incorrect secret key to reveal the hidden message. Furthermore, the attacker may employ the brute force attack that tries all possible permutation to reveal the hidden message. The total number of trials to reveal the hidden message are $37449 \times 2^{length\ of\ \xi}$ which are computationally infeasible for current

computers if length of ξ is greater than 80. The proposed scheme achieve stronger robustness against several attacks when compared with existing data hiding schemes. Furthermore, the secret information can be retrieved without encountering any loss of data and the original image can be recovered successfully from the dual images. But the payload is very small. To increase the payload one can apply three LSB Layers (LSB, LSB+1, LSB+2) to hide secret data bits using this scheme.

3.4 Enhanced Partial Reversible Data Hiding using Hamming Code (EPRDHHC)[4]

An enhanced partial reversible data hiding scheme using Hamming code has been proposed here. Three LSB layers (LSB, LSB+1, LSB+2) has been used to embed secret data within (7×7) block. Data bits are embedded using error creation at two different locations in each row of a LSB block using shared secret position and suitable location. During data extraction, the receiver detects the error position with the help of Hamming error correcting code and shared secret position. After data extraction, the receiver complements the bits at the data embedding positions to generate Hamming adjusted cover image. The changes made within the cover image during data embedding at the sender side has been removed after data extraction at the receiver end but the changes made due to odd parity adjustment can not be removed. So, the cover image is partially recovered.

3.4.1 Data Embedding Process

Consider (7×7) pixel block from the original image and convert it into binary number. Then collect three layers LSB, LSB+1 and LSB+2 of each pixel block separately. The redundant bits at $1^{st}, 2^{nd}$ and 4^{th} positions are adjusted using odd parity in each LSB block and a preprocessed image has been produced which is considered as Hamming adjusted cover image. Now, compute shared secret position κ using $\kappa = (\delta \mod 7) + 1$, where δ is a shared secret key. Complement the bit at the secret position κ in the first row of (7×7) LSB block of each layer.

[4]Published in the proceedings of the Sixth International Conference on Computer and Communication Technology, (ICCCT-2015), ACM digital library, ACM New York, NY, USA 2015, pp. 360-365, with title *An Efficient Data Hiding Scheme using Hamming Error Correcting Code*

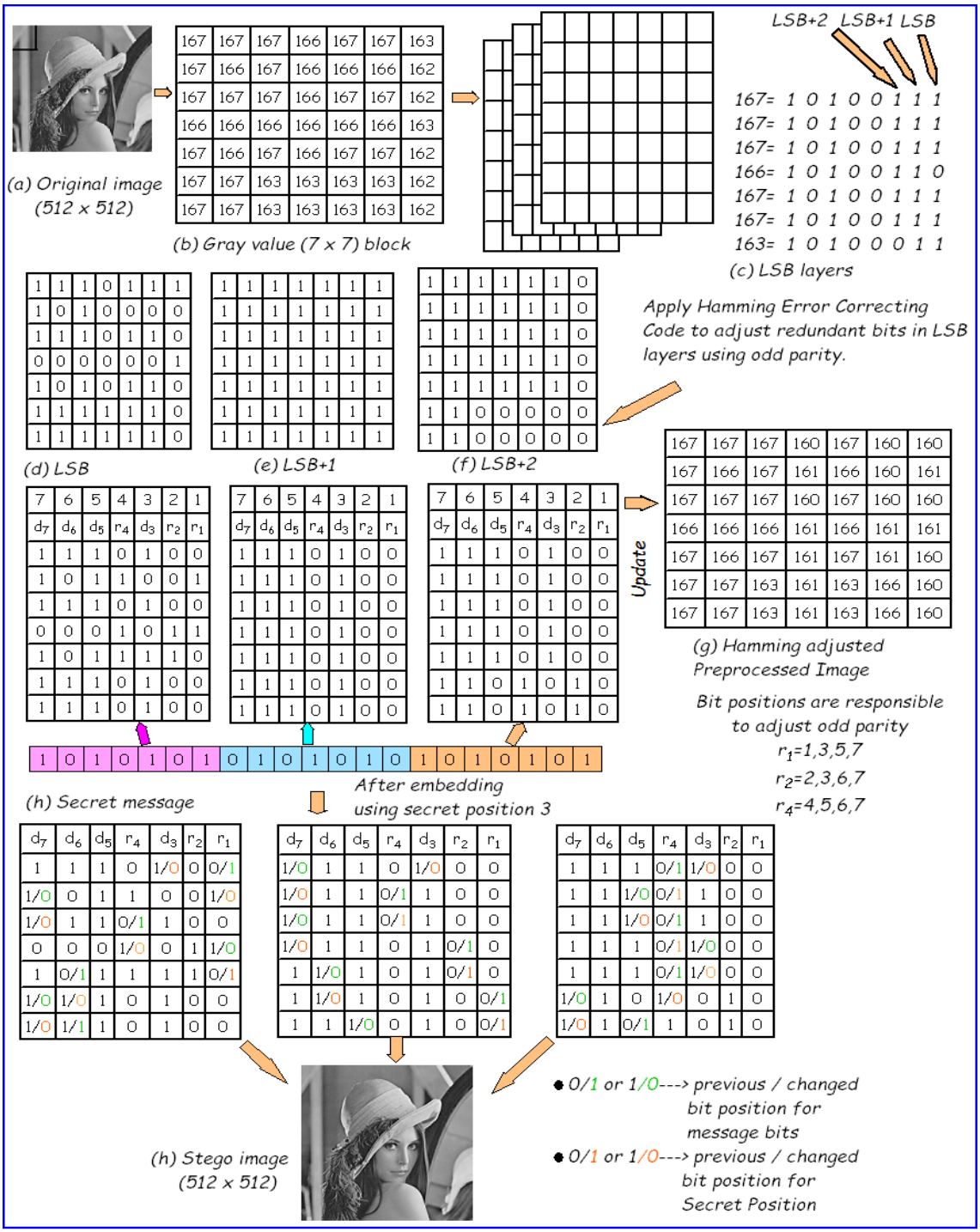


Figure 3.17: The schematic diagram of data embedding process in EPRDHHC

The same secret position has been used for three LSB blocks. Now, embed binary data bits by creating error at any suitable position except the secret position. In the second row, secret position is updated by the data embedding position of previous row. This process has been continued for the block by block and same process has been used for LSB, LSB + 1 and LSB + 2 layers. As a result, it is possible to embed 1,12,128 bits $(73 \times 3 \times 512)$ within a (512×512)

Input: Original image $I_{(m\times n)}$, Secret key δ, Calculate κ = $(\delta \mod 7) + 1$ (between 0 to 7), Secret data D;

Output: Stego image $S_{(m\times n)}$;

Initialization: $C = I$; $sq = 7$; $m = 512$; $n = 512$;

Step 1:

for $p = 1$ *to* m/sq **do**

  for $q = 1$ *to* n/sq **do**

    BC_{pq} from $I_{(m\times n)}$;

  end

  for $i = 1$ *to* sq **do**

    for $j = 1$ *to* sq **do**

      $LSB1_{(i,j)}$ = LSB bit of $BC_{pq(i,j)}$; $LSB2_{(i,j)}$ = 2nd LSB bit of $BC_{pq(i,j)}$; $LSB3_{(i,j)}$ = 3rd LSB bit of $BC_{pq(i,j)}$;

    end

  end

  for *Each LSB Matrix = LSB, LSB1, LSB2* **do** Adjust redundant bits using odd parity in row wise;

  ;

  for $i = (sq \times (p-1)) + 1$ *to* $(sq \times p)$ **do**

    for $j = (sq \times (q-1)) + 1$ *to* $(sq \times q)$ **do** Replace 3 LSBs of $C_{pq(i,j)}$ by LSB2,LSB1,LSB;

  end

  for *Each LSBs Hamming Code matrix* **do**

    for $i = 1$ *to* sq **do**

      if $i \leq sq$ **then** Complement LSB(i, κ) by $(0\ to\ 1\ /\ 1\ to\ 0)$;

    end

    for $j = 1$ *to* sq **do**

      if $j \neq \kappa$ *and LSB* $(i, j) \neq d_r$ **then**

        Create error at LSB (i, j) by $(0\ to\ 1\ /\ 1\ to\ 0)$; κ is updated by j ; break ;

      else

        Do not complement any bit; $\kappa = 1$;

      end

    end

    if $(r = length(D))$ **then** goto **Step 2**;

    Increase r and select next bit d_{r+1} ;

  end

  for $i = (sq \times (p-1))$ + 1 *to* $(sq \times p)$ **do**

    for $j = (sq \times (q-1))$ + 1 *to* $(sq \times q)$ **do** Replace LSB 3 bits of $S_{pq(i,j)}$ by LSB2, LSB1, LSB;

  end

  if $(r = length(D))$ **then** goto **Step 2**;

end

Step 2: Produce stego image $S_{(m\times n)}$;

Step 3: End.

Algorithm 5: Data embedding process of EPRDHHC

gray scale image. So, the payload is 0.426 (bpp) and PSNR is 32.14 (dB). The schematic diagram of data embedding procedure is shown in Fig. 3.17. The corresponding algorithm for data embedding is shown in Algorithm 5.

3.4.2 Data Extraction Process

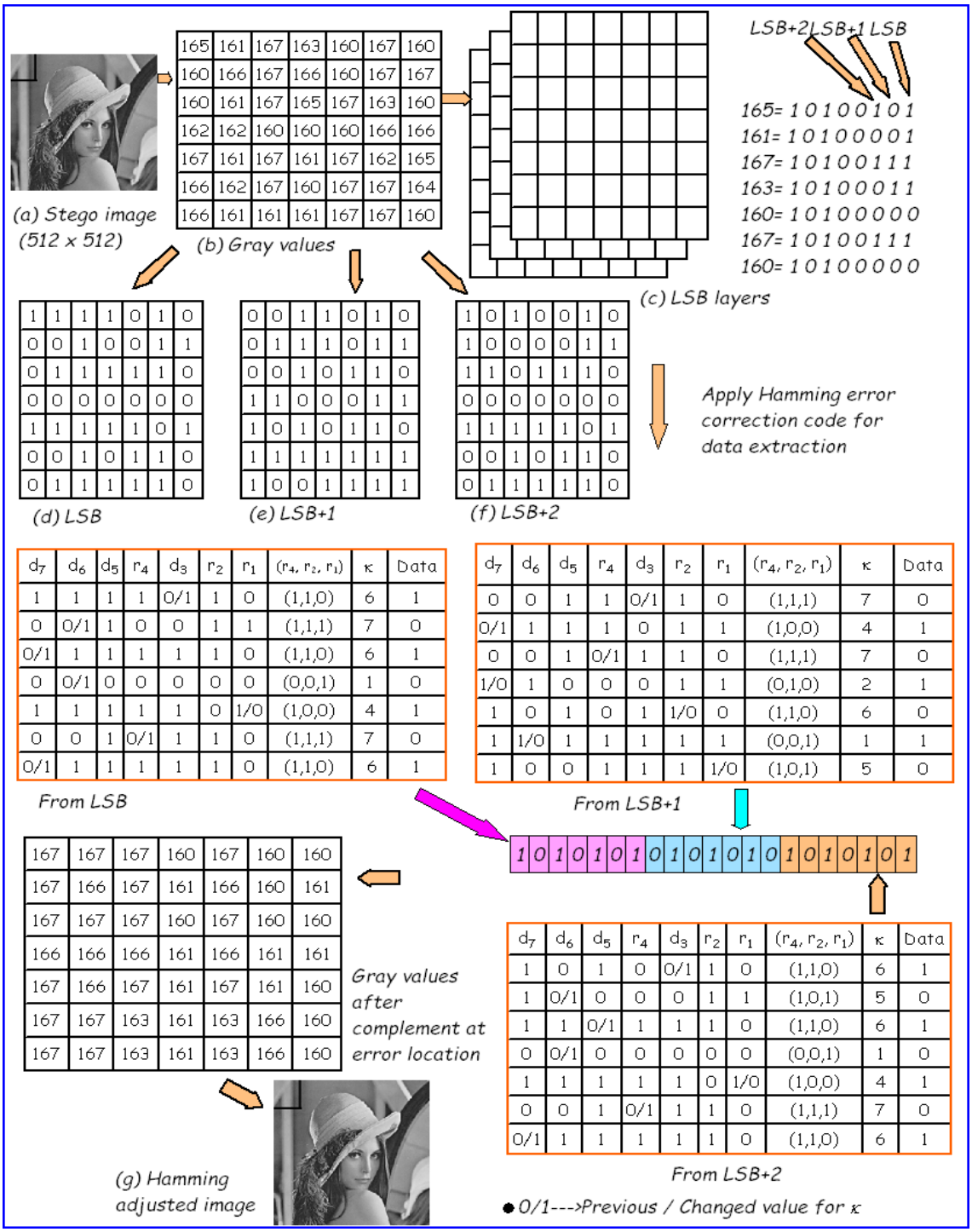


Figure 3.18: The schematic diagram of data extraction process in EPRDHHC

At the receiver end, we collect (7×7) three LSB layers block from stego image then complement the bit at the secret position in the first row of each LSB layer block. Then we apply Hamming error correcting code to find the error position for data bit in that row of LSB block.

Now, collect secret data bit from that error position then complement that bit position. In the second row, secret position will be updated by the error position of the previous row and continue data extraction. Repeat these processes for every block of the stego image until no message has been left. The process will be automatically stopped when we find the error at the secret position using Hamming error correcting code. As a result, no message length has been required to extract whole message. The corresponding schematic diagram is shown in Fig 3.18. The corresponding algorithm for data extraction is listed in Algorithm 6.

3.4.3 Experimental Results and Comparisons

The EPRDHHC data embedding and data extraction algorithms are implemented in MATLAB Version 7.6.0.324 (R2008a). Here, standard gray scale image of size (512×512) pixel has been used as cover image, which is shown in Fig 3.19. After embedding secret data using data

Figure 3.19: Standard cover images used in EPRDHHC

embedding process some stego images has been generated which are shown in Fig. 3.20. The

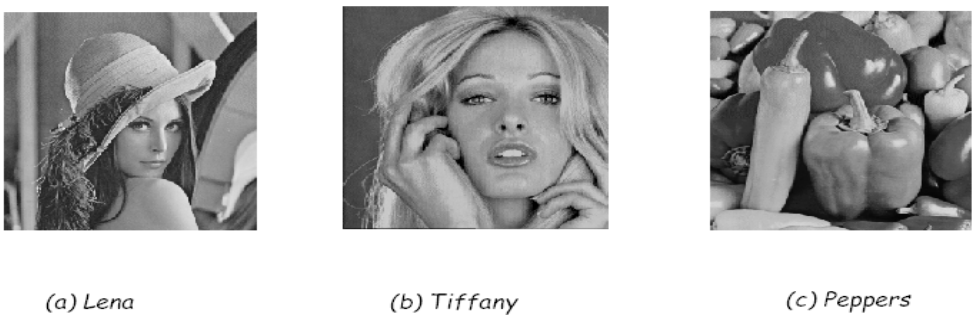


Figure 3.20: Stego image produced after data embedding in EPRDHHC

quality of the stego images have been measured after parity adjustment and after embedding secret data. The original image is considered as I, the parity adjustment image is considered as C and S is considered as the stego image. The PSNRs are measured after embedding 4096

Input: Stego image $S_{(m\times n)}$, Shared Secret key δ ;

Output: Cover image $C_{(m\times n)}$, Secret data (D);

Initialization: $sq = 7$, $m = 512$, $n = 512$, Calculate $\kappa = (\delta \mod 7) + 1$;

Step 1:

for $p = 1 to m/sq$ **do**

  for $q = 1 to n/sq$ **do**

    BS_{pq} from $S_{(m\times n)}$

  end

  for $i = 1$ *to* sq **do**

    for $j = 1$ *to* sq **do**

      $LSB1_{(i,j)}$ = LSB bit of $BS_{pq(i,j)}$;

      $LSB2_{(i,j)}$ = 2nd LSB bit of $BS_{pq(i,j)}$;

      $LSB3_{(i,j)}$ = 3rd LSB bit of $BS_{pq(i,j)}$;

    end

  end

  for *Each LSBs Stego matrix* **do**

    for $i = 1$ *to* sq **do**

      if $(i \leq sq)$ **then**

        Complement LSB (i, κ) by $(0\ to\ 1\ /\ 1\ to\ 0)$;

      end

    end

    Apply Hamming error correcting code in i^{th} row of LSB matrix to find the error at j^{th} column d_r = LSB (i, j) ;

    Store Secret data $D = d_r$;

    Complement LSB (i, j) by $(0\ to\ 1\ /\ 1\ to\ 0)$;

    if *No error found* **then**

      d_r = LSB $(i, j)(1)$; $\kappa = 1$; Do not complement;

    end

    if *Error found at the* κ *position* **then**

      goto **Step 2**;

    end

    $\kappa = j$;

    Increase r to retrieved next bit d_{r+1} ;

  end

  for $i = (sq \times (p-1))$ + 1 *to* $(sq \times p)$ **do**

    for $j = (sq \times (q-1))$ + 1 *to* $(sq \times q)$ **do**

      Replace LSB 3 bit of $C_{pq(i,j)}$ by LSB 2 LSB 1 LSB ;

    end

  end

end

Step 2: Recovered $C_{(m\times n)}$ and extract secret data (D) ;

Step 3: End

Algorithm 6: Data extraction process of EPRDHHC

and 16384 bits that are shown in Table 3.17. Table 3.18 shows the comparison of EPRDHHC with Kim et al.'s (DHHC) scheme [28] and Lien et al.'s (DDHHC) scheme [41]. In DHHC

Table 3.17: PSNR (dB) of stego image after embedding 4096 and 16384 bits in EPRDHHC

Embedded bits	4096			16384		
Image	PSNR(I & S)	PSNR(C & I)	PSNR(C & S)	PSNR(I & S)	PSNR(C & I)	PSNR(C & S)
Lena (512×512)	53.52	54.45	54.58	46.89	47.94	48.03
Barbara (512×512)	51.67	51.52	53.58	44.88	44.69	46.94
Tiffany (512×512)	53.29	54.48	54.57	46.87	47.98	48.04
Pepper (512×512)	52.31	51.97	53.43	45.85	45.68	47.04
Gold hill (512×512)	49.76	48.82	52.23	44.98	42.21	45.67

scheme, the MPSNR of lena image is 32.03 (dB) and 38.12 (dB) after embedding 16384 and 4096 bits respectively. In Lien et al.'s scheme, the MPSNR of lena image is 38.60 (dB) and 44.71 (dB) when embedded with 16384 and 4096 bits, but the PSNR of lena image in our proposed EPRDHHC scheme is 46.89 (dB) and 53.52 (dB) after embedding with 16384 and 4096 bits respectively. The quality is better than other existing schemes. The comparison graph of

Table 3.18: Comparison of EPRDHHC with existing schemes in terms of PSNR (dB)

	16384 bits			4096 bits		
Scheme	DHHC	DDHHC	EPRDHHC	DHHC	DDHHC	EPRDHHC
Lena (512×512)	32.03	38.60	46.89	38.12	44.71	53.52
Barbara (512×512)	32.03	38.57	44.88	38.14	44.77	51.67
Tiffany (512×512)	31.72	36.24	46.87	38.04	42.91	53.29
Pepper (512×512)	31.98	38.02	45.85	38.10	44.19	52.31
Gold hill (512×512)	32.02	38.66	44.98	38.13	44.96	49.76
Average	31.96	38.02	45.89	38.11	44.31	52.11

DHHC, DDHHC, and EPRDHHC are shown in Fig 3.21 and 3.22 when embedded 4096 and 16384 bits respectively.

From Table 3.18, it is observed that PSNR of our proposed scheme is greater than the MPSNR of other existing schemes. As a result, visual quality is better in our proposed EPRDHHC scheme. The payload is measured by

$$p = \frac{|\gamma|}{m \times n}(bpp) \tag{3.3}$$

where $|\gamma|$ is the number of secret data bits which are embedded within cover image and p denotes the payload in terms of bits per pixel (bpp). In a (512×512) image the embedding capacity is 1,12,128 bits. Hence, the payload p is 0.426 (bpp).

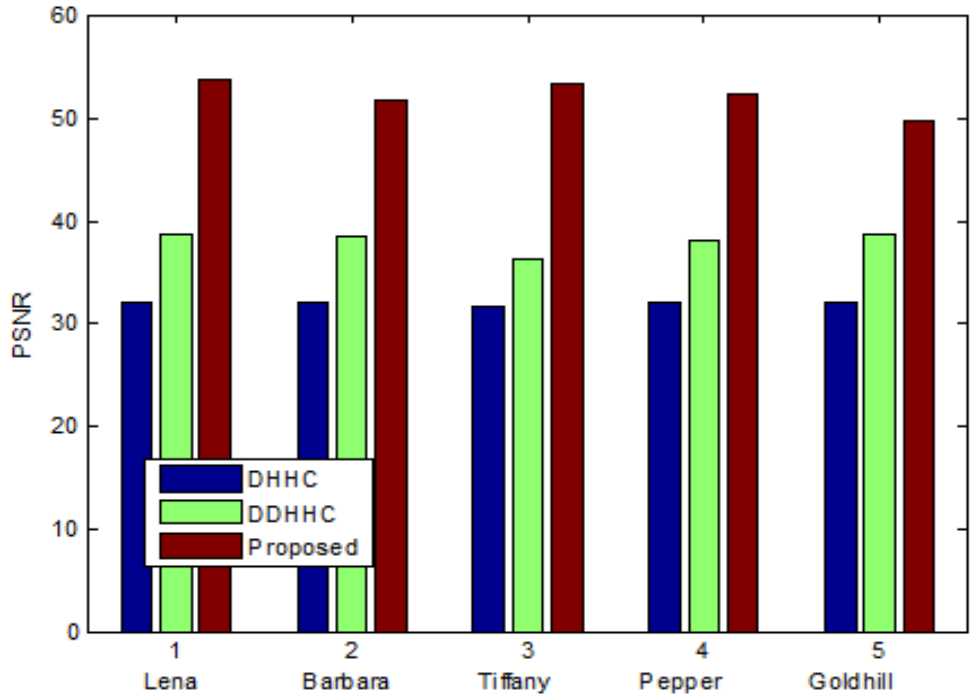


Figure 3.21: Comparison graph of PSNR (dB) after embedding 4096 bits in EPRDHHC

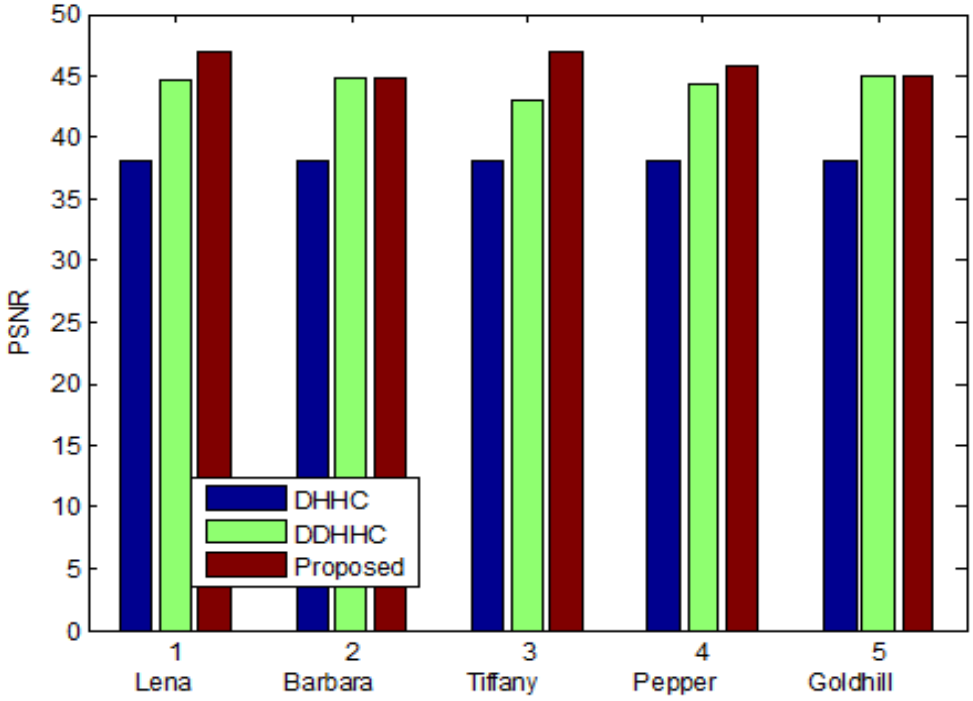


Figure 3.22: Comparison graph of PSNR (dB) after embedding 16384 bits in EPRDHHC

3.4.4 Steganalysis

To measure the imperceptibility and robustness of the proposed EPRDHHC scheme, some standard seganographic analysis and attack has been performed which are presented in this section.

3.4.4.1 RS Analysis

Stego image of this proposed EPRDHHC scheme is analyzed through RS analysis [18]. From Table 3.19, it is observed that the values of R_M = 6207, R_{-M} = 6035, S_M = 4997, S_{-M} = 5773 and RS value is 0.0667 after embedding 112,128 bits secret data within Lena (512×512) image. The RS value of all other images are nearer to zero which implies that the proposed EPRDHHC is not vulnerable against RS attacks.

Table 3.19: Experimental results of RS analysis in EPRDHHC

Image	Data (bits)	Stego image S(512×512)				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Lena (512×512)	50000	6768	6851	3944	3895	0.0123
	75000	6304	5947	4943	5079	0.0438
	112128	6207	6035	4997	5573	0.0667
Barbara (512×512)	50000	5563	5476	4291	4337	0.0135
	75000	5636	5539	4517	4689	0.0264
	112128	5641	5387	4509	4709	0.0447
Tiffany (512×512)	50000	5897	5975	5076	5131	0.0121
	75000	6018	5813	5107	5313	0.0369
	112128	5844	5986	5256	5623	0.0458
Pepper (512×512)	50000	6621	6498	5275	5231	0.0140
	75000	6419	6319	5107	5303	0.0256
	112128	6164	6067	4978	4678	0.0356
Gold hill (512×512)	20000	5756	5652	4896	4785	0.0201
	50000	5746	5675	4977	4901	0.0136
	75000	5518	5411	5141	5013	0.0320
	112128	5444	5186	5056	4881	0.0246

3.4.4.2 Relative Entropy

The relative entropy results between cover and stego image of the proposed EPRDHHC scheme are given in Table 3.20. The difference of relative entropy between original and stego images are nearer to zero which implies the proposed EPRDHHC scheme provides secure hidden data communication.

Table 3.20: Results of relative entropy in EPRDHHC

Cover image (I)	Data (bits)	Entropy of (I)	Entropy of (S)	Relative entropy
Lena (512×512)	75000	7.4451	7.4569	0.0118
	112128	7.4451	7.4663	0.0212
Barbara (512×512)	75000	7.0480	7.0612	0.0132
	112128	7.0480	7.0682	0.0202
Tiffany (512×512)	75000	7.2925	7.3028	0.0103
	112128	7.2925	7.3189	0.0264
Pepper (512×512)	75000	7.2767	7.2872	0.0105
	112128	7.2767	7.2971	0.0204
Gold hill (512×512)	75000	7.2367	7.2472	0.0105
	112128	7.2367	7.2575	0.0208

3.4.4.3 Statistical Analysis

The SD (σ) of before and after data embedding and CC(ρ) of original (I) and stego image (S) are calculated and summarized in Table 3.21. It is observed that the σ of the cover image and the stego image are near identical.

Table 3.21: Results of SD (σ) and CC (ρ) in EPRDHHC

Image	SD (σ)		CC (ρ)
	Image (I)	Stego image (S)	(I) and (S)
Lena (512×512)	47.8385	46.6524	0.9786
Barbara (512×512)	38.3719	37.8210	0.9720
Tiffany (512×512)	61.5978	60.3412	0.9713
Pepper (512×512)	52.1356	50.8421	0.9667
Gold hill (512×512)	58.8723	56.6423	0.9624

The main problem of this scheme is that it is not reversible. The scheme can not recover original cover image successfully but it can recover Hamming adjusted cover image.

3.5 Enhanced Dual Image based Reversible Data Hiding scheme using Hamming Code (EDRDHHC)

An enhanced dual image based reversible data hiding scheme using Hamming code (EDRD-HHC) has been proposed in this section. First, cover image is partitioned into (1×7) pixel blocks and copied into two arrays. After that collect LSBs and adjust redundant bits in three LSBs (LSB, LSB+1 and LSB+2) of each arrays on both the images separately using odd parity.

One shared secret position κ and one shared secret key ξ have been used to perform such data embedding and data extraction procedure. Now, complement the bit at the κ position in each three LSB blocks then embed secret data bit through error creation in any suitable position (ω) in the block except secret position κ. The suitable position is the position in the block which contains opposite value of the data bit. The κ is updated by ω and used as secret position for the next block. Continue this process to embed secret data bits within three LSBs of dual image. The key ξ is used to distribute pixel blocks among dual image. During extraction one can successfully extract the secret data from dual stego images using κ and ξ. Three LSBs are used to enhance payload. The data hiding capacity in this dual image based scheme is 224,256 bits.

3.5.1 Data Embedding Process

The detailed data embedding process is explained below:
First, partition the cover image into (1×7) consecutive pixel blocks then convert pixel into binary form and copy LSB bits into two arrays M and A. Three layers of LSB bits that is LSB, LSB+1 and LSB+2 are collected separately and used for data embedding which are shown in Fig. 3.23. Now, adjust redundant bits in both arrays separately using odd parity. The redundant bits r_1, r_2 and r_3 of M array are adjusted based on the number of one present in the bit position of M which are shown in Index 1 of Fig. 3.23. For example, the r_1 bit is set to one if the even number of one's are present in the position 3, 5 and 7 of M. The redundant bits r_1, r_2, r_3 and r_4 of A array are adjusted and updated in 3, 5, 6 and 7 positions of A depending on the number of one present in the bit positions shown in Index 2 of Fig. 3.23. Now, calculate shared secret position $\kappa = (\delta \mod 7) + 1$, where δ is the shared secret key. Then complement the bit at the position of κ (say 4 in Fig. 3.23) of M and embed secret data bit by error creation in any suitable position except the secret position. So, the positions 3, 5, 6 and 7 are suitable locations for error creation because if the data bit is 1 then suitable position should contain 0 only and vice versa. Here, we choose suitable position 3 (say). Now, the data embedding position (ω) of M is set to the secret position (κ) for data embedding in the array A. Then same process has been followed to embed next data bit with A array. After embedding in the LSB, we apply same process for two copies of LSB that is LSB+1 and LSB+2 into two arrays M and A separately. Now, store the modified pixel blocks within two stego images (SM) and (SA) by M and A respectively. After that update κ by ω for the next (1×7) pixel block using $\kappa_{i+1} = ((\kappa_i \times \omega) \mod 7) + 1$,

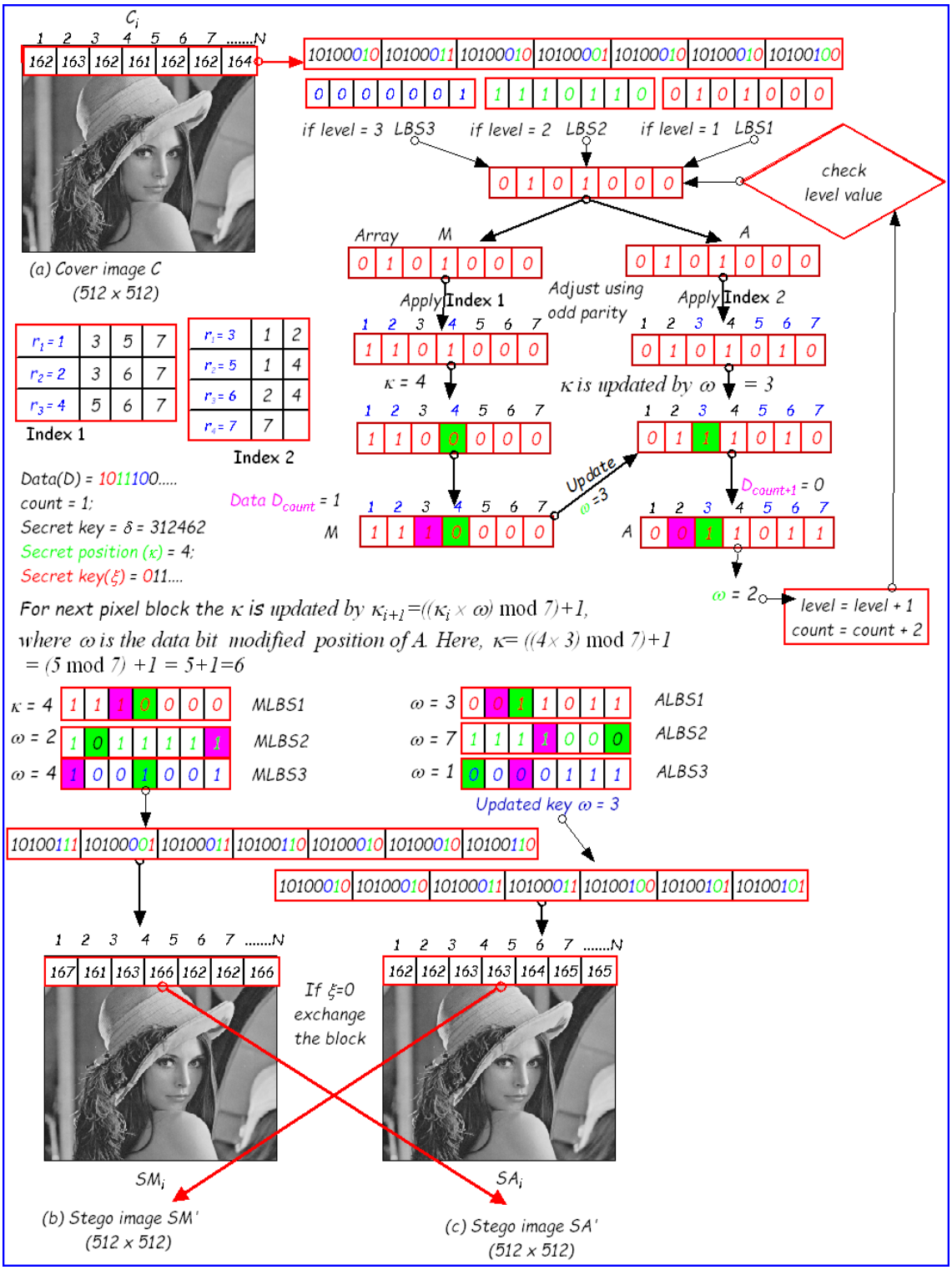


Figure 3.23: Schematic diagram of data embedding process in EDRDHHC

Input: Cover image C, Secret key δ and ξ, Secret data D;

Output: Dual stego image, $SM^{'}$ and $SA^{'}$;

Initialization: $count = 1$; Calculate $\kappa = (\delta \mod 7) + 1$;

Step 1:

for $i = 1$ *to* $(m \times (n/7))$ **do**

  for $j = 1$ *to* 7 **do** $tem \leftarrow dec2bin(C^i_{(1\times7)}(j), 8)$; $LSB1^i_{(1\times7)}(j) = str2num(tem(8))$;

  $LSB2^i_{(1\times7)}(j) = str2num(tem(7))$; $LSB3^i_{(1\times7)}(j) = str2num(tem(6))$;

end

Step 2:

for $level = 1$ *to* 3 **do**

  if $(level = 1)$ **then** $M_{(m\times n)} \leftarrow LSB1_{(m\times n)}$;$A_{(m\times n)} \leftarrow LSB1_{(m\times n)}$;

  if $(level = 2)$ **then** $M_{(m\times n)} \leftarrow LSB2_{(m\times n)}$;$A_{(m\times n)} \leftarrow LSB2_{(m\times n)}$;

  if $(level = 3)$ **then** $M_{(m\times n)} \leftarrow LSB3_{(m\times n)}$;$A_{(m\times n)} \leftarrow LSB3_{(m\times n)}$;

  for $i = 1$ *to* $(m \times (n/7))$ **do**

    (a) $M^i_{(1\times7)} \leftarrow$ modify redundant bit by odd parity using Index 1 of Fig 3.23;

    (b) $A^i_{(1\times7)} \leftarrow$ modify redundant bit by odd parity using Index 2 of Fig 3.23;

    (c) Complement $M^i_{(1\times7)}(\kappa)$, where κ is a shared secret value between 1 *to* 7;

    (d) Create an error by complement for data bit at any suitable position (ω) of $M^i_{(1\times7)}$ except κ position;

    if $((D_{count} == 0)$ *and* $(M^i_{(1\times7)} == 0))$ **then** Set $\omega \leftarrow 1$;

    {*** No suitable location found ***}

    (e) κ is updated by ω for data embedding in $A^i_{(1\times7)}$;

    (f) Complement $A^i_{(1\times7)}(\kappa)$, where κ is a shared secret position between 1 *to* 7;

    (g) Create an error by complement for data bit at any suitable position (ω) of $A^i_{(1\times7)}$ except κ position;

    if $((D_{count+1} == 1)$ *and* $(A^i_{(1\times7)} == 1))$ **then** Set $\omega \leftarrow 7$;

    {*** No suitable location found ***}

    (h) κ = $((\kappa \times \omega) \mod 7) + 1$;

  end

  if $(level = 1)$ **then** $MLSB1_{(m\times n)} \leftarrow M_{(m\times n)}$; $ALSB1_{(m\times n)} \leftarrow A_{(m\times n)}$;

  if $(level = 2)$ **then** $MLSB2_{(m\times n)} \leftarrow M_{(m\times n)}$; $ALSB2_{(m\times n)} \leftarrow A_{(m\times n)}$;

  if $(level = 3)$ **then** $MLSB3_{(m\times n)} \leftarrow M_{(m\times n)}$; $ALSB3_{(m\times n)} \leftarrow A_{(m\times n)}$;

end

Step 3:

for $i = 1$ *to* $(m \times (n/7))$ **do**

  for $j = 1$ *to* 7 **do**

    $temp \leftarrow dec2bin(C^i_{(1\times7)}(j), 8)$; $temp(8) = num2str(MLSB1^i_{(1\times7)}(j))$;

    $temp(7) = num2str(MLSB2^i_{(1\times7)}(j))$; $temp(6) = num2str(MLSB3^i_{(1\times7)}(j))$;

    $SM^i_{(1\times7)}(j) = bin2dec(temp)$; $temp(8) = num2str(ALSB1^i_{(1\times7)}(j))$;

    $temp(7) = num2str(ALSB2^i_{(1\times7)}(j))$; $temp(6) = num2str(ALSB3^i_{(1\times7)}(j))$;

    $SA^i_{(1\times7)}(j) = bin2dec(temp)$;

  end

end

Step 4: Apply ξ to distribute (1×7) pixel block;

if $(\xi_i = 0)$ **then** Exchange pixel block between SM^i and SA^i; **else** No exchange;

Step 5: Now two stego image $SM^{'} = SM$ and $SA^{'} = SA$ are produced.

Algorithm 7: Data embedding process of EDRDHHC

where i is the number of blocks. Then continue the process and update pixel values accordingly and produce two stego images stego SM and SA. Finally, we distribute stego pixel blocks, depending on another shared secret key ξ, among two stego image $SM^{'}$ and $SA^{'}$. If $(\xi = 1)$ then selected pixel block from SM is stored on $SM^{'}$ and pixel block from SA is stored on $SA^{'}$ otherwise pixel block from SM is stored on $SA^{'}$ and pixel block from SA is stored on $SM^{'}$.

3.5.2 Data Extraction Process

After receiving dual stego images $SM^{'}$ and $SA^{'}$ receiver applies the key ξ to rearrange pixel blocks which are shown in Fig. 3.24. If ($\xi = 1$) then selected pixel block from $SA^{'}$ is stored on SA_i and pixel block from $SM^{'}$ is stored on SM_i otherwise pixel block from $SM^{'}$ is stored on SA_i and pixel block from $SA^{'}$ is stored on SM_i. Collect LSBs of pixel blocks to perform data extraction process. LSB of the first block from both stego image SM_i and SA_i are copied into MLSB1 and ALSB1 respectively. Complement the bit at the κ position of the MLSB1 then apply Hamming error correcting code to detect the error position. In case of MLSB1, we use redundant bit r_1, r_2 and r_3 to detect the error as mentioned in Index 1 of Fig. 3.24. In this case, the error encountered at the position 3 of MLSB1, so ω is updated by 3 and then consider ω as secret position for ALSB1 array. The redundant bits r_1 and r_2, r_3, r_4 are used for ALSB1 shown in Index 2 of Fig. 3.24 to detect error. The error position of ALSB1 is the data embedding position and that position is again set by key ω for LSB2 bits which have to copied into two arrays MLSB2 and ALSB2. Apply the same extraction process for MLSB2 and ALSB2 and for MLSB3 and ALSB3 separately to extract secret bits. Now, the key κ is updated for the next block using formula $\kappa_{i+1} = ((\kappa_i \times \omega) \mod 7) + 1$, where, ω is the error position of ALSB3 and $i = 1, 2, \ldots,$ number of blocks. After extracting the message bit, complement the corresponding position that means correct the error and produce Hamming adjusted cover image. Put the MLSB1, MLSB2 and MLSB3 in proper position and construct M array of gray values and in the same way generate A array from ALSB1, ALSB2 and ALSB3. Now convert those bits into pixels and stored on SM^i and SA^i. Finally, collect pixels from the positions 3, 5, 6 and 7 from SM^i and pixels from the positions 1, 2 and 4 from SA^i and rearrange them to construct original image. The extraction process is to be stopped when receiver finds an error at secret position in any pixel block after applying Hamming code during extraction. As a result, one can send any arbitrary length of secret data through this approach. This scheme extracts

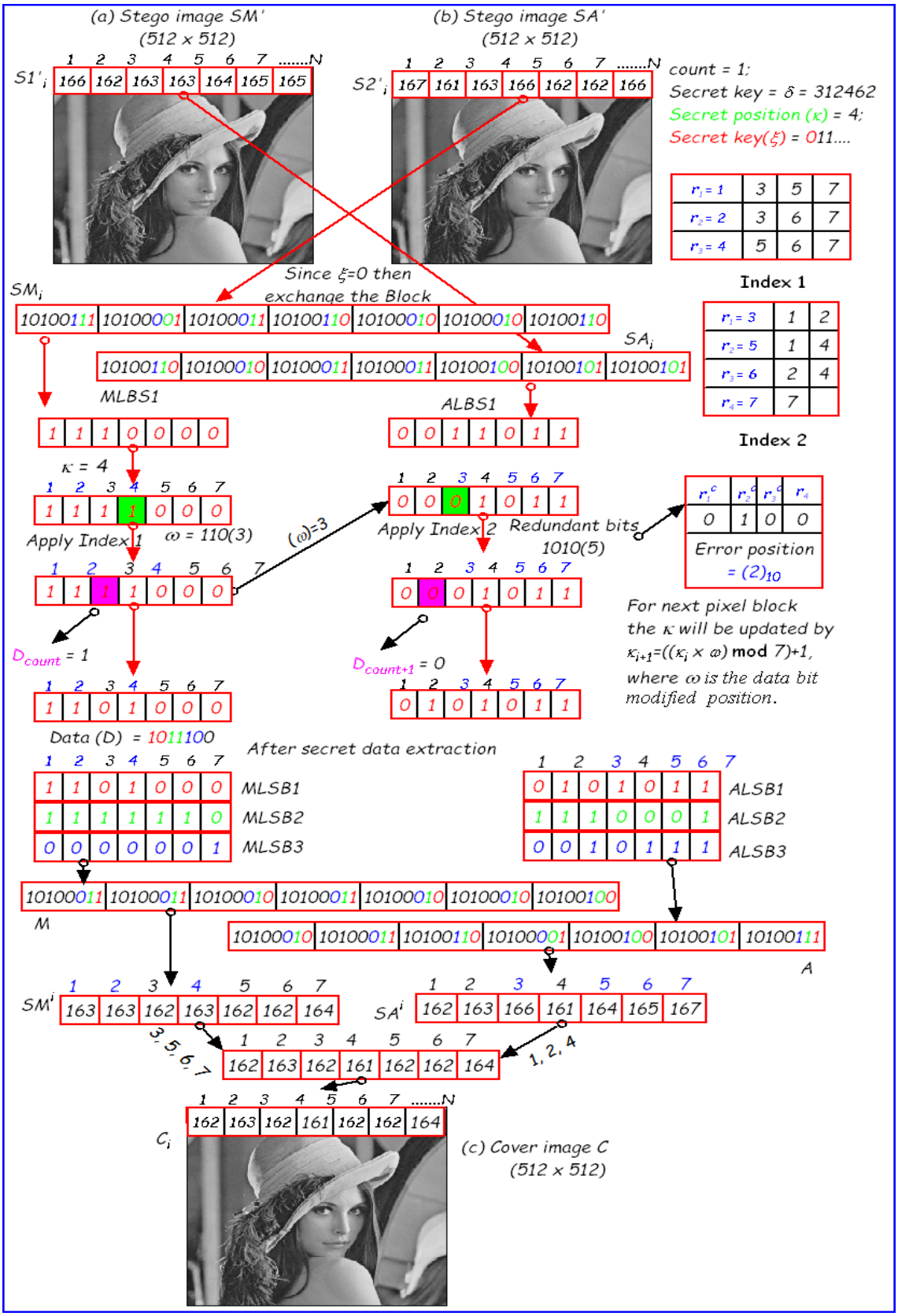


Figure 3.24: Schematic diagram of data extraction process in EDRDHHC

Input:Dual stego image $SM^{'}$ and $SA^{'}$ of hight m and width n, Secret keys δ and ξ;

Output: Secret data D, Cover image C; **Initialization**: $count = 1$; Calculate κ = $(\delta \mod 7) + 1$;

Step 1: **if** *(ξ(x)=0)* **then** $SM^{i}_{(1\times7)} \leftarrow SA^{'i}_{(1\times7)}$; $SA^{i}_{(1\times7)} \leftarrow SM^{'i}_{(1\times7)}$;

else $SM^{i}_{(1\times7)} \leftarrow SM^{'i}_{(1\times7)}$; $SA^{i}_{(1\times7)} \leftarrow SA^{'i}_{(1\times7)}$, where $i = 1, 2, 3, \ldots, (m \times (n/7))$ and $x = (reminder(i, length(\xi))+1)$;

Step 2:

for *($i = 1$ to $(m \times (n/7))$)* **do**

  for *($j = 1$ to 7)* **do** $temp = dec2bin(SM^{i}_{(1\times7)}(j),8)$; $MLSB1^{i}_{(1\times7)}(j) = str2num(temp(8))$;

  $MLSB2^{i}_{(1\times7)}(j) = str2num(temp(7))$; $MLSB3^{i}_{(1\times7)}(j) = str2num(temp(6))$; $temp = dec2bin(SA^{i}_{(1\times7)}(j),8)$;

  $ALSB1^{i}_{(1\times7)}(j) = str2num(temp(8))$; $ALSB2^{i}_{(1\times7)}(j) = str2num(temp(7))$;

  $ALSB3^{i}_{(1\times7)}(j) = str2num(temp(6))$;

end

Step 3:

for *($level = 1$ to 3)* **do**

  if *($level == 1$)* **then** $M_{(m\times n)} \leftarrow MLSB1_{(m\times n)}$;$A_{(m\times n)} \leftarrow ALSB1_{(m\times n)}$;

  if $(level == 2)$ **then** $M_{(m\times n)} \leftarrow MLSB2_{(m\times n)}$;$A_{(m\times n)} \leftarrow ALSB2_{(m\times n)}$;

  if $(level == 3)$ **then** $M_{(m\times n)} \leftarrow MLSB3_{(m\times n)}$;$A_{(m\times n)} \leftarrow ALSB3_{(m\times n)}$;

  for $i = 1$ to $(m \times (n/7))$ **do**

    (a) Complement $M^{i}_{(1\times7)}(\kappa)$, where κ is a shared secret position; (b) $r_{(1\times3)} \leftarrow M^{i}_{(1\times7)}$, where r is the redundant bits using Index 1 from Fig. 3.24; (c) $\omega \leftarrow decimal(r_{(1\times3)})$, where ω indicates the error position or data embedding position; (d) $D'_{count} = M^{i}_{(1\times7)}(\omega)$, if $\omega \neq 0$; **if** $((\omega == 0)\&(M^{i}_{(1\times7)} == 0))$ **then** $D_{count} = M^{i}_{(1\times7)}(1)$; $\omega = 1$;

    if $(\omega == \kappa)$ **then** goto **Step 5**;

    (e) $M^{i}_{(1\times7)}(\omega)$ = Complement($M^{i}_{(1\times7)}(\omega)$); κ is updated by ω;

    (f) Complement $A^{i}_{(1\times7)}(\kappa)$, where κ is a shared secret position; (g) $r_{(1\times3)} \leftarrow A^{i}_{(1\times7)}$, where r is the redundant bits using Index 2 from Fig. 3.24; (h) $\omega \leftarrow decimal(r_{(1\times3)})$, where ω indicates the error position or data embedding position; (i) $D'_{count} = A^{i}_{(1\times7)}(\omega)$, if $\omega \neq 0$; **if** $((\omega == 1)\&(A^{i}_{(1\times7)} == 1))$ **then** $D_{count} = A^{i}_{(1\times7)}(7)$; $\omega = 7$;

    if $(\omega == \kappa)$ **then** goto **Step 5**;

    (j) $A^{i}_{(1\times7)}(\omega)$ = Complement($A^{i}_{(1\times7)}(\omega)$); (k) Update κ as, $\kappa = ((\kappa \times \omega) \mod 7) + 1$;

  end

  if *($level == 1$)* **then** $MLSB1_{(m\times n)} \leftarrow M_{(m\times n)}$; $ALSB1_{(m\times n)} \leftarrow A_{(m\times n)}$;

  if *($level == 2$)* **then** $MLSB2_{(m\times n)} \leftarrow M_{(m\times n)}$; $ALSB2_{(m\times n)} \leftarrow A_{(m\times n)}$;

  if *($level == 3$)* **then** $MLSB3_{(m\times n)} \leftarrow M_{(m\times n)}$; $ALSB3_{(m\times n)} \leftarrow A_{(m\times n)}$;

end

Step 4:**for** *($i = 1$ to $(m \times (n/7))$)* **do**

  for *($j = 1$ to 7)* **do**

    $tem \leftarrow$ dec2bin($C^{i}_{(1\times7)}(j), 8$); $temp(8) = num2str(MLSB1^{i}_{(1\times7)}(j))$;

    $temp(7) = num2str(MLSB2^{i}_{(1\times7)}(j))$; $temp(6) = num2str(MLSB3^{i}_{(1\times7)}(j))$;

    $SM^{i}_{(1\times7)}(j) = bin2dec(temp)$; $temp(8) = num2str(ALSB1^{i}_{(1\times7)}(j))$;

    $temp(7) = num2str(ALSB2^{i}_{(1\times7)}(j))$; $temp(6) = num2str(ALSB3^{i}_{(1\times7)}(j))$;

    $SA^{i}_{(1\times7)}(j) = bin2dec(temp)$;

    if *(j=3 or j=5 or j=6 or j=7)* **then** $C'^{i}_{(1\times7)}(j) = SM^{i}_{(1\times7)}(j)$;

    if *(j=1 or j=2 or j=4)* **then** $C'^{i}_{(1\times7)}(j) = SA^{i}_{(1\times7)}(j)$;

  end

end

Step 5: Recover Cover image $C'_{(m\times n)}$ and secret data D'.

Algorithm 8: Data extraction process of EDRDHHC

secret data and recovers cover image successfully. The algorithm for data extraction and cover image reconstruction is given in Algorithm 8.

3.5.3 Experimental Results and Comparisons

Some standard gray scale images of size (512×512) are used in this study which are shown in Fig. 3.25. After embedding the secret data, dual stego image has been generated which are

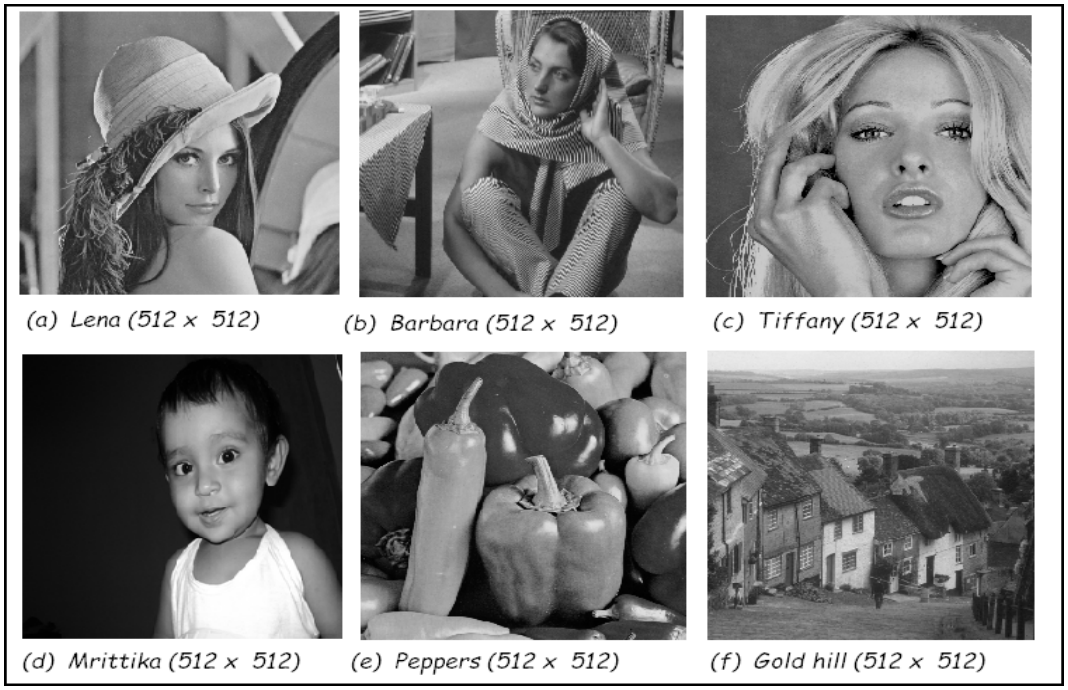


Figure 3.25: Standard cover images used in EDRDHHC

shown in Fig. 3.26. The qualities of stego images are measured using Peak Signal to Noise Ratio $(PSNR)$. Table 3.22 shows the image quality after embedding different amount of secret data bits. PSNR of (C & SM') and PSNR of (I & SA') represent the measurement of visual quality of the first and second stego image respectively while Avg. PSNR is the average of these two qualities of the stego images. From the Table 3.22 the average PSNR of proposed scheme is greater than 38 (dB) and the maximum embedding capacity is $6 \times (512 \times 512/7)$ = $2,24,256$ bits. The payload is measured by $p = \frac{\gamma}{(2 \times m \times n)}$ (bpp), where γ is the total number of bits of two stego images. The payload in this scheme is 0.426 (bpp). The proposed scheme is compared with Kim et al.'s [28] DHHC scheme and Lien et al.'s [41] DDHHC scheme shown in Table 3.23 In Kim et al.'s scheme, the MPSNR of lena image is 32.03 and 44.71 (dB) when they embed $16,384$ and $4,096$ bits respectively. In Lien et al.'s scheme, the MPSNR of lena image

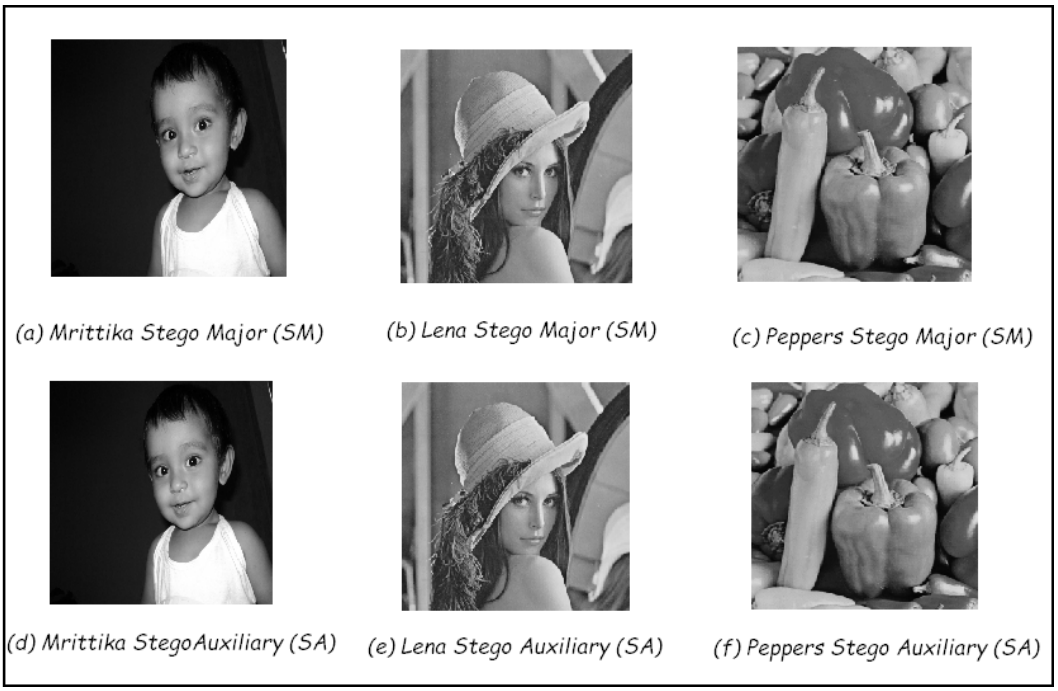


Figure 3.26: Dual stego images produced from EDRDHHC

Table 3.22: PSNR (dB) of stego images with different embedding capacity in EDRDHHC

PSNR (dB) with data embedding capacity (bits)				
Cover image C	Secret data (bits)	PSNR (C & $SM^{'}$)	PSNR (C & $SA^{'}$)	Avg. PSNR
Barbara (512×512)	74,752	40.84	40.88	39.72
	1,49,504	39.82	39.84	
	2,24,256	38.48	38.47	
Lena (512×512)	74,752	40.55	40.64	39.38
	1,49,504	39.27	39.34	
	2,24,256	38.18	38.29	
Peppers (512×512)	74,752	40.47	40.50	39.29
	1,49,504	39.22	39.20	
	2,24,256	38.15	38.25	
Mrittika (512×512)	74,752	40.27	40.70	39.50
	1,49,504	39.20	39.45	
	2,24,256	38.63	38.74	
Tiffany (512×512)	74,752	40.56	40.59	39.37
	1,49,504	39.29	39.28	
	2,24,256	38.21	38.31	
Gold hill (512×512)	74,752	40.11	40.90	39.34
	1,49,504	39.28	39.29	
	2,24,256	38.17	38.26	

is 38.60 and 44.71 (dB) when embeded $16,384$ and $4,096$ bits respectively, but in the proposed EDRDHHC scheme PSNR of lena image is 49.89 and 53.53 (dB) when embedded with secret

Table 3.23: Comparison with DHHC and DDHHC schemes in terms of PSNR (dB)

Embedded bits	$4,096$ bits			16,384 bits		
Scheme	DHHC	DDHHC	EDRDHHC	DHHC	DDHHC	EDRDHHC
Lena	32.03	38.60	53.53	38.12	44.71	49.89
Barbara	32.03	38.57	51.61	38.14	44.77	49.88
Tiffany	31.72	36.24	53.29	38.04	42.91	50.02
Pepper	31.98	38.02	52.31	38.10	44.19	49.85
Gold hill	32.02	38.66	49.76	38.13	44.96	48.98

bits $16,384$ and $4,096$ respectively. Other PSNR values are shown in Table 3.23. It is observed that PSNR of the proposed scheme is greater than the other existing schemes. As a result, visual quality is also better than other schemes. In a (512×512) cover image the embedding capacity is $2,24,256$ bits which implies the payload is 0.426 (bpp).

Another comparison has been presented with the existing Hamming code based data hiding schemes which considered gray scale image in their experiment. Matrix coding [67], Hamming $+1$ [73] and Hamming $+3$ [29] schemes have been taken into account for comparison with proposed scheme. The results are listed in the Table 3.24. The PSNR of the proposed scheme is lower than all other schemes. The PSNR and payload of proposed scheme is also lower

Table 3.24: Comparison of EDRDHHC with existing schemes in terms of PSNR (dB)

Images	Matrix coding	Hamming +1	Hamming +3	EDRDHHC scheme
Lena (512×512)	56.05	52.43	53.95	39.38
Barbara (512×512)	54.65	48.60	53.93	39.72
Tiffany (512×512)	53.98	47.46	53.96	39.37
Pepper (512×512)	54.01	47.26	53.95	39.29
Gold hill (512×512)	57.02	53.73	53.95	39.34

that other existing dual image based schemes but in terms of security it is better than existing dual image based schemes, because two shared secret keys δ and ξ have been used during data embedding where maximum possible numbers of κ is 37449. It is hard to guess the length of key ξ (l) which is used to shuffle the pixel block among dual image. So, total trails for choosing correct key will be $37449 \times 2^{Length\ of\ \xi}$. Reversibility has been achieved in Hamming code based data hiding schemes through this approach.

3.5.4 Steganalysis and Steganographic Attacks

All the stego images are analyzed through RS analysis [18] to test the vulnerability of proposed scheme and perform some steganographic attacks to prove the robustness of the proposed scheme.

3.5.4.1 RS Analysis

It is observed from Table 3.25 and 3.26 that the values of R_M = 6452, R_{-M} = 5321, S_M = 5412 , S_{-M} = 5214 for Lena image. Thus rule $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ are satisfied in this

Table 3.25: RS analysis of stego image $SM^{'}$ in EDRDHHC

Image	Data	$SM^{'}$				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Lena (512×512)	74752	6745	5942	5874	5687	0.0784
	149504	6591	5642	5641	5463	0.0921
	224256	6452	5321	5412	5214	0.1120
Barbara (512×512)	74752	6245	5621	5274	4952	0.0821
	149504	5942	5462	5124	4562	0.0941
	224256	5874	5287	4986	4356	0.1120
Tiffany (512×512)	74752	6874	6014	5749	5574	0.0819
	149504	6541	5324	5589	5264	0.1271
	224256	6124	5241	5472	5023	0.1148
Pepper (512×512)	74752	6754	6035	5962	5124	0.1224
	149504	6542	5941	5784	4963	0.1153
	224256	6324	5874	5632	4741	0.1121
Gold hill (512×512)	74752	6741	6039	5896	5034	0.1237
	149504	6536	5684	5674	4963	0.1280
	224256	6472	5963	5471	4762	0.1019

scheme. So, the proposed method is secure against RS analysis.

3.5.4.2 Relative Entropy

The relative entropy between the stego image and original cover image are measured and presented in Table 3.27.

3.5.4.3 Statistical Analysis

The SD (σ) before and after data embedding of cover and stego images and CC (ρ) between original and stego images are summarized in Table 3.28.

Table 3.26: RS analysis of stego image $SA^{'}$ in EDRDHHC

Image	Data	$SA^{'}$				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Lena (512×512)	74752	6674	5874	5764	5547	0.0817
	149504	6489	5592	5674	5547	0.0841
	224256	6487	5547	5468	5287	0.0937
Barbara (512×512)	74752	6245	5641	5236	4978	0.0750
	149504	5942	5462	5124	4562	0.0941
	224256	5874	5287	4986	4563	0.0930
Tiffany (512×512)	74752	6874	6014	5749	5574	0.0819
	149504	6541	5684	5589	5264	0.0974
	224256	6012	5324	5472	5023	0.0990
Pepper (512×512)	74752	6654	6125	5962	5362	0.0894
	149504	6542	6012	5784	5247	0.0865
	224256	6324	5874	5632	4962	0.0936
Gold hill (512×512)	74752	6654	6241	5896	5214	0.0872
	149504	6452	5745	5674	5247	0.0935
	224256	6324	5874	5632	4962	0.09367

Table 3.27: Experimental results of relative entropy in EDRDHHC

Image	Data(bits)	Entropy of I	Entropy of $SM^{'}$	Entropy of $SA^{'}$	$D(SM^{'}\|\|I)$	$D(SA^{'}\|\|I)$
Lena (512×512)	74752	7.4451	7.4512	7.4520	0.0061	0.0069
	149504	7.4451	7.4559	7.4565	0.0108	0.0114
	224256	7.4451	7.4596	7.4620	0.0145	0.0169
Barbara (512×512)	74752	7.0480	7.0540	7.0550	0.0060	0.0070
	149504	7.0480	7.0570	7.0580	0.0090	0.0100
	224256	7.0480	7.0610	7.0625	0.0130	0.0145
Tiffany (512×512)	74752	7.2925	7.3010	7.0390	0.0085	0.0165
	149504	7.2925	7.3120	7.33125	0.0195	0.0200
	224256	7.2925	7.3150	7.3160	0.0225	0.0235
Pepper (512×512)	74752	7.2767	7.2872	7.2971	0.0105	0.0204
	149504	7.2767	7.3172	7.3191	0.0406	0.0425
	224256	7.2767	7.3257	7.3298	0.0492	0.0533
Gold hill (512×512)	37720	7.2367	7.2387	7.2398	0.0020	0.0031
	64800	7.2367	7.2464	7.2485	0.0097	0.0118
	74752	7.2367	7.2490	7.2499	0.0123	0.0132

It is observed that there is no significant difference between the standard deviation of the cover image and the stego image. This study shows that the magnitude of change in stego-image based on image parameters is small from cover image. Since the image parameters have not changed much, so the method offers good concealment of secret data and reduces the chances of detection. Thus, it suggests a secure steganographic system.

Table 3.28: Experimental results of SD (σ) and CC (ρ) in EDRDHHC

Image	SD (σ)			CC (ρ)		
	Cover image (I)	Stego image ($SM^{'}$)	Stego image ($SA^{'}$)	I & $SM^{'}$	I & $SA^{'}$	$SM^{'}$ & $SA^{'}$
Lena (512×512)	47.8385	46.4867	46.6873	0.9752	0.9757	0.9754
Barbara (512×512)	38.3719	37.4945	37.8545	0.9751	0.9765	0.9756
Tiffany (512×512)	61.5978	60.4221	60.9442	0.9814	0.9774	0.9752
Pepper (512×512)	52.1356	51.8987	52.241	0.9793	0.9791	0.9784
Gold hill (512×512)	58.8723	57.8974	57.8776	0.9764	0.9728	0.9712

3.5.4.4 Histogram Attack

Fig 3.27 have shown the histogram of the cover and stego images and their difference histogram. The stego image is produced from cover image employing the maximum data hiding capacity. It is observed that the shape of the histogram is preserved after embedding the secret data. The difference of the histogram is very small. The bins that are close to zero are more in numbers and the bins which are away from zero are less in numbers. This confirms better quality of stego image. There is no step pattern observed which ensures the proposed method is robust against histogram analysis.

3.5.4.5 Brute Force Attack

Two shared secret keys δ and ξ have been used during both embedding and extraction stage. The scheme is secure to prevent possible malicious attacks. The proposed scheme constructs two stego images which protect original information by hiding secret information in both images $SM^{'}$ and $SA^{'}$. The Fig 3.28 shows the revelation example where both the keys are unknown. If the malicious attacker holds the original image and dual images and is fully aware of the proposed scheme, the hidden message still cannot be correctly revealed without knowing the correct secret keys. For example, Fig. 3.28 shows two stego derived from lena images using secret keys which are different from that used to construct without knowing secret key. The result indicates that the attacker only acquires noise -like image when applying incorrect secret key to reveal the hidden message. Furthermore, the attacker may employ the brute force attack that tries all possible permutation to reveal the hidden message. The total number of trials to revels the hidden message are $2 \times 7 \times 18 \times m \times n/B \times 2^{length\ of\ \xi}$, where $(m \times n)$ is the size of the cover image and B is the block size, which are computationally unfeasible for current computers. The proposed scheme is robust against several attacks. Furthermore, the secret

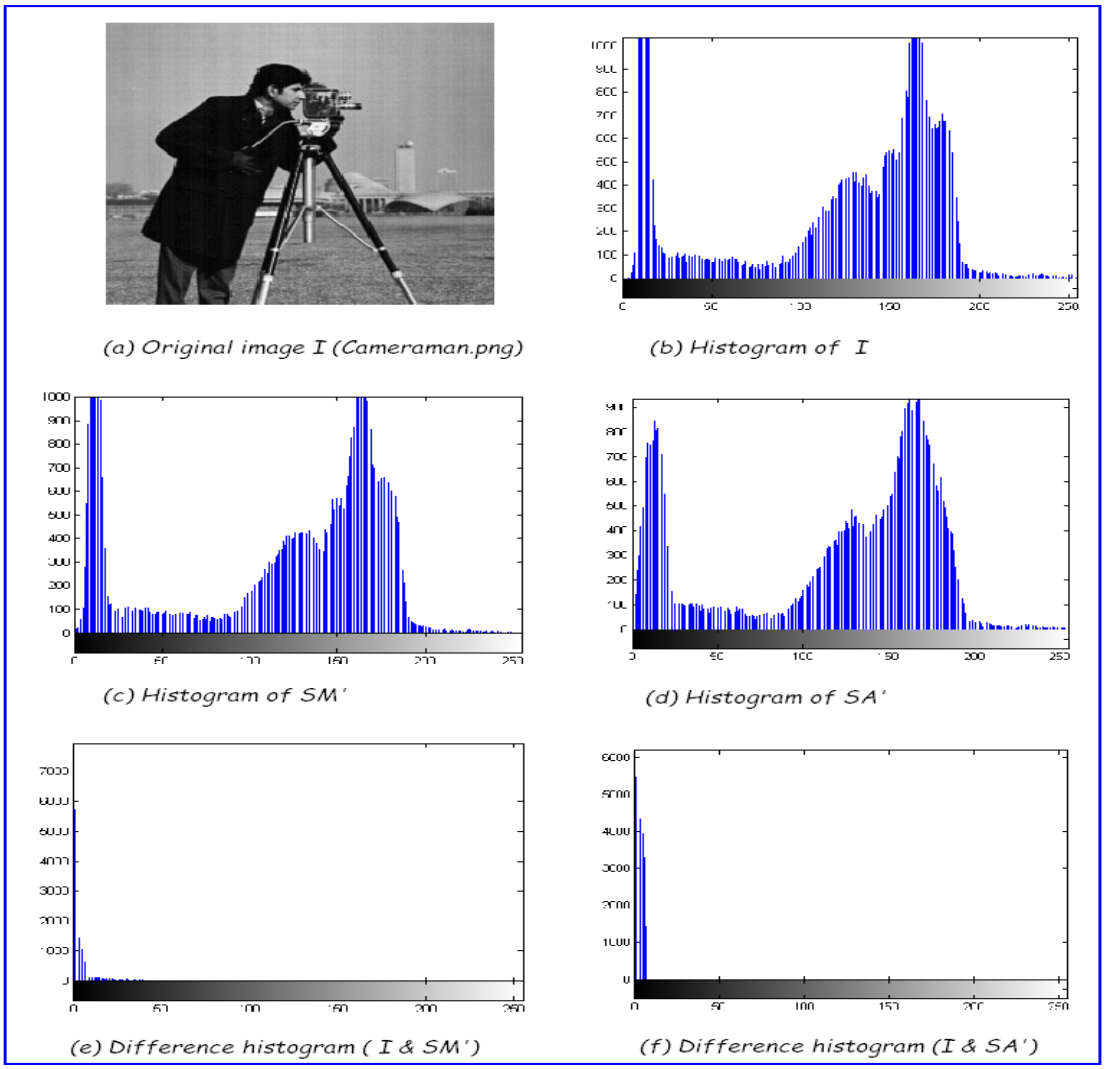


Figure 3.27: Histogram of original image, stego images and their difference in EDRDHHC

information can be retrieved without encountering any loss of data using correct secret keys and original image can be recovered successfully from dual image.

3.5.5 Key Space

Two shared secret keys δ and ξ have been used during data embedding and extraction stage. Another secret position κ and data embedding position ω is also used during data hiding and extraction stage but these are not shared secret keys. The κ has been updated by the data embedding position ω for each new row. One can use more than one key in a single row that will enhance security but increase computational cost. The possible number of keys are $2 \times 7 \times 18 \times m \times (n/B) \times 2^{length\ of\ \xi}$ which are explained below:

Corollary 3.5.1 *Possible number of blocks for a* $(m \times n)$ *cover image is* $\lfloor (m \times n/B) \rfloor$*, where*

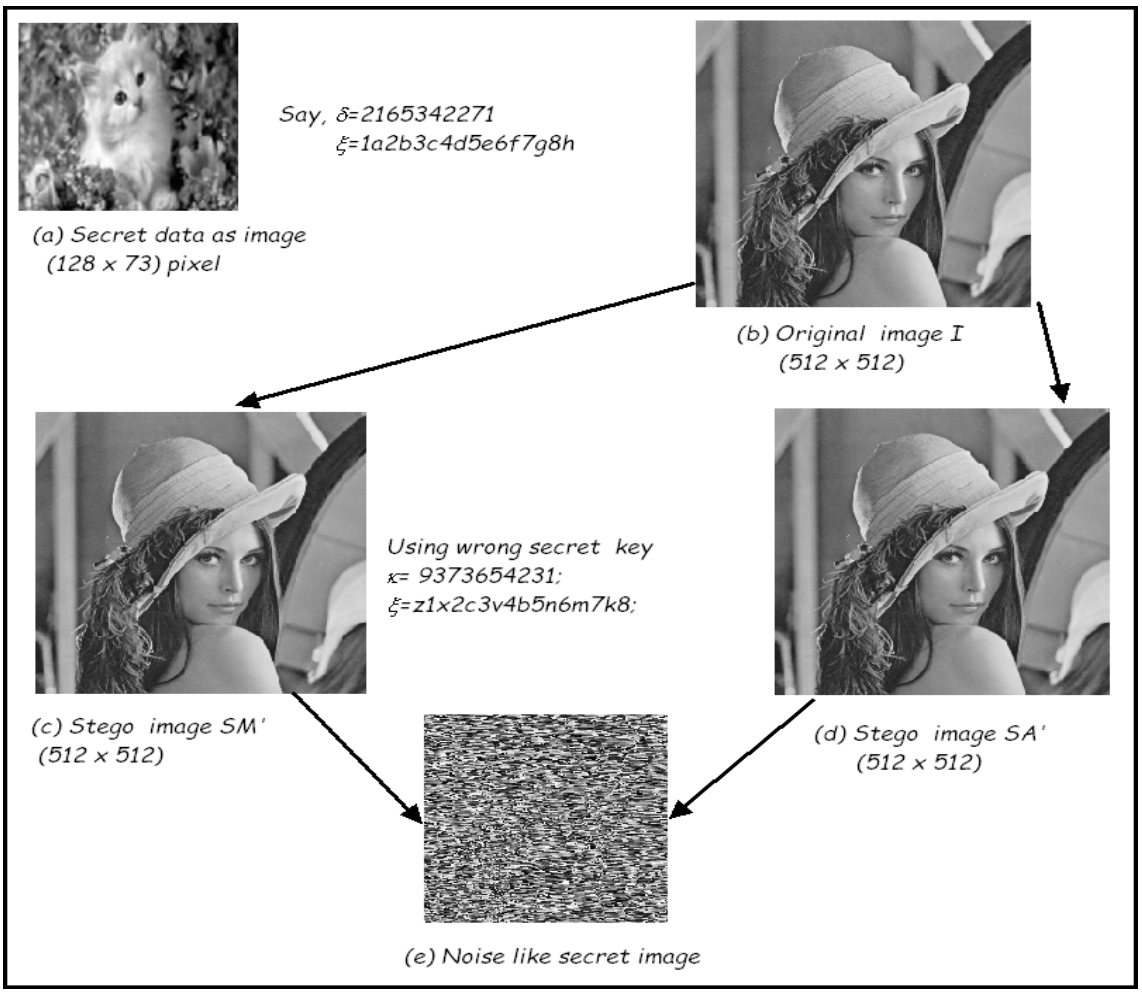


Figure 3.28: Experimental result of Brute Force Attack in EDRDHHC

B is the block size. ■

Corollary 3.5.2 *Possible number of secret key κ for a $(m \times n)$ cover image is $2 \times B \times B - 1 \times number\ of\ layers$ for dual image.* ■

Corollary 3.5.3 *Possible number of secret key ξ are $2^{length\ of\ \xi}$. Combining with κ, the maximum possible combination will be for a $(m \times n)$ cover image is $2 \times B \times B - 1 \times number\ of\ layers \times 2^{length\ of\ \xi}$.* ■

Example 3.5.1 *Consider a cover image of size (512×512), where m = 512 and n = 512 is the row and column of cover image and block size is 7 so, the possible number of keys κ is equal to $2 \times 7 \times (7-1) \times 3 \times \lfloor (512 \times 512/7) \rfloor$ = 9437184. Total possibility of secret keys ξ of length 128 (say) is equal to 2^{128}. So the combination of total number of possibilities to reveal the keys are equal to 9437184×2^{128}.* ■

3.6 Analysis and Discussions

The comparison of four proposed Hamming code based data hiding schemes are described below in Table 3.29. From this table it is observed that the payload is better in our reversible data hiding schemes because dual image has been used which increases data hiding capacity. The visual quality of PRDHHC and DRDHHC is better than EPRDHHC and EDRDHHC because only one LSB layer have been considered in earlier schemes but three LSB layers has been considered in the later scheme to increase payload. The maximum payload is 0.426 (bpp) in these proposed schemes but reversibility has been achieved. The secret keys play an important role in enhancing security.

Table 3.29: Comparisons of developed schemes in terms of PSNR (dB) and payload (bpp)

Schemes	Reversible/ Irreversible	Single/Dual	Capacity (bits)	PSNR (dB)	Payload (bpp)
PRDHHC	Irreversible	Single	37,306	50.13	0.142
DRDHHC	Reversible	Dual	74,606	51.75	0.142
EPRDHHC	Irreversible	Single	1,11,909	32.14	0.426
EDRDHHC	Reversible	Dual	2,23,818	38.23	0.426

The stego images of these schemes are analyzed through RS analysis, We calculate relative entropy and find the correlation coefficient (ρ) which are shown in Table 3.30. We have also shown the total number of trails required to reveal the secret message is $94, 37, 184 \times 2^{128}$ which are computationally unfeasible for current computers.

Table 3.30: Comparisons of proposed schemes in terms of steganalysis values

Proposed schemes	Capacity (bits)	PSNR (dB)	RS value	Relative Entropy	CC(ρ)	Payload (bpp)
PDRHHC	37,306	50.13	0.0357	0.0069	0.9864	0.142
DRDHHC	74,606	51.75	0.0578	0.0086	0.9834	0.142
EPRDHHC	1,11,909	32.14	0.667	0.0212	0.9786	0.426
EDRDHHC	2,23,818	38.23	0.1120	0.145	0.9752	0.426

To improve the payload some innovative reversible data hiding schemes have been proposed using Pixel Value Difference (PVD), Difference Expansion (DE) and Exploiting Modification Direction (EMD) method which are discussed in next chapter.

Chapter 4

RDH using PVD, DE and EMD

4.1 Introduction

Communication through data hiding is carried out by concealing the existence of secret information within cover media. If the existence of secret information is revealed, data hiding fails. It can meet both legal and illegal interests. For example, civilians may use it for protecting privacy while terrorists may use it for spreading terrorist information. For a sophisticated data hiding strategies, it has been proven in practice that one efficient style of increasing security is to reduce the number of changes that is inserted into the cover image. However, the embedding efficiency and payload cannot reach the best level when the schemes are used to deal with gray scale image. Recently, reversible data hiding plays very important role in medical image processing and military communications. Information can be embedded into image which contains ownership identification, authentication and copy right protection. Designing a novel data hiding system accomplishing good visual quality, high embedding capacity, robustness and protection is a technically challenging problem. After the confidential message extraction, the requirement for the image reversibility for the entire recovery of the original object without any distortion goes high.

To improve data hiding capacity in terms of the payload using pairs of pixel, some innovative dual-image based RDH schemes using PVD, DE and EMD method have been proposed. In the previous chapter, Hamming code based data hiding schemes have been developed with shared secret keys. The image quality and security was good but the embedding capacity was limited. To improve the data hiding capacity, three new dual image based reversible data hiding schemes have been introduced in this chapter. All these proposed schemes are based on PVD and combined with DE and EMD. The first data hiding scheme is based on PVD with DE (PVDDE). The PSNR of PVDDE is 38.95 (dB) and payload is 1.25 (bpp). To increase the data hiding capacity while preserving good visual quality another new dual image based RDH scheme has been presented using PVD with EMD (PVDEMD). In this approach, payload is 1.75 (bpp) with PSNR 40.43 (dB). Again an attempt has been made to increase the data hiding capacity with a tolerant level of visual quality. Then another new dual-image based RDH scheme has been proposed using TPVD with DE (TPVDDE). The payload of this approach is increased to 2.16 (bpp), but quality has been decreased and it is only 26.18 (dB).

4.2 Dual Image based RDH using PVD with DE (PVDDE)[5]

In this section, a new dual-image based RDH scheme has been proposed using PVD with DE (PVDDE). Here, the secret message is partitioned into sub-stream of size n bits, where $(n-1)$ bits are embedded using PVD and 1 bit is embedded using DE. First, we consider two consecutive pixels from cover image then calculate the difference between them. Now, embed $(n-1)$ secret data bits by modifying the pixel pair using PVD and embed 1 bit secret data using embedding function of DE. After embedding n bits secret data, two pair of pixel has been produced. Finally, we distribute these two stego pixel pair among dual stego images depending on a shared secret key. At the receiver end, secret message has been successfully extracted and original image has been recovered from dual stego images without any distortion. The experimental results and comparisons with other state-of-the-art methods are presented here. Analysis of stego images has been performed using RS analysis, Histogram analysis, Statistical analysis. Some steganographic attacks also has been shown here as case study.

4.2.1 Data Embedding Process

The schematic diagram of proposed PVDE method for data embedding process is shown in Fig 4.1. The corresponding data embedding algorithm is listed in Algorithm 9. The numerical illustration of data embedding also shown in Fig 4.2. According to this approach, first select two consecutive pixels x_i and x_{i+1} from original image I then calculate the pixel value difference d between them.

$$d = |x_i - x_{i+1}| \tag{4.1}$$

The number of secret data bits to be embedded within the selected pixel pair is determined by the sub-range of reference table R. The reference table R have equal sub-range $[lb, ub]$ having the length wb that is $wb = ub - lb + 1$. In this scheme, wb has been fixed as 16 unit. Hence, the contiguous sub-ranges are $\{0-15,\ 16-31,\ 32-47,\ \ldots,\ 240-255\}$ which have capabilities to embed four bits secret data within each pixel pair using PVD. Now to embed these four bits secret data, two new two parameters $d^{'}$ and $d^{''}$ have been introduced and calculated as

[5]Published in the **International Journal of Network Security**, Vol.18, No.4, pp.633-643, July 2016,(Impact Factor - 1.3921), with title *A Dual-image based Reversible Data Hiding Scheme using Pixel Value Difference Expansion*

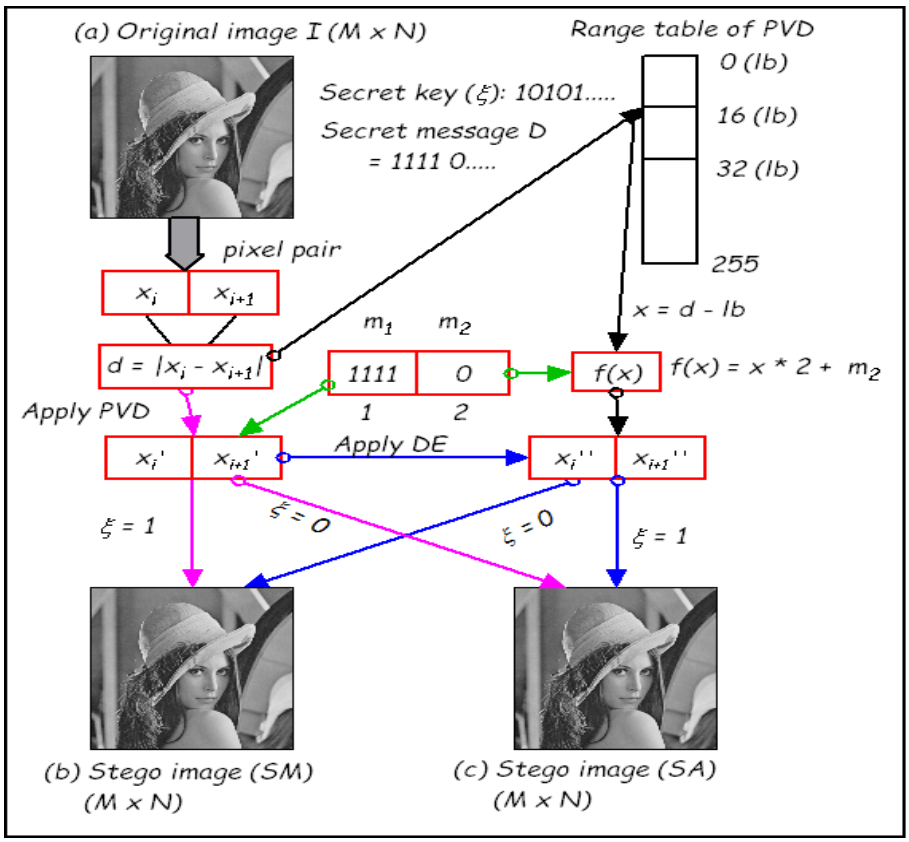


Figure 4.1: Schematic diagram of data embedding process in PVDDE scheme

follows:

$$d^{'} = lb + m_1 \tag{4.2}$$

$$d^{''} = d^{'} - d \tag{4.3}$$

where m_1 is the decimal value of four bits secret message. After that, the pixel values x_i and x_{i+1} are modified and two new pixel values $x_i^{'}$ and $x_{i+1}^{'}$ are produced as follows: If $x_i > x_{i+1}$ then

$$x_i^{'} = x_i + f \tag{4.4}$$

$$x_{i+1}^{'} = x_{i+1} - c$$

else

$$x_i^{'} = x_i - c \tag{4.5}$$

$$x_{i+1}^{'} = x_{i+1} + f$$

where $c = \lceil \frac{d^{''}}{2} \rceil$ and $f = \lfloor \frac{d^{''}}{2} \rfloor$. After that, DE method is applied on the pixel pair $x_i^{'}$ and $x_{i+1}^{'}$ to embed one bit data. Now, select the lower range (lb) from the sub-range of reference table R

Input: Original image I ($M \times N$), Secret data D, Shared secret key ξ.

Output: Two stego images, Stego Major (SM) and Stego Auxiliary (SA) of size $(M \times N)$.

Step 1: Select pixel pair (x_i, x_{i+1}) from I in raster scan order;

Step 2: Calculate the difference $d = |x_i - x_{i+1}|$;

Step 3: Select 4 bits secret data from D and convert it into decimal value m_1 and 1 bit as m_2;

Step 4: Calculate $d^{'} = m_1 + lb$; where, lb is the lower bound of the sub range of reference table R in which d is mapped;

Step 5: Calculate $d^{''} = d^{'} - d$;

Step 6: Compute $c = \lceil \frac{d^{''}}{2} \rceil$ and $f = \lfloor \frac{d^{''}}{2} \rfloor$;

Step 7:

if $(x_i > x_{i+1})$ **then**
  $x_i^{'} = x_i + f$; $x_{i+1}^{'} = x_{i+1} - c$;
else
  $x_i^{'} = x_i - c$; $x_{i+1}^{'} = x_{i+1} + f$;
end

Step 8: Calculate $h_1 = (d - lb)$;

Step 9: Calculate $h_1^{'} = 2 \times h_1 + m_2$; where, m_2 is 1 bit secret message;

Step 10: Calculate Average $A = \lfloor \frac{(x_i^{'} + x_{i+1}^{'})}{2} \rfloor$;

Step 11: Calculate $c_1 = \lceil \frac{h^{'}}{2} \rceil$; and $f_1 = \lfloor \frac{h^{'}}{2} \rfloor$;

Step 12:

if $(x_i^{'} > x_{i+1}^{'})$ **then**
  $x_i^{''} = A + c_1$; $x_{i+1}^{''} = A - f_1$;
else
  $x_i^{''} = A - f_1$; $x_{i+1}^{''} = A + c_1$;
end

Step 13:

if $(\xi = 1)$ **then**
  Store $(x_i^{'}, x_{i+1}^{'})$ within stego image SM and store $(x_i^{''}, x_{i+1}^{''})$ within stego image SA;
else
  Store $(x_i^{'}, x_{i+1}^{'})$ within stego image SA and store $(x_i^{''}, x_{i+1}^{''})$ within stego image SM;
end

Step 14: Repeat **Step 1** to **Step 13** until $length(D) = 0$;

Step 15: Dual stego image SM and SA are produced;

Step 16: End

Algorithm 9: Data embedding process of PVDDE

where the difference d is mapped. Then calculate the parameters h_1, A and $h_1^{'}$ as follows:

$$h_1 = (d - lb) \tag{4.6}$$

$$A = (x_i^{'} + x_{i+1}^{'})/2 \tag{4.7}$$

$$h_1^{'} = (2 \times h_1 + m_2) \tag{4.8}$$

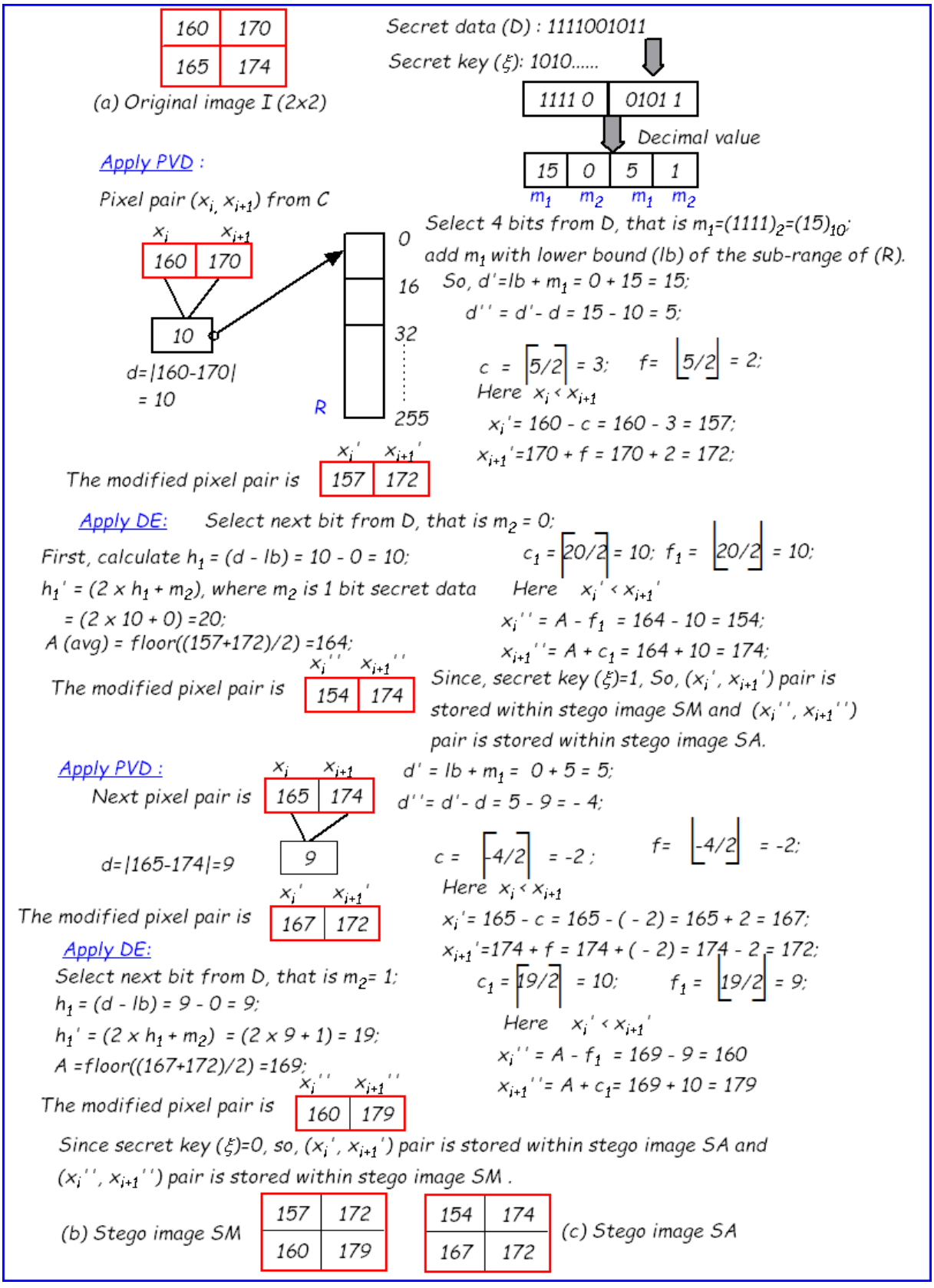


Figure 4.2: Numerical example of data embedding in PVDDE method

where m_2 is one bit secret message. Again, the pixel pair ($x_i^{'}$, $x_{i+1}^{'}$) is modified and produced $x_i^{''}$ and $x_{i+1}^{''}$ as follows: If ($x_i^{'} > x_{i+1}^{'}$) then

$$\begin{aligned} x_i^{''} &= A + c_1 \\ x_{i+1}^{''} &= A - f_1 \end{aligned} \tag{4.9}$$

else

$$\begin{aligned} x_i^{''} &= A - f_1 \\ x_{i+1}^{''} &= A + c_1 \end{aligned} \tag{4.10}$$

where $c_1 = \lceil (h_1^{'}/2) \rceil$ and $f_1 = \lfloor (h_1^{'}/2) \rfloor$. Finally, two stego pixel pairs $(x_i^{'}, x_{i+1}^{'})$ and $(x_i^{''}, x_{i+1}^{''})$ are distributed among dual stego images, Stego Major (SM) and Stego Auxiliary (SA) based on shared secret key bits ξ. If $\xi = 1$, then the pixel pair $(x_i^{'}, x_{i+1}^{'})$ is stored within the stego image SM and the pixel pair $(x_i^{''}, x_{i+1}^{''})$ is stored within the stego image SA else the pixel pair $(x_i^{'}, x_{i+1}^{'})$ is stored within the stego image SA and the pixel pair $(x_i^{''}, x_{i+1}^{''})$ is stored within the stego image SM. Finally, dual stego images have been produced.

4.2.2 Data Extraction Process

The data extraction procedure of proposed PVDDE scheme has been explained through a schematic diagram shown in Fig 4.3. At the receiver end, both the secret data extraction and original image reconstruction have been performed by considering pixel pair from both the stego images SM and SA. First, we apply the shared secret key ξ to select the pixel pair from both stego images SM and SA for specific operation either PVD or DE. If $\xi = 1$, then select pixel pair $(x_i^{'}, x_{i+1}^{'})$ from SM and apply data extraction procedure using PVD and at the same time select pixel pair $(x_i^{''}, x_{i+1}^{''})$ from SA and apply data extraction procedure using DE. If $\xi = 0$, then apply the pixel pair selection process opposite to the previous, that means select pixel pair $(x_i^{'}, x_{i+1}^{'})$ from stego image SA and $(x_i^{''}, x_{i+1}^{''})$ from stego image SM. The corresponding algorithm for data extraction and recovery of original image is listed in Algorithm 10.

Now, the secret data extraction and original image recovery process have been performed as

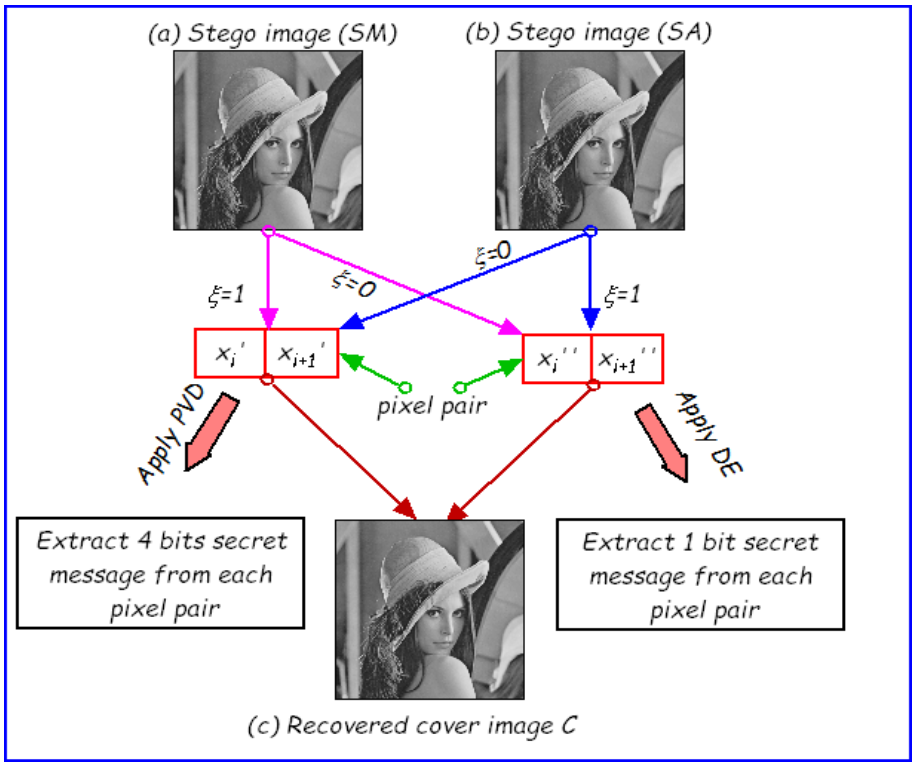


Figure 4.3: Schematic diagram of data extraction process in PVDDE scheme

follows: First, calculate the difference d between the pixel pair $(x_i^{'}, x_{i+1}^{'})$ and extract m_1 as

$$d = |x_i^{'} - x_{i+1}^{'}| \tag{4.11}$$

$$m_1 = d - lb, \tag{4.12}$$

where lb is the lower bound of the sub-range of the reference table R where d is mapped and m_1 is the 4 bits secret data. Now, we have perform

$$h_1^{'} = x_i^{''} - x_{i+1}^{''} \tag{4.13}$$

and collect one bit secret message (m_2) from LSB of $h_1^{'}$. To recover the original image, we have performed the following computations:

$$h_1 = \lfloor \frac{h_1^{'}}{2} \rfloor \tag{4.14}$$

$$d^{'} = (h + lb) \tag{4.15}$$

$$d^{''} = d^{'} - d \tag{4.16}$$

$$c = \lceil \frac{d^{''}}{2} \rceil \tag{4.17}$$

$$f = \lfloor \frac{d^{''}}{2} \rfloor \tag{4.18}$$

Input: Two stego images SM and SA, Shared secret key ξ.

Output: Original cover image $I(M \times N)$; Secret Data D;

Step 1: Select pixel pair from SM and SA in raster scan order;

Step 2:

if $(\xi = 1)$ **then**
  Collect $(x_i^{'}, x_{i+1}^{'})$ from SM and collect $(x_i^{''}, x_{i+1}^{''})$ from SA;
else
  Collect $(x_i^{'}, x_{i+1}^{'})$ from SA and collect$(x_i^{''}, x_{i+1}^{''})$ from SM;
end

Step 3: Calculate $d^{'} = |x_i^{'} - x_{i+1}^{'}|$;

Step 4: Extract secret message $m_1 = d^{'} - lb$, where lb is the lower bound of the sub-range of range table R;

Step 5: Calculate $h_1^{'} = (x_i^{''} - x_{i+1}^{''})$; (Extract secret message bit m_2 from LSB of $h_1^{'}$);

Step 6: Calculate $h_1 = \lfloor \frac{h_1^{'}}{2} \rfloor$;

Step 7: Calculate $d = (h_1 + lb)$; where lb is the lower bound of the sub-range of the reference table R where $d^{'}$ is mapped;

Step 8: Calculate $d^{''} = d^{'} - d$;

Step 9: Calculate $c = \lceil \frac{d^{''}}{2} \rceil$;

Step 10: Calculate $f = \lfloor \frac{d^{''}}{2} \rfloor$;

Step 11:

if $(x_i^{'} > x_{i+1}^{'})$ **then**
  $x_i = x_i^{'} - f$; $x_{i+1} = x_{i+1}^{'} + c$;
else
  $x_i = x_i^{'} + c$; $x_{i+1} = x_{i+1}^{'} - f$;
end

Step 12: Repeat **Step 1** through **Step 11** until all secret data are extracted;

Step 13: End

Algorithm 10: Data extraction process of PVDDE

Now, the pixel pair (x_i, x_{i+1}) has been recovered from the stego image using

$$(x_i, x_{i+1}) = \begin{cases} x_i^{'} - f,\ x_{i+1}^{'} + c & \text{if } x_i^{'} > x_{i+1}^{'} \\ x_i^{'} + c,\ x_{i+1}^{'} - f & \text{otherwise} \end{cases} \tag{4.19}$$

The corresponding numerical example is shown in Fig 4.4.

4.2.3 Overflow and Underflow Control

When stego pixel values become more than the upper limit of gray scale [0, 255] then overflow situation occurs and when stego pixel value becomes less than the lower limit of gray scale then underflow occurs. In this scheme, 8 bits gray scale image [0 - 255] have been used, where upper limit is 255 and lower limit is 0 (zero). Suppose a pixel pair (x_i, x_{i+1}) with pixel values $x_i = 250$ and $x_{i+1} = 255$ needs to embed 4 bits secret data value $(1101)_2$ that is $(13)_{10}$. The difference between pixel pair d is $\mid 250 - 255 \mid$ = 5 and the new difference $d^{'}$ is 13 + 0 = 13. Therefore, $d^{''}$ = 13 - 5 = 8, c = 4 and f = 4. After embedding secret message, the stego pixel

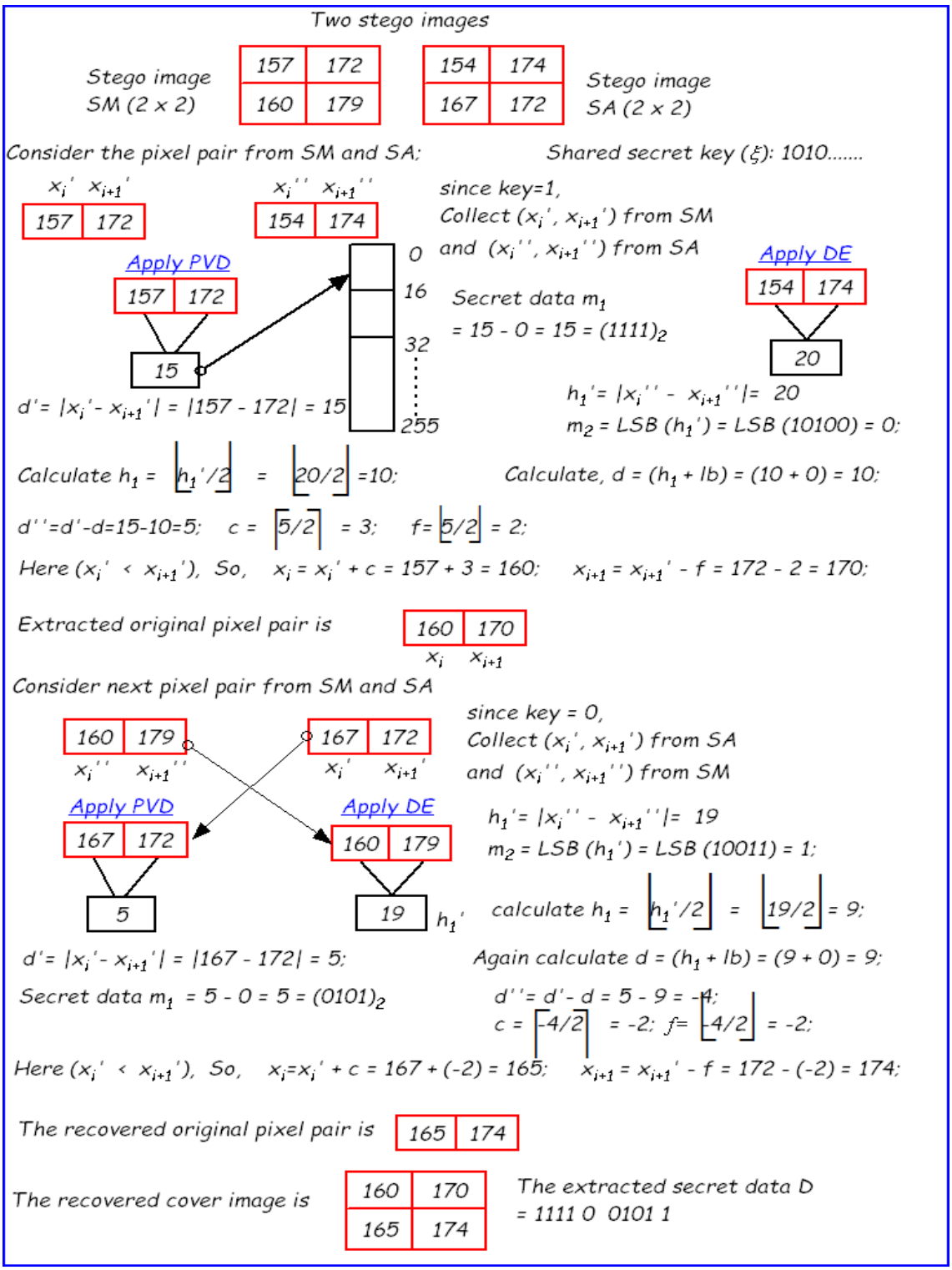


Figure 4.4: Numerical example of data extraction in PVDDE method

pair becomes $x_i^{'} = 246$ and $x_{i+1}^{'} = 259$ which crosses the upper limit of gray scale that means $x_{i+1}^{'} > 255$ which shows overflow condition. For underflow, suppose x_i = 0 and x_{i+1} = 7 and try to embed 4 bits secret data $(1010)_2$ that is $(10)_{10}$. The difference between pixel pair d is

$\mid 0 - 7 \mid = 7$ and the new difference $d^{'}$ is 10 + 0 = 10. Therefore, $d^{''}$ = 10 - 7 = 3, c = 2 and f = 1. The stego pixel pair becomes $x_i^{'} = -2$ and $x_{i+1}^{'} = 8$. It is observed that $x_i^{'} < 0$ shows underflow condition. To overcome this situation, do not embed any secret data bit within those specified pixel pair where overflow and underflow condition may occur. It is observed that after embedding, the difference between pixel pair is not greater than 31. To overcome the overflow problem, difference expansion method is used and set the difference 32 when data hiding by difference expansion is 0 and subtracting 32 from the average of two pixels. So, the modified pixel pair becomes $(x_i^{''} = avg - 32, x_{i+1}^{''} = x_{i+1}^{'})$ and set the difference 33 when data is 1 by subtracting 33 from the average of two pixels. So, the modified pixel pair becomes $(x_i^{''} = avg - 33, x_{i+1}^{''} = x_{i+1}^{'})$. To overcome the underflow problem, set the difference 32 when data is 0 by adding 32 with the average of two pixels. So, the modified pixel pair will be $(x_i^{''} = avg + 32, x_{i+1}^{''} = x_{i+1}^{'})$ and set the difference 33 when data is 1 by adding 33 with the average of two pixels. So, the modified pixel pair will be $(x_i^{''} = avg + 33, x_{i+1}^{''} = x_{i+1}^{'})$. In the receiver side, when difference between the pixels $x_i^{''}$ and $x_{i+1}^{''}$ is 32 or 33 the receiver understand that the secret message is not embedded within that pair $(x_i^{'}, x_{i+1}^{'})$ corresponding to $(x_i^{''}, x_{i+1}^{''})$.

4.2.4 Experimental Results and Comparisons

The proposed algorithm is verified using some standard gray scale images of size (256×256) pixels as cover image shown in Fig 4.5. After data embedding dual stego image SM and SA are produced which are shown in Fig 4.6. The data embedding and data extraction algorithms are implemented in MATLAB Version 7.6.0.324 (R2008a).

To measure the embedding capacity, we calculate payload (p) in terms of bits per pixel (bpp) using the following expression.

$$p = \frac{(\lfloor \frac{M}{2} \rfloor - 1) \times (\lfloor \frac{N}{2} \rfloor - 1)}{(2M \times 2N)} \tag{4.20}$$

Here, The payload (p) of this dual image based PVDDE scheme is 1.25 bpp. To measure the complexity, we assume that the size of the cover image is $(m \times n)$ and the data embedding process embed five secret bits within a pixel pair. Two copies of cover image are used as dual stego image and each pixel pair from cover image generate two copies of pixel pair. So, the time

Figure 4.5: Standard cover images of size (256×256) pixel used in PVDDE scheme

complexity of data embedding algorithm is $\mathcal{O}(mn)$. On the other hand, during data extraction, it is required to scan the pixel pair from dual image depending on the key. So, the time complexity of extraction algorithm is $\mathcal{O}(mn)$.

Visual quality of stego images are measured through PSNR shown in Table 4.1. Table 4.2 shows the PSNR values of existing dual image based RDH schemes. The PSNR of the stego images of the proposed PVDDE scheme is lower than the method proposed by Qin et al.'s [52], Lu et al.'s [45, 46], Chang et al.'s [5, 6] and Lee et al.'s [34, 36] schemes. But the PSNR of proposed PVDDE method is higher than Lee et al.'s [33] and Zeng et al.'s [75] schemes. The payload of this scheme is 1.25 (bpp), which is higher than the other existing dual image based schemes. The embedding capacity of the methods proposed by Qin et al. is approximately 0.09 (bpp) less than that of PVDDE method. The payload of Lu et al. and Chang et al. is approximately 0.25 (bpp) less than PVDDE method. It is observed that proposed PVDDE is superior than the other dual image based schemes in terms of embedding capacity. From the

Figure 4.6: Generated dual stego images of size (256×256) pixel from PVDDE scheme

Table 4.1: PSNR (dB) of stego images after data embedding in PVDDE scheme

Image	Data (bits)	PSNR (SM)	PSNR (SA)	Avg. PSNR
Cameraman	40,000	43.40	42.72	36.77
	80,000	35.75	38.84	
	1,60,000	30.77	36.19	
	1,63,592	30.35	36.14	
House	40,000	47.00	41.88	38.95
	80,000	40.59	38.53	
	1,60,000	35.84	36.01	
	1,63,592	35.79	35.97	
Lena	40,000	40.31	43.78	36.93
	80,000	35.31	40.19	
	1,60,000	30.77	37.28	
	1,63,592	30.67	37.18	
Peppers	40,000	39.67	43.47	37.27
	80,000	35.45	39.93	
	1,60,000	32.92	36.98	
	1,63,592	32.86	36.89	
Pirate	40,000	39.79	43.75	37.05
	80,000	35.29	40.28	
	1,60,000	31.58	37.15	
	1,63,592	31.48	37.09	
Girl	40,000	34.57	43.54	35.85
	80,000	32.39	40.47	
	1,60,000	30.50	37.51	
	1,63,592	30.44	37.42	
Tiffany	40,000	40.44	43.75	37.36
	80,000	36.32	40.20	
	1,60,000	32.00	37.19	
	1,63,592	31.92	37.12	
Zelda	40,000	42.20	43.76	38.87
	80,000	39.10	40.09	
	1,60,000	36.08	36.98	
	1,63,592	35.86	36.90	
Goldhill	40,000	45.84	42.85	38.66
	80,000	39.77	39.48	
	1,60,000	34.09	36.80	
	1,63,592	34.06	36.76	

above discussion, one can conclude that PVDDE is better than other existing schemes in terms of payload, and the PSNR is also reasonable which implies that the quality of the stego image is good.

Table 4.2: Comparison of PVDDE with existing dual image based RDH schemes

Scheme	PSNR (dB)	Payload (bpp)
Chang et al. [5]	45.1225	1.00
Chang et al. [6]	48.14	1.00
Lee et al. [36]	52.3098	0.74
Lee et al. [33]	34.38	0.91
Zeng et al. [75]	32.74	1.04
Lee and Huang [34]	49.6110	1.07
Qin et al. [52]	52.11	1.16
Lu et al. [45]	49.20	1.00
Proposed PVDDE	38.95	1.25

4.2.5 Steganalysis and Steganographic Attacks

Table 4.3: RS analysis of stego image SM in PVDDE schemes

Image	Data (bits)	SM				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Cameraman	80000	6745	6542	5462	5364	0.0246
	163592	6524	6254	5341	5149	0.0389
House	80000	6345	6147	5632	5486	0.0287
	163592	6452	6241	5624	5241	0.0490
F16	80000	6125	5943	5210	5003	0.0343
	163592	6314	6014	5347	5123	0.0449
Lake	80000	5863	5642	5874	5684	0.0350
	163592	5963	5632	5748	5562	0.0441
Lena	80000	5687	5469	5684	5478	0.0372
	163592	5987	5647	5841	5623	0.0471
Living Room	80000	5512	5263	5547	5689	0.0353
	163592	6001	5789	5641	5426	0.0366
Peppers	80000	5987	5784	5487	5236	0.0395
	163592	6024	5789	5641	5327	0.0470
Pirates	80000	5789	5569	5364	5166	0.0374
	163592	6354	6004	5476	5123	0.0594
Girl	80000	5741	5478	5123	4957	0.0394
	163592	5941	5741	5247	4968	0.0428
Tiffany	80000	6687	6486	5874	5647	0.0340
	163592	6841	6421	5964	5741	0.0502
Zelda	80000	6541	6347	5784	5569	0.0331
	163592	6254	5962	5942	5762	0.0387
Gold hill	80000	6452	6243	5674	5441	0.0364
	163592	6214	6001	5946	5562	0.0490

Steganalysis is the art of discovering whether a secret message exists or not within a suspected

image. The stego image of PVDDE scheme has been analyzed through RS analysis [18]. Other analysis and attacks are depicted here.

4.2.5.1 RS Analysis

When the value of RS analysis is closed to zero it means the scheme is secure. The stego images are tested under the RS analysis. It is observed from Table 4.3 and 4.4 that the values of R_M and R_{-M}, S_M and S_{-M} are nearly equal for stego image SM and SA. Thus rule $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ is satisfied for the stego image in this scheme. So, the proposed PVDDE method is secure against RS attack. In this experiment, the ratio of R and S lies between 0.0246 to 0.0502

Table 4.4: RS analysis of stego image SA in PVDDE scheme

Image	Data (bits)	SA				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Cameraman	80000	5784	5964	5624	5247	0.0488
	163592	6241	6014	5421	5014	0.0543
House	80000	6541	6214	5784	5547	0.0457
	163592	6547	6241	5416	5326	0.0331
F16	80000	6284	5989	5684	5247	0.0611
	163592	6574	6015	5246	5124	0.0576
Lake	80000	6014	5741	5984	5746	0.0425
	163592	6241	5634	5742	5641	0.0590
Lena	80000	6741	6254	5647	5214	0.0742
	163592	5987	5684	5874	6324	0.0634
Living Room	80000	5324	5621	5641	5247	0.0630
	163592	6584	6125	6002	5762	0.0555
Peppers	80000	5987	5784	5487	5236	0.0395
	163592	6354	6014	5641	5230	0.0626
Pirates	80000	5789	5569	5364	5166	0.0374
	163592	6325	6004	5426	5123	0.0531
Girl	80000	5641	5478	5214	4957	0.0386
	163592	5698	5241	5247	5123	0.0530
Tiffany	80000	6475	6521	5874	5647	0.0221
	163592	6874	6541	5964	5475	0.0640
Zelda	80000	6254	6347	5784	5684	0.01603
	163592	5742	6452	5942	5742	0.0778
Gold hill	80000	6328	6354	5674	5541	0.0132
	163592	6412	6241	5946	5641	0.0385

for SM and 0.0132 to 0.0778 for SA of Cameraman image. Other values are shown in the Table 4.3 and 4.4.

4.2.5.2 Relative Entropy

In this experiment, it is shown that when the number of data bits in the secret message increases, the relative entropy of stego image is directly proportional to it. The relative entropy between cover image and SM of lena image varies between 0.006 to 0.0331 when embedded with 40000 to 161992 data bits and SA varies between 0.004 to 0.0286 which implies the proposed scheme provides secure hidden communication. Other relative entropy values with SM and SA are listed in Table 4.5.

Table 4.5: Relative entropy of SM and SA in PVDDE scheme

Image	Data(bits)	Entropy I	Entropy SM	Difference (I& SM)	Entropy SA	Difference (I& SA)
Lena	40000	7.4451	7.4511	0.006	7.4491	0.004
	80000		7.4581	0.013	7.4576	0.0125
	160000		7.4779	0.0328	7.4729	0.0278
	161992		7.4782	0.0331	7.4737	0.0286
Barbara	40000	7.0480	7.051	0.003	7.062	0.014
	80000		7.0575	0.0095	7.0687	0.0207
	160000		7.0623	0.0143	7.0712	0.0232
	161992		7.0723	0.0243	7.0892	0.0412
Tiffany	40000	7.2925	7.2936	0.0011	7.2985	0.006
	80000		7.2998	0.0073	7.3214	0.0289
	160000		7.3125	0.02	7.3621	0.0696
	161992		7.3254	0.0329	7.3685	0.076
Pepper	40000	7.2767	7.2798	0.0031	7.2845	0.0078
	80000		7.2841	0.0074	7.2954	0.0187
	160000		7.3154	0.0387	7.3015	0.0248
	161992		7.3254	0.0487	7.3125	0.0358
Gold hill	40000	7.2367	7.2398	0.0031	7.2445	0.0078
	80000		7.2541	0.0174	7.2654	0.0287
	160000		7.3054	0.0687	7.2815	0.0448
	161992		7.3154	0.0787	7.3025	0.0658

4.2.5.3 Statistical Analysis

The proposed PVDDE scheme is also assessed through statistical analysis. The Standard Deviation (σ) of before and after data embedding and Correlation Coefficient (ρ) of cover and stego images are summarized in Table 4.6. From this table it has been observed that there is no substantial divergence between the standard deviation of the cover image and the stego image. This study shows that the magnitude of change in stego image based on image parameters is small

Table 4.6: Result of SD (σ) and CC (ρ) in PVDDE scheme

Image	SD (σ)			CC (ρ)		
	I	SM	SA	I& SM	I& SA	SM & SA
Baboon	38.37	37.85	38.54	0.98	0.99	0.97
Cameraman	61.59	61.12	61.73	0.99	0.99	0.99
Lena	47.83	47.43	47.97	0.98	0.99	0.98

from a cover image. Since the image parameters have not changed much, the method offers a good concealment of data and reduces the chances of detection. Thus, it indicates a perfectly secure steganographic system.

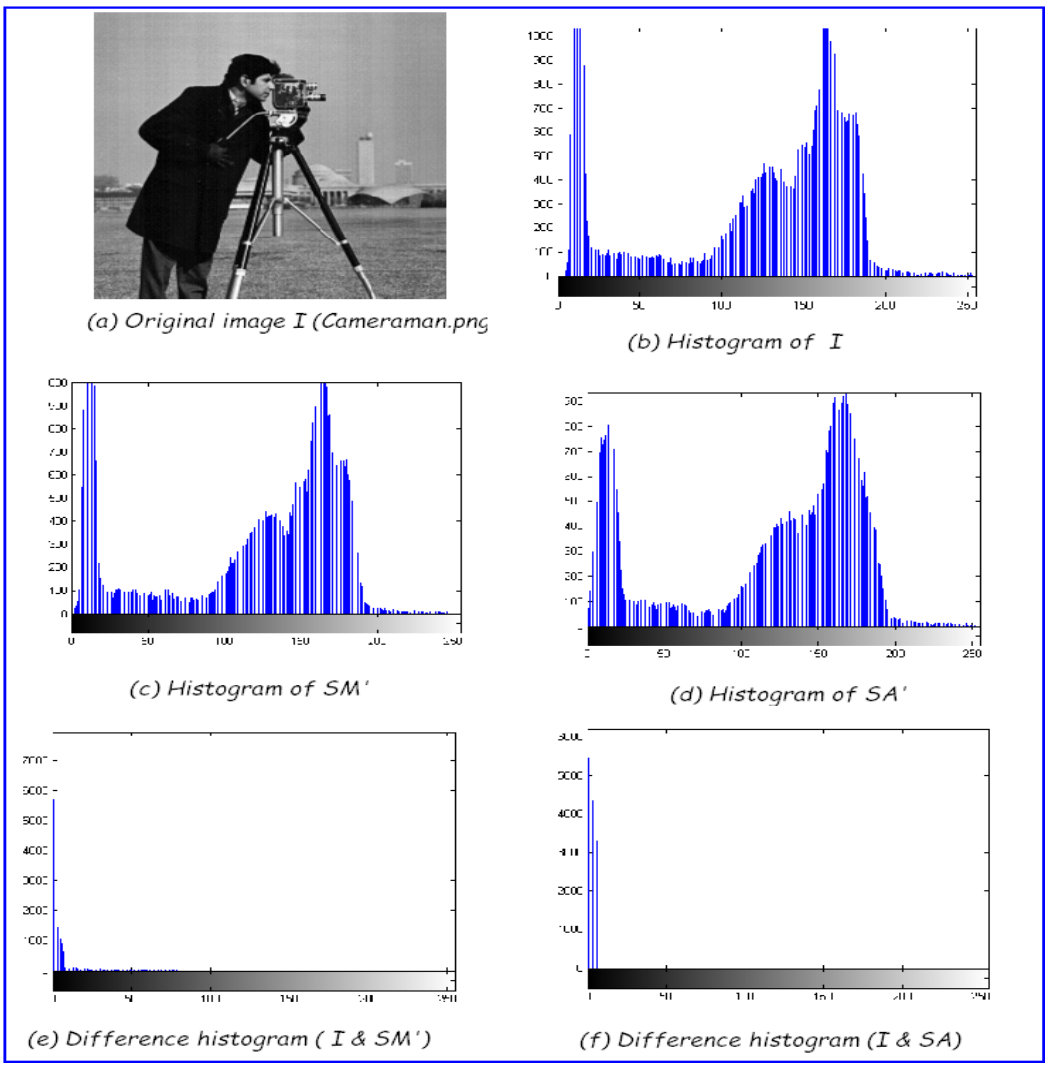


Figure 4.7: Result of Histogram attack in PVDDE scheme

4.2.5.4 Histogram Attack

Fig 4.7 depicted the histogram of the cover and stego image and their difference histogram. The stego image are produced from cover image employing the maximum data hiding capacity. It is observed that the shape of the histogram is preserved after embedding the secret data. The difference of the histogram is very small. It is observed that, bins close to zero are more in number and the bins which are away from zero are less in number. This confirms the good quality of stego images. There is no step pattern observed which ensures that the proposed method is robust against histogram attacks.

4.2.5.5 Brute Force Attack:

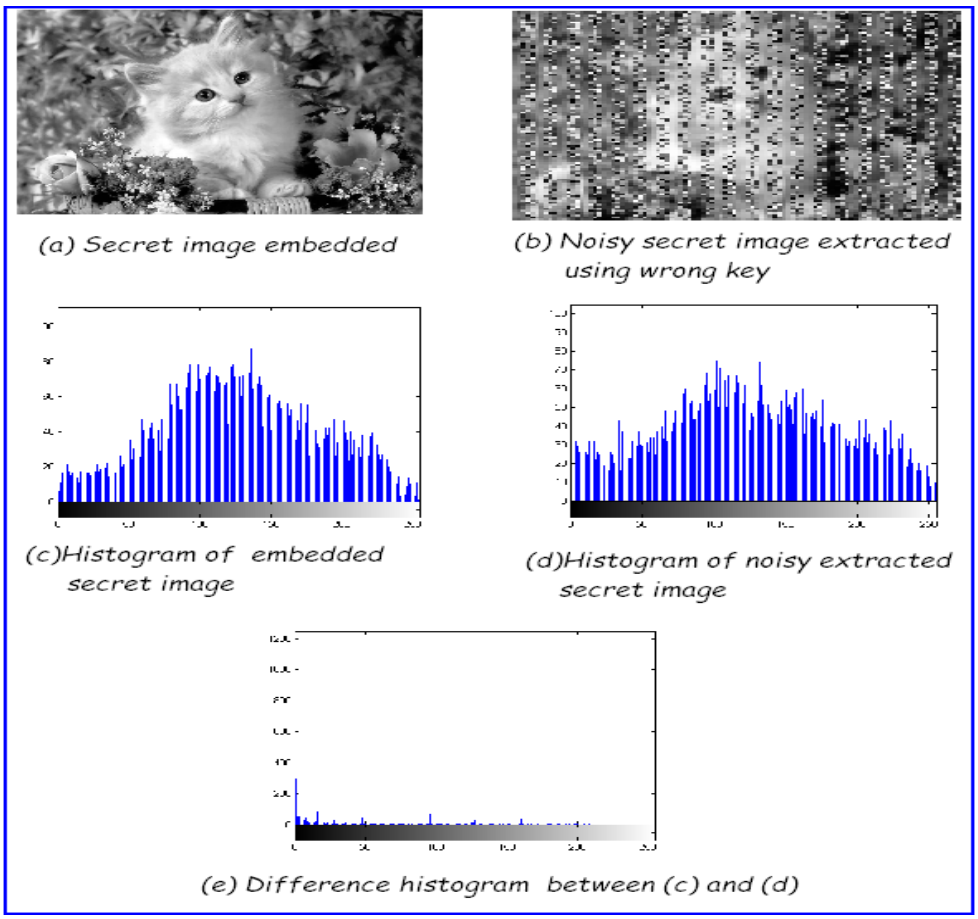


Figure 4.8: Result of Case Study - 1 in PVDDE scheme

In this experiment, a shared secret key ξ is used to distribute the stego pixel among dual images. Here the revelatory example is explained with unknown key.

Case Study - 1: To extract the secret image, if someone uses the wrong key then he can't re-

trieve the exact image but a noisy image. Fig. 4.8 shows the experimental result of case study - 1 of PVDDE method. From Table 4.7 it is observed that the SD (σ) of secret image is 56.2454 and SD (σ) of extracted noisy image is 62.4046. So, the image deviates 6.1592 from original image. Also the CC (ρ) between these two images is 0.6452. Although the CC (ρ) are not equal to 1 but it has been shown that the noisy image and the original secret image are visually identified. To improve the results and get more noisy image, Case Study - 2 is being performed XOR operation between secret key and secret data.

Case Study - 2: To improve the security of the proposed method, XOR operation has been performed between secret message bits and share secret key bits stream. Then the result of XOR operation is embedded within cover image. Then we try to extract secret data with unknown key. Fig. 4.9 shows the experimental result of case study - 2. From Table

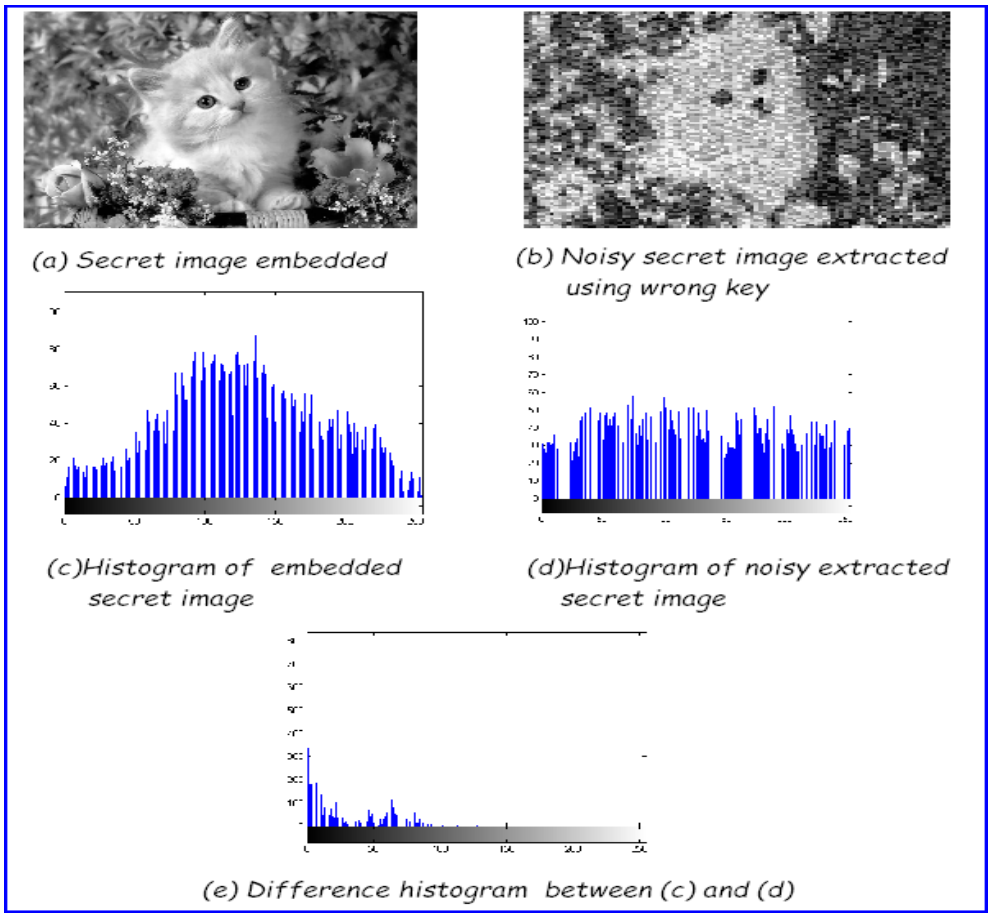


Figure 4.9: Result of Case Study - 2 in PVDDE scheme

4.7 it has been observed that the SD (σ) of secret image is 56.2454 and noisy image is 66.6004. So, the image deviates 10.355 from original image. The CC (ρ) between these

two images is 0.5230. Although case study - 2 gives better results than case study - 1 but some portion of the noisy image is same as the original secret image. So, to enhance security we distribute each pixel within dual stego images depending on shared secret key.

Case Study - 3: In case study - 3, after performing XOR operation, the stego pixels are distributed depending on shared secret key ξ. For this method, if ξ_i is one, the pixel $x_i^{'}$ is stored within the stego image SM and the pixel $x_i^{''}$ is stored within the stego image SA. If ξ_i is zero then the pixel $x_{i+1}^{'}$ is stored within the stego image SM and the pixel $x_{i+1}^{''}$ is stored within the stego image SA. Fig. 4.10 shows the experimental result of case study

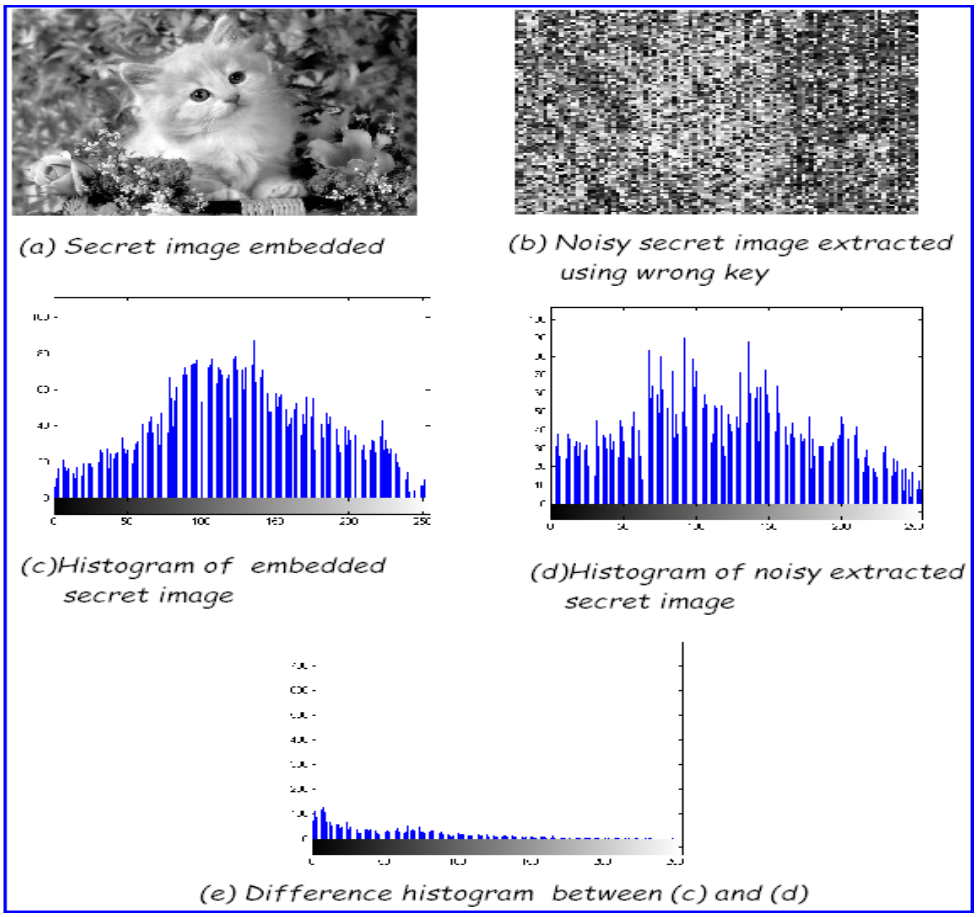


Figure 4.10: Result of Case Study - 3 in PVDDE scheme

- 3 of PVDDE. From Table 4.7 it is observed that the SD (σ) of secret image is 56.2454 and noisy image is 70.9560. So, the image data deviates 14.7106. The CC (ρ) between these two images is 0.3495. The CC (ρ) is tense to 0 and the original stego image and noisy image are not visually identified. In case study - 3, the secret image is not recovered using unknown secret key.

Table 4.7: Result of SD (σ) and CC (ρ) after Brute Force Attack in PVDDE scheme

SD (σ)				CC (ρ)		
Secret Image (SI)	Case - 1	Case - 2	Case - 3	SI & Case - 1	SI & Case - 2	SI & Case - 3
56.2454	62.4046	66.6004	70.9560	0.6452	0.5230	0.3495

Here, image has been used as secret data instead of text documents. The eavesdropper may extract the secret data without knowing the parameters which is identified in case study - 1 and case study - 2. But it is not visually identified in case study - 3 where noisy image is extracted. But when the text documents are used as a secret data then it is hard to extract original secret message because any single bit change will effect on the ASCII value of text document. For image as secret data it is visually observed through open eye but in case of text document it is difficult to identifying it through open eyes.

4.3 Dual Image based RDH using PVD with EMD (PVDEMD)[6]

To improve data hiding capacity in terms of payload using pairs of pixel, a new dual image based RDH scheme has been proposed using PVD and EMD method (PVDEMD), which provides data hiding capacity up to 1.75 (bpp) with good visual quality measured by PSNR greater than 40 (dB). The details of data embedding and extraction procedures are described below:

4.3.1 Data Embedding Process

Consider an original image I of size $(M \times N)$, where M is the height and N is the width of the original image. Interpolate the original image I and generate cover image C of size $(2M \times 2N)$

[6]Published in the proceedings of the First International Conference on Intelligent Computing and Communication ($ICIC^2$-2016), Advances in Intelligent Systems and Computing (AISC), Springer AISC Series, with title *Dual-Image Based Reversible Data Hiding Scheme Using Pixel Value Difference with Exploiting Modification Direction*

using the following equation.

$$\begin{cases} C(i,j) = I(p,q) \\ \quad \{\text{where }, p = 1\ldots M,\ q = 1\ldots N,\ i = 1,3,\ldots,(2M-1),\ j = 1,3,\ldots,(2N-1)\} \\ C(i,j) = (C(i,j-1) + C(i,j+1))/2 \\ \quad \{\text{if } (((i \bmod 2) \neq 0)\&((j \bmod 2) = 0)), \forall\, i = 1\ldots,(2M-1), j = 1\ldots,(2N-1)\} \\ C(i,j) = (C(i-1,j) + C(i+1,j))/2 \\ \quad \{\text{if } \quad (((i \bmod 2) = 0)\&((j \bmod 2) \neq 0)),\ \forall\, i = 1\ldots,(2M-1), j = 1\ldots,(2N-1)\} \\ C(i,j) = (C(i-1,j-1) + C(i-1,j+1) + C(i+1,j-1) + C(i+1,j+1))/4 \\ \quad \{\text{if } \quad (((i \bmod 2) = 0)\&((j \bmod 2) = 0)), \forall\, i = 1\ldots,(2M-1), j = 1\ldots,(2N-1)\} \end{cases} \tag{4.21}$$

The cover image C is generated using image interpolation with size $(2M-1) \times (2N-1)$. To get the size ($2M \times 2N$) image, we copy the pixel value of $(2M-1)$ row to $2M$ row and $(2N-1)$ column to $2N$ column value. This is the boarder line of the cover image which is not visually distorted. Now, consider the secret data D which is divided into blocks of 7 bits that is $D = \sum\{D_i | i = 1,\ 2,\ \ldots, D/7\}$. Each 7 bits are again divided into two parts, where 4 bits are embedded through PVD method and 3 bits are embedded using EMD method. The schematic diagram of data embedding using PVD and EMD is depicted in Fig 4.12. To embed 4 bits secret data, first select pixel pair (C_a, C_b) from the cover image C in raster scan order. Then compute the difference d as follows:

$$d = |C_a - C_b| \tag{4.22}$$

The difference value $|d|$ belongs to the range between 0 to 255. In this scheme, we use the range table R with n contiguous sub-range R_b, $\{R_b | m = 1,\ 2,\ \ldots,\ n\}$ which are used in both embedding and extraction process. The range table R is shown in Fig. 4.11.

Each sub-range R_b has a lower and a upper bound, namely lb and ub respectively and $R_b \in [lb, ub]$. The width wb of each sub-range R_b is obtained using

Lower bound	0	16	32	48	.....	224	128
Upper bound	15	31	47	63		239	255
Number of bits Embedded	4	4	4	4	.....	4	4

Figure 4.11: Range table R of the proposed PVDEMD method

$$wb = ub - lb + 1 \tag{4.23}$$

Here, each sub-range R_b has same width wb that is 16. The number of bits (t) to be embedded are decided by the following equation.

$$t = \lfloor \log_2(wb) \rfloor \tag{4.24}$$

Since, the length of each sub-range of range table is 16, that means 4 bits secret data are to be embedded within each pixel pair. Now, select 4 bits from secret data D_i and convert it into

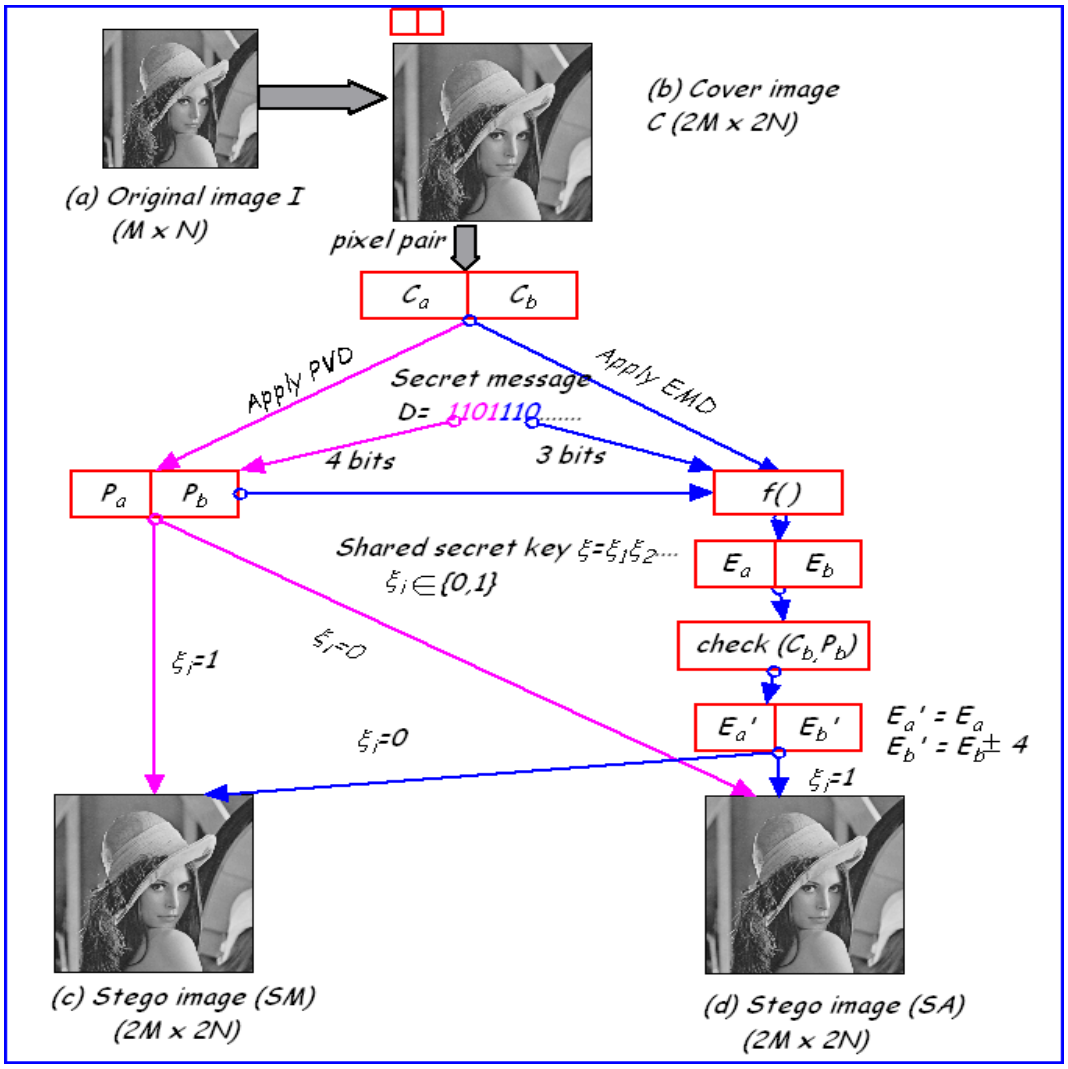


Figure 4.12: Schematic diagram of data embedding process in PVDEMD scheme

decimal value m_1. To embed secret message v, compute new difference $|d^{'}|$ using the following equation.

$$d^{'} = m_1 + lb \tag{4.25}$$

Now, calculate some parameters $d^{''}$, c and f using the following equation.

$$\begin{cases} d^{''} = d^{'} - d \\ c = \lceil d^{''}/2 \rceil \\ f = \lfloor d^{''}/2 \rfloor \end{cases} \tag{4.26}$$

New pixel pair (P_a,P_b) is generated by modifying the pixel pair (C_a,C_b) either by adding or subtracting parameters c and f by which 4 bits secret data are embedded. If $(C_a > C_b)$, then

the modified pixel pair (P_a, P_b) is obtained by

$$\begin{cases} P_a = C_a + f \\ P_b = C_b - c \end{cases} \tag{4.27}$$

If $(C_a < C_b)$, then the modified pixel pair (P_a, P_b) is obtained by

$$\begin{cases} P_a = C_a - c \\ P_b = C_b + f \end{cases} \tag{4.28}$$

After that we apply EMD to embed next 3 bits of secret message within two pixel pairs (C_a,C_b) taken from cover image and generated pixel pair (P_a,P_b). Now, select next 3 bits secret message from D_i and convert into decimal value m_2.

We compute the embedding function $f()$ using two pixel pair (C_a,C_b, P_a,P_b) as follows.

$$f(C_a, C_b, P_a, P_b) = (C_a \times 1 + C_b \times 2 + P_a \times 3 + P_b \times 4) \bmod 9 \tag{4.29}$$

If $f(C_a, C_b, P_a, P_b) = m_2$, then the new pixel pair (E_a, E_b) have been updated by the pixel pair (C_a, C_b) that is $E_a = C_a$ and $E_b = C_b$. If f(C_a,C_b,P_a,P_b) $\neq m_2$ then we calculate $f_1()$ such that $f_1()$ = m_2. The new pixel E_a will contain the pixel value C_a to achieve reversibility. Now, the $f_1()$ function is calculated using the pixels (C_a, C_b, P_a, P_b) as follows.

$$f_1() = [1 \times C_a + 2 \times (C_b - (x \times sign(P_b - C_b))) + P_a \times 3 + P_b \times 4] \bmod 9 \tag{4.30}$$

Now, one have to adjust the value of x in such a way that the value of $f_1()$ will be equal to m_2, where, x is an integer, $x \in \{1, 2, \ldots, 5\}$ and $sign()$ will return 1 or -1 depending on the value of $(P_b - C_b)$. So, the modified stego pixel pair (E_a, E_b) is calculated as follows.

$$E_a = C_a; E_b = (C_b - (x \times sign(P_b - C_b))) \tag{4.31}$$

The following modification on pixel pair (E_a, E_b) has been applied to enhance stego image quality. The modified pixel pair ($E_a^{'}$, $E_b^{'}$) has been calculated using following equation.

$$(E_a^{'}, E_b^{'}) = \begin{cases} (E_a, E_b + 4), & \text{if } P_b > C_b \\ (E_a, E_b - 4), & \text{otherwise} \end{cases} \tag{4.32}$$

Finally, two stego pixel pairs (P_a, P_b) and $(E_a^{'}, E_b^{'})$ are generated. To enhance the security, we did not store the modified pixel into one stego image but rather distributed among dual image

Input: Original image I ($M \times N$), Secret data D, Shared secret key ξ, Range table R.

Output: Two stego images, SM and SA of size $(2M \times 2N)$;

Step 1 : Produce the cover image C of size $(2M \times 2N)$ from the original image I using equation (4.21);

Step 2 : Secret data D, which is divided into $D_i\{D_i|i = 1, 2, \ldots, D/7\}$;

Step 3 : Select pixel pair (C_a, C_b) from C in raster scan order ;

{ ***Data embedding using PVD (4 bits)*** }

Step 4 :

(a) Calculate $d = |C_a - C_b|$;

(b) Select 4 bits data from D_i and convert into decimal value m_1;

(c) Calculate $d^{'} = m_1 + lb$; where, lb is the lower bound of the sub range of range table R where d mapped.;

(d) Calculate $d^{''} = d^{'} - d$;

(e) Calculate $c = \lceil \frac{d^{''}}{2} \rceil$ and $f = \lfloor \frac{d^{''}}{2} \rfloor$;

(f)

if *($C_a > C_b$)* **then**

$P_a = C_a + f$; $P_b = C_b - c$;

else

$P_a = C_a - c$; $P_b = C_b + f$;

end

{ ***Data embedding using EMD (3 bits)*** }.

Step 5 : Select next 3 bits from D_i and convert into decimal value m_2;

Step 6 : Calculate f(C_a,C_b,P_a,P_b) using equation (4.29).

Step 7 :

if *f*(C_a,C_b,P_a,P_b) = v_1 **then**

E_a = C_a ; E_b = C_b;

else

Calculate $f_1()$ using equation (4.30) such that $f_1()$ = v_1 then.

E_a = C_a ; E_b = $(C_b - (x \times sign(P_b - C_b)))$;

end

Step 8 :The modified pixel pair ($E_a^{'}$,$E_b^{'}$) is calculated as follows:

if $P_b > C_b$ **then**

$E_a^{'}$ = E_a; $E_b^{'}$ = E_b+4;

else

$E_a^{'}$ = E_a; $E_b^{'}$ = E_b-4;

end

Step 9 : Distribute modified pixel pair among dual image depending on shared secret key ξ.

if *($\xi = 1$)* **then**

Store (P_a, P_b) within stego image SM and store (E_a, E_b) within stego image SA;

else

Store (P_a, P_b) within stego image SA and store (E_a, E_b) within stego image SM;

end

Step 10 : Repeat **Step** 3 to **Step** 9 until all secret data bits are embedded.

Step 11 : Two stego images SM ($2M \times 2N$) and SA ($2M \times 2N$) are produced.

Step 12 : End.

Algorithm 11: Data embedding process of PVDEMD

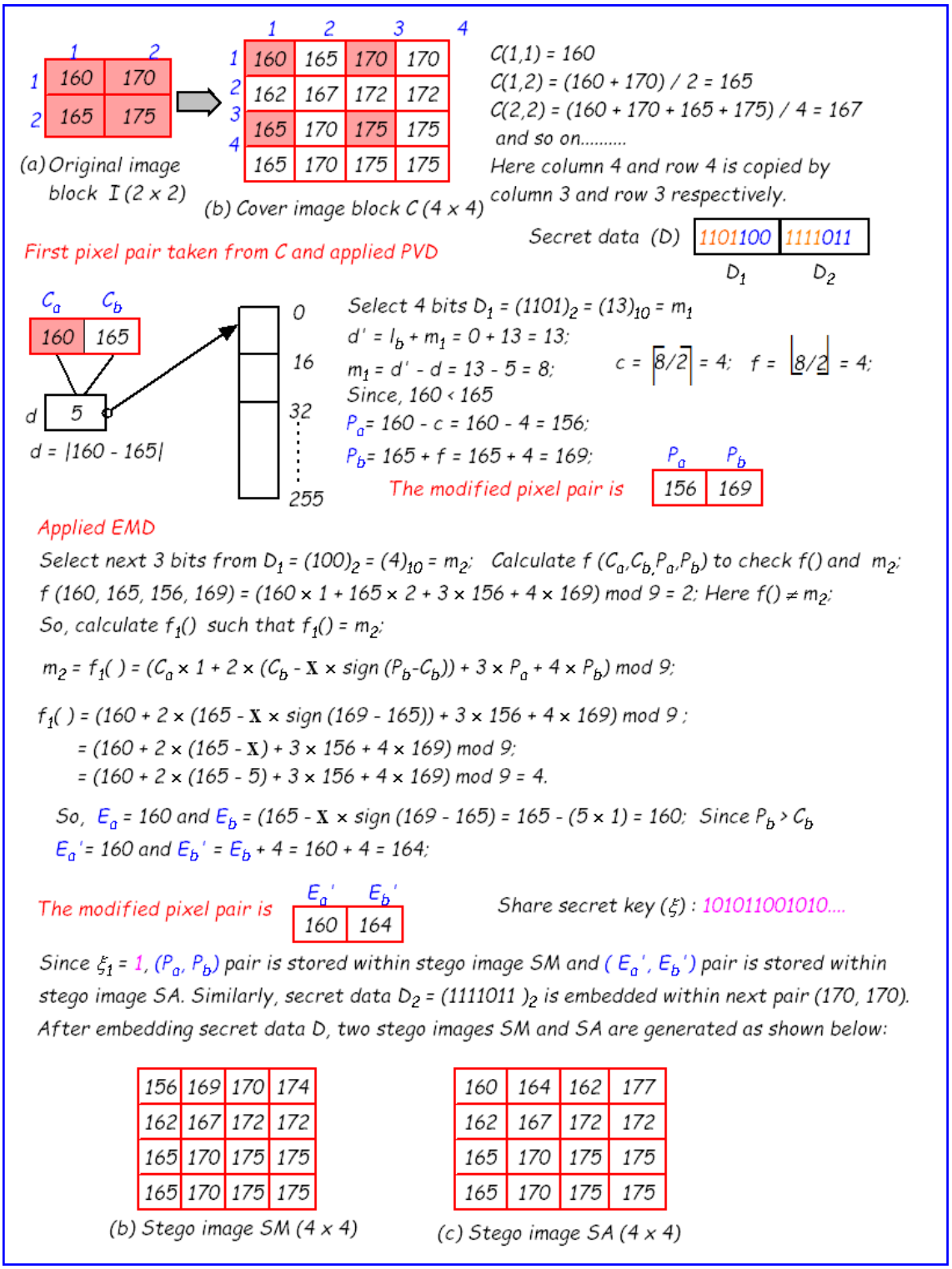


Figure 4.13: Numerical illustration of data embedding process in PVDEMD scheme

depending on a share secret key stream ξ. If (ξ = 1) then the pixel pair (P_a, P_b) has been stored within the stego major (SM) image and the pixel pair $(E_a^{'}, E_b^{'})$ has been stored within the stego auxiliary (SA) image. If (ξ = 0) then the pixel pair (P_a, P_b) has been stored within the SA and the pixel pair $(E_a^{'}, E_b^{'})$ has been stored within the SM.

Example 4.3.1 *Fig.* 4.13 *shows the example using some numerical values. The original image block of size* (2×2) *has been taken and the secret data* D *is* $(11011001111011)_2$ *is considered. The data bits are divided into two data units* $D_1 = (1101100)_2$ *and* $D_2 = (1111011)_2$*. Fig.* $4.13(b)$ *shows the cover image block* C *of size* (4×4) *which is produced using equation* (4.21)*.* D_1 *is embedded within the first pixel pair that is* $(160, 165)$ *and* D_2 *is embedded within the second pixel pair that is* $(170, 170)$*. In this example, Fig.* $4.13(b)$ *and* $4.13(c)$ *shows two stego images SM and SA which are generated after data embedding.* ■

4.3.2 Data Extraction Process

During data extraction, we first rearrange the pixel pair from stego images SM and SA using shared secret key ξ. If ($\xi = 1$) then pixel pair $(E_a^{'}, E_b^{'})$ is collected from the stego image SA otherwise, pixel pair $(E_a^{'}, E_b^{'})$ is collected from the stego image SM. Then we recover the original image I from pixel pair $(E_a^{'}, E_b^{'})$. To recover the pixel value (I_a) of the original image I, first we check the pixel pair $(E_a^{'}, E_b^{'})$ which belongs to the odd row of SM or SA. If the pixel pair is collected from odd row, then the original pixel value (I_a) is obtained by

$$I_a = E_a^{'} \tag{4.33}$$

The schematic diagram of data extraction and original image recovery is depicted in Fig. 4.14, where Fig. $4.14(a)$ and $4.14(b)$ shows two stego image SM ($2M \times 2N$) and SA ($2M \times 2N$) respectively. Fig. $4.14(c)$ is the original image I of size $(M \times N)$. Fig. $4.14(d)$ is cover image C of size $(2M \times 2N)$.

After recovering the original image I, we generate the cover image C using equation (4.21). The last row and last column of the cover image C are copied by the value of $(2M-1)$ row and $(2N-1)$ column of cover image C. To retrieve the secret data from dual image, first we select pixel pair from the stego images SM and SA in raster scan order and apply PVD or modified

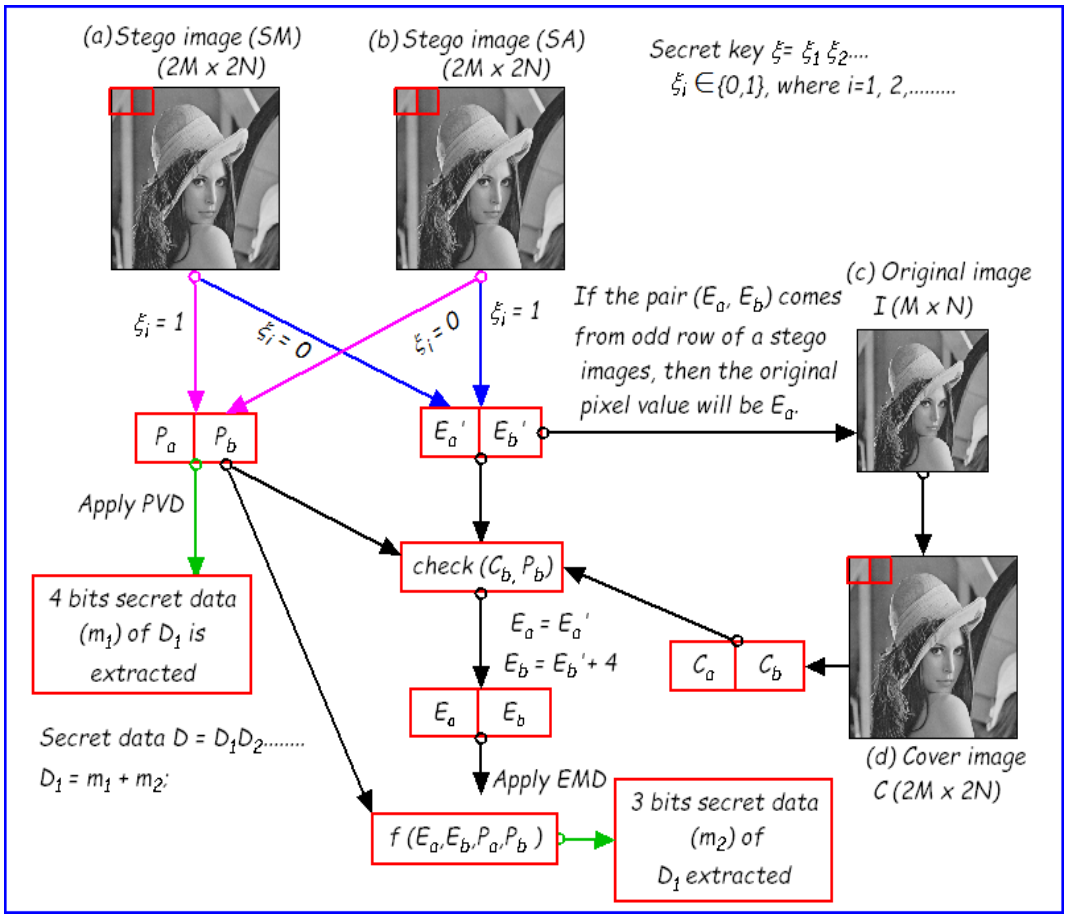


Figure 4.14: Schematic diagram of data extraction process in PVDEMD scheme

EMD method depending on the share secret key ξ. If (ξ = 1) then the pixel pair (P_a, P_b) is retrieved from the stego image SM and the pixel pair $(E_a^{'}, E_b^{'})$ is retrieved from the stego image SA. If (ξ= 0) then the pixel pair (P_a, P_b) is retrieved from the stego image SA and the pixel pair $(E_a^{'}, E_b^{'})$ is retrieved from the stego image SM. Also we select pixel pair (C_a, C_b) from the cover image C in raster scan order. Then we apply data extraction procedure using PVD method on the pixel pair (P_a, P_b) to extract 4 bits secret data from each pixel pair. To do this first compute the difference $d^{'}$ using the following equation.

$$d^{'} = |P_a - P_b| \tag{4.34}$$

The number of bits t, which are to be extracted from the pixel pair are to be decided by the range table R, where the difference $d^{'}$ is mapped.

$$t = \lfloor \log_2(wb) \rfloor, \tag{4.35}$$

where $wb = (ub - lb + 1)$. Here, lb and ub will be the lower bound and upper bound of each

Input: Two stego images SM ($2M \times 2N$) and SA ($2M \times 2N$), Shared secret key ξ, Range table R;

Output: Original image $I(M \times N)$; Secret data D ;

Step 1: For each q, where $q = 1, 2, \ldots, (2M \times 2N/2)$;

if $(\xi = 1)$ **then**

  select q^{th} pixel pair $(E_a^{'}, E_b^{'})$ from odd row of stego image SA; I_a = $E_a^{'}$';

else

  Select q^{th} pixel pair $(E_a^{'}, E_b^{'})$ from odd row of stego image SM; I_a = $E_a^{'}$;

end

Step 2: After executing Step 1, the original image I is recovered. Then generate cover image C using equation (4.21).

Step 3: Select pixel pair from SM and SA in raster scan order;

if $\xi = 1$ **then**

  select pixel pair (P_a, P_b) from stego image SM and select pixel pair $(E_a^{'}, E_b^{'})$ from stego image SA

else

  Select pixel pair (P_a, P_b) from stego image SA and select pixel pair $(E_a^{'}, E_b^{'})$ from stego image SM

end

Step 4: Calculate $d^{'} = |P_a - P_b|$;

Step 5: Secret data $m_1 = d^{'} - lb$, where lb is the lower bound of the sub-range of range table R.

Step 6: Convert m_1 into binary form of 4 bits.

Step 7: Modify pixel pair ($E_a^{'}$,$E_b^{'}$) to (E_a,E_b) as follows:

if $P_b > C_b$ **then**

  E_a = $E_a^{'}$; E_b = $E_b^{'}$ - 4;

else

  E_a = $E_a^{'}$; E_b = $E_b^{'}$ + 4;

end

Step 8: Calculate $f()$ function to retrieve the secret message m_2 using equation (4.38).

Step 9: Convert m_2 into binary form of 3 bits, Concatenate m_1 and m_2 to get secret data $D_1 = m_1 m_2$.

Step 10: Repeat Step 3 to Step 9 until all data are extracted;

Step 11: End.

Algorithm 12: Data extraction process of PVDEMD

sub-range R_b respectively. The secret data m_1 in decimal form will be extracted using

$$m_1 = d^{'} - lb \tag{4.36}$$

Now, convert m_1 into binary form and get t bits secret data of D_1. To get the pixel pair (E_a, E_b) from the modified pixel pair ($E_a^{'}$,$E_b^{'}$), follow the equation below.

$$(E_a, E_b) = \begin{cases} (E_a^{'}, E_b^{'} - 4), & \text{if } P_b > C_b \\ (E_a^{'}, E_b^{'} + 4), & \text{if } P_b \leq C_b \end{cases} \tag{4.37}$$

Then secret data m_2 is retrieved from the pixel values (E_a, E_b, P_a, P_b) by calculating the $f()$ function using the following formula

$$m_2 = f(E_a, E_b, P_a, P_b) = (E_a \times 1 + E_b \times 2 + P_a \times 3 + P_b \times 4) \mod 9 \tag{4.38}$$

Now, convert m_2 into 3 bits binary form to get the last part of D_1.

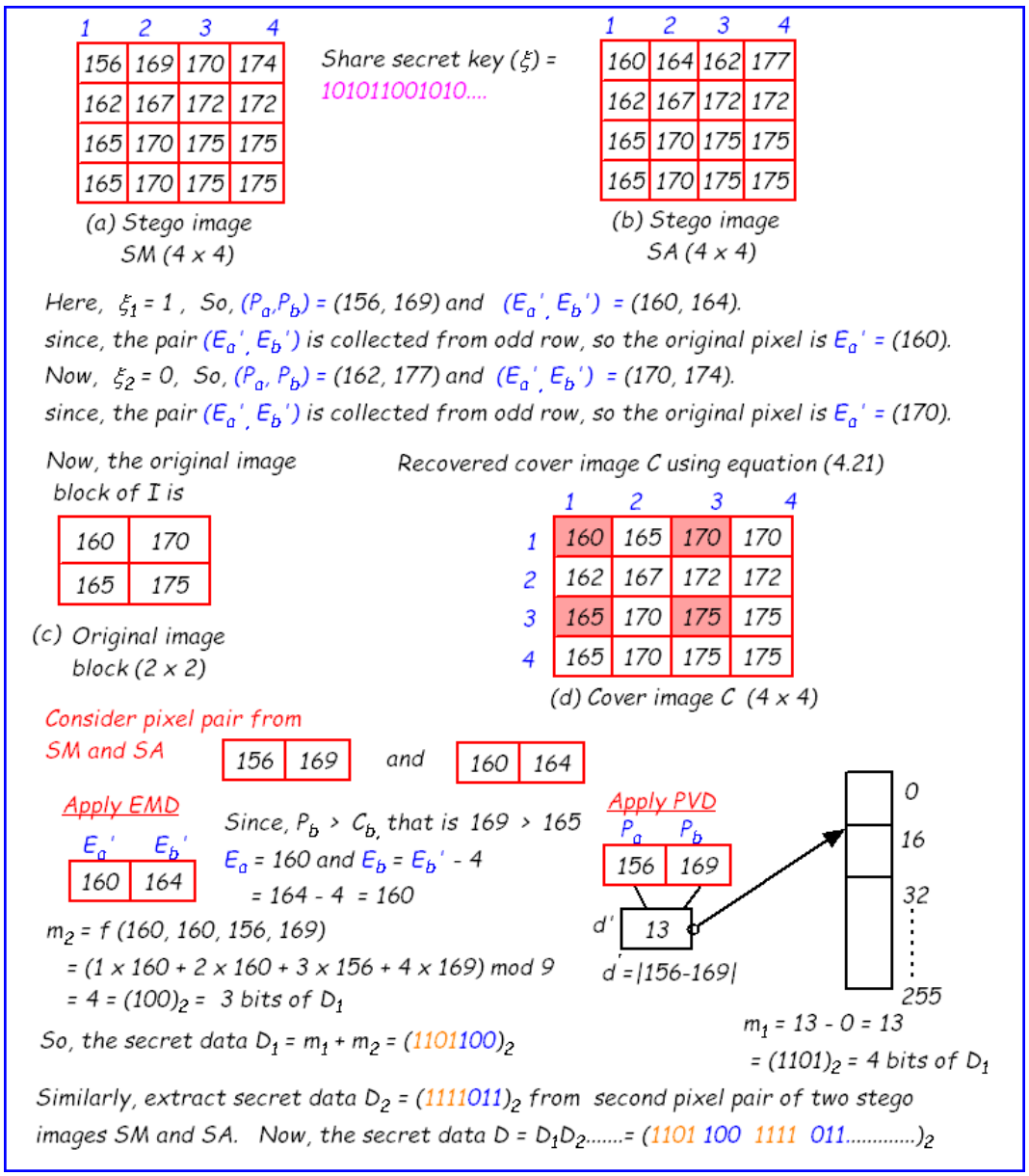


Figure 4.15: Numerical illustration of data extraction in PVDEMD scheme

Example 4.3.2 *Fig.* 4.15 *shows the numerical illustration of data extraction procedure. Fig.* $4.15(a)$ *and Fig.* $4.15(b)$ *shows two stego images SM and SA of size* (4×4) *respectively. Since secret key bit* $\xi = 1$*, so, the pixel pair* (P_a, P_b) *that is* $(156, 169)$ *is selected from stego image SM and the pixel pair* (E_a', E_b') *that is* $(160, 164)$ *is selected from stego image SA. The pixel pair* (E_a', E_b') *belongs to odd row of the stego images which is the original pixel value, that is* $(I_a) =$ 160*. Again we check shared secret key bit* $\xi = 0$*, so, the pixel pair* (P_a, P_b) *that is* $(162, 177)$ *is selected from stego image SA and the pixel pair* (E_a', E_b') *that is* $(170, 174)$ *is selected from*

stego image SM. Since the pixel pair (E_a, E_b) *belongs to odd row which indicates the original pixel value* (I_a) = 170 *from the pixel pair. The Fig.* $4.15(c)$ *shows the original image block* I *of size* (2×2) *which is extracted from stego images and Fig.* $4.15(d)$ *shows the cover image block* C *of size* (4×4) *which is generated using equation (4.21). Now, extract secret data* $(1101)_2$ *from pixel pair* $(156, 169)$ *applying PVD and extract secret data* $(100)_2$ *from pixel pairs* $(160, 160, 156, 169)$ *using* $f()$ *function of EMD. So, the secret data* D_1 = $(1101100)_2$ *has been extracted. Again, the extraction of secret data* $(1111)_2$ *from pixel pair* $(162, 177)$ *and the secret data* $(011)_2$ *from pixel pair* $(170, 170, 162, 177)$ *has been performed which are shown here. So,* D_2 *is* $(1111011)_2$ *is extracted. Combining these two data unit* D_1 *and* D_2*, we can get secret data* D=$(11011001111011)_2$ *which are extracted successfully from both the stego images.* ■

4.3.3 Overflow and Underflow Control

In this approach, overflow and underflow situation may occur during data embedding. When the original pixel value is nearer to 255 and then if we modify that by addition then the pixel value may exceed the maximum gray value (255) then this situation is called overflow situation. When the pixel value is nearer to zero (0) and if we modify that one by subtraction then the pixel value may reach below the gray scale (0) that is negative then that situation is called underflow situation. In this scheme, any pixel modification depends on the embedding function which is modulus of 9, So maximum possible modification will be done by 9. To overcome the overflow and underflow situation, we modify the pixel value of the cover image to less than 246 or greater 9 respectively.

Overflow Control: Suppose a pixel pair with pixel value $C_a = 253$ and $C_b = 252$ and 4 bits secret data that is $(1110)_2$ in decimal $(14)_{10}$ needs to be embedded. The difference between two pixels d = $\mid 253-252 \mid$ = 1. The new difference $d^{'}$ is $14+0 = 14$. Therefore, $d^{''} = 14 - 1 = 13$ and the value of c and f are 7 and 6 respectively. The modified pixel $P_a = C_a + f = 259 > 255$ and $P_b = C_b - c = 245$. That means overflow problem occur at the pixel P_a. It has been possible to solve this problem by taking the help of image interpolation technique. At the time of image interpolation we can set a threshold value in such a way that no pixel value falls into the overflow situation meaning it can not exceed

246 because the maximum width of the sub-range of range table is 16. The maximum value of $f()$ will not be greater than 8. So, addition of $f()$ with the pixel value causes overflow situation. Therefore, if we fix the interpolate pixel value 246 then it overcomes overflow situation. For two consecutive pixels, one is original pixel C_a and another one interpolates pixel C_b. So, it may perform addition operation on interpolating pixel C_b which is not greater than 246. Again, pixel P_a or P_b may fall under overflow or underflow situation after adjusting C_a and C_b by c and f during data embedding. To overcome this situation, always modify C_b by the parameter $d^{''}$ and at the overflow situation C_a will be unchanged. The $d^{''}$ is subtracted from C_b to keep $d^{'}$ unaffected between the pixel pair (P_a, P_b). The pixel pair (P_a, P_b) can be computed using the following formula.

$$\begin{cases} P_a = C_a \\ P_b = C_b - d^{''} \end{cases} \tag{4.39}$$

For example, consider two pixel C_a = 253 and C_b = 252. C_b is the interpolated pixel and fix it by 246. Now the difference $d = |253 - 246| = 7$ and $d^{''} = (14 - 7) = 7$. Parameter $c = 4$ and $f = 3$. According to equation (4.27) the pixel pair $P_a = 253 + 3 = 256$ and $P_b = (246 - 4) = 242$. Here, P_a falls into overflow situation. To solve this overflow problem again, we use the equation (4.39). That means the pixel pair $P_a = 253$ and $P_b = (246 - 7) = 239$. In case of EMD there is no chance to occur overflow situation on pixel pair (E_a, E_b). Because after apply this technique E_a is same as C_a which belongs to [0,255] and C_b is modified to E_b by k, where $k = 1, 2, \ldots, 9$. If k is 9 then $E_b = (246 + 9) = 255$ or $E_b = (246 - 9) = 237$. So, this scheme successfully handles the overflow situation.

Underflow control: Consider an example where underflow problem may occur during data embedding through pixel value difference on the pixel pair (C_a, C_b). Assume that the value of the pixel pair is $(C_a, C_b) = (0, 2)$. 4 bits secret data $(1110)_2$ (in decimal $(14)_{10}$) needs to be embedded. The difference between two pixels d is $|0 - 2| = 2$ and $d^{'}$ is $14 + 0 = 14$. The parameters $d^{''} = (14 - 2) = 12$ and the value of $c = 6$ and $f = 6$ are calculated. Now the updated pixel pair will be $P_a = -6$ and $P_b = 8$. Here underflow problem occurs because $(P_a < 0)$. To overcome this problem, we adjust the interpolated pixel at the time of interpolation. When the interpolated values lie less than 9 then fix the

interpolated pixel C_b by 9. Similarly any pixel P_a or P_b may fall in underflow situation after adjusting C_a and C_b by c and f during data embedding. To overcome this problem we modify C_b by $d^{''}$ (at the underflow situation C_a will be same and m is added with C_b to keep $d^{'}$ unaffected between the pixel pair (P_a, P_b). So, the modified pixel pair (P_a, P_b) is computed using the following formula.

$$\begin{cases} P_a = C_a \\ P_b = C_b + d^{''} \end{cases} \tag{4.40}$$

Therefore the underflow problem of the above situation is solved by setting interpolated pixel C_b = 9. The new pixel pair $C_a = 0$ and $C_b = 9$. Difference d = | $0 - 9$ | = 9 and $d^{''}$ = (14 - 9) = 5. Therefore the pixel pair $P_a = -3$ and $P_b = 11$. It is observed that P_a is in underflow situation. So, it is required to apply equation (4.40) to calculate new pixel P_a = 0 and P_b = 14. In case of EMD, no underflow problem may occur on the pixel pair E_a, E_b for the reason that C_b is set to 9. Since C_b is modified to E_b by k, where $k = 1, 2, \ldots, 9$. If k is 9 then E_b = 9 + 9 = 18 or E_b = 9 - 9 = 0. No underflow situation may occur in this proposed scheme.

4.3.4 Experimental Results and Comparisons

The proposed scheme is verified and tested using gray scale image of size (256×256) pixels which is shown in Fig. 4.16. After embedding the secret messages, dual stego image, stego major (SM) and stego auxiliary (SA) of size (512×512) have been generated which are shown in Fig. 4.17. The developed algorithm Algorithm 11 for data embedding and Algorithm 12 for data extraction are implemented in MATLAB Version 7.6.0.324 (R2008a).

The analysis in terms of PSNR is shown in Table 4.8. The PSNR of the stego images in proposed scheme varies 44 (dB) to 37 (dB) when data embedding capacity varies from $1,60,000$ bits to $9,16,656$ bits. To calculate payload in terms of bits per pixel (bpp), the following expression is used

$$p = \frac{2M \times N \times r}{2M \times 2N \times s}, \tag{4.41}$$

where $M = 256$ and $N = 256$, r is the number of bits which are embedded within each pixel pair and s is the number of stego images. According to equation (4.41) the payload $p = 1.75$

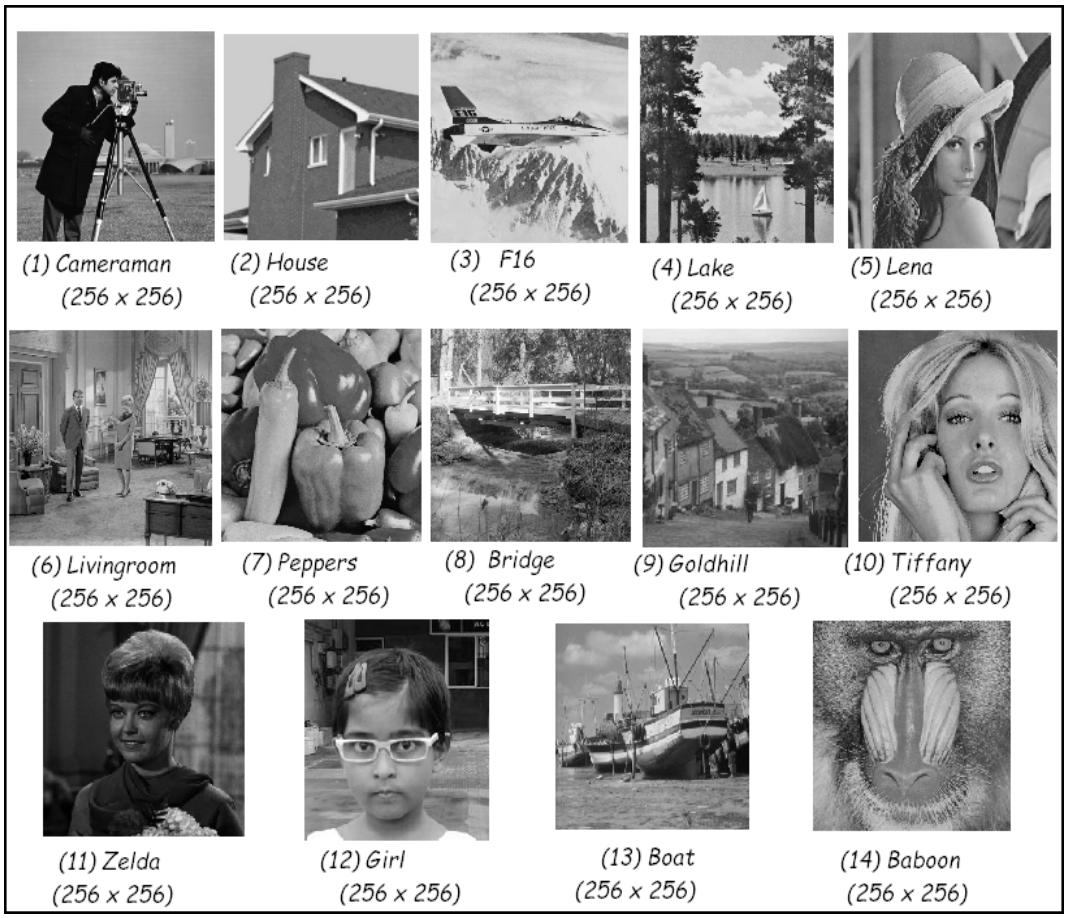


Figure 4.16: Standard input images are used in PVDEMD scheme

(bpp). It is observed that the average PSNR is more than 40 (dB) which assures the quality of the stego images.

Comparisons with other dual image based existing data hiding schemes are shown in Table 4.9. From this table it is observed that the average PSNR of the stego images of our proposed method is higher than 40 (dB) which is lower than the scheme proposed by Qin et al. [52], Lu et al. [45], Chang et al. [5] and Lee et al. [36] [34]. But the PSNR of proposed scheme is 0.54 (dB) higher than Chang et al.'s [10] scheme.

The payload of the PVDEMD method is 1.75 (bpp) which is higher than other existing schemes. The embedding capacity of Chang et al.'s [10] scheme is 1.53 (bpp) which is 0.22 (bpp) less than PVDEMD scheme. It is observed that the embedding capacity of PDVEMD is higher than the other existing dual image based RDH methods. Table 4.10 presents the comparison of the proposed scheme with other single image based data hiding schemes. The PSNR of

Figure 4.17: Generated dual stego images SM and SA of PVDEMD scheme

Table 4.8: PSNR (dB) with data embedding capacity of PVDEMD scheme

Image	Data (bits)	PSNR (SM& C)	PSNR(SA & C)	Avg. PSNR(dB)
Cameraman	1,60,000	44.8730	44.6110	40.4289
	4,00,000	40.8789	40.6836	
	6,00,000	39.1323	38.9067	
	9,16,656	37.2853	37.0608	
House	1,60,000	44.8721	44.6118	40.4281
	4,00,000	40.8752	40.6821	
	6,00,000	39.1334	38.9054	
	9,16,656	37.2841	37.0611	
F16	1,60,000	44.8733	44.6128	40.4281
	4,00,000	40.8746	40.6773	
	6,00,000	39.1364	38.9071	
	9,16,656	37.2818	37.0621	
Lake	1,60,000	44.8733	44.6128	40.4274
	4,00,000	40.8740	40.6795	
	6,00,000	39.1321	38.9098	
	9,16,656	37.2795	37.0584	
Lena	1,60,000	44.8695	44.6172	40.4285
	4,00,000	40.8788	40.6769	
	6,00,000	39.1403	38.9077	
	9,16,656	37.2816	37.0564	
Peppers	1,60,000	44.8709	44.6074	40.4284
	4,00,000	40.8831	40.6774	
	6,00,000	39.1292	38.9118	
	9,16,656	37.2869	37.0612	
Boat	1,60,000	44.8787	44.6066	40.4278
	4,00,000	40.8772	40.6792	
	6,00,000	39.1322	38.9077	
	9,16,656	37.2822	37.0589	
Gold hill	1,60,000	44.8681	44.6141	40.4267
	4,00,000	40.8729	40.6795	
	6,00,000	39.1302	38.9050	
	9,16,656	37.2851	37.0592	
Zelda	1,60,000	44.8814	44.6053	40.4330
	4,00,000	40.8817	40.6877	
	6,00,000	39.1332	38.9097	
	9,16,656	37.2951	37.0705	
Babbon	1,60,000	44.8719	44.6109	40.4330
	4,00,000	40.8774	40.6146	
	6,00,000	39.1379	38.9053	
	9,16,656	37.2855	37.0634	

proposed method is lower than the method proposed by Shen and Huang [54] but higher than the method proposed by Lee et al. [33] and Zeng et al. [75]. This method is superior in terms

Table 4.9: Comparison of PVDEMD with existing dual image based RDH scheme

Methods	Measure	Images					
		Lena	Peppers	Boat	Goldhill	Zelda	Baboon
Chang et al. [5]	PSNR(1)	45.12	45.14	45.12	45.13	45.13	45.11
	PSNR(2)	45.13	45.15	45.13	45.14	45.11	45.13
	PSNR(Avg.)	45.13	45.15	45.13	45.14	45.12	45.12
	Capacity(bpp)	1	0.99	1	1	0.99	0.99
Chang et al. [6]	PSNR(1)	48.13	48.11	48.13	48.13	48.15	48.13
	PSNR(2)	48.14	48.14	48.12	48.15	48.13	48.14
	PSNR(Avg.)	48.14	48.13	48.13	48.14	48.14	48.14
	Capacity(bpp)	1	1	1	1	1	1
Lee et al. [36]	PSNR(1)	51.14	51.14	51.14	51.14	51.14	51.14
	PSNR(2)	54.16	54.17	54.16	54.16	54.17	54.14
	PSNR(Avg.)	52.65	52.66	52.65	52.65	52.66	52.64
	Capacity(bpp)	0.75	0.75	0.75	0.75	0.75	0.75
Lee and Huang [34]	PSNR(1)	49.76	49.75	49.76	49.77	49.77	49.77
	PSNR(2)	49.56	49.56	49.57	49.57	49.58	49.56
	PSNR(Avg.)	49.66	49.66	49.67	49.67	49.68	49.77
	Capacity(bpp)	1.07	1.07	1.07	1.07	1.07	1.07
Chang et al. [10]	PSNR(1)	39.89	39.94	39.89	39.9	39.89	39.91
	PSNR(2)	39.89	39.94	39.89	39.9	39.89	39.91
	PSNR(Avg.)	39.89	39.94	39.89	39.9	39.89	39.91
	Capacity(bpp)	1.53	1.52	1.53	1.53	1.53	1.53
Qin et al. [52]	PSNR(1)	52.11	51.25	51.11	52.11	52.06	52.04
	PSNR(2)	41.34	41.52	41.57	41.34	41.57	41.56
	PSNR(Avg.)	46.72	46.39	46.84	46.72	46.82	46.80
	Capacity(bpp)	1.16	1.16	1.16	1.16	1.16	1.16
Lu et al. [45]	PSNR(1)	49.20	49.19	49.20	49.23	49.19	49.21
	PSNR(2)	49.21	49.21	49.21	49.18	49.21	49.20
	PSNR(Avg.)	49.21	49.20	49.21	49.21	49.20	49.21
	Capacity(bpp)	1	0.99	1	1	0.99	0.99
Lu et al.(k=2) [45]	PSNR(1)	49.89	49.89	49.89	49.90	49.89	49.89
	PSNR(2)	52.90	52.92	52.90	52.90	52.88	52.87
	PSNR(Avg.)	51.40	51.41	51.40	51.40	51.39	51.38
	Capacity(bpp)	1	0.99	1	1	0.99	0.99
PVDEMD	PSNR(1)	40.54	40.54	40.54	40.54	40.54	40.54
	PSNR(2)	40.31	40.31	40.31	40.31	40.31	40.31
	PSNR(Avg.)	40.43	40.43	40.43	40.43	40.43	40.43
	Capacity(bpp)	1.75	1.75	1.75	1.75	1.75	1.75

of embedding capacity than the other existing methods. In terms of security, it is also better because the secret key is used to distribute the embedded pixel among dual stego image. Again without simultaneous dual images and secret key it is hard to retrieve the secret message.

Table 4.10: Comparison of PVDEMD with existing single image based RDH scheme

Methods	Measure	Images			
		Lena	Boat	Goldhill	Babbon
Lee et al. [33]	PSNR	34.38	33.12	32.08	30.03
	Capacity(bpp)	0.91	0.86	0.84	0.62
Zeng et al. [75]	PSNR	32.74	32.96	31.82	30.97
	Capacity(bpp)	1.04	1.04	0.80	0.51
Shen and Huang. [54]	PSNR	42.46	41.60	41.80	38.88
	Capacity(bpp)	1.53	1.55	1.54	1.69
PVDEMD	PSNR	40.43	40.43	40.43	40.43
	Capacity(bpp)	1.75	1.75	1.75	1.75

4.3.5 Steganalysis and Steganographic Attacks

The goal of steganalysis is to gather enough evidence about the presence of embedded message and to break the security of its carrier. The importance of steganalytic techniques that can reliably detect the presence of hidden information in images. Steganalysis finds its use in computer forensics, cyber warfare, tracking the criminal actions over the internet and collecting evidence for investigations especially in case of anti-social components. Apart from this law enforcement and anti-social implication steganalysis also has a peaceful application improving the security of steganographic tools by judging and recognizing their weaknesses.

Table 4.11: Results of RS analysis for stego image SM in PVDEMD scheme

Image	Data (bits)	SM				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Cameraman	160000	7118	7107	3551	3594	0.0051
	400000	6768	6851	3944	3895	0.0123
	600000	6304	5947	4943	5279	0.0616
	916656	6207	6035	4997	5173	0.0311
Lena	160000	5617	5607	4067	4068	0.0011
	400000	5563	5476	4291	4337	0.0135
	600000	5636	5539	4517	4589	0.0166
	916656	5641	5387	4509	4709	0.0447
Baboon	160000	5893	5815	4960	5105	0.0205
	400000	5897	5875	5076	5131	0.0070
	600000	6018	5813	5107	5313	0.0369
	916656	5844	5986	5256	5123	0.0248

4.3.5.1 RS Analysis

We have analyzed the stego images by RS analysis [16]. When the value of RS analysis is closer to zero means the scheme is secure. It is observed from Table 4.11 and 4.12 that the values of R_M and R_{-M}, S_M and S_{-M} are nearly equal for stego image SM and SA. In this experiment, the ratio of R and S lies between 0.0051 to 0.0616 for SM and 0.0043 to 0.0267 for SA for Cameraman image. Thus rule $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ is satisfied for the stego image in our proposed scheme. So, the proposed method is secure against RS attack. Other experimental values are shown in the Table 4.11 and 4.12.

Table 4.12: Results of RS analysis for stego image SA in PVDEMD scheme

Image	Data (bits)	SA				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Cameraman	160000	6945	7078	3877	3721	0.0267
	400000	6506	6535	4490	4472	0.0043
	600000	6514	6528	4287	4224	0.0071
	916656	6538	6647	4283	4225	0.0154
Lena	160000	5575	5565	4139	4133	0.0016
	400000	5590	5514	4239	4299	0.0138
	600000	5587	5442	4579	4665	0.0227
	916656	5652	5621	4592	4553	0.0123
Baboon	160000	5876	5881	4995	5092	0.0094
	400000	5821	5878	5121	5147	0.0076
	600000	5895	5827	5196	5283	0.0140
	916656	5874	5830	5194	5206	0.0051

4.3.5.2 Relative Entropy

The relative entropy between the original and stego major SM is shown in Table 4.13. The values of entropy difference is nearly zero, which implies the proposed scheme provides secure hidden data communication. Other relative entropy values are depicted in Table 4.13.

4.3.5.3 Statistical Analysis

The Standard Deviation SD (σ) of before and after data embedding and Correlation Coefficient CC (ρ) of cover and stego images are summarized in Table 4.14. From this table, it is seen that there is no significant difference between the σ of the cover image and the stego images. This study shows that the ratio of change in stego images based on image parameters is small from

Table 4.13: Relative entropy between I and SM in PVDEMD scheme

Image	Data (bits)	Entropy I	Entropy SM	Entropy Difference
Lena	160000	7.4451	7.4451	0.0027
	400000	7.4451	7.4452	0.0058
	600000	7.4451	7.4452	0.0105
	916656	7.4451	7.4453	0.0131
Barbara	160000	7.0480	7.0480	0.0031
	400000	7.0480	7.0482	0.0064
	600000	7.0480	7.0485	0.0112
	916656	7.0480	7.0486	0.0134
Tiffany	160000	7.2925	7.2925	0.0029
	400000	7.2925	7.2925	0.0057
	600000	7.2925	7.2926	0.0122
	916656	7.2925	7.2926	0.0129
Pepper	160000	7.2767	7.2767	0.0039
	400000	7.2767	7.2768	0.0077
	600000	7.2767	7.2770	0.0142
	916656	7.2767	7.2771	0.0169
Gold hill	160000	7.2367	7.2367	0.0034
	400000	7.2367	7.2371	0.0056
	600000	7.2367	7.2375	0.0112
	916656	7.2367	7.2379	0.0143

a cover image. Since the image parameters have not changed much, the method offers a good concealment of data and reduces the chance of the secret data being detected. Thus, it indicates a secure data hiding scheme.

Table 4.14: Experimental results of SD (σ) and CC (ρ) in PVDEMD scheme

Image	SD (σ)			CC (ρ)		
	I	SM	SA	I&SM	I&SA	SM & SA
Baboon	38.3719	37.8500	38.5478	0.9820	0.9965	0.9745
Cameraman	61.5978	61.1221	61.7313	0.9931	0.9982	0.9913
Lena	47.8385	47.4358	47.9772	0.9894	0.9976	0.9864

4.3.5.4 Histogram Attack

Fig. 4.18 depicted the histogram of the cover and stego images and their difference histogram are shown in Fig. 4.19. The stego images are produced from cover image employing the maximum data hiding capacity. It is observed that the shape of the histogram is preserved after embedding the secret data. The difference of the histogram bins are very small. It is observed

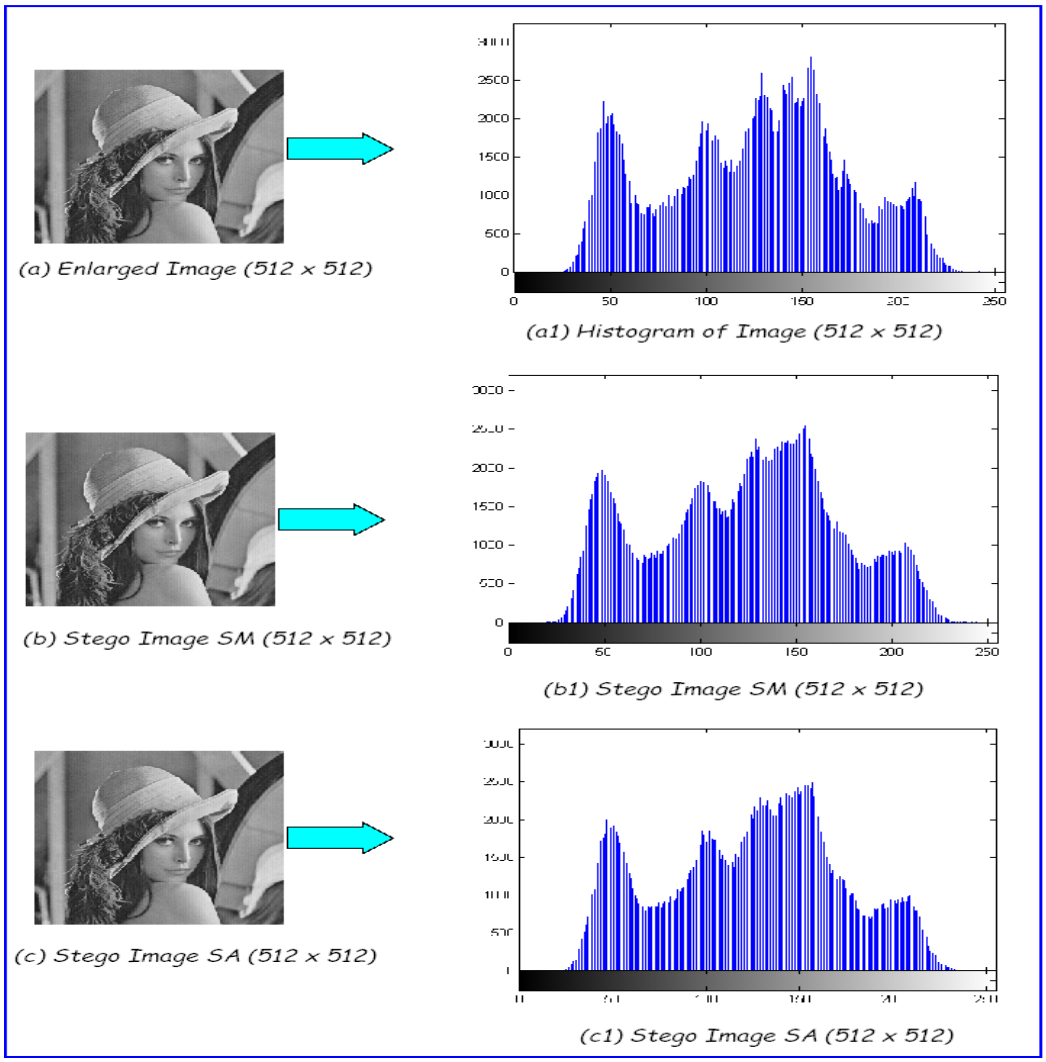


Figure 4.18: Histogram of original image, SM and SA of Lena image in PVDEMD scheme

that, bins close to zero are more in number and the bins which are away from zero are less in number. This confirms the quality of stego images. There is no step pattern observed which ensures the proposed method is robust against histogram analysis.

4.3.5.5 Brute Force Attack

The brute force attack with unknown secret key is tested in this section. The case study in three different ways is taken and it tries to improve the protection the secret data when it has been extracted by an eavesdropper with unknown key. Three cases are described below:

Case Study - 1: An image is considered as secret data and embedded within the cover image using a shared secret key ξ. Now, data extraction has been performed using wrong secret

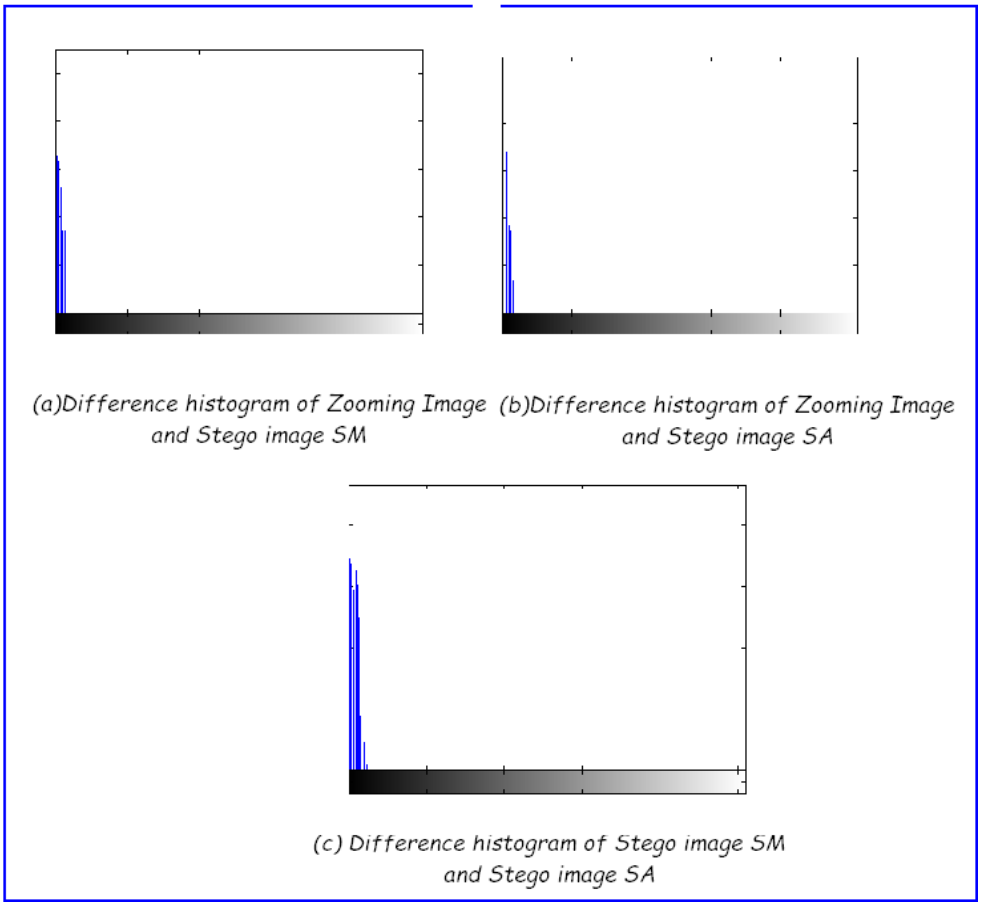


Figure 4.19: Difference Histogram of Lena image in PVDEMD scheme

key and it is found that a noisy image is retrieved. Fig 4.20 shows the experimental result with wrong secret key. From Table 4.15, it is observed that the σ of secret image is 59.7078 and extracted noisy image is 65.1471 which deviates 5.441 units. The ρ between these two images is 0.5789. It is observed that the noisy image and the original secret image are visually identified through open eyes. To improve the security of secret image data we performed XOR operation with the secret message bit and shared secret key bit.

Case Study - 2: To improve the visual deformation which improve security in this scheme. The XOR operation has been performed between the secret message bits with the share secret key bits before data embedding operation. Then embed the results within the stego images. Now extraction has been performed using wrong secret key. It is observed that this approach gives good results than previous study. Fig 4.21 shows the experimental result of case study - 2 using this approach. From Table 4.15, it is observed that the σ of secret image and noisy image is 59.7078 and 70.3203 respectively and the difference is

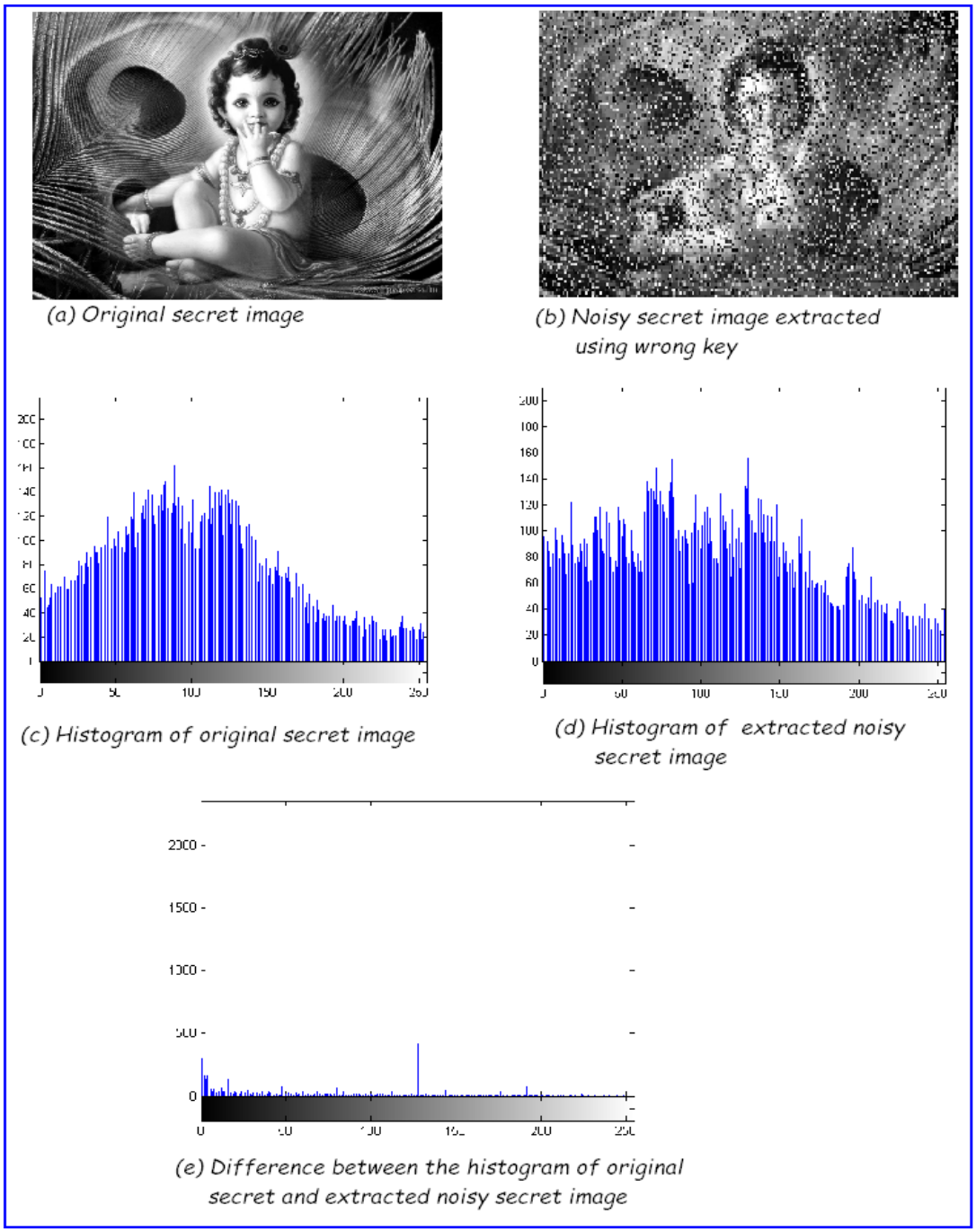


Figure 4.20: Result of Case Study - 1 in PVDEMD scheme

10.6125. Also the ρ between these two images is 0.5002. Although case study - 2 gives better results than case study-1 it shows some similarity with the original secret image. To improve the results from the previous study we shuffle stego pixel among dual images which is described in case study - 3.

Case Study - 3: To enhance the security through visual distortion, first we perform XOR operation with bits of secret message and bits of secret key then distribute the pixel among

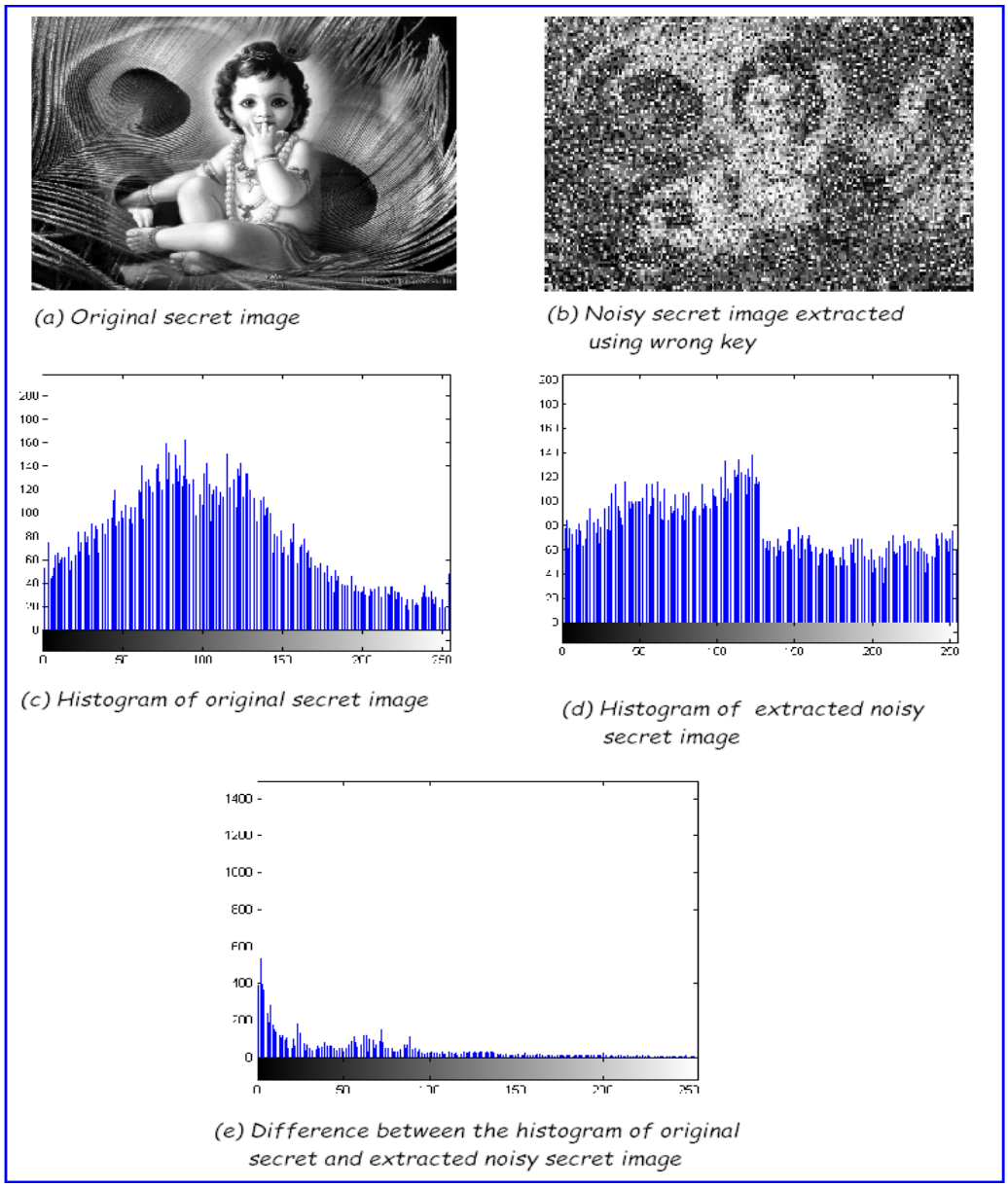


Figure 4.21: Result of Case Study - 2 in PVDEMD scheme

dual image depending on the shared secret key. We shuffle each pixel among dual image depending on shared secret key. For this method, if ξ_i is 1 then the pixel P_a is stored within the stego image SM and the pixel $E_a^{'}$ is stored within the stego image SA. If ξ_i is 0 then the pixel P_b is stored within the stego image SM and the pixel $E_b^{'}$ is stored within the stego image SA.

Fig 4.22 shows the result of case study-3 in this approach. From Table 4.15, the σ of secret image is 59.7078 and σ of noisy image is 71.3753. So, the difference is 11.6675 unit which is not similar. The ρ between these two images is 0.3080, which indicates high dissimilarity

Table 4.15: SD (σ) and CC (ρ) for Case Study 1, 2 and 3 of PVDEMD scheme

SD (σ)				CC (ρ)		
Secret Image (SI)	Case - 1	Case - 2	Case - 3	SI& Case - 1	SI & Case - 2	SI & Case - 3
59.7078	65.1471	70.3203	71.3753	0.5789	0.5002	0.3080

between them. This value is tense to zero and noisy image is not visually identified. In this approach, it is much robust against brute force attack where more than 70% image information is noisy that means corrupted and only 30% image information is retrieved or matched.

Here, image has been used as secret data, so visual identification is possible and noisy image is extracted. But when the text documents are used as secret messages then it is hard to extract original secret message because change of any bits can not be recovered during extraction using unknown secret key. For image as secret message it is visually observed through our open eyes but in case of text documents the actual information has been changed due to little changes of bits during extraction with unknown key and not identified by our open eyes.

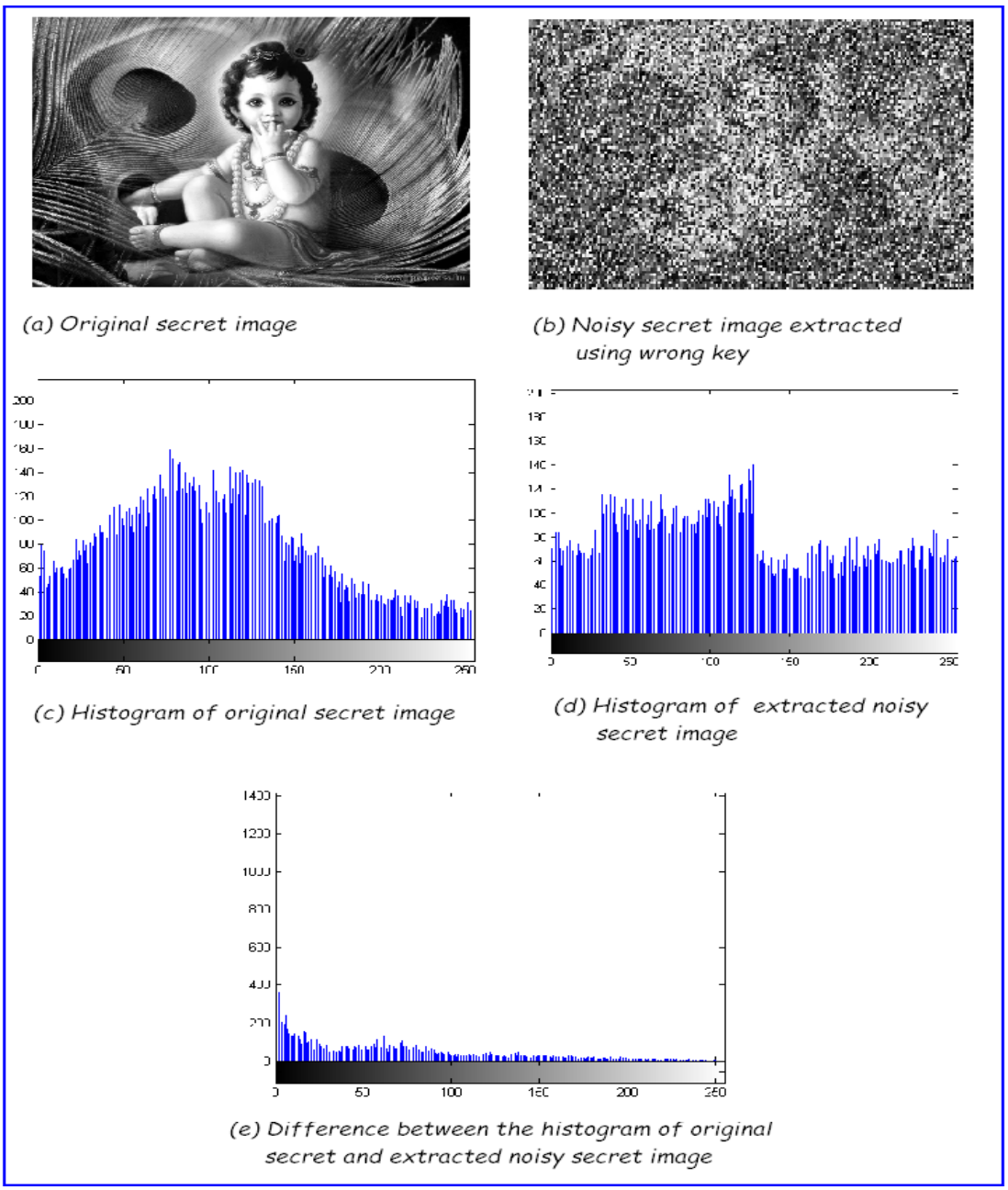


Figure 4.22: Result of Case Study - 3 in PVDEMD scheme

4.4 Dual Image based RDH using Three PVD (TPVD) with DE (TPVDDE)[7]

The payload of the previous scheme was 1.75 (bpp) and PSNR was nearer 40 dB. To improve the payload, a dual image based RDH scheme using Three Pixel Value Differences (TPVD) with Difference Expansion (DE) has been proposed called TPVDDE. Three consecutive pixels from the original image have been taken and 13 bits secret data are embedded by modifying these pixels in pairs using TPVD with DE method. Modified three pixel values are generated after embedding secret bits through TPVD and another three modified pixel values are generated after embedding secret data using DE. These two sets of three pixel values are stored on dual stego images. At the receiver end, two sets of three pixel values are collected from dual stego images then the secret data is extracted successfully using TPVD an DE method. The TPVD was not reversible data hiding scheme, but using dual image we achieve reversibility in TPVD based data hiding schemes. The proposed scheme has been compared with other existing state-of-the-art methods where it exhibits reasonably better performance in terms of data embedding capacity.

4.4.1 Data Embedding Process

A new dual image based RDH scheme has been proposed by combining TPVD with DE. According to this method, first we select three consecutive pixels P_1, P_b and P_2 from cover image I. Consider the secret data $D = D_i|i = 1, 2, 3, ...D/13$. Each D_i contain 13 bits secret data. Now, embed first 3 bits secret data by directly replacing last three bits from P_b pixel and a new $P_b^{'}$ pixel has been produced. Then we calculate two pixel value difference d_a and d_b between the pixel pair (P_1, $P_b^{'}$) and ($P_b^{'}$, P_2) using

$$d_a = |P_1 - P_b^{'}|;\ d_b = |P_b^{'} - P_2| \tag{4.42}$$

The range table R has been divided into equal width sub-range $[lb, ub]$ having length wb that is $wb = ub - lb + 1$. In this scheme, the number of bits to be embedded is determined using

[7]Published in the proceedings of the Third International Conference on Information System Design and Intelligent Application (INDIA 2016), Information Systems Design and Intelligent Applications, Springer, Vol. 434, pp. 403-412, with title *Dual Image Based Reversible Data Hiding Scheme Using Three Pixel Value Difference*

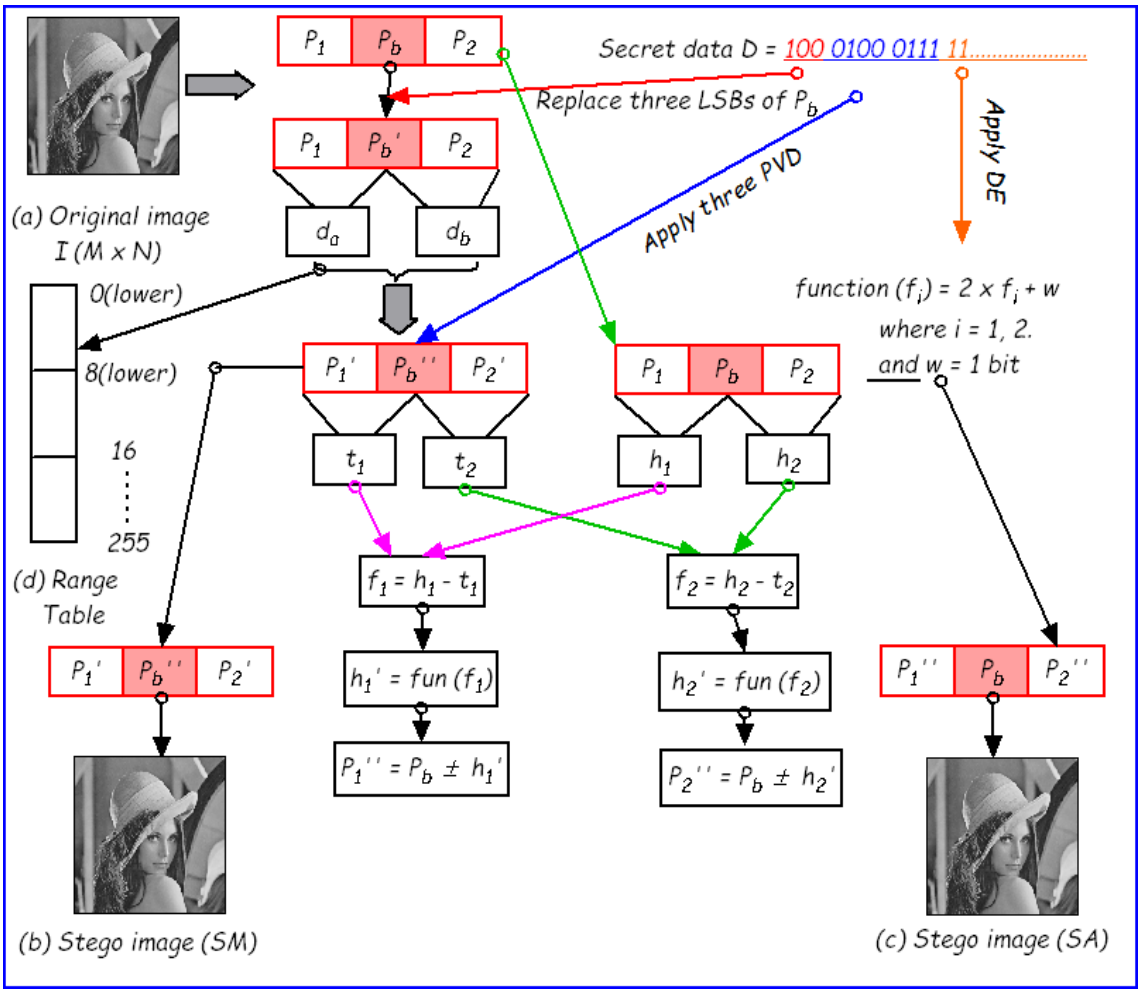


Figure 4.23: Block diagram of data embedding process in TPDVDE scheme

the width of sub-range of range table R. Here, wb has been taken as 8. Hence, the contiguous sub-ranges are {0-8, 8-16, 16-24, . . . , 248-255} which has the capability to embed two 4 bits secret data within each pixel pair. After embedding two 4 bits secret data between two pixel pair, three stego pixels $P_1^{'}$, $P_b^{'}$ and $P_2^{'}$ are produced. Then DE is applied on the four differences which has been calculated using

$$t_1 = |P_1^{'} - P_b^{'}|;\ t_2 = |P_b^{'} - P_2^{'}|;\ h_1 = |P_1 - P_b|;\ h_2 = |P_b - P_2| \tag{4.43}$$

and then we calculate

$$f_1 = |h_1 - t_1|;\ f_2 = |h_2 - t_2| \tag{4.44}$$

Now one bit secret data is embedded through each operation

$$P_1^{''} = |P_b(+/-)h_1^{'}|;\ P_2^{''} = |P_b(+/-)h_2^{'}| \tag{4.45}$$

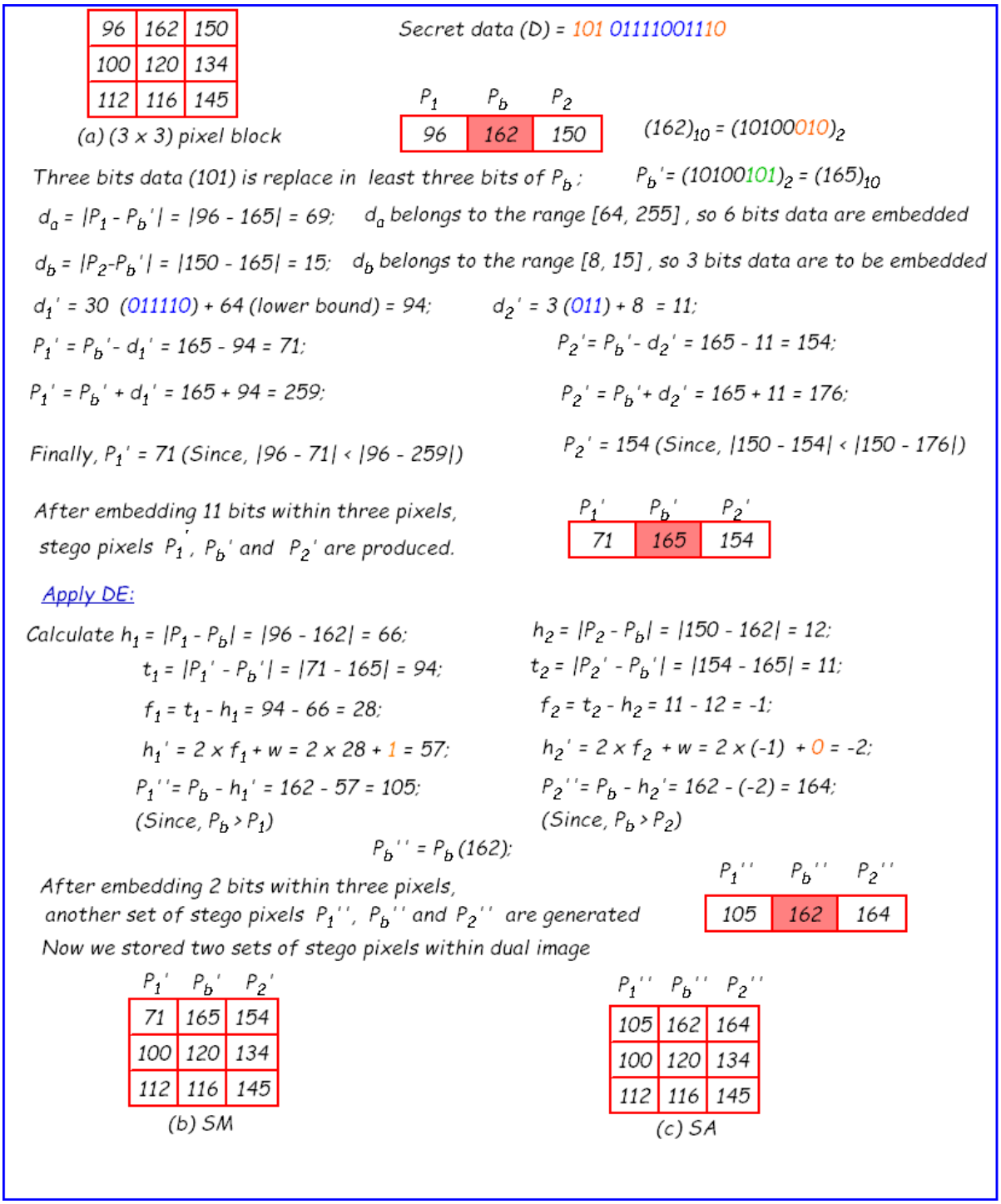


Figure 4.24: Numerical example of data embedding process in TPVDDE scheme

where, $h_1' = f(f_1)$, $h_2' = f(f_2)$, $f(f_i) = 2 \times f(i) + w$ and $w = 1$ bit secret data and $i = 1, 2$. Finally, three modified pixels, P_1'', $P_b'' = P_b$, P_2'' are stored on stego image SA and P_1', P_b' and P_2' are stored on SM. The detailed schematic diagram of proposed method for data embedding process and an illustration with some numerical values are shown in Fig. 4.23 and Fig. 4.24 respectively.

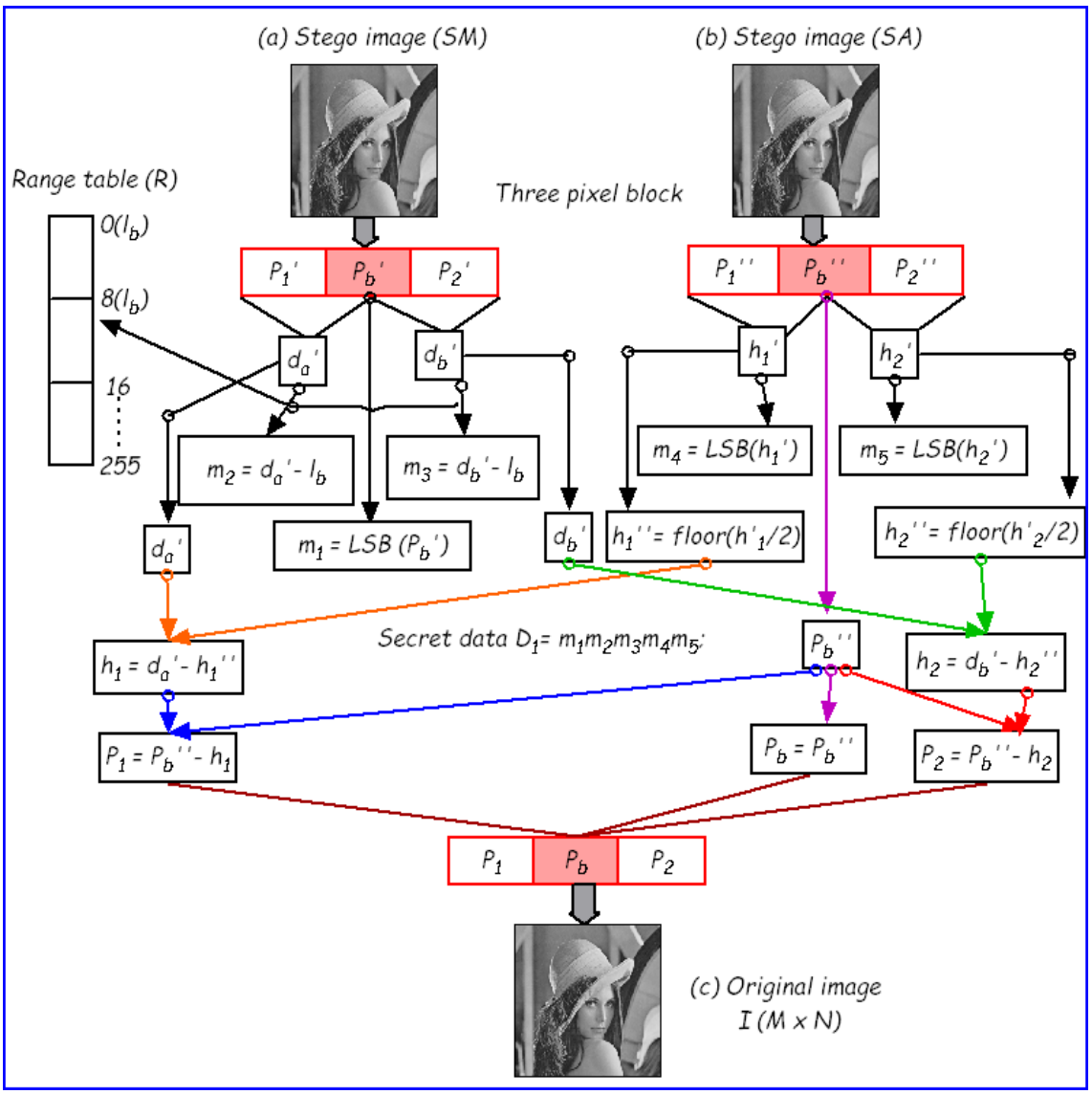


Figure 4.25: Block diagram of data extraction process in TPVDDE scheme

4.4.2 Data Extraction Process

At the receiver end, both the data extraction and original image reconstruction has been performed by taking three consecutive pixels from both the stego images SM and SA. The extraction process and numerical example is shown in the Fig. 4.25 and Fig. 4.26 respectively. First we calculate the differences between pixels collection from SM.

$$d_a^{'} = |P_1^{'} - P_b^{'}|; d_b^{'} = |P_b^{'} - P_2^{'}| \tag{4.46}$$

Then first 3 bits secret data (m_1) has been extracted from three LSB of the $P_b^{'}$. Then we collect the each 4 bits secret data by subtracting lower bound of the sub-range of specific range table

R.

$$m_2 = |d_a^{'} - l_b|; m_3 = |d_b^{'} - l_b| \tag{4.47}$$

Now, collect three consecutive pixels from SA and perform the following operations.

$$h_1^{'} = |p_1^{''} - p_b^{''}|; h_2^{'} = |p_b^{''} - p_2^{''}| \tag{4.48}$$

The secret message m_4 = LSB of $(h_1^{'})$ and m_5 = LSB of $(h_2^{'})$. Hence, the secret data $D_1 = m_1m_2m_3m_4m_5$ has been extracted successfully. The original image has been recovered as shown in Fig. 4.25.

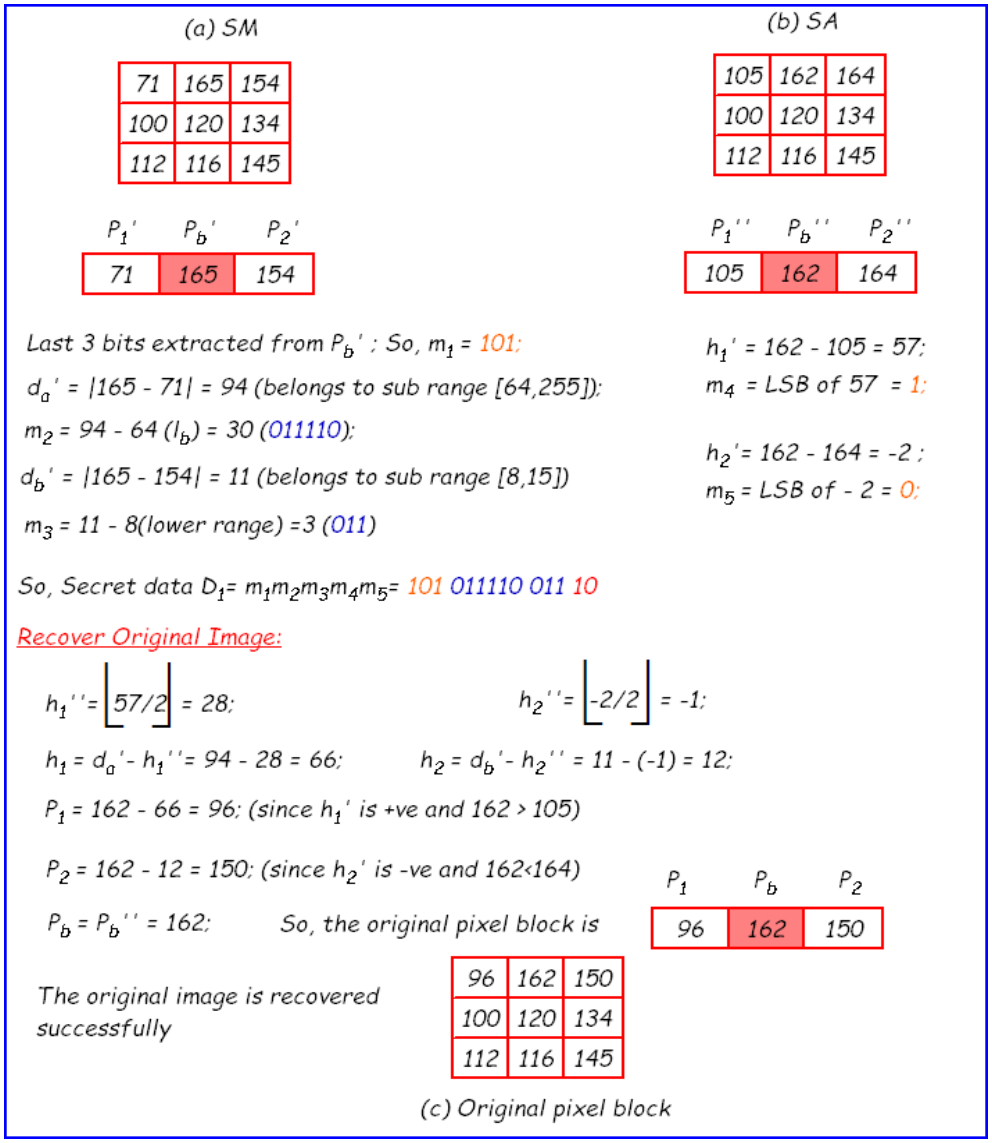


Figure 4.26: Numerical illustration of data extraction in TPVDDE scheme

4.4.3 Experimental Results and Comparisons

In this section, the proposed method has been verified and tested using gray scale image of size (256×256) which is shown in Fig. 4.27. The original image and the image after embedding the secret messages, dual stego images, SM and SA have been generated which are shown in Fig 4.28. Our developed embedding and extraction Algorithms are implemented in MATLAB Version 7.6.0.324 (R2008a). Distortion is measured by means of two parameters namely, Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR). The MSE is calculated by using $MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[X(i,j)-Y(i,j)]}{M \times N}$, where M and N denote the total number of pixels in the horizontal and the vertical dimensions of the image. $X(i,j)$ represents the pixels in the cover image and $Y(i,j)$ represents the pixels of the stego image. The Peak Signal to Noise Ratio $(PSNR)$ is calculated using $PSNR = 10\ log_{10}\frac{I_{max}^2}{MSE}$, where I_{max} is the maximum intensity value of the pixel. Higher value of PSNR can be the better image quality.

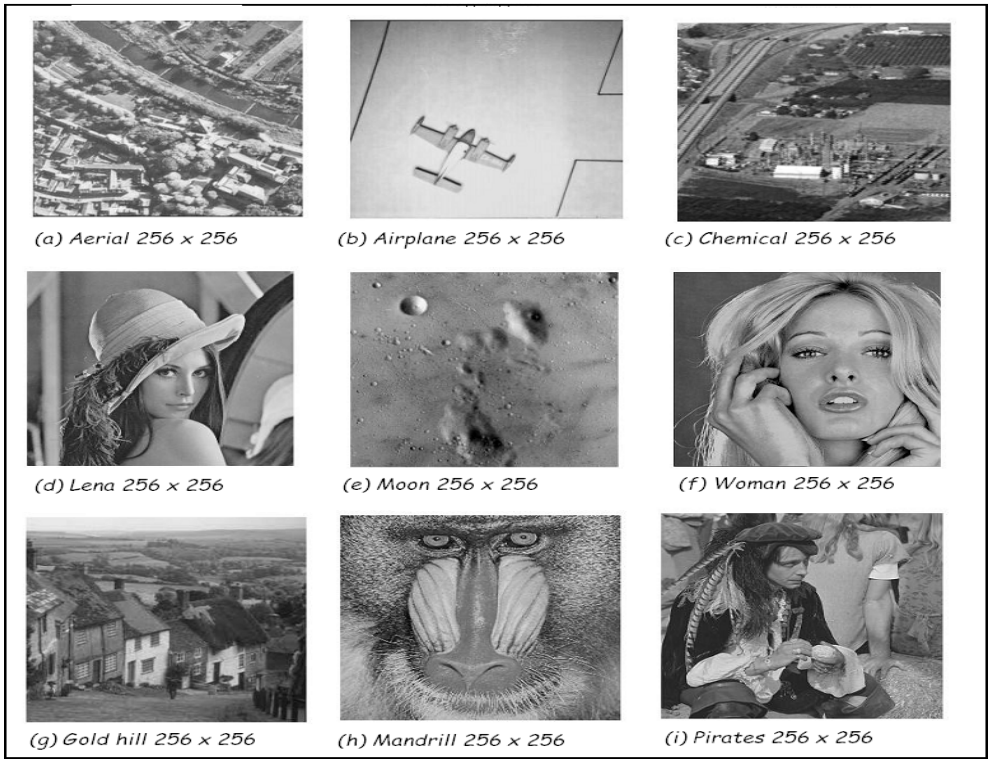


Figure 4.27: Standard cover images are used in TPVDDE scheme

The analysis of stego images in terms of $PSNR$ (dB) has given good results which are shown in Table 4.16. The average $PSNR$ of TPVDDE scheme varies 36.70 dB to 37.80 dB when data embedding capacity varies from $40,000$ bits to $1,61,992$ bits. To calculate the payload in terms

of bits per pixel (bpp), total bits = $256 \times 256/3 \times 13$ = 851968 bits. Here gray scale dual image of size (256×256) has been used. So, payload p = 851968/$(2 \times 256 \times 256)$ = 2.166 (bpp).

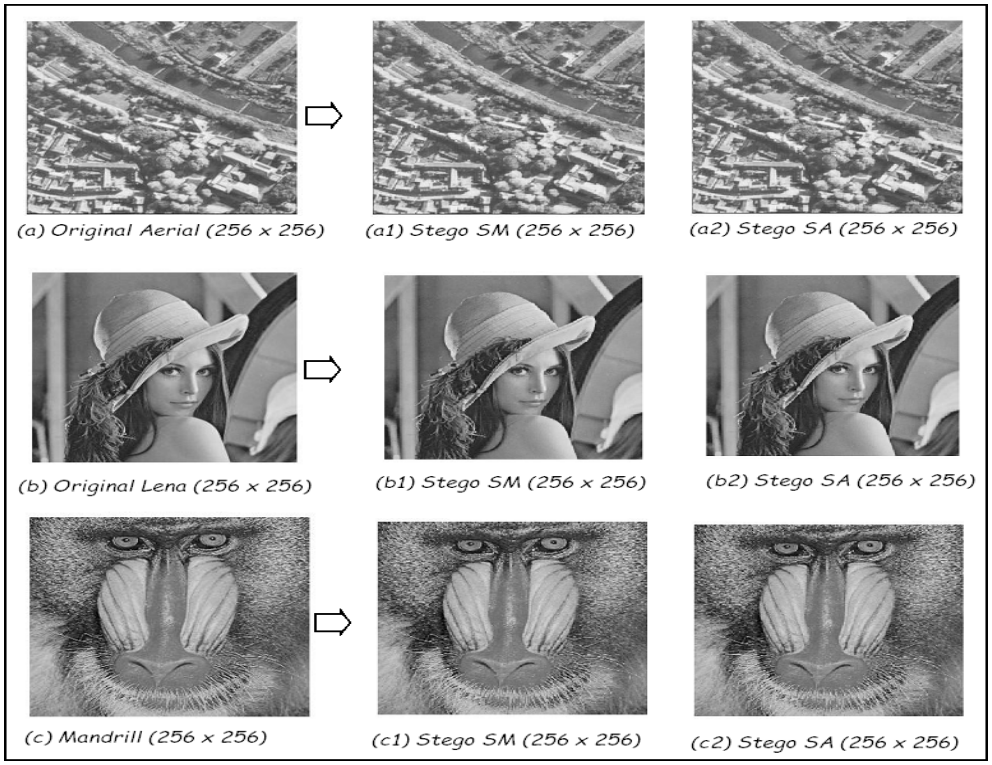


Figure 4.28: Dual stego images are generated after data embedding in TPVDDE scheme

To measure the security in our proposed method, the relative entropy (RE) between the probability distributions of the original image (P) and the stego image (Q) has been calculated by $D(Q||P) = \sum q(x) log \frac{q(x)}{p(x)}$.

The average PSNR (dB) of the stego images of the TPVDDE method is 26.18 (dB) when we embed maximum secret data that is 851968 bits, which is lower than the method proposed by Qin et al.'s [52], Lu et al.'s [45], Chang et al.'s [5], Lee et al.'s [36], Zeng et al.'s [75] and Chang et al.'s [6] schemes. The embedding capacity (payload) of the proposed TPDVDE method is 2.166 (bpp) which is higher than the other existing dual image based RDH schemes that are shown in Table 4.17.

Table 4.16: PSNR (dB) of stego images after embedding secret data in TPVDDE scheme

PSNR (dB) with Data embedding capacity (in bits)				
Image I	Capacity(bits)	PSNR(I and SM)	PSNR(I and SA)	Avg. PSNR
Jetplane	40000	47.07	43.29	37.88
	80000	37.18	39.36	
	160000	31.65	36.48	
	161992	31.62	36.41	
Lena	40000	40.31	43.78	36.93
	80000	35.31	40.19	
	160000	30.77	37.28	
	161992	30.67	37.18	
Living Room	40000	38.93	43.47	36.70
	80000	34.18	40.02	
	160000	31.37	37.19	
	161992	31.31	37.11	
Pirates	40000	39.79	43.75	37.05
	80000	35.29	40.28	
	160000	31.58	37.15	
	161992	31.48	37.09	
Woman	40000	40.44	43.75	37.37
	80000	36.32	40.20	
	160000	32.00	37.19	
	161992	31.92	37.12	

Table 4.17: Comparison of TPVDDE scheme with existing dual image based RDH schemes

Scheme	Average PSNR (dB)	Capacity (bpp)
Chang et al. [5]	45.1225	1.00
Cnang et al. [6]	48.14	1.00
Lee et al. [36]	34.38	0.91
Zeng et al. [75]	32.74	1.04
Lee and Huang [34]	49.6110	1.07
Qin et. al. [52]	52.11	1.16
Lu et al. [45]	49.20	1.00
TPVDDE	26.18	2.166

Table 4.18: Comparison of the proposed schemes in terms of payload (bpp) and PSNR (dB)

Proposed Schemes	Single/Dual Image	Capacity (in bits)	PSNR (dB)	Payload (bpp)
PVDDE	Dual	1,63,840	38.95	1.250
PVDEMD	Dual	9,16,656	40.43	1.750
TPVDDE	Dual	11,31,520	26.18	2.150

4.5 Analysis and Discussion

The comparison of proposed RDH schemes is presented in Table 4.18. It is observed that the PSNR of PVDEMD is higher than other proposed schemes but payload (bpp) is higher in TPVDDE. The PSNR of TPVDDE is less than the PSNR of PVDEMD. Here, three RDH schemes are proposed according to their the payload (bpp). In the PVDDE scheme, the payload is 1.25 (bpp), In PVDEMD scheme, the payload is 1.75 (bpp) and in the TPVDDE scheme, the payload is 2.16 (bpp). All these schemes are dual image based RDH schemes. Shared secret key has been used to enhance the security in PVDDE and PVDEMD. These schemes are designed in such a way that it is possible to handle the overflow and underflow situations. Steganalysis has been performed with stego image of all these schemes and their results are compared which are shown in Table 4.19.

Table 4.19: Comparison of proposed schemes in terms of steganalysis values

Proposed Schemes	Capacity (bits)	PSNR (dB)	RS value	Relative Entropy	CC (ρ)	Payload (bpp)
PVDDE	1,63,840	38.95	0.0123	0.0131	0.9840	1.250
PVDEMD	9,16,656	40.43	0.0447	0.0131	0.9820	1.750
TPVDDE	11,31,520	26.18	0.531	0.139	0.9682	2.150

Chapter 5

Reversible Data Hiding using Weighted Matrix

5.1 Introduction

Secure, high payload, reversible data hiding scheme with good visual quality is still an important research issue in data hiding and designing such scheme is a technically challenging problem. Tseng et al. [64] proposed a secure weighted matrix based data hiding scheme for binary image which can hide only two bits secret data within a (3×3) pixel block (B). Fan et al. [14] suggested an improved weighted matrix based data hiding scheme for gray scale image which can hide four bits secret data within a (3×3) pixel block. Both of these weighted matrix based data hiding schemes [64] [14] performed only one modular sum of entry-wise-multiplication operation between image block and weighted matrix.

There exists high-risk security vulnerability in special case, because an attacker will be able to estimate the form of weighted matrix by using brute force attack. In order to overcome the drawbacks of data embedding by matrix method, we developed an improved embedding strategy by modifying the weighted matrix W which are used for every block B of (3×3) original pixel. The W is updated using the formula $W_{i+1} = (W_i \times \kappa - 1) \mod 9$, where $i = 0, 1, 2, \ldots, 2^r$ and $gcd\ (\kappa, 9) = 1$. The sender will send a weighted matrix and κ to the receiver during data communication. Then sender can modified by increasing or decreasing the pixel value of the original image at the d^{th} position of the weighted matrix in the pixel location of the cover image which means if B increases by one then the modular sum $SUM(B \otimes W)$ will increase by $W \in \{0, 1, 2, \ldots, 2^{r-1} + 1\}$ and if B decreases by one then the modular sum $SUM(B \otimes W)$ will decrease by $W \in \{0, 1, 2, \ldots, 2^{r-1} + 1\}$. In extraction phase, the receiver only needs to calculate $SUM(B^{'} \otimes W)(\mathrm{mod}\ 2^r)$.

There is a scope to increase data embedding capacity by performing more than one modular sum of entry-wise-multiplication operation between image block and weighted matrix. The modifications performed within pixels at the time of data embedding through weighted matrix are hard to recover during data extraction at the receiver end. That means weighted matrix based data hiding scheme is not reversible which is one of the important requirements in many human centric application areas. After data extraction, the requirement of the image reversibility for the entire recovery of the cover image without any distortion goes high. In this chapter, three new weighted matrix based reversible data hiding schemes have been designed and solved using

dual image and image interpolation.

At first, Dual Image based RDH using Weighted Matrix (DRDHWM) has been proposed. The data hiding capacity of this approach is 1.98 (bpp) with PSNR greater than 37 (dB). To increase the payload, another new Interpolated Image based RDH using Weighted Matrix (IRDHWM) has been introduced. The data hiding capacity of IRDHWM increases and it is 2.97 (bpp) with PSNR greater than 37 (dB). Finally, an Interpolated Dual Image based RDH scheme using Weighted Matrix (IDRDHWM) has been further introduced to achieve high embedding capacity. The data hiding capacity of IDRDHWM increases; it is 3.462 (bpp) with PSNR greater than 35 (dB).

5.2 Dual Image based RDH using Weighted Matrix (DRDHWM)

In this approach, original image has been partitioned into (3×3) pixel block and performed modular sum of entry-wise-multiplication operation with a predefined weighted matrix. After that we calculate the positional value (pv) by subtraction between the result of modular sum and secret data unit. Then we increase or decrease the pixel value by one unit at the corresponding pixel value of image block depending on the sign of the pv. After that we store the pv within dual image using addition or subtraction operation with the original pixel value of cover image. Repeat this process nine times to embed thirty six bits secret data within a (3×3) pixel block. Finally, original and modified pixels are distributed among dual image depending on a shared secret key. At the receiver end, secret data bits have been extracted by performing modular sum of entry-wise-multiplication operation with weighted matrix. After data extraction, cover image can be recovered from dual stego image without any distortion because the original pixels are not effected during data embedding.

5.2.1 Data Embedding Process

Here, data embedding process has been described in details. Consider the weighted matrix (W) and original image block (B) of size (3×3). Perform modular sum of entry-wise-multiplication between (B) and (W) that is $val = SUM(B \otimes W)(\mathrm{mod}\, 2^r)$. Then calculate positional value

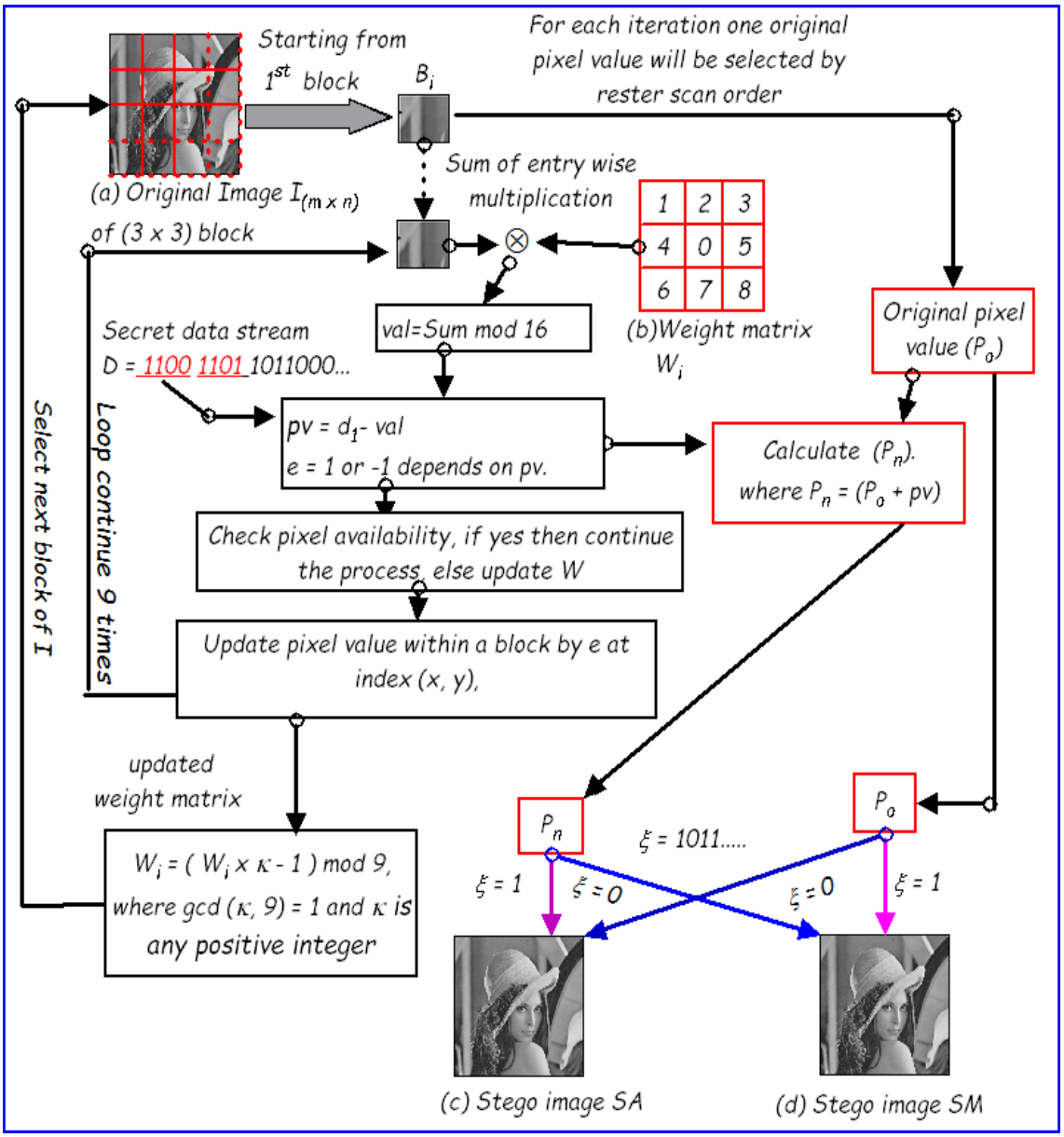


Figure 5.1: Schematic diagram of data embedding process in DRDHWM

(pv) through the subtraction of (val) from secret data unit d_i, where, $D = d_1, d_2, \ldots$, and $i = 1, 2, \ldots$, that is, $pv = d_i - val$. Now, check the sign of pv. If the sign of pv is positive then increase the corresponding pixel value of the block B by one. The corresponding pixel of B is the pixel which is mapped with the pv of the weighted matrix W; If the sign of pv is negative then decrease the corresponding pixel value by one unit. This indicates that if B increases by one then the modular sum $SUM(B \otimes W)$ will also increase by $W \in \{0, 1, 2, \ldots, 2^{r-1} + 1\}$ and if B decreases by one then the modular sum $SUM(B \otimes W)$ will decrease by $W \in \{0, 1, 2, \ldots, 2^{r-1} + 1\}$. Now, calculate P_n = (P_o) + pv; if pv is positive; else P_n = (P_o) - pv; Then distribute P_o and P_n between dual image. The SM or SA holds P_o

Input: Cover Image $I_{(m\times n)}$, Weight matrix $W_{(3\times 3)}$), Data $D = \{d_1, d_2, d_3, \ldots,$ where $d_i = 4$ bits each$\}$, Two shared secret key ξ and κ;

Output: Two stego image $SM_{(m\times n)}$ and $SA_{(m\times n)}$;

Initialize: $Dcount = Kcount = 1$; $sq = 3$; $SM = I$; $SA = I$;

Step 1:

for *(*$p = 1$ *to* (m/sq)*)* **do**
  for *(*$q = 1$ *to* (n/sq)*)* **do**
    $B_{pq}(3 \times 3) \leftarrow I_{(m\times n)}$;
    for *(*$i = (sq * (p-1)) + 1$ *to* $(sq * p)$*)* **do**
      for *(*$j = (sq * (p-1)) + 1$ *to* $(sq * q)$*)* **do**
        $SUM = B_{pq} \otimes W_{pq}$;
        $val = SUM (\mathrm{mod}\ 16)$; $pv = d_{Dcount} - val$;
        if *(*$pv > 0$*)* **then**
          if *(*$pv > 8$*)* **then** $pv = (16 - pv)$; $e = -1$;
          else $e = 1$;
        else
          if *(*$pv < -8$*)* **then** $pv = abs(16 + pv)$; $e = 1$;
          else $pv = abs(pv)$; $e = -1$;
        end
        $B_{pq}(x, y) = B_{pq}(x, y) + e$; if $W_r(x, y) = pv$, where $x = 1, 2, 3$ and $y = 1, 2, 3$;
        $P_o = I_{(m\times n)}(i, j)$; $P_n = I_{(m\times n)}(i, j) + (pv \times e)$;
        if $(\xi(Kcount) = 1)$ **then**
          $SM_{(m\times n)}(i, j) = P_o$; $SA_{(m\times n)}(i, j) = P_n$;
        else
          $SM_{(m\times n)}(i, j) = P_n$; $SA_{(m\times n)}(i, j) = P_o$;
        end
        $Dcount = Dcount + 1$; $Kcount = Kcount + 1$;
        if *(*$Kcount > length(\xi)$*)* **then**
          $Kcount = 1$;
        end
        if *(*$Dcount > length(D)$*)* **then**
          goto **Step 2**
        end
      end
    end
    $W_{pq+1} = ((W_{pq} \times \kappa - 1) \bmod 9)$; where $gcd\,(\kappa, 9) = 1$.
  end
end

Step 2: Produced two stego images $SM_{(m\times n)}$ and $SA_{(m\times n)}$;

Algorithm 13: Data embedding process of DRDHWM

or P_n that has been decided by $\xi_{(\ \mathrm{mod}\ (j, length(\xi)))+1}$. Since ξ is the secret key in binary form, $\mathrm{mod}(j, length(\xi)) + 1$ indicates the index value where $j = 1, 2, 3, \ldots$. If $\xi_{(\ \mathrm{mod}\ (j, length(\xi)))+1}$ = 1 then P_o is stored within SM and P_n is stored within SA; otherwise, P_o is stored within SA and P_n is stored within SM. We repeat this embedding process nine times and embed thirty six bits secret data within a block of dual image. After that W has been updated for new image

block using the following equation

$$W_{i+1} = (W_i \times \kappa - 1) \mod 9, \tag{5.1}$$

where $i = 0,\ 1,\ 2,\ \ldots,\ 2^r$ and $gcd\ (\kappa, 9) = 1$. We apply the same data embedding process using new weighted matrix for new image block. Continue the process to hide a good amount of secret data within dual image using this scheme. In the extraction phase, the receiver only calculates $SUM(B^{'} \otimes W)(\mathrm{mod}\ 2^r)$ and extracts the secret message.

After embedding all secret data bits, two stego images SM and SA are generated. ξ, W and κ plays an important role during data embedding and data extraction process. ξ is used to distribute stego and original pixels between dual image. The κ is any positive integer which is used to update weighted matrix. The schematic diagram of data embedding is shown in Fig. 5.1 and the corresponding algorithm is listed in Algorithm 13. The numerical illustration of data embedding is shown in Fig 5.2.

Example 5.2.1 *Now, consider one example of embedding process shown in* $Fig.5.2$*. Suppose a non-overlapping block* B_1 $(Fig.5.2(a))$*, the secret data bits are* $D = d_1d_2d_3d_4d_5\ldots$ = 1001 1100 0010 1110 0111 0001 1001 11100011 *....* *To embed secret data within the pixel value* $P_o = 10$ *at location* $B_1(1,1)$*, we calculate the sum of entry-wise-multiplication as* $SUM[(B_1(1,1) \times W(1,1))$ *+* $(B_1(1,2) \times W(1,2))$ *+* $(B_1(1,3) \times W(1,3))$ *+* $(B_1(2,1) \times W(2,1))$ *+* $(B_1(2,2) \times W(2,2))$ *+* $(B_1(2,3) \times W(2,3))$ *+* $(B_1(3,1) \times W(3,1))$ *+* $(B_1(3,2) \times W(3,2))$ *+* $(B_1(3,3) \times W(3,3))]$ *=* $SUM[(10\times1)$ *+ (12×2) + (14×3) + (16×4) + (18×0) + (22×5) + (20×6) + (27×7) + (30×8)] =* 799*. Then perform* $val = 799 \mod 16 = 15$ *which is not equal to* $d_1 = (1001)_2 = (9)_{10}$*. So* B_1 *is modified to* B_1' *by calculating the* $pv = -6$*. and* $e = -1$*. Then* P_n *= (10 + (-6)) = 4. Since* $\xi(1) = 1$*,* $SM_1(1,1) = P_o = 10$ *and* $SA_1(1,1) = P_n = 4$*. Similarly at location* $B_1(1,2)$*, the value is 12. Perform sum of entry - wise - multiplication as given below.* $SUM[(10\times1)+(12\times2)+(14\times3)$ *+(16×4)+(18×0)+(22×5)+(19×6)+ (27×7)+(30×8)] = 793. Then perform* $val = 793\ mod\ 16 = 9$ *which is not equal to* $d_2 = (1100)_2 = (12)_{10}$*. So, modify* $B_1^{'}$ *to* $B_1^{''}$ *by the help of* pv *and* e*. Here secret key bit stream* $\xi(2) = 0$*. So,* $SM_1(1,2) = P_n = (12 + (12 - 9)) = 15$ *and* $SA_1(1,2) = P_o = 12$*. In this way, dual stego image* SM *and* SA *are generated which are shown in Fig.* 5.2*.* ■

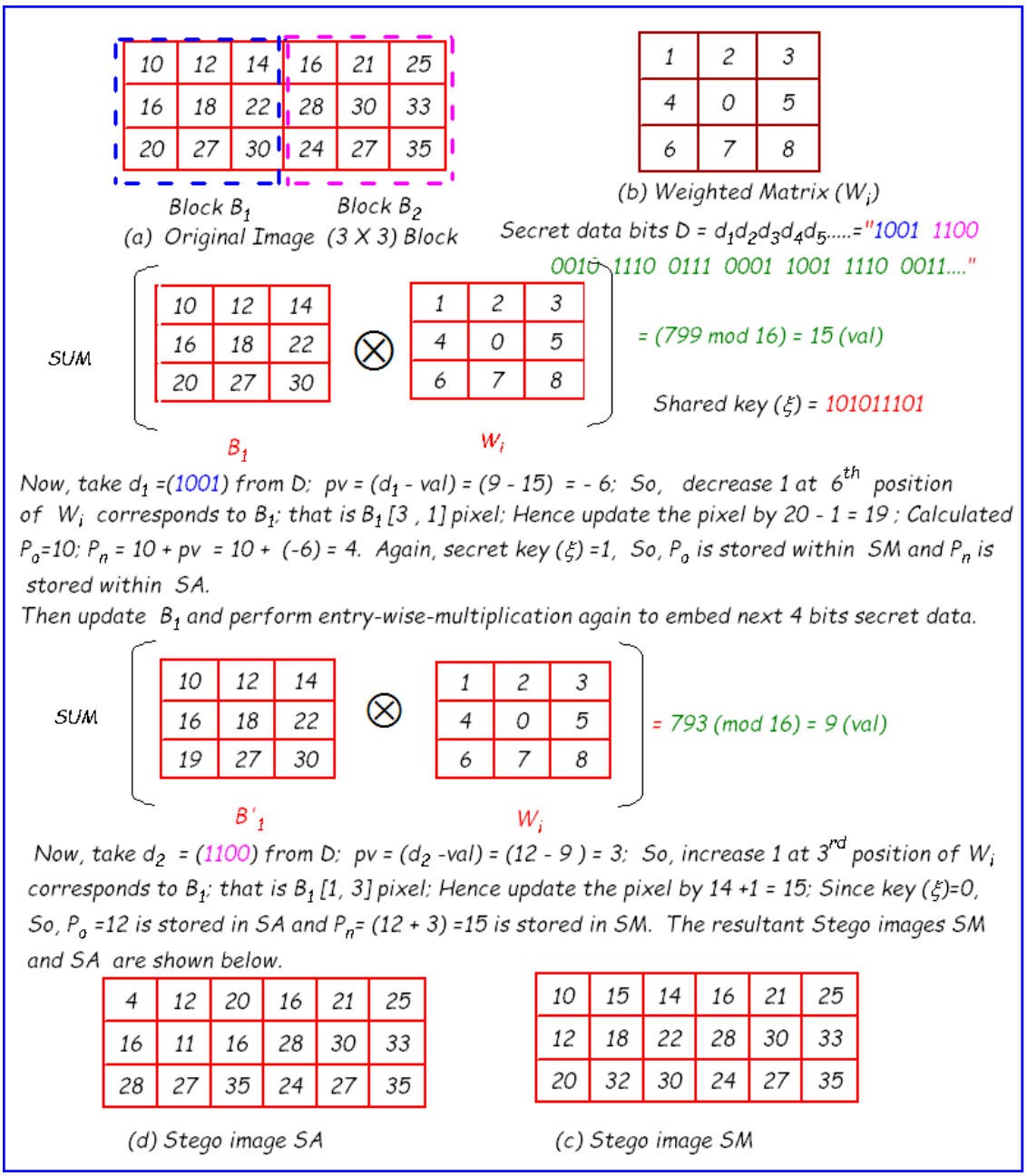


Figure 5.2: Numerical example of data embedding in DRDHWM

5.2.2 Data Extraction Process

At the receiver end, data extraction process has been performed from dual stego images SM and SA with the help of secret keys ξ, κ and weighted matrix (W). First, rearrange the original pixel (P_o) and new pixel (P_n) by selecting (3×3) pixel block B and generate the original image matrix (I) and new image matrix (NI) respectively with the help of ξ. Then calculate the difference and store in matrix (DM) using

$$DM = NI - I \tag{5.2}$$

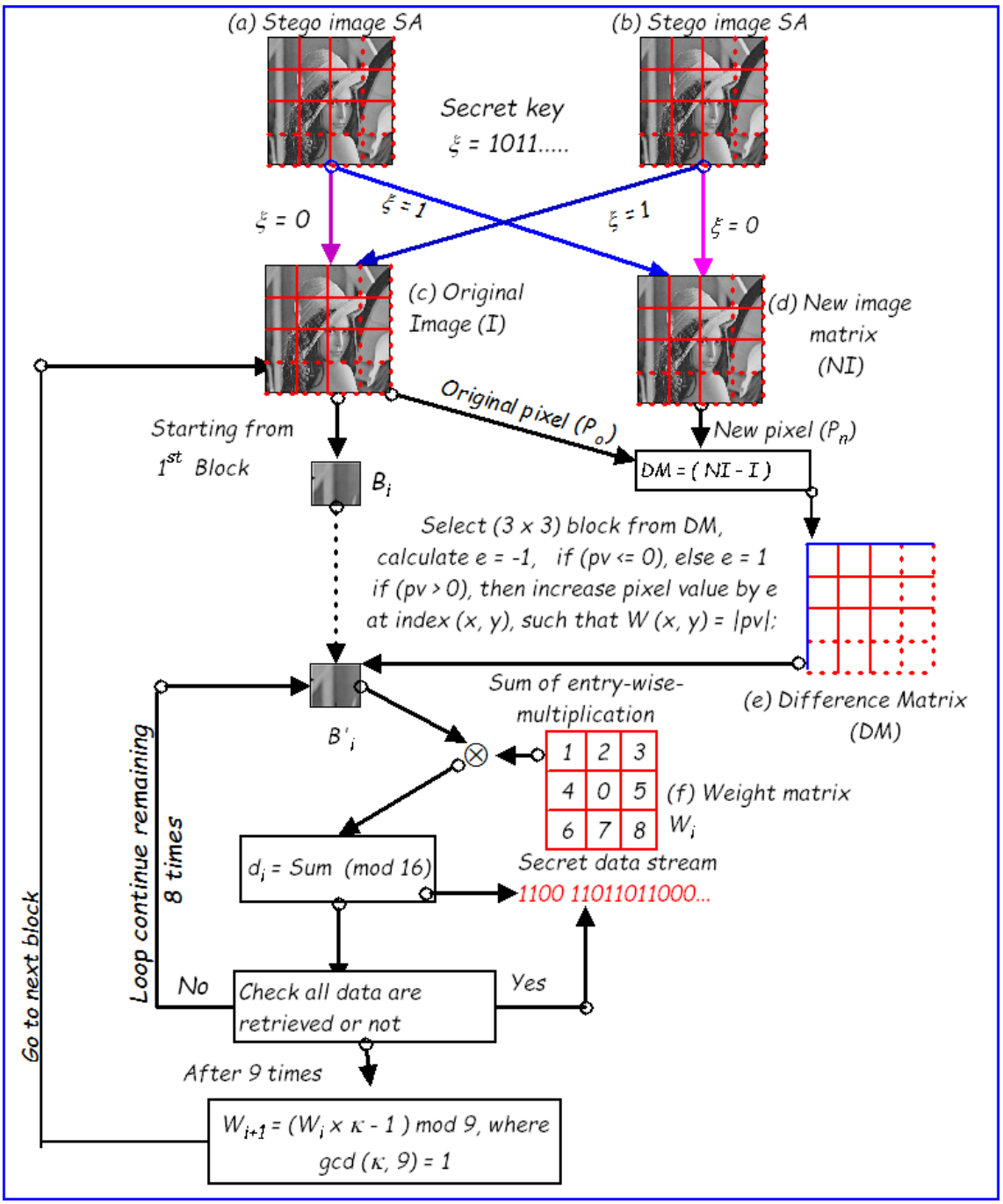


Figure 5.3: Schematic diagram of data extraction process in DRDHWM

The equation (5.2) is a simple matrix subtraction between NI and I. Now, modify the block B to B' depending on the difference value of $DM_i(x, y)$ where $x = 1, 2, 3$ and $y = 1, 2, 3$. The B has been modified to B' by increasing or decreasing one which will be depended on the sign of the value of DM_i(1,1) and the corresponding position of W. Then perform the sum of entry-wise-multiplication operation between B and W. The r bits secret data will be extracted using $SUM(B^{'} \oplus W) \pmod{2^r}$. Continue the operation nine times then modify W

for each new image block (B_i), where i = 1, 2, 3, ..., N_B, N_B is the number of block, using $W_{i+1} = (W_i \times \kappa - 1) \mod 9$, where $gcd\,(\kappa, 9) = 1$. Then select B_{i+1} for next iteration. The schematic diagram for extraction is shown in Fig. 5.3. The algorithm of secret data extraction is shown in Algorithm 14. The numerical illustration is also shown in Fig. 5.4.

Example 5.2.2 *The numerical example of data extraction and original image recovery process has been shown in Fig. 5.4. Consider a shared secret key ξ in binary form say, ξ = 101011101. If ξ = 1 then P_o = 10 collected from SM and P_n = 4 collected from SA. Store all P_o within I and all P_n within NI. With the help of I and NI, we generate a list of pv using subtraction operation as $\{$(4 - 10),(15 - 12),(20 - 14),(12 - 16),(11 - 18),(16 - 22),(28 - 20),(32 - 27),(35 - 30)$\}$ equals to $\{$-6, 3, 6, -4, -7, -6, 8, 5, 5$\}$. Before going to calculate sum of entry-wise-multiplication, the block B_1 is modified to B_1' due to the pv value = - 6. Then find the sum as $SUM[(10\times1)$ + (12×2) + (14×3) + (16×4) + (18×0) + (22×5) + (19×6) + (27×7) + $(30\times8)]$ = 793 and find the result of $\mod$ 16 which is actually the hidden four bits data $(9)_{10} = (1010)_2$. Again modify B_1' for the next pv value by 3 and start to extract the next four bits secret data using same the process. Continue the process to extract all hidden data D.* ■

5.2.3 Overflow and Underflow Control

Here, P_o is the original image where overflow or underflow situation may not occur but overflow occurs in P_n because it has been updated by addition or subtraction operation using pv. For example, consider $P_o = 248$ and $e = 1$, $pv = 8$, then P_n = (248 + 8) = 256 which is greater then 255. So overflow situation may arise. Similarly, if P_o = 6 and $e = -1$ and pv = 7, then P_n = (6 - 7) = - 1 which is less then 0. So underflow situation may arise. To overcome this problem, we calculate P_n using equation (5.3).

$$P_n = \begin{cases} 247 + (pv \times e), & \text{if } P_o > 247 \\ 8 + (pv \times e), & \text{if } P_o < 8 \\ P_o + (pv \times e), & \text{otherwise.} \end{cases} \tag{5.3}$$

Here, we set P_o as 247 if it is greater then 247 and set 8 if it is less then 8. At the receiver end, receiver can easily find P_o and P_n by the key ξ. Now, to extract pv in this case, we follow the

Input: Two stego image $SM_{(m\times n)}$ and $SA_{(m\times n)}$; Weighted matrix $W_{(3\times 3)}$, Two shared secret keys ξ and κ; $Dlen$ is the data length;

Output: Cover image $I'_{(m\times n)}$, Data $D' = \{d'_1, d'_2, d'_3, \ldots,$ where $d'_i = 4$ bits each$\}$;

Initialize: $Dcount = Kcount = 1$; $sq = 3$; DM is a matrix that holds the pv values; $I' = SM$;

Step 1:

for *(*$p = 1$ *to* (m/sq)*)* **do**
  for *(*$q = 1$ *to* (n/sq)*)* **do**
    for *(*$i = (sq * (p-1)) + 1$ *to* $(sq * p)$*)* **do**
      for *(*$j = (sq * (q-1)) + 1$ *to* $(sq * q)$*)* **do**
        if *(*$\xi(Kcount) = 1$*)* **then**
          $P_o = SM_{(m\times n)}(i,j)$; $P_n = SA_{(m\times n)}(i,j)$; $I'_{pq}(i,j) = P_o$;
        else
          $P_o = SA_{(m\times n)}(i,j)$; $P_n = SM_{(m\times n)}(i,j)$; $I'_{pq}(i,j) = P_o$;
        end
        $DM_{pq}(i,j) = (P_n - P_o)$;
      end
      $Kcount = Kcount + 1$;
      if *(*$Kcount > length(\xi)$*)* **then** $Kcount = 1$;
    end
  end
end

Step 2:

for *(*$p = 1$ *to* (m/sq)*)* **do**
  for $(q = 1$ *to* $(n/sq))$ **do**
    $B_{pq}(3\times 3) \leftarrow I'_{(m\times n)}$;
    for *(*$i = (sq * (p-1)) + 1$ *to* $(sq * p)$*)* **do**
      for *(*$j = (sq * (q-1)) + 1$ *to* $(sq * q)$*)* **do**
        if *(*$DM_{pq}(i,j) > 0$*)* **then** $pv = DM_{pq}(i,j)$; $e = 1$;
        if *(*$DM_{pq}(i,j) \leq 0$*)* **then** $pv = abs(DM_{pq}(i,j))$; $e = -1$;
        $B_{pq}(x,y) = B_{pq}(x,y) + d$, if $W_r(x,y) = pv$, where $x = 1,2,3$ and $y = 1,2,3$;
        $SUM = B_{pq} \otimes W_{pq}$;
        $d'_{Dcount} = SUM(\mathrm{mod}16)$; $pv = d_{Dcount} - val$;
        $Dcount = Dcount + 1$;
        if *(*$Dcount > Dlen$*)* **then** goto **Step 3**;
      end
    end
    $W_{pq+1} = (W_{pq} \times \kappa - 1) \bmod 9$; where $gcd\,(\kappa, 9) = 1$
  end
end

Step 3: Produced $I'_{(m\times n)}$ and $D' = \{d'_1, d'_2, d'_3, \ldots,$ where $d'_i = 4$ bits each$\}$

Step 4: End

Algorithm 14: Data extraction process of DRDHWM

equation (5.4).

$$pv = \begin{cases} P_n - 247, & \text{if } P_o > 247 \\ P_n - 8, & \text{if } P_o < 8 \\ P_n - P_o, & \text{otherwise.} \end{cases} \tag{5.4}$$

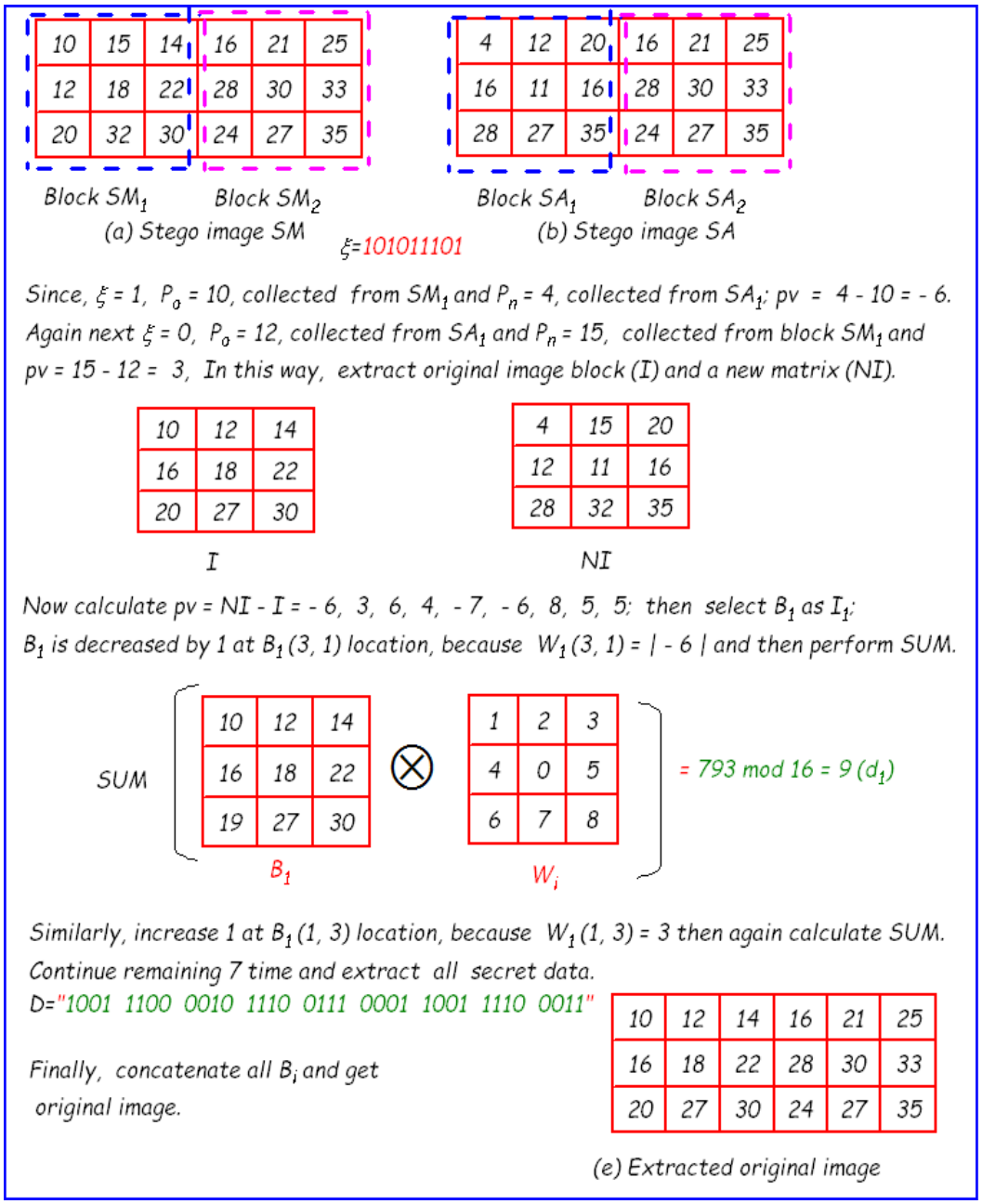


Figure 5.4: Numerical example of data extraction in DRDHWM

5.2.4 Experimental Results and Comparisons

Data embedding and data extraction algorithms of DRDHWM scheme are implemented through MATLAB Version 7.6.0.324 (R2008a). The payload in terms of bits per pixel (bpp) is calculated by the following equation

$$p = \frac{\frac{m}{x} \times \frac{n}{y} \times ((x \times y) \times r)}{(m \times n \times s)}, \tag{5.5}$$

Figure 5.5: Standard input images are used in DRDHWM

where m and n represent the size of input image, x and y represent the size of block, r represents the number of bits which are to be embedded within each block, s represents the number of stego images (here it is 2 for dual image). Consider $m = 256$, $n = 256$, $x = 3$, $y = 3$, $r = 4$ and $s = 2$. So, the payload $p = 1.98$ (bpp).

The standard cover images are used for experiment which are shown in Fig 5.5 and Fig 5.6 shows generated dual stego images after embedding 2,60,096 bits secret data. It is observed from the Table 5.1 that the average PSNR of the stego images is around 37.7 (dB) when embedding capacity is 2,60,096 bits. The payload is 1.98 (bpp). The data embedding capacity of

Figure 5.6: Generated dual stego images after data embedding in DRDHWM

Table 5.1: PSNR (dB) of stego images with capacity in DRDHWM

Image (I)	Embedded Data (bits)	PSNR of SM	PSNR of SA	Avg. PSNR
Camera Man	80,000	42.6752	43.1809	42.92805
	1,60,000	39.7952	40.1716	39.9834
	2,40,000	37.9778	38.4299	38.20035
	2,60,096	37.6072	38.0798	37.8435
House	80,000	42.6101	43.1188	42.86445
	1,60,000	39.6894	40.1066	39.898
	2,40,000	37.9109	38.3755	38.1432
	2,60,096	37.5586	38.0262	37.7924
Jet Plane	80,000	42.6988	43.2139	42.95635
	1,60,000	39.7976	40.1907	39.99415
	2,40,000	37.9901	38.4492	38.21965
	2,60,096	37.6146	38.0896	37.8521
Lake	80,000	42.7090	43.1874	42.9482
	1,60,000	39.7833	40.1606	39.97195
	2,40,000	37.9790	38.4355	38.20725
	2,60,096	37.6212	38.0835	37.85235
Lena	80,000	42.7090	43.1928	42.9509
	1,60,000	39.7856	40.1739	39.97975
	2,40,000	37.9790	38.4502	38.2146
	2,60,096	37.6379	38.0760	37.85695
Peppers	80,000	42.7149	43.2015	42.9582
	1,60,000	39.7838	40.1705	39.97715
	2,40,000	37.9876	38.4235	38.20555
	2,60,096	37.6140	38.0699	37.84195
Bridge	80,000	42.6612	43.1739	42.9175
	1,60,000	39.7634	40.1405	39.9519
	2,40,000	37.9640	38.4279	38.1959
	2,60,096	37.6078	38.0475	37.8276
Tiffany	80,000	42.6908	43.1750	42.9329
	1,60,000	39.7634	40.1471	39.9552
	2,40,000	37.9148	38.3954	38.1551
	2,60,096	37.5652	38.0385	37.8018
Little Lady	80,000	42.2023	42.6494	42.4258
	1,60,000	39.5056	39.9096	39.7076
	2,40,000	37.4163	37.8754	37.6458
	2,60,096	36.9781	37.4429	37.2105

this scheme is higher than other mentioned schemes in Table 5.2. So, in terms of payload this method is superior than others, but the PSNR is slightly dropped.

Table 5.2: Comparison of DRDHWM with existing dual image based schemes

Methods	PSNR (dB)	Images			
		Lena	Peppers	Boat	Goldhill
Chang et al. [5]	PSNR (1)	45.12	45.14	45.12	45.13
	PSNR (2)	45.13	45.15	45.13	45.14
	Avg. PSNR	45.13	45.15	45.13	45.14
	Payload (bpp)	1	0.99	1	1
Chang et al. [6]	PSNR (1)	48.13	48.11	48.13	48.13
	PSNR (2)	48.14	48.14	48.12	48.15
	Avg. PSNR	48.14	48.13	48.13	48.14
	Payload (bpp)	1	1	1	1
Lee et al. [34]	PSNR (1)	49.76	49.75	49.76	49.77
	PSNR (2)	49.56	49.56	49.57	49.57
	Avg. PSNR	49.66	49.66	49.67	49.67
	Payload (bpp)	1.07	1.07	1.07	1.07
Chang et al. [10]	PSNR (1)	39.89	39.94	39.89	39.9
	PSNR (2)	39.89	39.94	39.89	39.9
	Avg. PSNR	39.89	39.94	39.89	39.9
	Payload (bpp)	1.53	1.52	1.53	1.53
Qin et al. [52]	PSNR (1)	52.11	51.25	51.11	52.11
	PSNR (2)	41.34	41.52	41.57	41.34
	Avg. PSNR	46.72	46.39	46.84	46.72
	Payload (bpp)	1.16	1.16	1.16	1.16
Lu et al. [45]	PSNR (1)	49.20	49.19	49.20	49.23
	PSNR (2)	49.21	49.21	49.21	49.18
	Avg. PSNR	49.21	49.20	49.21	49.21
	Payload (bpp)	1	0.99	1	1
DRDHWM	PSNR (1)	37.63	37.61	37.58	37.59
	PSNR (2)	38.07	38.06	38.01	38.08
	Avg. PSNR	37.85	37.83	37.79	37.83
	Payload (bpp)	**1.98**	**1.98**	**1.98**	**1.98**

5.2.5 Steganalysis and Steganographic Attacks

The steganalysis has been performed through RS analysis proposed by J. Fridrich [18], statistical analysis and relative entropy has been calculated. The perceptibility of the proposed scheme have been measured through various steganographic attacks including Jeremiah J. Harmsena's Histogram attack and Brute Force Attack.

5.2.5.1 RS Analysis

The stego images are tested through the J. Fridrich's RS steganalysis [18]. It is observed from Table 5.3 and Table 5.4 that the values of R_M and R_{-M}, S_M and S_{-M} are nearly equal. The RS

value for lena image of SM is 0.0094 after embedding 2,60,096 bits. Thus rule $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ are satisfied for the stego image in this scheme. So, this is secure against RS attack.

Table 5.3: Results of RS analysis for stego images SM in DRDHWM

Image	Data	SM				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Cameraman	80000	6896	6922	3807	3817	0.0034
	160000	6403	6451	4270	4237	0.0076
	240000	6074	6138	4496	4468	0.0087
	260096	6152	6122	4527	4479	0.0073
Lena	80000	5490	5586	4142	4050	0.0195
	160000	5427	5550	4280	4149	0.0262
	240000	5422	5586	4424	4312	0.0280
	260096	5484	5535	4406	4448	0.0094
Baboon	80000	5872	5812	5010	5141	0.0176
	160000	5800	5815	5116	5112	0.0017
	240000	5851	5770	5118	5219	0.0166
	260096	5856	5757	5109	5215	0.0187

Table 5.4: Results of RS analysis for stego images SA in DRDHWM

Image	Data	SA				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Cameraman	80000	6961	6954	3788	3770	0.0023
	160000	6537	6422	4190	4248	0.0161
	240000	6179	6278	4426	4405	0.0113
	260096	6278	6244	4420	4447	0.0057
Lena	80000	5572	5483	4071	4100	0.0122
	160000	5476	5570	4308	4214	0.0192
	240000	5475	5452	4299	4354	0.0080
	260096	5409	5602	4495	4338	0.0353
Baboon	80000	5813	5831	5004	5091	0.0097
	160000	5841	5823	5077	5135	0.0070
	240000	5915	5701	5022	5251	0.0405
	260096	5803	5793	5088	5134	0.0051

5.2.5.2 Relative Entropy

In this experiment, it has been observed that the relative entropy is directly proportional to the number of secret data bits. The relative entropy varies from 0.01 to 0.18 when 80,000

to 2,60,096 bits are embedded, which is very small and implies that the DRDHWM scheme provides secure hidden data communication which are shown in Table 5.5.

Table 5.5: Relative entropy of stego images SM and SA in DRDHWM

Image	Data	Entropy I	Entropy SM	Difference (I & SM)	Entropy SA	Difference (I&SA)
Cameraman	80000	7.0299	7.0572	0.04	7.1143	0.02
	160000		7.1220	0.01	7.1143	0.03
	240000		7.1547	0.14	7.1458	0.18
	260096		7.1555	0.14	7.1452	0.12
Lena	80000	7.4429	7.4491	0.02	7.4494	0.01
	160000		7.4550	0.02	7.4562	0.01
	240000		7.4653	0.03	7.4622	0.03
	260096		7.4668	0.04	7.4654	0.03
Baboon	80000	7.2371	7.2393	0.04	7.2394	0.04
	160000		7.2438	0.05	7.2437	0.05
	240000		7.2471	0.05	7.2461	0.05
	260096		7.2469	0.05	7.2475	0.05

5.2.5.3 Statistical Analysis

The SD (σ) and CC (ρ) of cover and stego images has been calculated and depicted in Table 5.6. It is observed that the σ of cover image and stego image is nearly equal and the ρ is nearly 0.99. It is also observed that there is no substantial divergence between the σ of the cover image and the stego images. This study shows that the magnitude of change in stego images based on image parameters is small from a cover image. Since the image parameters have not changed a lot, the method offers good concealment of secret data and reduces the chances of the secret data bit being detected. Thus, it indicates secure data hiding scheme.

Table 5.6: Results of SD (σ) and CC (ρ) in DRDHWM

Image	SD (σ)			CC (ρ)		
	I	SM	SA	I & SM	I & SA	SM & SA
Cameraman	61.58	61.70	61.67	0.99	0.99	0.99
Lena	47.83	47.96	47.94	0.99	0.99	0.99
Baboon	38.37	38.48	38.50	0.99	0.99	0.99

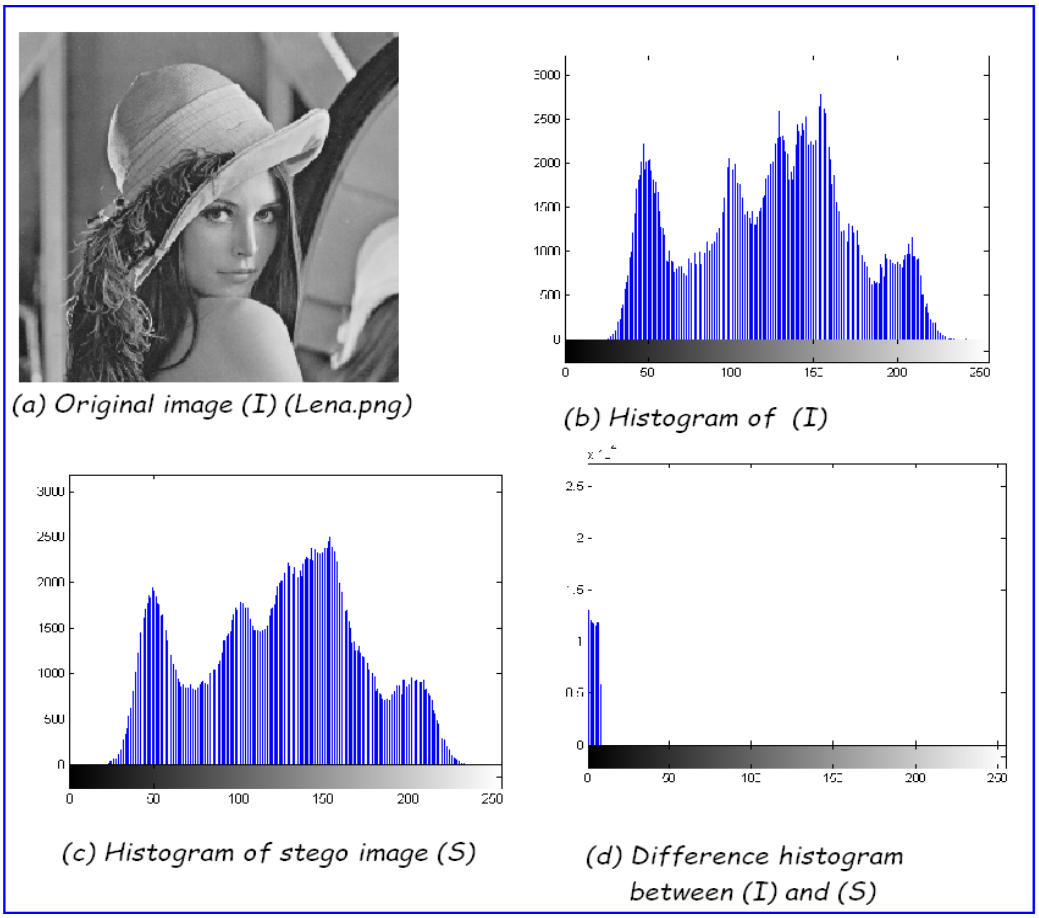


Figure 5.7: Results of Histogram attack in DRDHWM

5.2.5.4 Histogram Attack

A new steganalytic attack has been proposed by Harmsena [22] which is based on noise adding in the spatial domain corresponding to low-pass filtering of the histogram. Fig. 5.7 describes the histogram of the cover and the stego image and their difference histogram. The stego image is produced from cover image employing the maximum data hiding capacity. It is observed from Fig. 5.7 that the shape of the histogram is preserved after embedding secret data bits. Histogram of cover image is represented as h whereas histogram of stego image is represented as $h^{'}$. The change of histogram can be measured by $D_h = \sum_{m=1}^{255} |h^{'}_m - h_m|$. The difference of the histogram in this approach is very small. It is also observed that the bins close to zero are more in number and the bins which are away from zero are less in numbers. This confirms that the quality of stego image is preserved. There is no step pattern observed which ensures that the proposed method is robust against histogram analysis.

5.2.5.5 Brute Force Attack

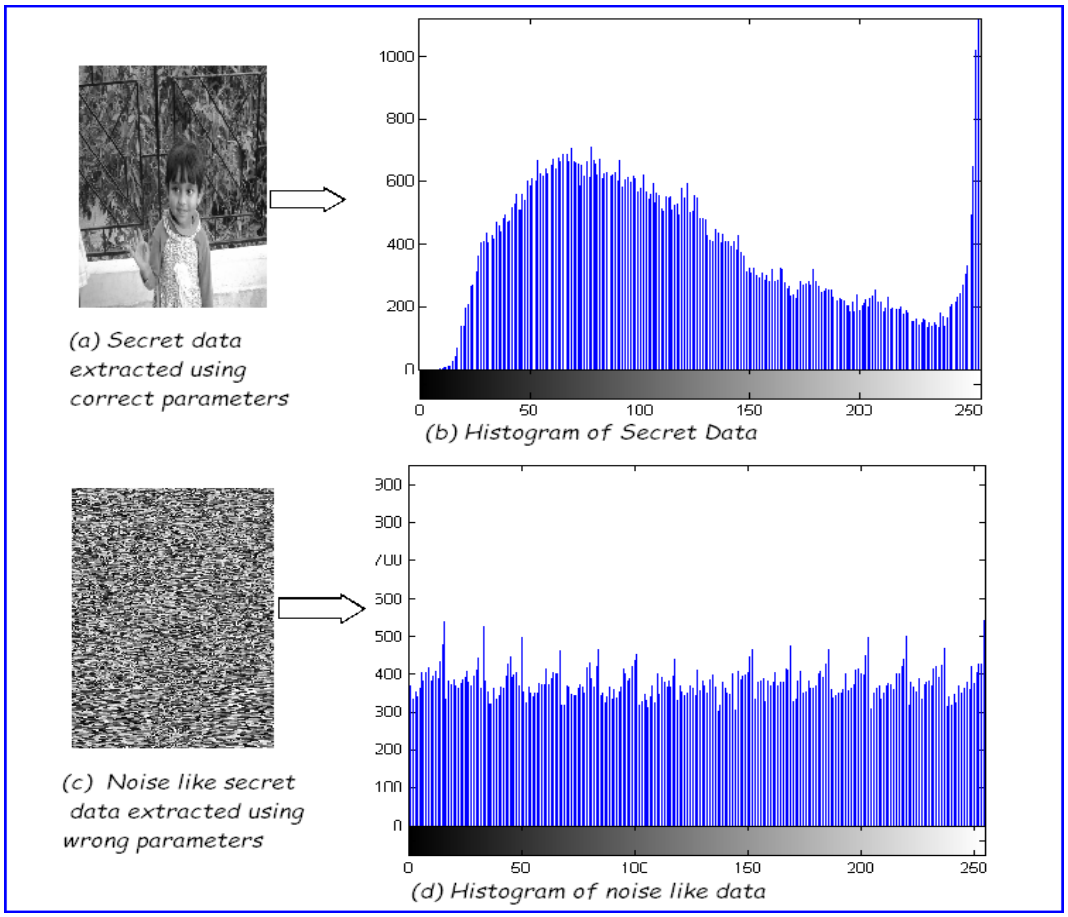


Figure 5.8: Results of Brute Force Attack in DRDHWM

The secret information is distributed among dual images. The proposed scheme protects secret information due to its distributed nature. The weighted matrix based data hiding scheme only hides embedding position value (pv) within dual image and not the actual data bits. Also the weighted matrix is updated for each new selected block using secret key κ. The method DRDHWM is secure to prevent possible malicious attacks. The Fig. 5.8 shows the example of getting noise like secret data when applied with a wrong key and wrong weighted matrix to revel the hidden message. If the malicious attackers hold the original image and stego image and are fully aware about the proposed scheme, the hidden message still cannot be revealed without secret key and weighted matrix. Similarly, if the malicious attacker is fully aware about the formation of weighted matrix of the proposed scheme then the hidden message still cannot be revealed without secret keys ξ and κ. Furthermore, the attacker may employ the brute force attack that tries all possible permutations and combinations to reveal the hidden message. The

possible number of weighted matrix to embed r bits secret data in each block are $(2^{r-1}+1)!$. The original matrix $(m \times n)$ has been used and partitioned into (3×3) blocks. Total number of blocks are $\lfloor \frac{m}{3} \rfloor \times \lfloor \frac{n}{3} \rfloor$ and each block is used as a modified weighted matrix. So, the number of trials to reveal the hidden message are $(2^{r-1}+1)! \times \lfloor \frac{m}{3} \rfloor \times \lfloor \frac{n}{3} \rfloor$. In this scheme, key ξ is used for pixel distribution among dual image. If the key length of ξ is l, then the possible combination of ξ will be 2^l. So total number of trials = $(2^{r-1}+1)! \times \lfloor \frac{m}{3} \rfloor \times \lfloor \frac{n}{3} \rfloor \times 2^l$.

Example 5.2.3 *Consider the image of* (256×256). *with* $r = 4$ *bits and the key length is* l *bits. The number of trials to reveal the secret message will be* $(2^l \times 3,62,880) \times 7281$ *which is computationally infeasible by an adversary using current computers.* ■

The proposed scheme has achieved stronger robustness against several attacks. Furthermore, the secret information can be retrieved without encountering any loss of data and original image can be recovered successfully from stego image with a valid key and weighted matrix.

5.3 Interpolated Image based RDH using Weighted Matrix (IRDHWM)[8,9]

High payload weighted matrix based reversible data hiding scheme has been proposed in this section. A secure data hiding scheme for binary images using a key matrix and a weight matrix has been proposed by Tseng et al. [64] which can hide only 2 bits secret data within a (3×3) pixel block. After that, Fan et al. [14] suggested an improved efficient data hiding scheme using weighted matrix for gray scale images which can hide 4 bits in a (3×3) block. In the literature, it is found that weighted matrix based data hiding schemes performed only one modular sum of entry wise multiplication between weighted matrix W and a pixel block of original image. Only one embedding operation is performed with a single block and only 4 bits secret data are embedded within a block. High-capacity is still one of important research issue in data hiding. After message extraction, the requirement for the image reversibility without any distortion

[8]Published in the proceedings of the International Conference on Computational Intelligence in Data Mining (ICCIDM-2015), Advances in Intelligent Systems and Computing, Springer India, Vol. 411, pp. 239-248, with title *Weighted Matrix Based Reversible Data Hiding Scheme Using Image Interpolation.*

[9]Published in the **Optik-International Journal for Light and Electron Optics, Elsevier, (Impact Factor-0.677) (2016), 127(6), 3347-3358**, with title *High Payload Reversible Data Hiding Scheme using Weighted matrix.*

goes high. Here, we proposed a high capacity reversible data hiding scheme where twelve time multiplication operations have been performed in each block to hide forty eight bits secret data within the block. The developed scheme provides average embedding payload 2.97 (bpp) with good visual quality measured by PSNR which is higher than 37.97 (dB).

5.3.1 Data Embedding Process

First, enlarge the original image I of size $(M \times N)$ using image interpolation technique which produces cover image C of size $((2M-1) \times (2N-1))$ using equation (5.6).

$$\begin{cases} C(i,j) = I(p,q) \\ \quad \{\text{ where }, p = 1 \ldots M,\ q = 1 \ldots N,\ i = 1,3,\ldots,(2M-1),\ j = 1,3,\ldots,(2N-1)\} \\ C(i,j) = (C(i,j-1) + C(i,j+1))/2 \\ \quad \{\text{ where } (((i \bmod 2) \neq 0) \& ((j \bmod 2) = 0)), \forall\, i = 1 \ldots,(2M-1), j = 1 \ldots,(2N-1)\} \\ C(i,j) = (C(i-1,j) + C(i+1,j))/2 \\ \quad \{\text{ where } (((i \bmod 2) = 0) \& ((j \bmod 2) \neq 0)),\ \forall\, i = 1 \ldots,(2M-1), j = 1 \ldots,(2N-1)\} \\ C(i,j) = (C(i-1,j-1) + C(i-1,j+1) + C(i+1,j-1) + C(i+1,j+1))/4 \\ \quad \{\text{ where } (((i \bmod 2) = 0) \& ((j \bmod 2) = 0)), \forall\, i = 1 \ldots,(2M-1), j = 1 \ldots,(2N-1)\} \end{cases} \tag{5.6}$$

Then partition the (I) into (3×3) pixel block $B_{(3\times3)}$ and (C) into (5×5) pixel block $C_{(5\times5)}$. Consider the predefined weighted matrix W of size (3×3). Now, perform modular sum of entry-wise-multiplication of $B_{3\times3}$ with W and get the result (val). Then calculate data embedding position (pv) through subtraction of (val) from secret data unit (D) that is $pv = D - val$.

If the sign of pv is positive then increase the pixel value by one unit at the corresponding position of image block $B_{(3\times3)}$ otherwise decrease the pixel value. At the same time, data embedding position (pv) is stored within the interleaved pixel of the cover image block $C_{(5\times5)}$. This sum of entry-wise-multiplication operation has been repeated twelve times and each time the pixel value at $B_{(3\times3)}$ has been increased / decreased and at the same time pv has been stored within the interleaved pixel of $C_{(5\times5)}$ to achieve high data embedding capacity. In each operation, four bits secret data is embedded without a block through only one unit change. As a result, the scheme can hide $(12 \times 4) = 48$ bits secret information within the single block. After finishing data embedding within a particular block, update weighted matrix for next block using $W_{i+1} = (W_i \times \kappa - 1) \mod 9$, where $gcd\ (\kappa, 9) = 1$, $i = 1, 2, \ldots, N_B$, N_B is the number of block. The Fig. 5.9 shows the schematic diagram of data embedding process. Numerical

Input: Original Image $I[M][N]$, Weighted matrix $W[3][3]$, Secret Data $D = \{d_1, d_2, d_3, \ldots,$ where $d_i = 4$ bits each$\}$, and shared secret key κ, $count = 1$;

Output: Stego image $S[2 \times M - 1][2 \times N - 1]$;

Step 1: Generate cover image $C[2 \times M - 1][2 \times N - 1]$ from $I[M][N]$ using equation (5.6); $S = C$;

Step 2: Partition $I[M][N]$ into (3×3) overlapping blocks ;

Step 3: $Brow = \lfloor \frac{M+1}{sq-1} \rfloor - 1$; $Bcol = \lfloor \frac{N+1}{sq-1} \rfloor - 1$;

for $p = 1$ *to* $Brow$ **do**
- **for** $q = 1$ *to* $Bcol$ **do**
  - $B_{pq}3 \times 3 \Leftarrow I[M][N]$; where sq = 3;
  - **if** *(*$p \neq Brow$ & $q \neq Bcol$*)* **then** $sqr = (2 \times (sq-1) \times p)$; $sqc = (2 \times (sq-1) \times q)$;
  - **if** *(*$p \neq Brow$ & $q = Bcol$*)* **then** $sqr = (2 \times (sq-1) \times p)$; $sqc = (2 \times (sq-1) \times q) + 1$;
  - **if** *(*$p = Brow$ & $q \neq Bcol$*)* **then** $sqr = (2 \times (sq-1) \times p) + 1$; $sqc = (2 \times (sq-1) \times q)$;
  - **if** *(*$p = Brow$ & $q = Bcol$*)* **then** $sqr = (2 \times (sq-1) \times p) + 1$; $sqc = (2 \times (sq-1) \times q) + 1$;
  - **for** $i = (2 \times (sq-1) \times (p-1))$ *to* sqr **do**
    - **for** $j = (2 \times (sq-1) \times (q-1))$ *to* sqc **do**
      - **if** *(*$i \mod 2$ = *0 or* $j \mod 2$ = *0)* **then**
        - **if** *(*$count \leq length(D)$*)* **then**
          - $SUM = B_{pq} \otimes W_r$; $dec = BCD(d_{count})$; $val = SUM(\mathrm{mod}16)$; $pv = dec - val$;
          - **if** *(*$pv > 0$*)* **then**
            - **if** *(*$pv > 8$*)* **then**
              - $pv = (16 - pv)$; e = -1;
            - **else**
              - e = 1;
            - **end**
          - **end**
          - **if** *(*$pv < 0$*)* **then**
            - **if** *(*$pv <$ - *8)* **then**
              - $pv = abs(16 + pv)$; e = 1;
            - **else**
              - $pv = abs(pv)$; e = -1;
            - **end**
          - **end**
          - $B_{pq}(x, y) = B_{pq}(x, y) + e$ if $W_r(x, y) = pv$, where $x = 1, 2, 3$ and $y = 1, 2, 3$;
          - $S_{pq}(i, j) = C_{pq}(i, j) + (pv \times e)$; $count = count + 1$;
        - **else**
          - goto **Step 4**;
        - **end**
      - **end**
    - **end**
  - $W_{i+1} = (W_i \times \kappa - 1) \mod 9$, where $gcd\,(\kappa, 9) = 1$;
- **end**

end

Step 4: Generate stego image $S[2M - 1 \times 2N - 1]$;

Step 5: End.

Algorithm 15: Data embedding process of IRDHWM

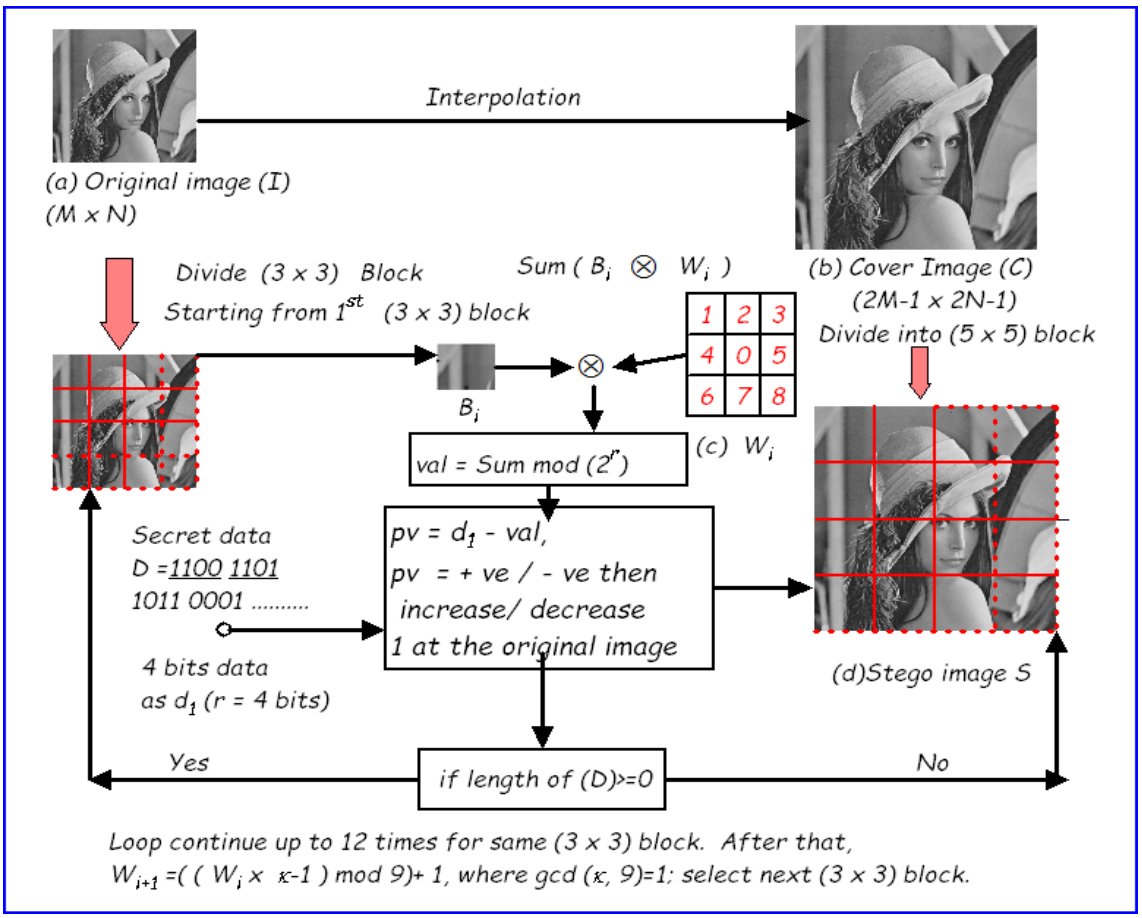


Figure 5.9: Schematic diagram of data embedding process in IRDHWM

illustration of data embedding process is described in Fig. 5.10 and the corresponding algorithm is listed in Algorithm 15.

$$\begin{cases} Inpo = (S(i,j-1) + S(i,j+1))/2; \text{ where } (i \mod 2) \neq 0 \text{ and } (j \mod 2) = 0; \\ Inpo = (S(i-1,j) + S(i+1,j)/2; \text{ where } (i \mod 2) = 0 \text{ and } (j \mod 2) \neq 0; \\ \qquad i = 1 \ldots (2 \times M - 1) \text{ and } j = 1 \ldots (2 \times N - 1); \\ Inpo = (S(i-1,j-1) + S(i-1,j+1) + \\ \qquad S(i+1,j-1) + S(i+1,j+1))/4; \text{ where } (i \mod 2) = 0 \text{ and } (j \mod 2) = 0; \end{cases} \tag{5.7}$$

5.3.2 Data Extraction Process

At first, extract the original image (I) by selecting pixels from odd row and odd column of stego image (S) which is shown in Fig. 5.10. Now, consider $BI_{(3\times3)}$ of size (3×3) from (I) and $BC_{(5\times5)}$ of size (5×5) from (S). Then calculate interpolation value $(Inpo)$ using equation (5.7) and position value (pv) using $pv = Inpo - current\ value$, where $current\ value$ is the present pixel value of the stego-image. After that, we check the sign of pv. If $(pv \leq 0)$ then

Input: Stego Image $S[2 \times M - 1][2 \times M - 1]$; Weighted matrix $W[3][3]$; shared secret key κ;
$count = 1$; length of hidden data ($stlen$); $M = \frac{(2\times M-1+1)}{2}$;$N = \frac{(2\times M-1+1)}{2}$;

Output: Original Image $I[M][N]$; Secret Data D ;

Step 1: Extract original Image $I[M][N]$ from Stego Image $S[2 \times M - 1][2 \times M - 1]$;

$i = 1; p = 1;$

while *($i \leq 2 \times M - 1$)* **do**

  $j = 1; q = 1;$

  while *($j \leq 2 \times M - 1$)* **do**

    $I[p][q] = S[i][j]; j + 2; q = q + 1;$

  end

  $i = i + 2; p = p + 1;$

end

Step 2: $Brow = \lfloor \frac{M+1}{sq-1} \rfloor - 1$; $Bcol = \lfloor \frac{N+1}{sq-1} \rfloor - 1$;

for $p = 1$ *to* $Brow$ **do**

  for $q = 1$ *to* $Bcol$ **do**

    $B_{pq}3 \times 3 \Leftarrow I[M][N]$; where $sq = 3$;

    if *($p \neq Brow$ & $q \neq Bcol$)* **then** $sqr = (2 \times (sq - 1) \times p)$; $sqc = (2 \times (sq - 1) \times q)$;

    if *($p \neq Brow$ & $q = Bcol$)* **then** $sqr = (2 \times (sq - 1) \times p)$; $sqc = (2 \times (sq - 1) \times q) + 1$;

    if *($p = Brow$ & $q \neq Bcol$)* **then** $sqr = (2 \times (sq - 1) \times p) + 1$; $sqc = (2 \times (sq - 1) \times q)$;

    if *($p = Brow$ & $q = Bcol$)* **then** $sqr = (2 \times (sq - 1) \times p) + 1$; $sqc = (2 \times (sq - 1) \times q) + 1$;

    for $i = (2 \times (sq - 1) \times (p - 1))$ *to* sqr **do**

      for $j = (2 \times (sq - 1) \times (q - 1))$ *to* sqc **do**

        if *($i \mod 2 = 0$ or $j \mod 2 = 0$)* **then**

          if $(count \leq stlen)$ **then**

            calculate interpolate value **(Inpo)** using equation (5.7) at location (i, j) with the help of $S[2 \times M - 1] \times [2 \times M - 1]$

            if $(Inpo < S(i, j))$ **then**

              $pv = S(i, j) - Inpo$; $e = 1$;

            else

              $pv = Inpo - S(i, j)$;$e = -1$;

            end

            if $pv \neq 0$ **then**

              $B_{pq}(x, y) = B_{pq}(x, y) + e$ if $W_r(x, y) = pv$, where $x = 1, 2, 3$ and $y = 1, 2, 3$;

            end

            $SUM = B_{pq} \otimes W_r$;

            d_{count} = $SUM(\mathrm{mod}16)$; $count = count + 1$;

          else

            goto **Step 3**

          end

        end

      end

    end

    $W_{i+1} = (W_i \times \kappa - 1) \mod 9$, where $gcd\,(\kappa, 9) = 1$;

  end

end

Step 3:Produce Original image $I[M \times N]$ and Data D = (d_1, d_2, ...), where d_i is the 4 bits data

Step 4: End.

Algorithm 16: Data extraction process of IRDHWM

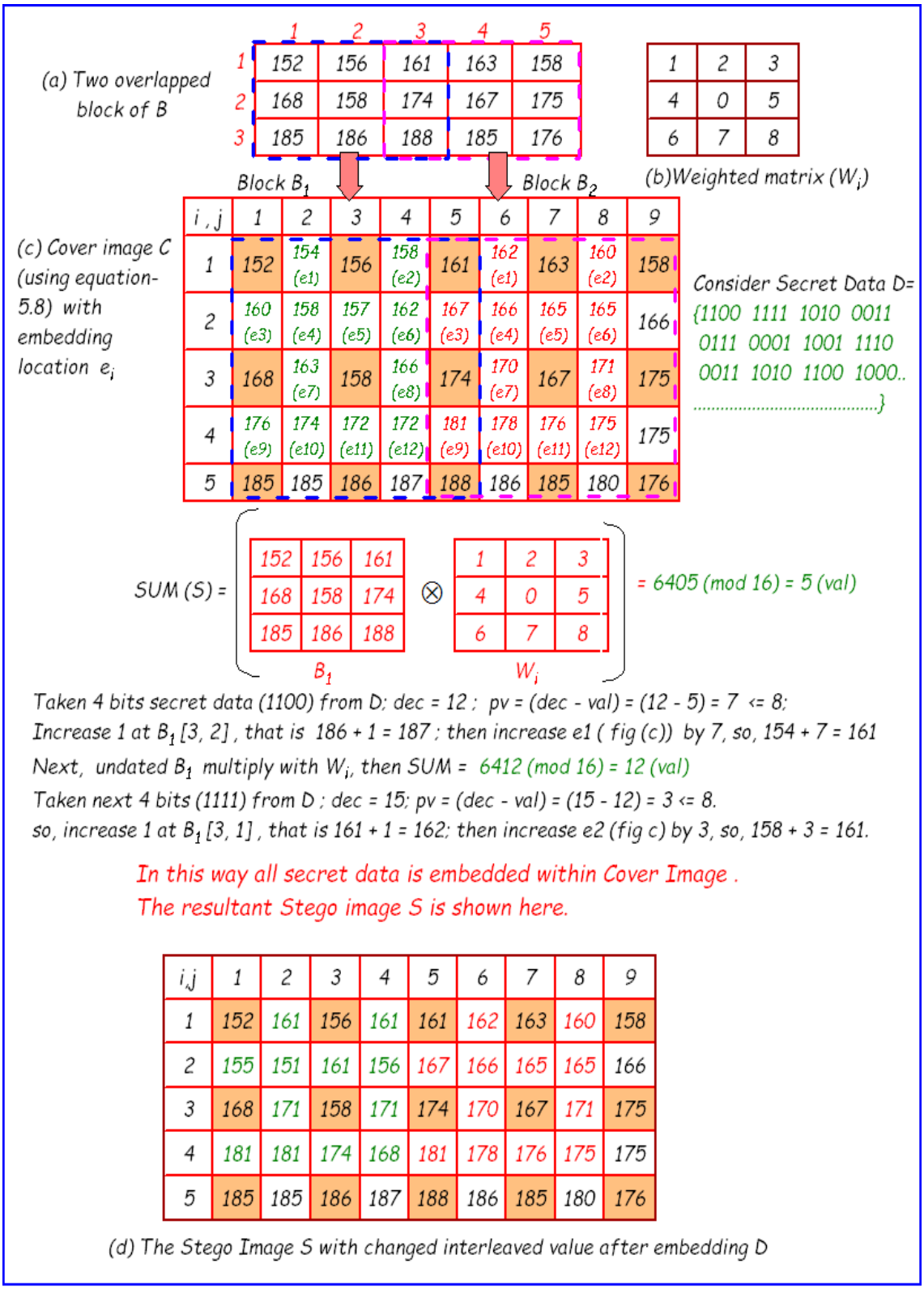


Figure 5.10: Numerical example of data embedding in IRDHWM

$e = 1$ else $e = -1$. Then we update $BI_{(3\times3)}$ by $BI_{(3\times3)} + e$ by updating the pixel value at the position value pv of the weighted matrix. Finally, we extract 4 bits secret data using d_i =

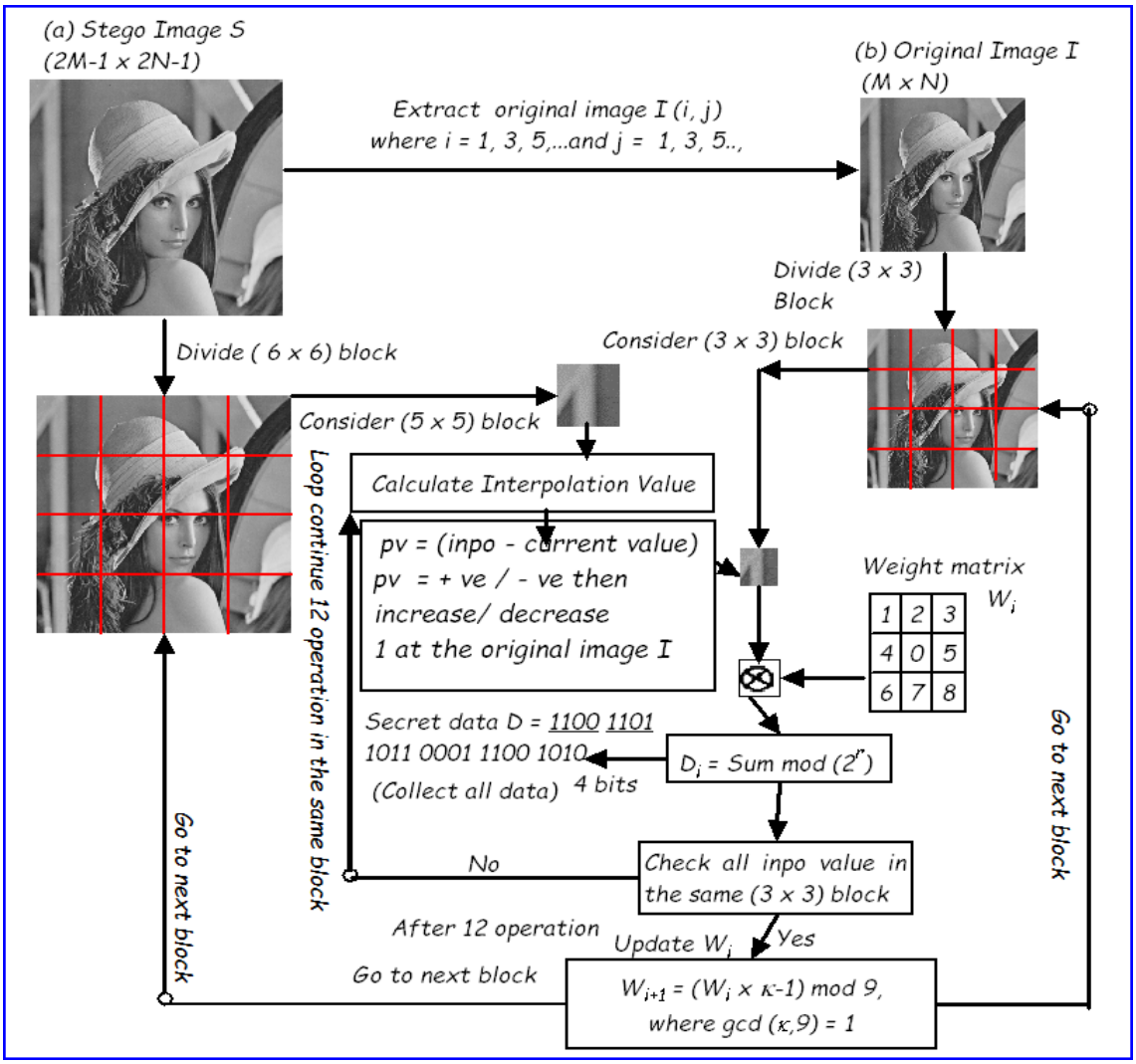


Figure 5.11: Schematic diagram of extraction process in IRDHWM

$(BI_{(3\times3)} \otimes W_i)(\text{mod } 16)$, where $i = 0, 1, 2 \ldots, 12$. The entry-wise-multiplication operation is performed twelve times on the same block and 48 bits secret data is extracted. The schematic diagram of data extraction is shown in Fig 5.10 and numerical illustration is shown in Fig. 5.12. Extraction algorithm is listed in Algorithm 16.

5.3.3 Overflow and Underflow Control

Overflow situation may occur when interpolated pixel value exceeds the maximum intensity value of gray scale image after addition with pv. For example, consider the pixel pair $(254, 255)$. After interpolation it becomes $(254, 254, 255)$. If the pv value 8 is added with 254, it becomes 262 which is greater than 255, this situation is called overflow situation and it may occur during data embedding. When the pv is subtracted from pixel then there may be a chance that under-

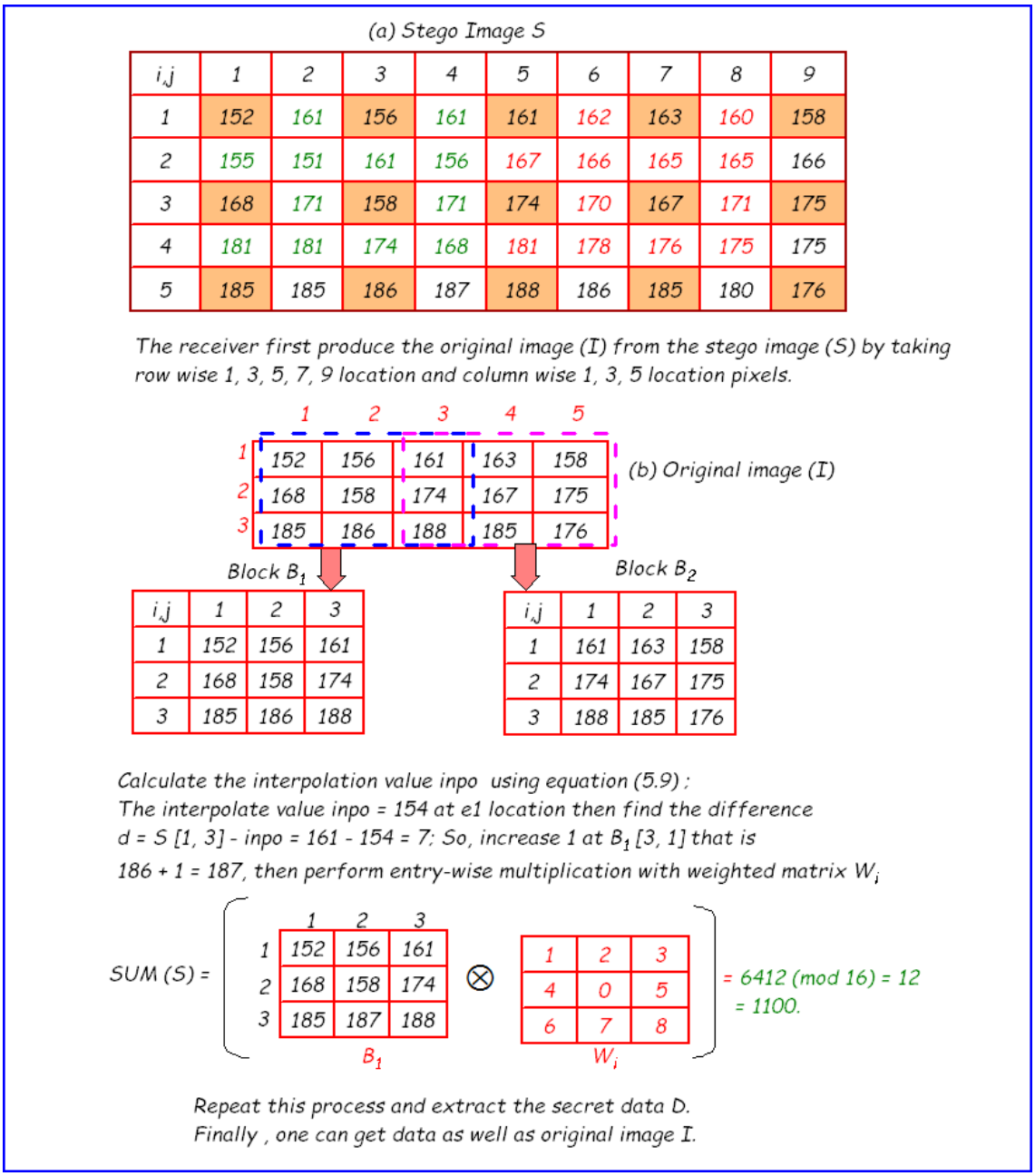


Figure 5.12: Numerical example of data extraction in IRDHWM

flow occurs. For example, consider that the pixel pair is $(2, 4)$, after interpolation it becomes $(2, 3, 4)$. Consider the pv is equal to 4. If 4 is subtracted from 3 then it becomes negative, which is not a valid pixel value. This is an underflow situation. To overcome these situations we can adjust pixel values as follows: It is observed that the maximum value of pv is 8. If we adjust interpolated pixel value at 247 (that means 255 - 8) then it is possible to overcome the overflow

situations.

$$Inpo = \begin{cases} 247 & \text{if } Inpo > 247 \\ 8 & \text{if } Inpo < 8 \end{cases} \tag{5.8}$$

To overcome the underflow situation, we fix interpolated pixel value to 8 when it is below eight. For example, consider $(2, 8, 4)$ pixel values. If $e = -1$ and pv is any value between the range $0 - 8$ then the interpolated pixel never crosses the limit 0.

5.3.4 Experimental Results and Comparisons

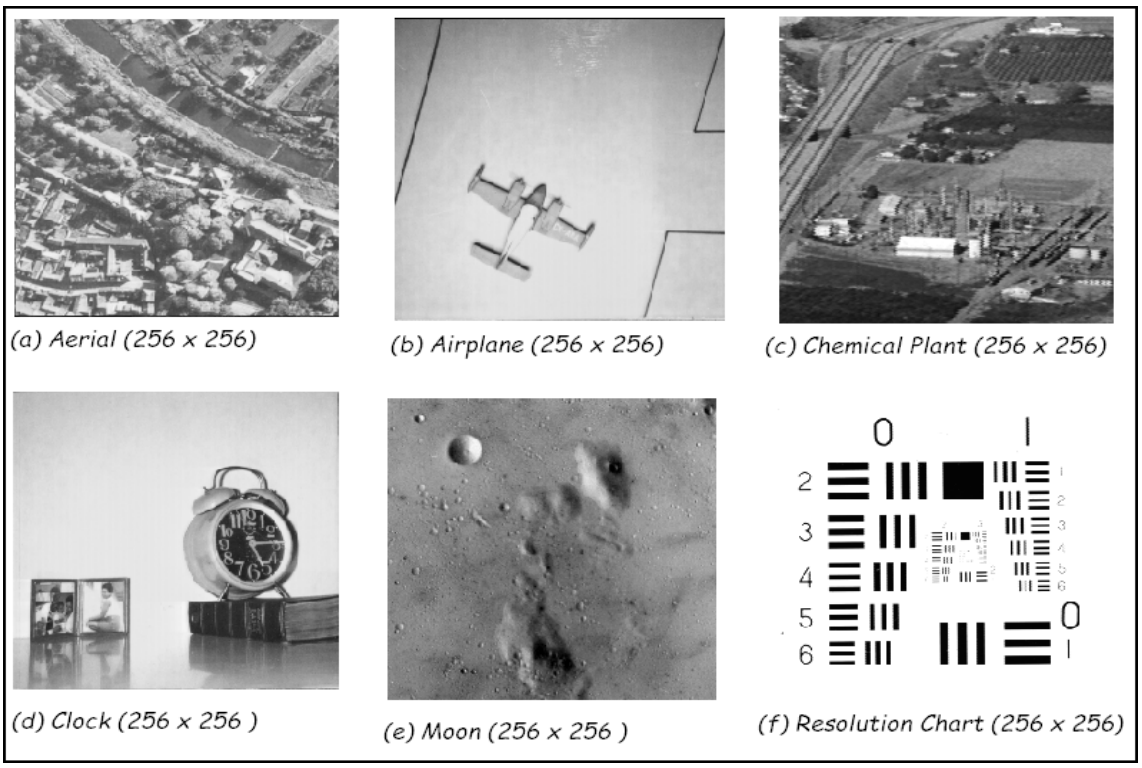


Figure 5.13: Standard cover images of size (256×256) pixel used in IRDHWM

The proposed scheme is tested and verified using gray scale images Lena, Moon Surface, Aerial, Airplane, Clock, Resolution Chart and Chemical Plant which are collected from the USC-SIPI image database, University of Southern California [65] shown in Fig. 5.13. After image interpolation and data embedding, stego image are generated which are shown in Fig. 5.14. In this experiment, we have considered the original image (I) of (256×256) = $65,536$ bytes. After image interpolation the size of cover image (C) and stego image (S) becomes (511×511) = $2,62,121$ bytes. The secret data has been considered as image of size (382×254) = 7,76,224 bits which is shown in Fig. 5.15. The quality of the original and stego

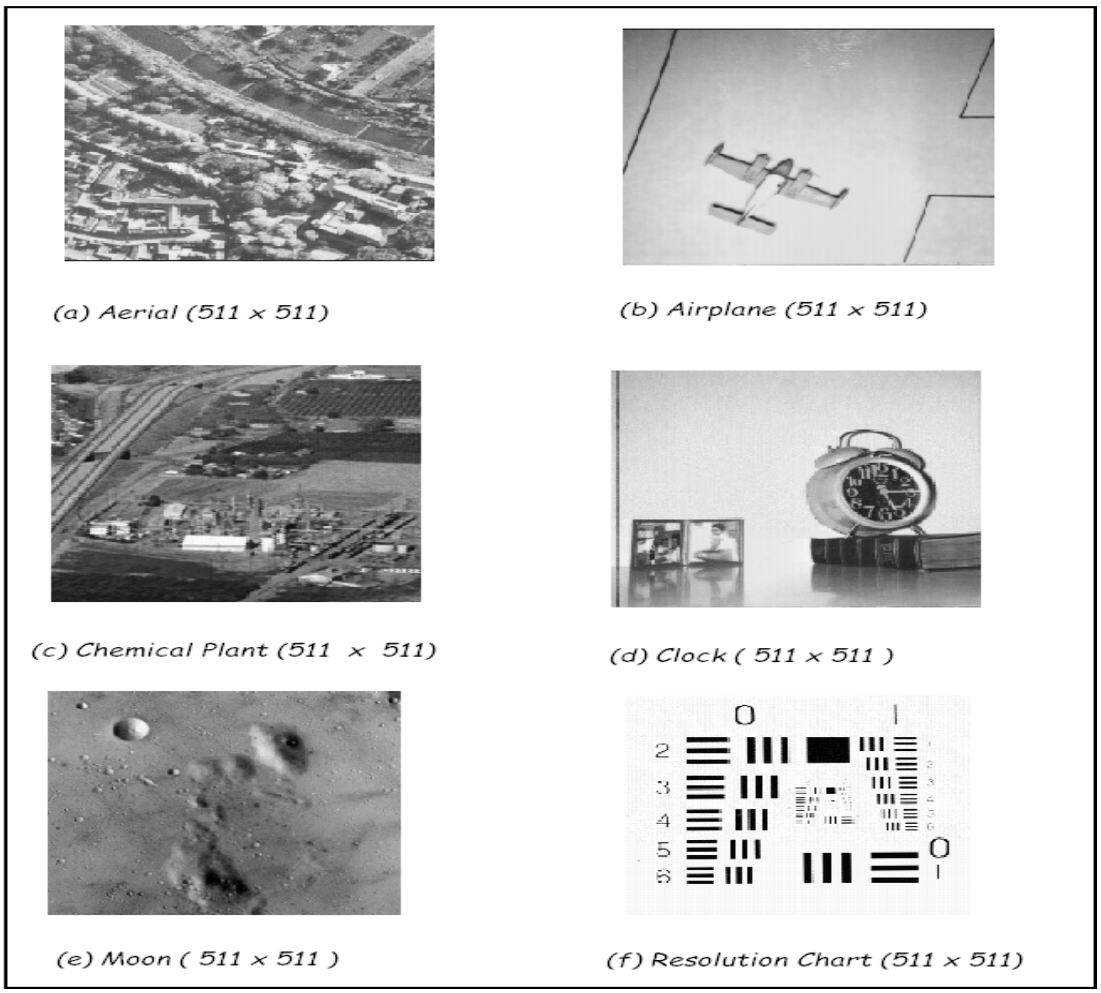


Figure 5.14: Generated stego image after embedding (7,76,224) bits in IRDHWM

image are assessed by the Peak Signal to Noise Ratio $(PSNR)$. The experimental results in terms of PSNR are shown in Table 5.7. Higher values of PSNR between two images indicate better quality and low PSNR demonstrates the opposite.

To calculate payload in terms of bits per pixel (bpp), following equation (5.9) has been used.

$$p = \frac{(\lfloor \frac{M+1}{2} \rfloor - 1) \times (\lfloor \frac{N+1}{2} \rfloor - 1) \times 48}{(2M-1)(2N-1)}, \tag{5.9}$$

here, $M = 256$, $N = 256$, and payload $p = 2.96$ (bpp).

Comparisons with other existing schemes are listed in Table 5.8. The PSNR of Ni et al.'s [51] scheme is 30.88 (dB) which is 7.09 (dB) less and payload is 1.11 (bpp) which is 1.85 units less than IRDHWM scheme. Both PSNR and payload of IRDHWM scheme are higher than other existing schemes shown in Table 5.8. Average PSNR of this scheme is 37.97 (dB) which is

Table 5.7: PSNR (dB) of stego image with data embedding capacity in IRDHWM

PSNR (dB) with data hiding capacity (bits)			
Image I	Capacity(bits)	PSNR	Avg. PSNR
Lena	2,60,096	40.85	37.97
	5,60,000	37.24	
	7,76,224	35.80	
Moon Surface	2,60,096	40.87	37.97
	5,60,000	37.24	
	7,76,224	35.81	
Arial	2,60,096	40.83	37.96
	5,60,000	37.24	
	7,76,224	35.8	
Airplane	2,60,096	40.82	37.96
	5,60,000	37.24	
	7,76,224	35.81	
Clock	2,60,096	40.85	37.96
	5,60,000	37.23	
	7,76,224	35.8	
Resolution	2,60,096	40.85	38.02
	5,60,000	37.41	
	7,76,224	35.81	
Chemical	2,60,096	40.86	37.97
	5,60,000	37.24	
	7,76,224	35.81	

Table 5.8: Comparisons with other existing schemes

Scheme	Average PSNR (dB)	Payload (bpp)
Ni et al.'s [51]	30.88	1.11
Jung et al.'s [26]	33.24	0.96
Lee et al.'s [38]	33.79	1.59
Tang et al. [57]	33.85	1.79
IRDHWM	37.97	2.96

more than existing weighted matrix and interpolation based data hiding schemes which guarantees good visual quality. The payload is very high compared to other existing schemes and it is 2.96 (bpp). To enhance security, weighted matrix has been updated using κ for each new block where κ is a shared secret key.

Figure 5.15: Image cameraman treated as secret information in IRDHWM

5.3.5 Steganalysis and Steganographic Attacks

Steganalysis is an art of discouraging covert communications. Its basic requirement is to determine accurately whether a secret message is hidden in the testing medium or not. To estimate the presence of secret message one has to test stego images through different steganalysis techniques. The perceptibility of the proposed scheme have been assessed through various steganographic attacks including Jeremiah J. Harmsena's Histogram attack, and Brute Force Attack.

5.3.5.1 RS Analysis

In this scheme, stego images are analyzed through RS analysis and results are shown in Table 5.9. It is observed from the table that the values of R_M and R_{-M}, S_M and S_{-M} are nearly equal. Thus rule $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ is satisfied for the stego image. So, the method is secure against RS attack. In this experiment, the ratio of R and S lies between 0.0034 to 0.0065 for the lena stego image. Other values are shown in the Table 5.9.

5.3.5.2 Relative Entropy

The relative entropy has been calculated and shown in Table 5.10. From this table, it is observed that after embedding 776224 bits secret data within lena image, the relative entropy between I and S are 7.4429 and 7.4622 respectively. The difference is 0.0193 unit which is nearer to zero indicating good results.

Table 5.9: RS analysis of stego image of IRDHWM

Image	Secret data (bits)	RS values of stego image				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Lena	260096	22838	22763	11743	11701	0.0034
	560000	22356	22119	15041	15058	0.0068
	776224	21788	21641	17258	17366	0.0065
Moon Surface	260096	24051	24169	14628	14503	0.0063
	560000	23503	23543	16833	16883	0.0022
	776224	22934	22942	18381	18376	0.0010
Arial	260096	19801	19465	11506	11618	0.0043
	560000	20192	19788	14115	14337	0.0182
	776224	19978	20124	15995	16097	0.0069
Airplane	260096	33190	33666	8359	8190	0.0155
	560000	28070	28319	14078	13951	0.0089
	776224	23202	23765	18698	16231	0.0246
Clock	260096	29089	29541	9901	9843	0.0131
	560000	24499	24522	15302	15323	0.0011
	776224	22998	22738	18032	18298	0.0128

Table 5.10: Relative entropy between original image and stego image in IRDHWM

Image	Data(bits)	Entropy (I)	Entropy (S)	Entropy Difference
Lena	260096	7.4429	7.4380	0.0049
	560000		7.4524	0.0009
	776224		7.4622	0.0193
Moon surface	260096	6.6752	6.6866	0.0114
	560000		6.6967	0.0215
	776224		6.7044	0.0292
Arial	260096	7.2988	7.2985	0.0003
	560000		7.3091	0.0103
	776224		7.3185	0.0197
Airplane	260096	6.4568	6.4624	0.0056
	560000		6.5419	0.0851
	776224		6.6587	0.2019
Clock	260096	6.7004	6.7566	0.0562
	560000		6.9253	0.2249
	776224		6.9484	0.248

5.3.5.3 Statistical Analysis

The SD (σ) of before and after data embedding and CC (ρ) of cover and stego images are summarized in Table 5.11. From the table 5.11, it is seen that there is no substantial divergence between the standard deviation of the cover image and the stego image. This study shows that

Table 5.11: Experimental results of SD (σ) and CC (ρ) of IRDHWM

Image	$SD(\sigma)$		CC (ρ)
	Image I	Stego Image S	I & S
Lena	47.8255	47.3553	0.9823
Moon Surface	27.1998	27.4773	0.9895
Arial	45.0844	44.1284	0.9686
Airplane	32.0451	32.3063	0.9924
Clock	57.3003	57.4202	0.9976

the magnitude of change in stego image based on image parameters is small from a cover image. Since the image parameters have not changed much, the method offers good concealment of data and reduces the chance of the secret data being detected. Thus, it indicates a perfectly secure steganographic scheme.

5.3.5.4 Histogram Attack

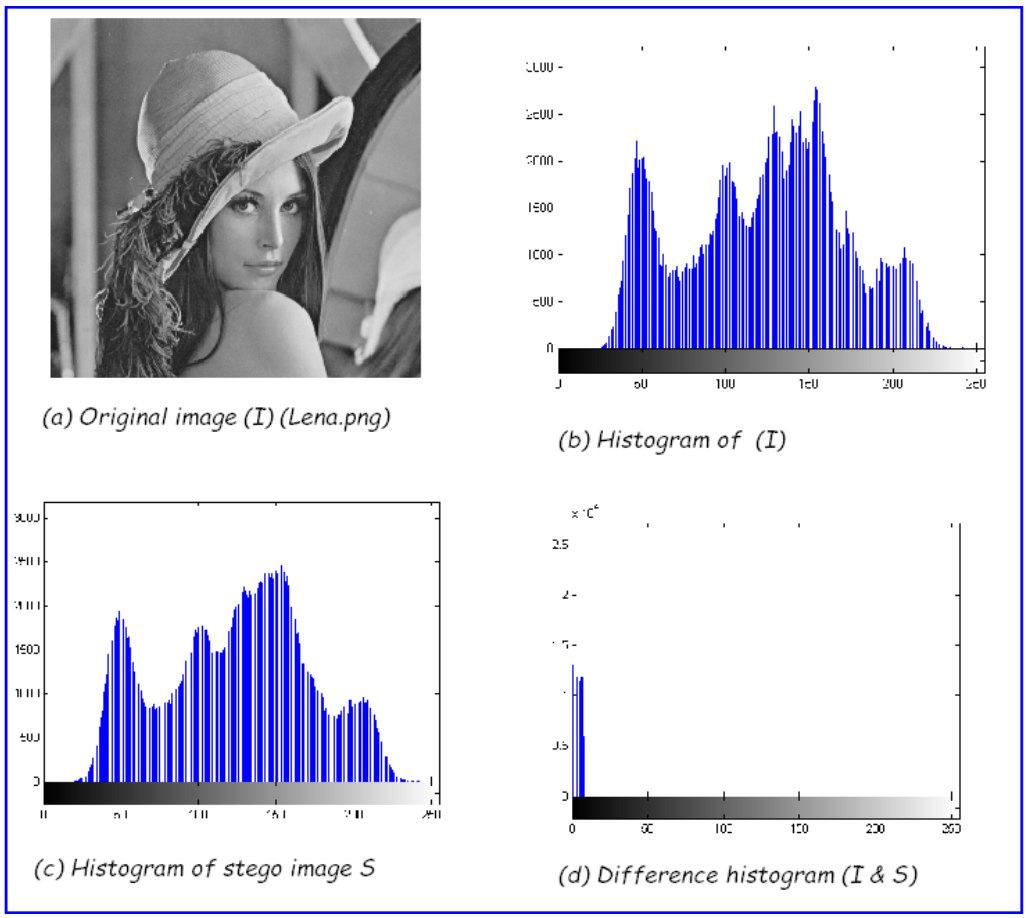


Figure 5.16: Histogram of original image, stego image and their difference in IRDHWM

The histogram of the cover and stego image and their difference histogram has been depicted

on Fig. 5.16. The stego image is produced from cover image employing maximum data hiding. It is observed that the shape of the histogram is preserved after embedding huge secret data. Histogram of cover image is represented as h whereas histogram of stego image is represented as $h^{'}$. The change of histogram can be measured by $D_h = \sum_{m=1}^{255} |h'_m - h_m|$. The difference of the histogram is very small. It is observed that, bins close to zero are more in number and the bins which are away from zero are less in number. This confirms that the quality of the stego image is preserved. There is no step pattern observed which ensures that the proposed method is robust against histogram analysis.

5.3.5.5 Brute Force Attack

The IRDHWM scheme protects secret data by embedding only position value (pv) within dual stego images but it does not embed actual secret data. The weighted matrix is updated for each new block using secret key κ. The strategy is secure to prevent possible malicious attacks. The Fig. 5.17 shows the revelation example where wrong key and wrong weighted matrix are used to revels the hidden message. If the malicious attacker holds the original image and stego image and is fully aware about the scheme, the hidden message still cannot be correctly revealed without knowing the correct secret key and correct weighted matrix. For example, Fig. 5.17 shows stego image derived from Lena image using correct weighted matrix and secret key which are different from that used to construct without knowing the weighted matrix and secret key. The result indicates that the attacker only acquires noise like image when applying incorrect weighted matrix and secret key to reveal the hidden message. Furthermore, the attacker may employ the brute force attack that tries all possible permutation to reveal the hidden message. Maximum possible weighted matrix to embed r bits data length in each block are $(2^{r-1}+1)!$, using $(M \times N)$ original matrix and partitioned (3×3) blocks. Total number of blocks are $(M/3 \times N/3)$ and each block uses unique weighted matrix. So, the number of required trials to reveal the hidden message are $(2^{r-1}+1)! \times (M/3 \times N/3)$. In this scheme, for (256×256) image with $r = 4$, number of trials will be $(3, 62, 880) \times 7, 225$ which is computationally infeasible for current computers. The IRDHWM scheme achieves stronger robustness against several attacks when compared with existing data hiding schemes. Furthermore, the secret information can be retrieved without encountering any loss of data and the original image can be successfully recovered from the stego image with valid keys and weighted matrix.

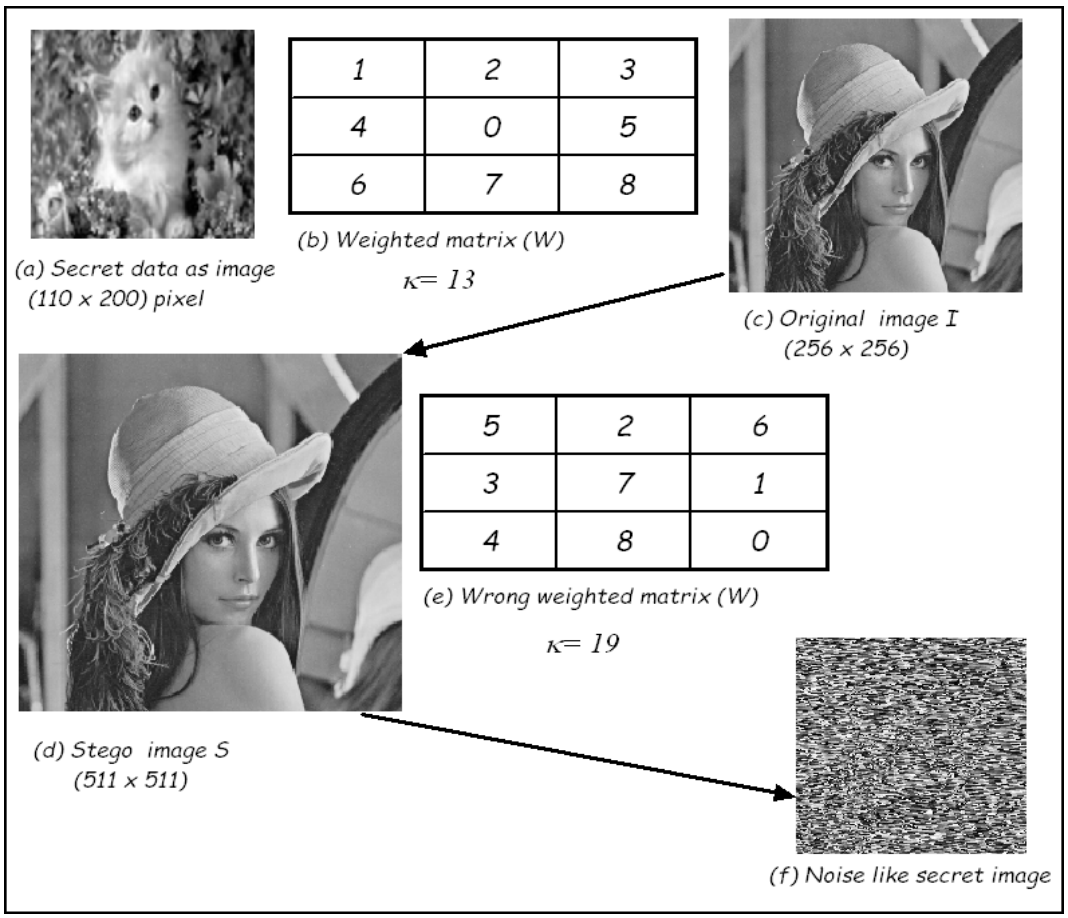


Figure 5.17: Result of Brute Force Attack in IRDHWM

5.4 Interpolated Dual Image based RDH using Weighted Matrix (IDRDHWM)[10]

In this section, a new weighted matrix based reversible data hiding scheme has been proposed using interpolated dual image. The scheme has been divided into two stages. At the first stage, Modular sum of entry-wise-multiplication operation has been performed with each element of original image block and weighted matrix. The data embedding position is identified by the subtraction between the r-bits secret data unit and modular sum. Then store this positional value by updating the original pixel and distribute these original and stego pixel within dual stego image depending on a shared secret key ξ. Repeat the embedding process nine times to embed thirty- six bits at this stage for a single block. In the next stage, interpolate the dual

[10]Published in the proceedings of the International Conference on Computers and Management (ICCM-2015), Jaipur, Rajasthan, December 16-17, 2015, with title *An Efficient Weighted Matrix based Reversible Data Hiding Scheme*

stego image and store more positional value by updating the interpolated pixel value. Repeat the embedding process twenty four times to embed ninety six bits secret data within each pixel block of interpolated dual image at this stage. After hiding one hundred and thirty-two bits secret data within one block of dual image we update the weighted matrix. For i-th block ($i = 1, 2, \ldots, N_B$, where N_B is the number of block), the weighted matrix W_{i+1} can be updated as $W_{i+1} = (W_i \times \kappa - 1) \mod 9$, where $gcd\,(\kappa, 9) = 1$ and κ is a shared secret key.

In the extraction process, first extract the positional value from interpolated dual image and then recover the original data by performing modular sum of entry-wise- multiplication between weighted matrix and original pixel block. Again rearrange the pixel using shared secret ξ from dual image and perform the same extraction operation thirty-three times and extract one hundred thirty-two bits secret data from only one single pixel block of dual image. The IDRDHWM scheme provides average embedding payload 3.46 bits per pixel (bpp) with good visual quality measured by peak signal to noise ratio (PSNR) greater than 35 (dB). The method provides high payload and good visual quality than other existing schemes.

5.4.1 Data Embedding Process

The block diagram of data embedding process is shown in Fig 5.18. The secret message is partitioned into three sections or one can hide three different messages in this scheme. Consider D is the secret message which is divided into three parts D_1, D_2 and D_3. $D_1 = d_1, d_2, \ldots, d_x$; length of $(D_1) = (M \times N)$; $D_2 = d_{x+1}, d_{x+2}, \ldots, d_y$; length of $(D_2) = (y - (x + 1)) + 1 = (\lfloor \frac{(M+1)}{(B-1)} \rfloor - 1) \times (\lfloor \frac{(N+1)}{(B-1)} \rfloor - 1) \times 12$ and $D_3 = d_{y+1}, d_{y+2}, \ldots, d_z$; length of $(D_3) = (z - (y + 1)) + 1 = (\lfloor \frac{(M+1)}{(B-1)} \rfloor - 1) \times (\lfloor \frac{(N+1)}{(B-1)} \rfloor - 1) \times 12$, where d_i = 4 bits decimal value and $i = 1, 2, \ldots, z$; B = image block size. The algorithm for data embedding in Stage 1 is described in Algorithm 17 (I, D_1, W, ξ, κ), where I is the original image and D_1 is the first part of secret data unit D. The corresponding numerical illustration regarding D_1 unit data embedding of Stage 1 is shown in Fig. 5.19. The embedding process are divided in two stages. In each stage data bits are embedded through modular sum operation $SUM(B_i \otimes W)$ with (3×3) image block B_i and (3×3) predefined weighted matrix W for $i = 1, 2, 3, \ldots, N_B$, where N_B is the number of block. After that apply the following equation

$$pv = (b_1 b_2 \ldots b_r)_2 - (SUM(B_i \otimes W) \bmod 2^r), \tag{5.10}$$

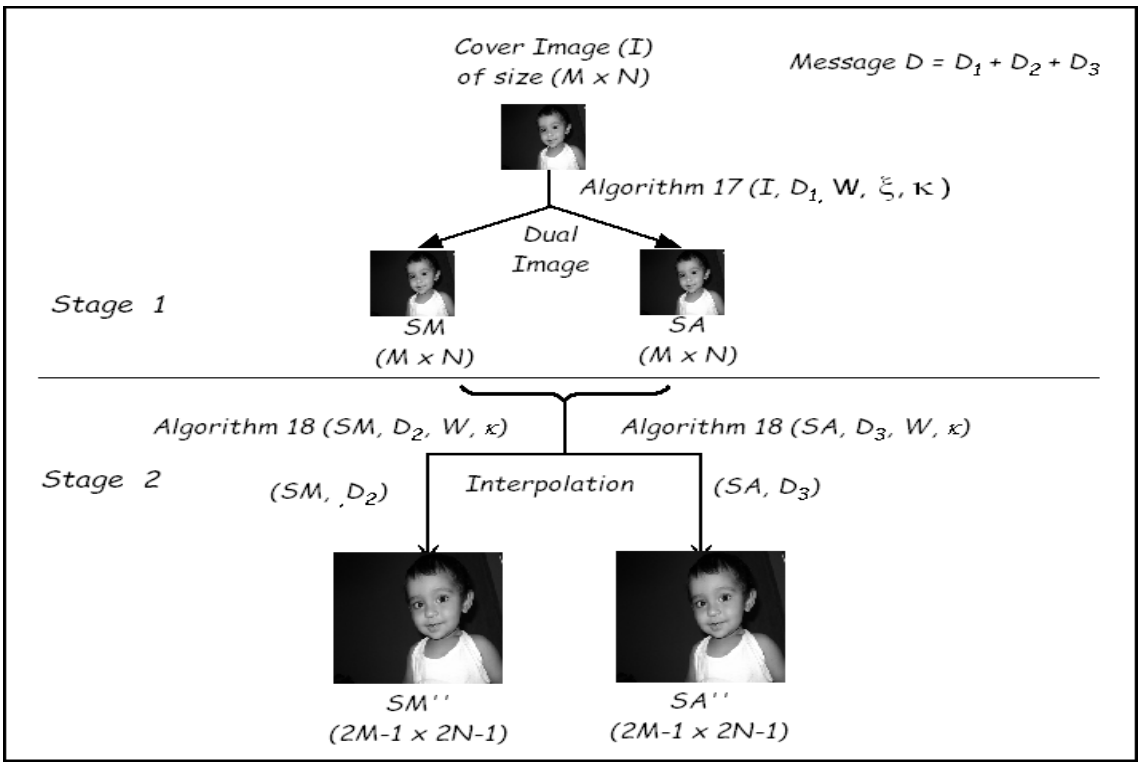


Figure 5.18: Schematic diagram of data embedding process in IDRDHWM

and we get the data embedding position pv using equation (5.10). The pixel in the cover image is incremented or decremented by one depending on the positive or negative sign of position value pv. The pv has been stored by adding or subtracting with existing pixel. Keep original pixel in one image and stego pixel in another image depending on ξ. As a result, the receiver can retrieve the pv by simple subtracting two same location pixel from dual image depending on ξ. Again we apply $SUM(B_i^{'} \otimes W) \mod 2^r$ and store another pv within dual image. Repeat this operation nine times to embed thirty-six bits secret data within a block.

To enhance the embedding capacity (payload), again we consider two stego images SM and SA and apply Stage 2 embedding process. Enlarge the stego images using equation (5.6) by considering SM or SA as image $I_{(M\times N)}$ and generate cover image as $C_{((2M-1)\times(2N-1))}$. So, the $SM^{'}$ is generated from SM using interpolation and $SA^{'}$ is generated from SA. Now, calculate $SUM(B_i \otimes W) \mod 2^r$, where B_i is the (3×3) pixel block of SM. Compute pv using equation (5.10) and then store it within interpolated image $SM^{'}$ and updated stego image $SM^{''}$ is produced. In this way, embed the next part of original secret data D_2 within $SM^{''}$ using Algorithm 18 $(SM^{'}, D_2, W, \kappa)$ and last part of secret data D_3 is embedded within $SA^{''}$ using

Algorithm (I, D_1, W, ξ, κ)

Input: Original Image $I_{(M\times N)}$, Data $D = \{D_1, D_2, D_3\}$, Weighted matrix $W_{(3\times 3)}$, Two shared secret key ξ and κ;

Output: Two stego image $SM_{(M\times N)}$ and $SA_{(M\times N)}$;

Initialize: $Dcount = Kcount$ = 1; sq = 3; $SM = I$; $SA = I$; $W_{11} = W$;

Step 1:

for *(p = 1 to* (M/sq)*)* **do**

  for *(q = 1 to* (N/sq)*)* **do**

    $B_{pq}(3 \times 3) \leftarrow I_{(M\times N)}$;

    for *(i = (sq*(p* $-$ *1))+1 to* $(sq * p)$*)* **do**

      for *(*$j = (sq * (q-1)) + 1$ *to* $(sq * q)$*)* **do**

        $SUM = B_{pq} \otimes W_{pq}$;

        val = $SUM(\mathrm{mod}16)$; pv = (D_{Dcount} - val);

        if *(*$pv > 0$*)* **then**

          if *(*$pv > 8$*)* **then**

            $pv = (16 - pv)$; $e = -1$;

          else

            $e = 1$;

          end

        else

          if *(*$pv <$ - *8)* **then**

            $pv = abs(16 + pv)$; $e = 1$;

          else

            $pv = abs(pv)$; $e = -1$;

          end

        end

        $B_{pq}(x, y) = B_{pq}(x, y) + e$; if $W_{pq}(x, y) = pv$, where $x = 1,\ 2,\ 3$ and $y = 1,\ 2,\ 3$;

        P_o = $I_{(M\times N)}(i, j)$; P_n = $I_{(M\times N)}(i, j)$ + $(pv \times e)$;

        if $(\xi(Kcount) = 1)$ **then**

          $SM_{(M\times N)}(i, j) = P_o$; $SA_{(M\times N)}(i, j) = P_n$;

        else

          $SM_{(M\times N)}(i, j) = P_n$; $SA_{(M\times N)}(i, j) = P_o$;

        end

        $Dcount = Dcount + 1$; $Kcount = Kcount + 1$;

        if $(Kcount > length(\xi))$ **then** $Kcount = 1$;

        if *(*$Dcount > length(D_1)$*)* **then** goto **Step 2**;

      end

    end

    $W_{pq+1} = (W_{pq} \times \kappa - 1)(\mathrm{mod}9)$, where $gcd(\kappa, 9) = 1$;

  end

end

Step 2: Produced $SM_{(M\times N)}$ and $SA_{(M\times N)}$ dual stego images;

Algorithm 17: Data embedding process of Stage 1 in IDRDHWM

the Algorithm 18 $(SA^{'}, D_3, W, \kappa)$ The corresponding numerical example is shown in Fig. 5.20.

Finally, two stego image $SM^{''}$ and $SA^{''}$ are generated which contain a good amount of secret

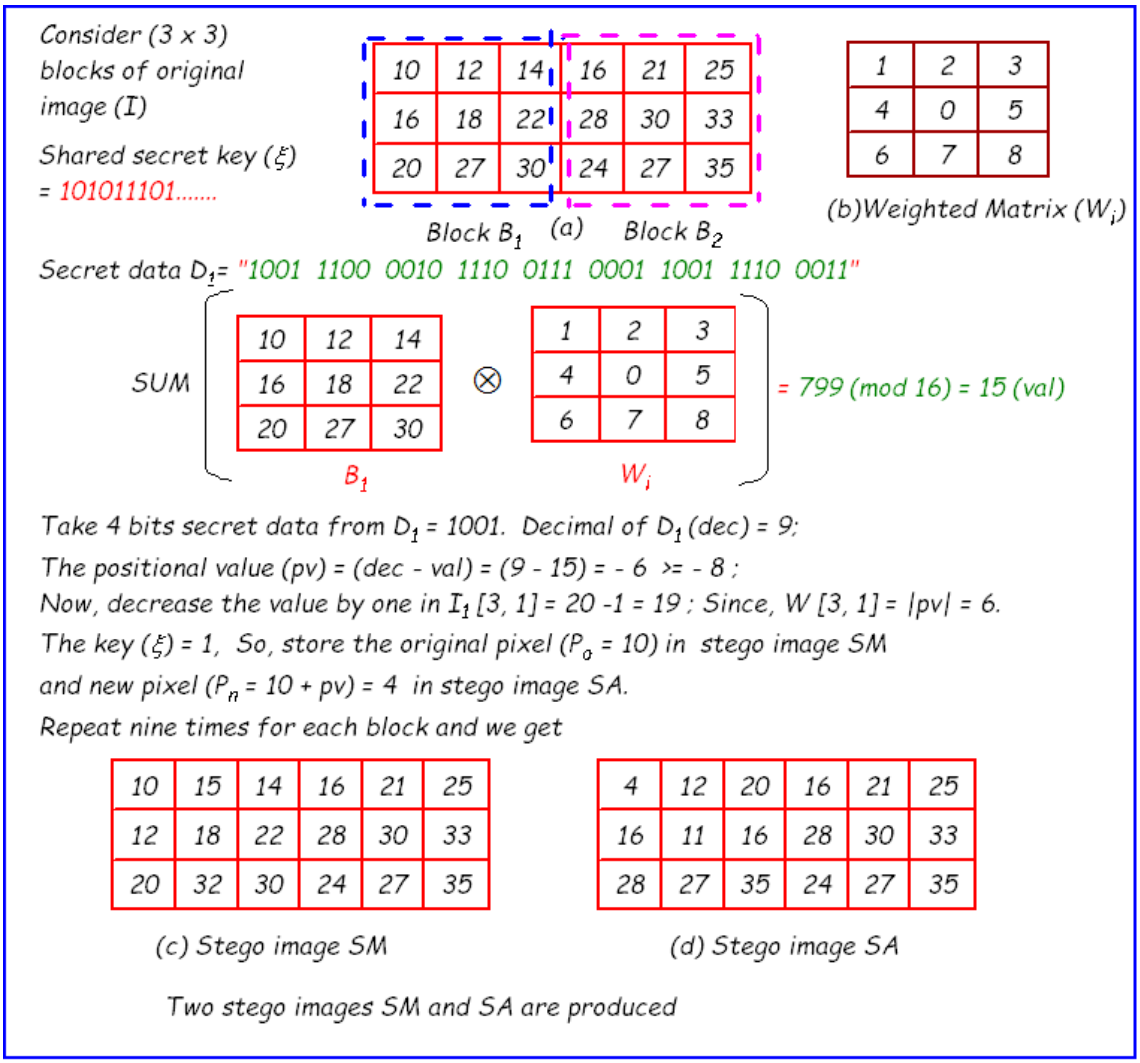


Figure 5.19: Numerical example of data embedding process of Stage 1 in IDRDHWM

data which has been calculated by $(M \times N) \times 4 + 2((\lfloor \frac{(M+1)}{(B-1)} \rfloor - 1) \times (\lfloor \frac{(N+1)}{(B-1)} \rfloor - 1) \times 48)$ bits. The proposed scheme achieve 3.47 (bpp) payload. In addition, security vulnerability exists because an attacker may guess the form of weighted matrix using brute-force attack. In order to enhance the security, weighted matrix W_{i+1} has been changed after every nine operations in Stage 1 and every twelve operation in Stage 2 using the formula $W_{i+1} = (W_i \times \kappa - 1) \mod 9$, where $gcd\ (\kappa, 9) = 1$ and $i = 1, 2, \ldots, N_B$, N_B is the number of blocks.

5.4.2 Data Extraction Process

Two stage data extraction has been performed in this scheme. At first, extract secret data D_2 and D_3 from the dual stego image SM'' and SA'' respectively. The block diagram of extraction process is shown in Fig. 5.21. To extract D_2 from SM'' we use Algorithm 19 $(S, W, \kappa, Dlen)$,

Algorithm (I, D, W, κ)

Input: Original Image $I_{(M\times N)}$, Data $D = \{D_1, D_2, D_3\}$, Weighted matrix $W_{(3\times 3)}$, Shared secret key κ, $count = 1$;

Output: Stego image $S_{(2M-1)\times(2N-1)}$;

Step 1: Generate cover image $C_{(2M-1)\times(2N-1)}$ using equation (5.6) from $I_{(M\times N)}$ and $S = C$;

Step 2: Partition $I_{(M\times N)}$ into (3×3) overlapping blocks ;

Step 3: $W_{11} = W$; $Brow = \lfloor \frac{M+1}{sq-1} \rfloor - 1$; $Bcol = \lfloor \frac{N+1}{sq-1} \rfloor - 1$;

for $p = 1$ *to* $Brow$ **do**
- **for** $q = 1$ *to* $Bcol$ **do**
  - $B_{pq}(3 \times 3) \Leftarrow I_{(M\times N)}$; where sq = 3;
  - **if** *(*$p \neq Brow$ & $q \neq Bcol$*)* **then** $sqr = (2 \times (sq-1) \times p)$; $sqc = (2 \times (sq-1) \times q)$;
  - **if** *(*$p \neq Brow$ & $q = Bcol$*)* **then** $sqr = (2 \times (sq-1) \times p)$; $sqc = (2 \times (sq-1) \times q) + 1$;
  - **if** *(*$p = Brow$ & $q \neq Bcol$*)* **then** $sqr = (2 \times (sq-1) \times p) + 1$; $sqc = (2 \times (sq-1) \times q)$;
  - **if** *(*$p = Brow$ & $q = Bcol$*)* **then** $sqr = (2 \times (sq-1) \times p) + 1$; $sqc = (2 \times (sq-1) \times q) + 1$;
  - **for** $i = (2 \times (sq-1) \times (p-1))$ *to sqr* **do**
    - **for** $j = (2 \times (sq-1) \times (q-1))$ *to sqc* **do**
      - **if** *((*$i \mod 2$ = *0) or (*$j \mod 2$ = *0))* **then**
        - **if** *(*$count \leq length(D)$*)* **then**
          - $SUM = B_{pq} \otimes W_{pq}$;
          - dec = BCD(D_{count}); val = $SUM \pmod{16}$; $pv = dec - val$;
          - **if** *(*$pv > 0$*)* **then**
            - **if** *(*$pv > 8$*)* **then** pv = (16 - pv); e = -1;
            - **else** e = 1;
          - **end**
          - **if** *(*$pv < 0$*)* **then**
            - **if** *(*$pv <$ - *8)* **then** pv = abs(16 + pv); e = 1;
            - **else** pv = abs(pv); e = -1;
          - **end**
          - $B_{pq}(x, y) = B_{pq}(x, y) + e$; if $W_{pq}(x, y)$ = pv, where x = 1, 2, 3 and y = 1, 2, 3;
          - $S_{pq}(i, j) = C_{pq}(i, j) + (pv \times e)$; $count = count + 1$;
        - **else**
          - goto **Step 4**;
        - **end**
      - **end**
    - **end**
  - **end**
  - $W_{pq+1} = (W_{pq} \times \kappa - 1) \bmod 9$, where $gcd(\kappa, 9) = 1$;
- **end**

end

Step 4: Produced stego image $S_{(2M-1)\times(2N-1)}$

Step 5: End.

Algorithm 18: Data embedding process of Stage 2 in IDRDHWM

where $Dlen$ is the length of the secret data D_2 and parameter $S = SM''$. Then call Algorithm 19 $(S, W, \kappa, Dlen)$ with the parameter $S = SA''$ to extract D_3. The corresponding numerical example is shown in Fig. 5.22. To calculate the interpolation value $(Inpo)$, equation (5.7) has been used.

Algorithm ($S, W, \kappa, Dlen$)

Input: Stego Image $S_{(2M-1)\times(2N-1)}$; Weighted matrix $W_{(3\times3)}$; Shared secret key κ;

$count$ = 1; Length of hidden data of D_2 and D_3 separately treated as $(Dlen)$;

Output: Original Image $I_{(M\times N)}$; Secret Data $D = \{D_1, D_2, D_3\}$;

Step 1: i = 1; p = 1;

while *(*$i \leq (2M-1)$*)* **do**

  j = 1; q = 1;

  while *(*$j \leq (2N-1)$*)* **do**

    $I(p,q) = S(i,j)$; $j+2$; $q = q+1$;

  end

  $i = i+2$; $p = p+1$;

end

Step 2: $W_{11} = W$;

$Brow = \lfloor \frac{M+1}{sq-1} \rfloor - 1$; $Bcol = \lfloor \frac{N+1}{sq-1} \rfloor - 1$;

for $p = 1$ *to* $Brow$ **do**

  for $q = 1$ *to* $Bcol$ **do**

    $B_{pq}(3\times3) \Leftarrow I_{(M\times N)}$; where $sq = 3$;

    if *(*$p \neq Brow$ & $q \neq Bcol$*)* **then** $sqr = (2\times(sq-1)\times p)$; $sqc = (2\times(sq-1)\times q)$;

    if *(*$p \neq Brow$ & $q = Bcol$*)* **then** $sqr = (2\times(sq-1)\times p)$; $sqc = (2\times(sq-1)\times q)+1$;

    if *(*$p = Brow$ & $q \neq Bcol$*)* **then** $sqr = (2\times(sq-1)\times p)+1$; $sqc = (2\times(sq-1)\times q)$;

    if *(*$p = Brow$ & $q = Bcol$*)* **then** $sqr = (2\times(sq-1)\times p)+1$; $sqc = (2\times(sq-1)\times q)+1$;

    for *i* = *(*$2\times(sq-1)\times(p-1)$*) to sqr* **do**

      for *j* = *(*$2\times(sq-1)\times(q-1)$*) to sqc* **do**

        if *((*$i \mod 2$ = *0) or (*$j \mod 2$ = *0))* **then**

          if *(count*$\leq$ *Dlen)* **then**

            calculate (**Inpo**) using equation (5.7) at location (i,j) with the help of $S_{(2M-1)\times(2N-1)}$

            if $(Inpo < S(i,j))$ **then**

              $pv = S(i,j) - Inpo$; $e = 1$;

            else

              $pv = Inpo - S(i,j)$; $e = -1$;

            end

            if $pv \neq 0$ **then** $B_{pq}(x,y) = B_{pq}(x,y) + e$; if $W_{pq}(x,y) = pv$, where x = 1,2,3 and y = 1,2,3;

            $SUM = B_{pq} \otimes W_{pq}$; D_{count} = $SUM \pmod{16}$; $count = count + 1$;

          else

            goto **Step 3**

          end

        end

      end

    end

    $W_{pq+1} = ((W_{pq}\times\kappa) \bmod 9)$, where $gcd(\kappa, 9) = 1$;

  end

end

Step 3: Produced $I_{(M\times N)}$ and Data $D = \{D_1, D_2, D_3\}$

Step 4: End.

Algorithm 19: Data extraction process of Stage 2 in IDRDHWM

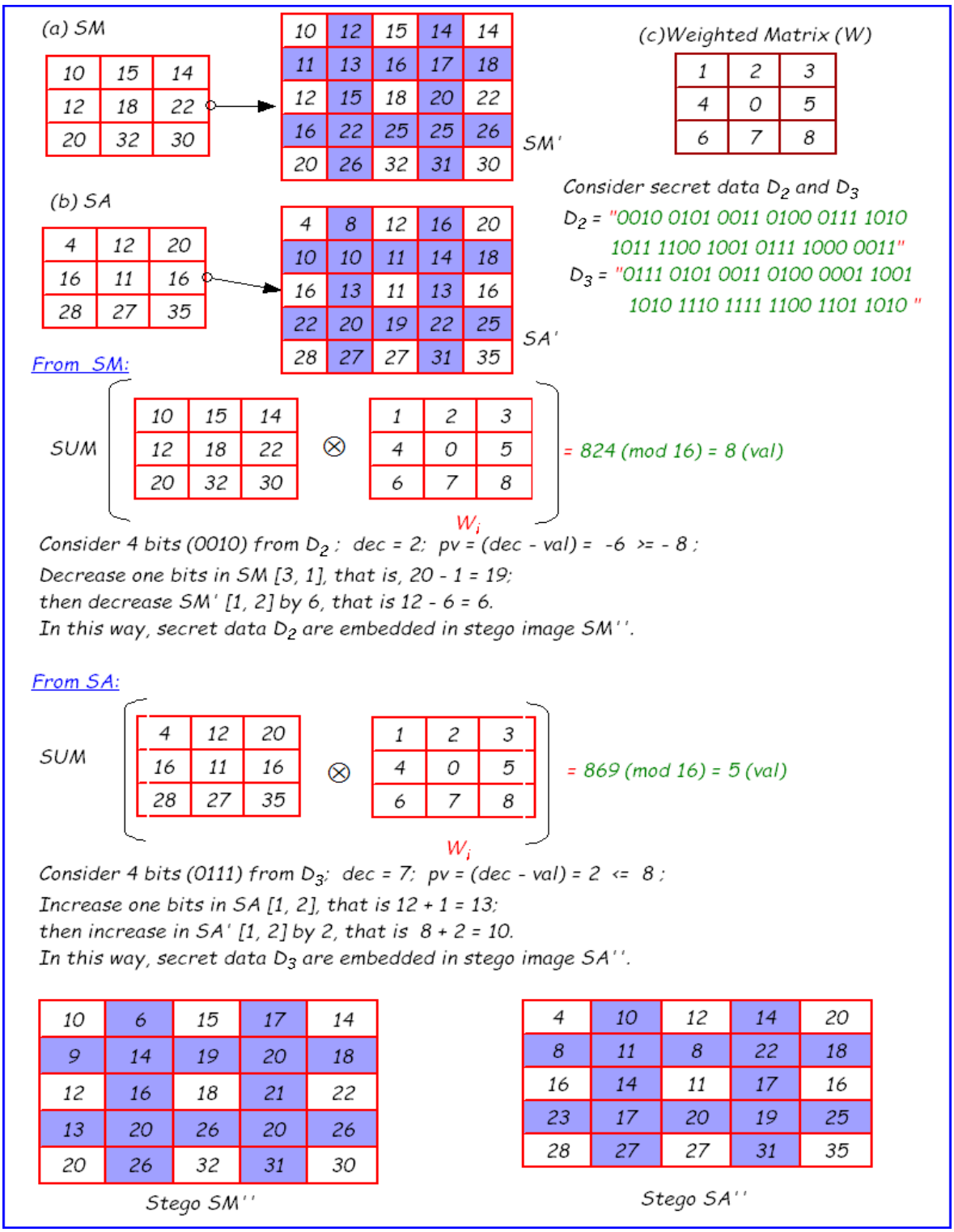


Figure 5.20: Numerical example of data embedding process of Stage 2 in IDRDHWM

Finally, using Algorithm 20 $(SM,\ SA,\ W,\ Dlen,\ \xi,\ \kappa)$, where $SM = SM^{'}$, and $SA = SA^{'}$), data D_1 has been extracted and original image is recovered. The entire secret message

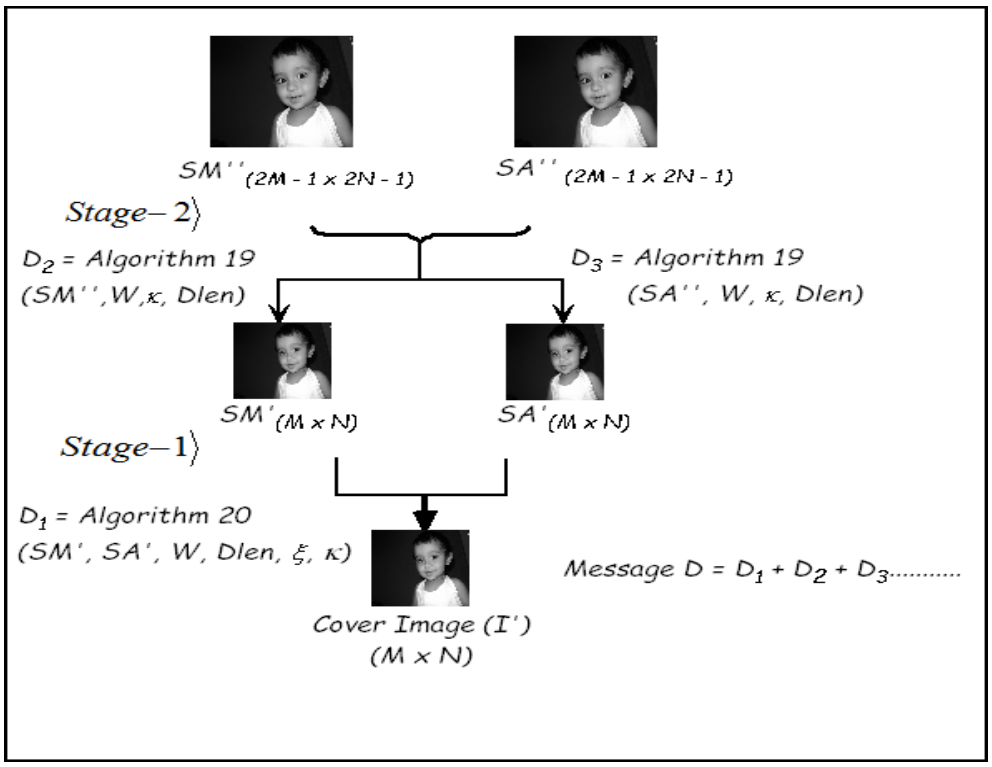


Figure 5.21: Schematic diagram of data extraction process in IDRDHWM

(D_1, D_2 and D_3) is extracted successfully. The corresponding numerical example is shown in Fig. 5.23. The proposed scheme improves the data embedding capacity and achieves reversibility.

5.4.3 Overflow and Underflow Control

Overflow will be occur when the updated pixel value exceed the range of the gray value, that is 255. Underflow may be caused when updated pixel show any negative values after subtraction the pv. To overcome these overflow and underflow problem it is possible to adjust pixel value before data embedding. The new pixel (P_n) is generated by adding or subtracting pv with original pixel (P_o). Now, store P_n pixel in one image and P_o in another image depending on ξ. So, P_o never falls on overflow or underflow situation but P_n may falls in overflow or underflow situation. For example, consider P_o = 248, pv = 8 and e = 1, then $P_n = P_o + pv$ = (248 + 8) = 256, which is greater than the maximum gray level 255. So, overflow situation arises. Similarly, consider P_o = 6, pv = 7 and e = -1, then $P_n = P_o - pv = (6 - 7) = -1$, which is less than 0 which means underflow occurs. To overcome this overflow-underflow situation, adjust P_n as

Algorithm $(SM, SA, W, Dlen, \xi, \kappa)$

Input: Two stego image $SM''_{(M\times N)}$ and $SA''_{(M\times N)}$; Weight matrix $W_{(3\times 3)}$, $Dlen$ is the data length of D_1, Two shared secret key ξ and κ;

Output: Cover Image $I'_{(M\times N)}$, Data $D = \{D_1, D_2, D_3\}$;

Initialize:$Dcount = Kcount = 1$; $sq = 3$; DM is a matrix hold the pv value; $I' = SM''$; $W_{11} = W$;

Step 1:

for *(*$p = 1\ to\ (M/sq)$*)* **do**
  for *(*$q = 1\ to\ (N/sq)$*)* **do**
    for *(*$i = (sq * (p-1)) + 1\ to\ (sq * p)$*)* **do**
      for *(*$j = (sq * (q-1)) + 1\ to\ (sq * q)$*)* **do**
        if *(*$\xi(Kcount) = 1$*)* **then**
          $P_o = SM''_{(M\times N)}(i,j)$; $P_n = SA''_{(M\times N)}(i,j)$; $I'_{pq}(i,j) = P_o$;
        else
          $P_o = SA''_{(M\times N)}(i,j)$; $P_n = SM''_{(M\times N)}(i,j)$; $I'_{pq}(i,j) = P_o$;
        end
        $DM_{pq}(i,j) = (P_n - P_o)$;
      end
      $Kcount = Kcount + 1$;
      if *(*$Kcount > length(\xi)$*)* **then**
        $Kcount = 1$;
      end
    end
  end
end

Step 2:

for *(*$p = 1\ to\ (M/sq)$*)* **do**
  for *(*$q = 1\ to\ (N/sq)$*)* **do**
    $B_{pq}(3\times 3) = I'_{(M\times N)}$;
    for *(*$i = (sq * (p-1)) + 1\ to\ (sq * p)$*)* **do**
      for *(*$j = (sq * (q-1)) + 1\ to\ (sq * q)$*)* **do**
        if *(*$DM_{pq}(i,j) > 0$*)* **then** $pv = DM_{pq}(i,j)$; $e = 1$;
        if *(*$DM_{pq}(i,j) \leq 0$*)* **then** $pv = abs(DM_{pq}(i,j))$; $e = -1$;
        $B_{pq}(x,y) = B_{pq}(x,y) + e$ if $W_{pq}(x,y) = pv$, where $x = 1,\ 2,\ 3$ and $y = 1,\ 2,\ 3$;
        $SUM = B_{pq} \otimes W_{pq}$;
        $D'_{Dcount} = SUM \pmod{16}$; $pv = D_{Dcount} - val$;
        $Dcount = Dcount + 1$; **if** *(*$Dcount > Dlen$*)* **then** goto **Step 3**;
      end
    end
    $W_{pq+1} = W_{pq} \times \kappa \pmod 9$, where $gcd(\kappa, 9) = 1$;
  end
end

Step 3: Produced $I'_{(M\times N)}$ and $D = \{D_1, D_2, D_3\}$;

Algorithm 20: Data extraction process of Stage 1 in IDRDHWM

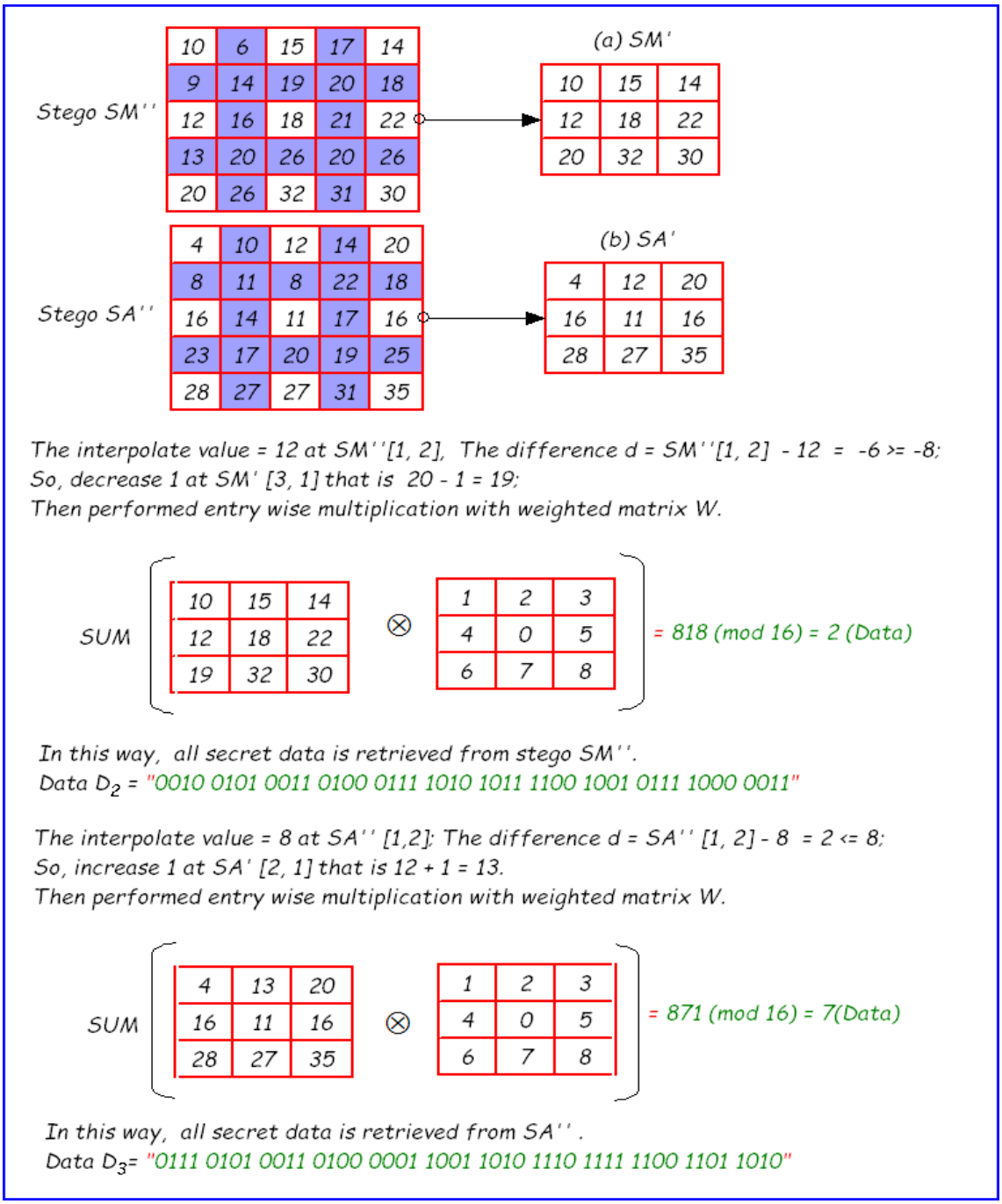


Figure 5.22: Numerical example of data extraction process of Stage 2 in IDRDHWM

follows:

$$P_n = \begin{cases} 247 + (pv \times e), & \text{if } P_o > 247 \\ 8 + (pv \times e), & \text{if } P_o < 8 \\ P_o + (pv \times e), & \text{otherwise} \end{cases} \tag{5.11}$$

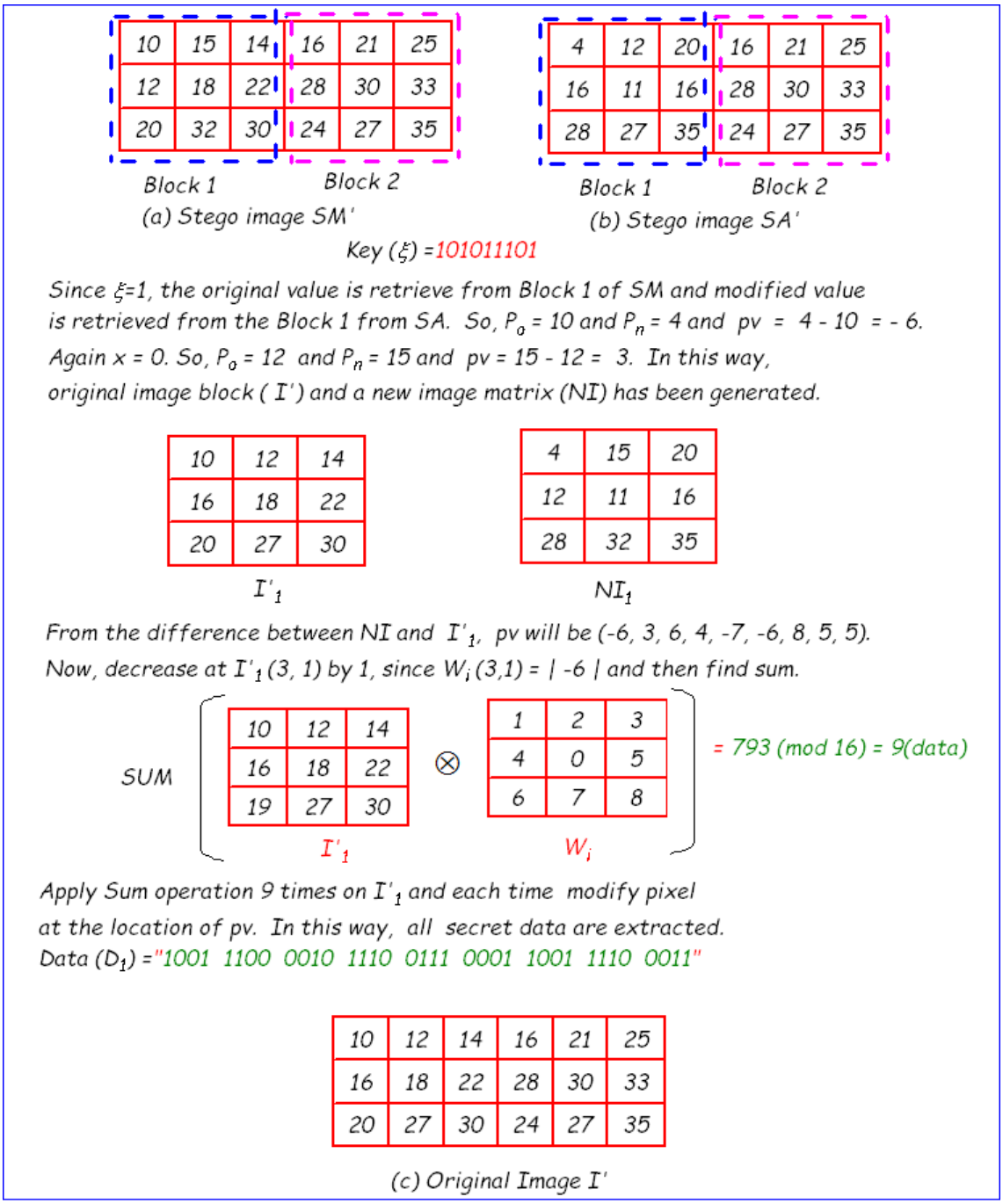


Figure 5.23: Numerical example of data extraction process of Stage 1 in IDRDHWM

The receiver can easily found P_o and P_n by using the key ξ. During extraction of pv follow the equation in case of overflow and underflow situation.

$$pv = \begin{cases} P_n - 247, & \text{if } P_o > 247 \\ P_n - 8, & \text{if } P_o < 8 \\ P_n - P_o, & \text{otherwise} \end{cases} \tag{5.12}$$

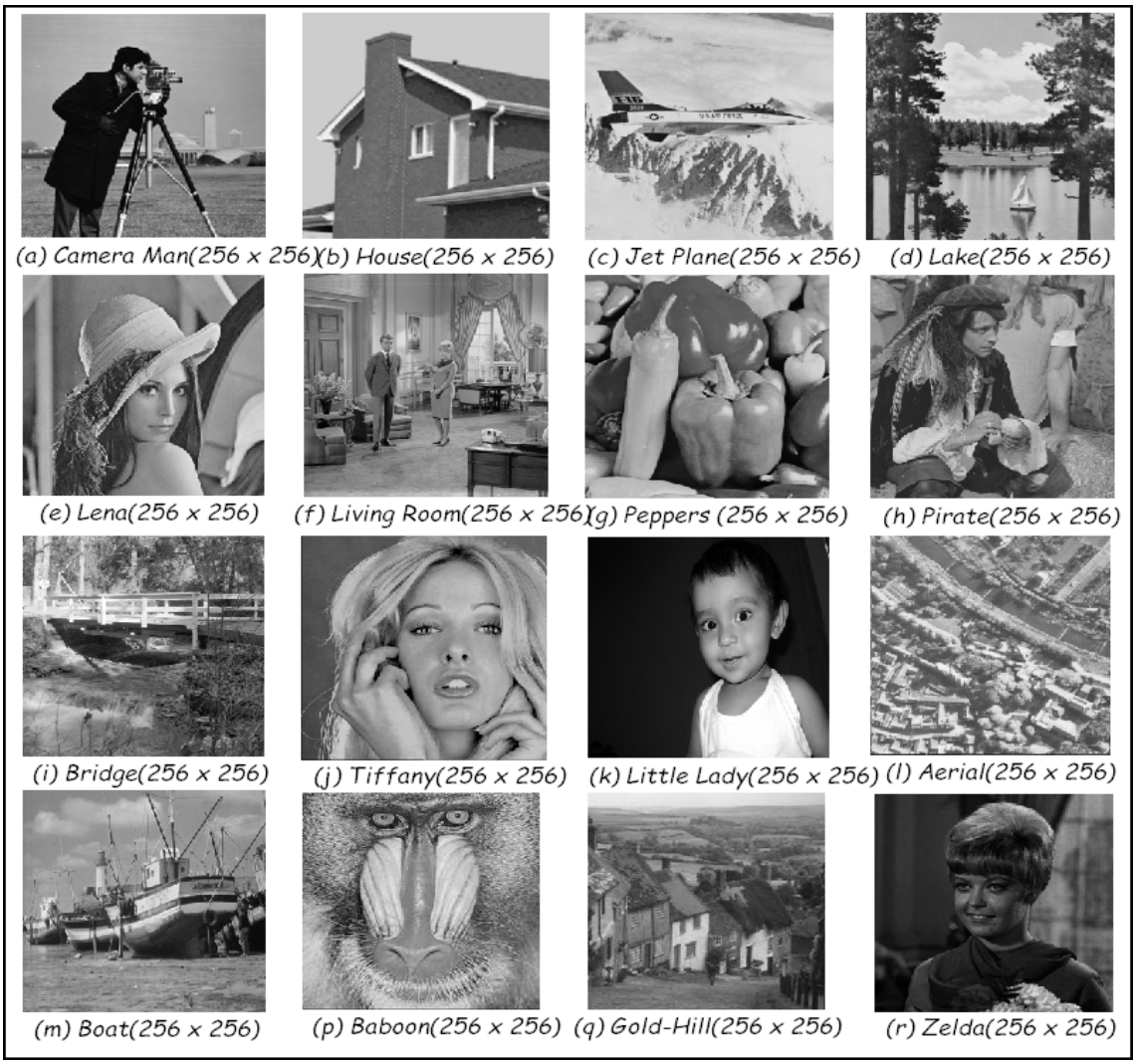


Figure 5.24: Standard cover images are used in IDRDHWM of size (256×256)

5.4.4 Experimental Results and Comparisons

The developed algorithms: data embedding and extraction are implemented in MATLAB Version 7.6.0.324 (R2008a). Here, the distortion is assessed by means of two factors namely, Mean Square Error (MSE) and Peak Signal to Noise Ratio $(PSNR)$. The number of bits are embedded in Stage 1 which are calculated by the following formula.

$$U = \frac{M}{B} \times \frac{N}{B} \times (B^2 \times r), \tag{5.13}$$

where, M and N represent the size of the input image, r represents the number of bits which are to be hidden by each operation, B is the block size. Again number of bits are embedded in Stage 2 are calculated by

$$V = 2 \times (\lfloor \frac{(M+1)}{(B-1)} \rfloor - 1) \times (\lfloor \frac{(N+1)}{(B-1)} \rfloor - 1) \times 12 \tag{5.14}$$

Figure 5.25: Generated stego images of size (511 $\times$ 511) in IDRDHWM

Now, total number of bits embedded within interpolated dual stego image are $U + V$. The payload has been calculated using the following equation.

$$p = \frac{(U+V)}{s \times (2M-1) \times (2N-1)}, \tag{5.15}$$

Consider $M = 256$, $N = 256$, $r = 4$ and $s = 2$. So, $p = 3.46$ (bpp). The input images which are used for this experiment are shown in Fig. 5.24. After embedding 18,12,544 secret bits dual stego images are generated which are shown in Fig. 5.25.

The analysis in terms of PSNR of cover image and stego image shows reasonable good results which is shown in Table 5.12. Higher the values of PSNR between two images, better the quality of the stego image and very similar to the cover image where as low PSNR demonstrates the opposite. The Table 5.13 presents the comparison of the proposed method with existing dual

Table 5.12: PSNR (dB) of stego images after embedding secret data through IDRDHWM

Input Image ($I_(256 \times 256)$)	Capacity Bits	PSNR (I vs SM)	PSNR (I vs SA)	Average PSNR
Cameraman	1248000	36.25	36.35	35.46
	1380096	35.51	35.66	
	1680448	35.05	35.19	
	1812544	34.52	34.67	
Lena	1248000	36.27	36.30	35.39
	1380096	35.52	35.64	
	1680448	35.03	35.16	
	1812544	34.51	34.67	
Peppers gray	1248000	36.27	36.32	35.40
	1380096	35.51	35.67	
	1680448	35.07	35.11	
	1812544	34.58	34.67	
Pirate	1248000	36.24	36.30	35.38
	1380096	35.51	35.64	
	1680448	35.02	35.19	
	1812544	34.53	34.61	
Tiffany	1248000	36.23	36.31	35.38
	1380096	35.52	35.64	
	1680448	35.04	35.15	
	1812544	34.51	34.63	
Little leady	1248000	36.26	36.34	35.40
	1380096	35.59	35.62	
	1680448	35.03	35.18	
	1812544	34.52	34.68	
Boat	1248000	36.27	36.37	35.39
	1380096	35.52	35.61	
	1680448	35.01	35.15	
	1812544	34.57	34.61	
Baboon	1248000	36.28	36.31	35.38
	1380096	35.54	35.60	
	1680448	35.00	35.13	
	1812544	34.50	34.68	
Gold Hill	1248000	36.26	36.32	35.39
	1380096	35.57	35.62	
	1680448	35.01	35.15	
	1812544	34.51	34.65	
Zelda	1248000	36.22	36.30	35.37
	1380096	35.56	35.67	
	1680448	35.04	35.11	
	1812544	34.53	34.63	

based schemes. It is observed that the average PSNR of the stego images of IDRDHWM method is around 35 (dB) and payload is 3.46 (bpp). The payload of dual image based scheme proposed by Chang et al. [10] achieves 1.53 (bpp) only. The payload of proposed scheme is higher than the existing scheme proposed by Chang et al.'s [5] [6], Lee et al.'s [34] and Chang et al. [10], Qin et al. [52] and Lu et al. [45]. So, in terms of payload IDRDHWM method is extremely high (3.46 bpp) compared to 1 (bpp) of existing schemes but the PSNR is slightly dropped because of the high payload.

5.4.5 Steganalysis and Steganographic Attacks

Here, steganalysis has been performed through the RS analysis and statistical analysis. The results of relative entropy, Histogram Attack and Brute Force Attack are presented below.

5.4.5.1 RS analysis

The results of RS analysis has been given in Table 5.14 and 5.15. It is observed that the values of R_M and R_{-M}, S_M and S_{-M} are nearly equal. Thus rule $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ are satisfied for the stego image in this scheme. So, the proposed method is secure against RS attack.

5.4.5.2 Relative Entropy

Relative entropy values are listed in Table 5.16. Here, difference column shows the difference of relative entropy between original image and stego image. The entropy difference belongs to 0.001 to 0.1 which is very small.

5.4.5.3 Statistical Analysis

The security of the proposed scheme has been analyzed using various known attacks. The SD (σ) of before and after data embedding and CC (ρ) of cover and stego images are calculated and summarized in Table 5.17. It is observed that there is no significant difference between the standard deviation of the cover image and the stego images. This study shows that the magnitude of change in stego images based on image parameters is small from a cover image. Since the image parameters have not changed much, the method offers a good concealment of

Table 5.13: Comparison of IDRDHWM scheme with existing methods

Methods	Measure	Images				
		Lena	Peppers	Boat	Goldhill	Zelda
Chang et al. [5]	PSNR(1)	45.12	45.14	45.12	45.13	45.13
	PSNR(2)	45.13	45.15	45.13	45.14	45.11
	Avg. PSNR	45.13	45.15	45.13	45.14	45.12
	Payload (bpp)	1	0.99	1	1	0.99
Chang et al. [6]	PSNR(1)	48.13	48.11	48.13	48.13	48.15
	PSNR(2)	48.14	48.14	48.12	48.15	48.13
	Avg. PSNR	48.14	48.13	48.13	48.14	48.14
	Payload (bpp)	1	1	1	1	1
Lee et al. [36]	PSNR(1)	51.14	51.14	51.14	51.14	51.14
	PSNR(2)	54.16	54.17	54.16	54.16	54.17
	PSNR(Avg.)	52.65	52.66	52.65	52.65	52.66
	Capacity(bpp)	0.75	0.75	0.75	0.75	0.75
Lee et al. [34]	PSNR(1)	49.76	49.75	49.76	49.77	49.77
	PSNR(2)	49.56	49.56	49.57	49.57	49.58
	Avg. PSNR	49.66	49.66	49.67	49.67	49.68
	Payload (bpp)	1.07	1.07	1.07	1.07	1.07
Chang et al. [10]	PSNR(1)	39.89	39.94	39.89	39.9	39.89
	PSNR(2)	39.89	39.94	39.89	39.9	39.89
	Avg. PSNR	39.89	39.94	39.89	39.9	39.89
	Payload (bpp)	1.53	1.52	1.53	1.53	1.53
Qin et al. [52]	PSNR(1)	52.11	51.25	51.11	52.11	52.06
	PSNR(2)	41.34	41.52	41.57	41.34	41.57
	Avg. PSNR	46.72	46.39	46.84	46.72	46.82
	Payload (bpp)	1.16	1.16	1.16	1.16	1.16
Lu et al. [45]	PSNR(1)	49.20	49.19	49.20	49.23	49.19
	PSNR(2)	49.21	49.21	49.21	49.18	49.21
	Avg. PSNR	49.21	49.20	49.21	49.21	49.20
	Payload (bpp)	1	0.99	1	1	0.99
Lu et al. [46]	PSNR(1)	49.89	49.89	49.89	49.90	49.89
	PSNR(2)	52.90	52.92	52.90	52.90	52.88
	Avg. PSNR	51.40	51.41	51.40	51.40	51.39
	Payload (bpp)	1	0.99	1	1	0.99
IDRDHWM	PSNR(1)	35.33	35.35	35.34	35.33	35.33
	PSNR(2)	35.44	35.44	35.43	35.43	35.42
	Avg. PSNR)	35.38	35.39	35.38	35.38	35.37
	Payload (bpp)	**3.46**	**3.46**	**3.46**	**3.46**	**3.46**

Table 5.14: RS analysis of stego images SM'' in IDRDHWM

Image	Data	SM''				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Cameraman	1248000	23623	23524	15777	15619	0.0065
	1380096	23045	23369	16776	16500	0.0151
	1680448	22406	22778	18072	17794	0.0161
	1812544	22507	22841	18303	17817	0.0201
Lena	1248000	22169	22312	15076	14906	0.0084
	1380096	22304	21925	15728	16065	0.0188
	1680448	21221	21432	17057	16949	0.0083
	1812544	21910	21855	17457	17593	0.0049
Baboon	1248000	21072	20968	15922	15944	0.0034
	1380096	21044	21174	16341	16182	0.0077
	1680448	21181	21217	17430	17452	0.0015
	1812544	21215	21120	17296	17410	0.0054

Table 5.15: RS analysis of stego images SA'' in IDRDHWM

Image	Data	SA''				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Cameraman	1248000	23834	23305	15642	15825	0.0180
	1380096	23237	23120	16474	16532	0.0044
	1680448	22579	22559	17891	17837	0.0018
	1812544	22535	22515	17929	17940	0.0007
Lena	1248000	22399	22114	14860	15053	0.0128
	1380096	22241	22140	15817	15894	0.0047
	1680448	21610	21330	16905	17185	0.0145
	1812544	21858	21896	17430	17492	0.0025
Baboon	1248000	20963	21120	16036	15861	0.0090
	1380096	21196	21109	16211	16283	0.0043
	1680448	20967	21319	17462	17273	0.0141
	1812544	21267	21082	17294	17500	0.0101

data and reduces the chance of the secret data being detected. Thus, it indicates a perfectly secure steganographic system.

5.4.5.4 Histogram Attack

Fig. 5.26 shows the original image, dual stego image and their histogram. The difference of these Histogram is shown in Fig. 5.27. It is observed that the shape of the histogram is preserved after embedding 18,12,544 bits secret data. It is observed that, bins close to zero are more in numbers and the bins which are away from zero are less in numbers. This preserve the quality

Table 5.16: Relative entropy of stego images SM'' and SA'' in IDRDHWM

Image	Data	SM''		SA''	
		Entropy	Difference	Entropy	Difference
Cameraman	1248000	7.0572	0.04	7.1143	0.0227
	1380096	7.1220	0.0148	7.1143	0.0295
	1680448	7.1547	0.1375	7.1458	0.1192
	1812544	7.1555	0.1383	7.1452	0.128
Lena	1248000	7.4491	0.0224	7.4494	0.0062
	1380096	7.4550	0.0283	7.4562	0.0147
	1680448	7.4653	0.0386	7.4622	0.0355
	1812544	7.4668	0.0401	7.4654	0.0387
Baboon	1248000	7.2393	0.0471	7.2394	0.0472
	1380096	7.2438	0.0561	7.2437	0.0515
	1680448	7.2471	0.0549	7.2461	0.0539
	1812544	7.2469	0.0547	7.2475	0.0553

Table 5.17: The result of SD (σ) and CC (ρ) of stego image in IDRDHWM

Image	SD (σ)			CC (ρ)		
	I	SM	SA	I & SM	I & SA	SM & SA
Cameraman	61.58	61.70	61.67	0.9986	0.9988	0.9979
Lena	47.83	47.96	47.94	0.9977	0.9980	0.9957
Baboon	38.37	38.48	38.50	0.9965	0.9968	0.9933

of stego image. There is no step pattern observed, which ensures the proposed method is robust against histogram attack.

5.4.5.5 Brute Force Attack

The proposed scheme protect secret information using shared secret key ξ and κ. The κ is used to update weighted matrix for each new block and ξ is used to distribute the stego pixel among dual image. We embed the positional value (pv) within stego image. The scheme is secure from possible malicious attacks. The Fig. 5.28 shows the noise like result when wrong weighted matrix and keys are used to reveal the hidden message. If the malicious attacker holds the original image and stego image and is fully aware of the proposed scheme, the hidden message still cannot be correctly revealed without knowing the correct secret key and weighted matrix. Similarly, if the malicious attacker is fully aware about the weighted matrix of the proposed scheme, the hidden message still cannot be correctly revealed without knowing the correct key ξ and κ. Furthermore, the attacker may employ the brute force attack that tries all

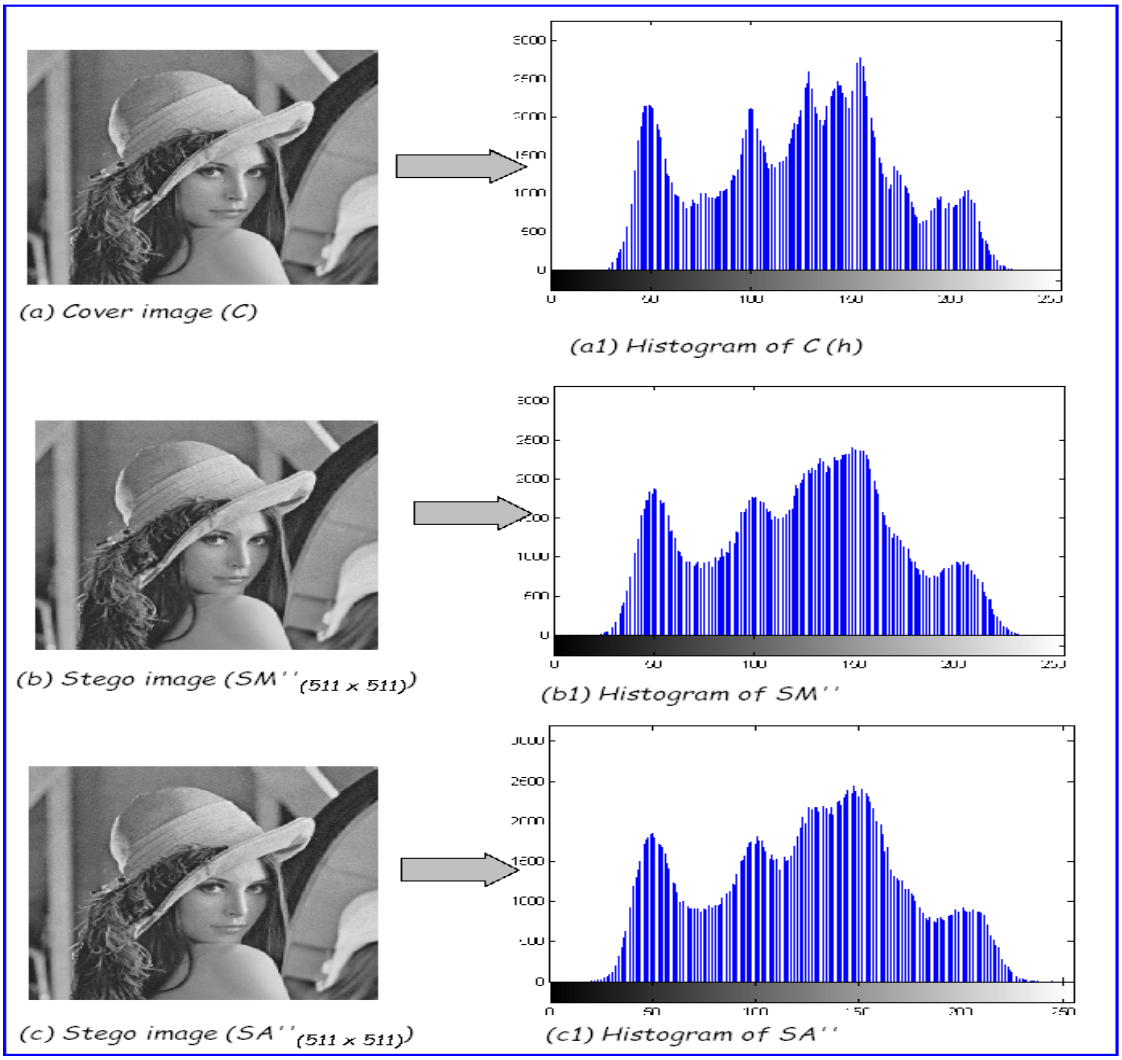


Figure 5.26: Histogram of cover and dual stego image in IDRDHWM

possible permutation to reveal the hidden message. It enhances security because the number of attempts to reveals the secret in a $(M \times N)$ image are $(2^{r-1}+1)! \times (M/3 \times N/3)$, where r is the number of bits to be embedded in each operation. Again, distribution of original and stego pixel among two stego images has been performed depending on the bit pattern of shared secret key ξ of length 128 bits and the possible combinations of ξ are 2^{128}. The maximum possible number of weighted matrix are $(2^{r-1}+1)! \times (M/3 \times N/3)$ and 2^{128} combinations of ξ is required to retrieve secret message, which is hard to compute by any high computing machine. It is robust against brute force attack because such huge number of attempts are computationally impractical for current computers.

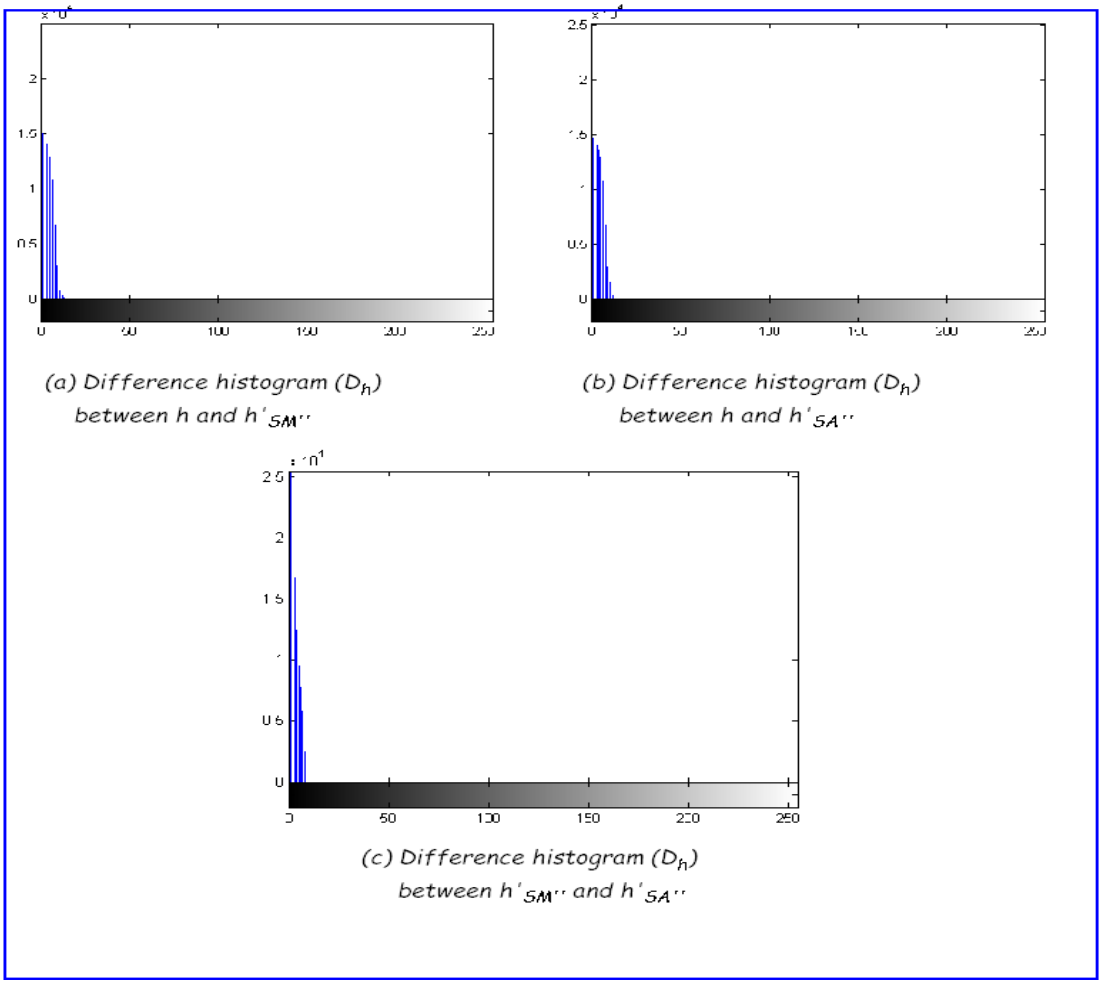


Figure 5.27: Histogram difference of cover and dual stego image in IDRDHWM

5.5 Analysis and Discussions

The comparison of these proposed methods through weighted matrix based data hiding schemes are shown in Table 5.18. From the table it is observed that payload of IDRDHWM is highest among all other proposed schemes and PSNR is reasonably high. The overflow and underflow has been controlled in these proposed schemes. All the schemes are robust and reversible.

Table 5.18: Comparison of proposed RDH schemes in terms of PSNR (dB) and Payload

Schemes	Reversible/ Irreversible	Single/Dual	Capacity (bits)	PSNR (dB)	Payload (bpp)
DRDHWM	Reversible	Dual	2,60,100	39.73	1.984
IRDHWM	Reversible	Single	7,74,192	37.96	2.965
IDRDHWM	Reversible	Dual	18,08,484	35.39	3.462

The analysis has been performed through some steganographic analysis and attacks which are presented in Table 5.19. In IDRDHWM, it is observed that after embedding 18,08,484

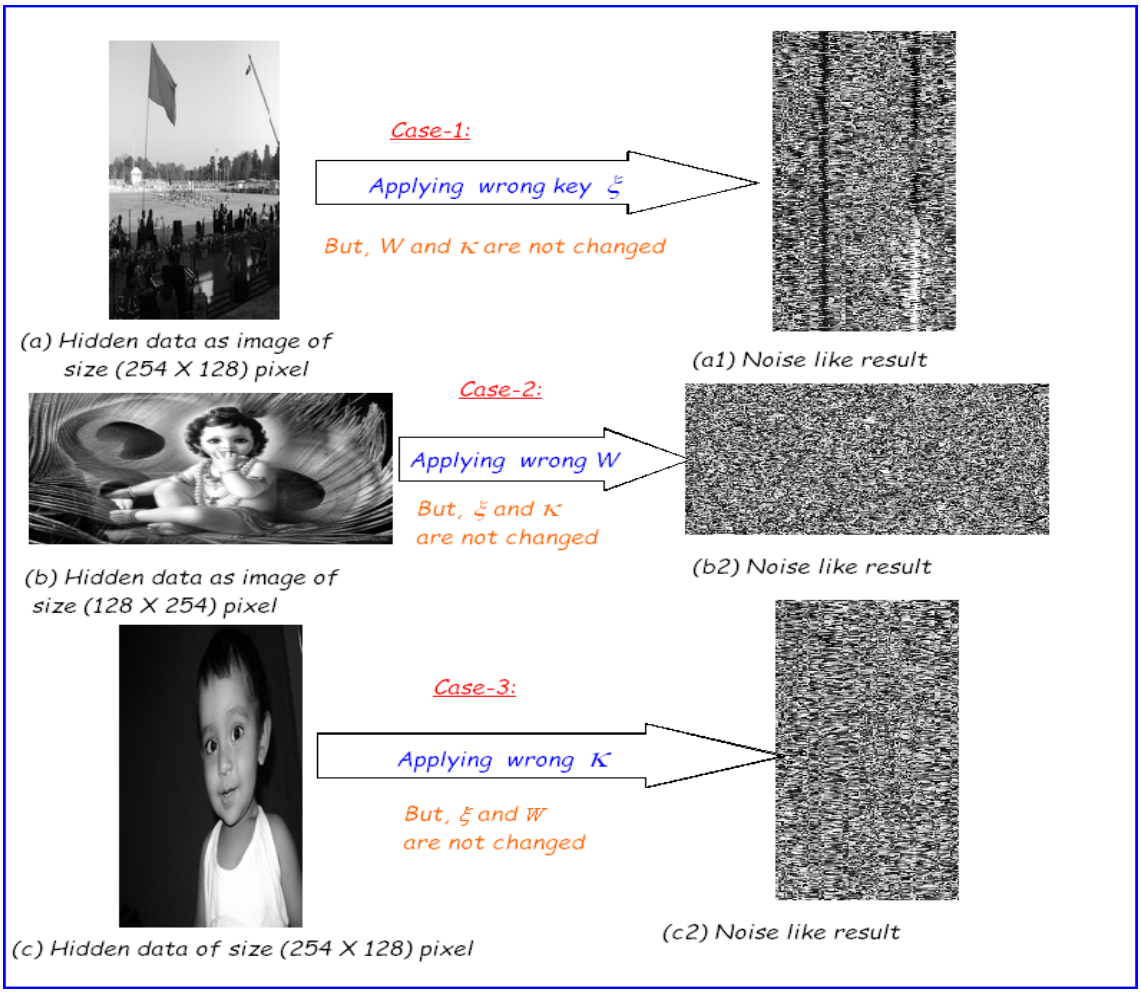


Figure 5.28: Result of Brute Force Attack in IDRDHWM

bits within cover image the corresponding RS value , Relative entropy and CC (ρ) values are 0.0049, 0.0387 and 0.9986 respectively. It is robust against all these steganographic analysis and attacks.

Table 5.19: Comparison of proposed RDH schemes in terms of steganalysis values

Proposed Scheme	Capacity (bits)	PSNR (dB)	RS value	Relative Entropy	CC (ρ)	Payload (bpp)
DRDHWM	2,60,100	39.73	0.0094	0.041	0.9901	1.984
IRDHWM	7,74,192	37.96	0.0065	0.0193	0.9823	2.965
IDRDHWM	18,08,484	35.39	0.0049	0.0387	0.9986	3.462

Chapter 6

Analysis and Discussions

In this dissertation, some new data hiding techniques have been devloped using Hamming code, PVD, DE, EMD and Weighted matrix. Dual image and image interpolation techniques play an important role in solving these data hiding techniques. Three core data hiding techniques are developed and described earlier.

PRDHHC deals with Hamming code based data hiding technique using shared secret key. This is a partially reversible data hiding scheme with PSNR 50.13 (dB) and payload 0.142 (bpp). To develop reversible data hiding scheme through Hamming code, dual image has been used in DRDHHC. The PSNR of DRDHHC is 51.75 (dB) and payload is 0.142 (bpp). To enhance the data hiding capacity, three LSB layers (LSB, LSB+1 and LSB+2) are used in EPRDHHC, where PSNR is 32.14 (dB) and payload 0.426 (bpp) but the scheme is not reversible. Again in order to achieve reversibility, dual images are used in EDRDHHC, where PSNR is 38.23 (dB) and payload 0.426 (bpp).

The data embedding capacities of all these Hamming code based data hiding methods vary between 0.142 to 0.426 (bpp) but security is enhanced due to the use of shared secret keys. The keys are updated for every new block. Any arbitrary length of secret message can be communicated through proposed Hamming code based data hiding schemes where receiver can extract the message successfully without knowing the length of it. The quality of stego images has been analyzed through standard steganographic analysis which obtains satisfactory results. The experimental outcomes of steganalysis and steganographic attacks are illustrated which shows these proposed schemes are good concealment strategies in hidden data communication.

To increase the data embedding capacity, PVD with DE is considered in PVDDE scheme and developed using dual image with shared secret key. In this approach, data embedding capacity has been enhanced (1.25 bpp) while preserving good visual quality (38.95 dB PSNR) which is shown in Table 6.1. Again to improve the data embedding capacity, a new PVD and EMD based data hiding scheme has been designed in PVDEMD scheme using dual image with shared secret key. In this scheme, payload is 1.75 (bpp) with PSNR 40.43 (dB). Another new reversible data hiding scheme has been developed through TPVDDE using TPVD with DE. In this approach, the payload is increased up to 2.16 (bpp), but the quality is slightly deteriorated.

Table 6.1: Comparison of proposed RDH schemes with experimental results

Proposed Model	Reversibility	Single/Dual Image	Capacity (bits)	PSNR (dB)	Payload (bpp)
PRDHHC	Irreversible	Single	37,306	50.13	0.142
DRDHHC	Reversible	Dual	74,606	51.75	0.142
EPRDHHC	Irreversible	Single	1,11,909	32.14	0.426
EDRDHHC	Reversible	Dual	2,23,818	38.23	0.426
PVDDE	Reversible	Dual	1,63,840	38.95	1.250
PVDEMD	Reversible	Dual	9,16,656	40.43	1.750
TPVDDE	Reversible	Dual	11,31,520	26.18	2.150
DRDHWM	Reversible	Dual	2,60,100	39.73	1.984
IRDHWM	Reversible	Single	7,74,192	37.96	2.965
IDRDHWM	Reversible	Dual	18,08,484	35.39	3.462

In these new proposed data hiding methods, dual image and shared secret key have been used to improve data hiding capacity, achieve reversibility, enhance security with good visual quality. The experimental results are numerically illustrated and outcomes are analyzed through some standard steganalysis techniques which indicates the stability of the developed schemes.

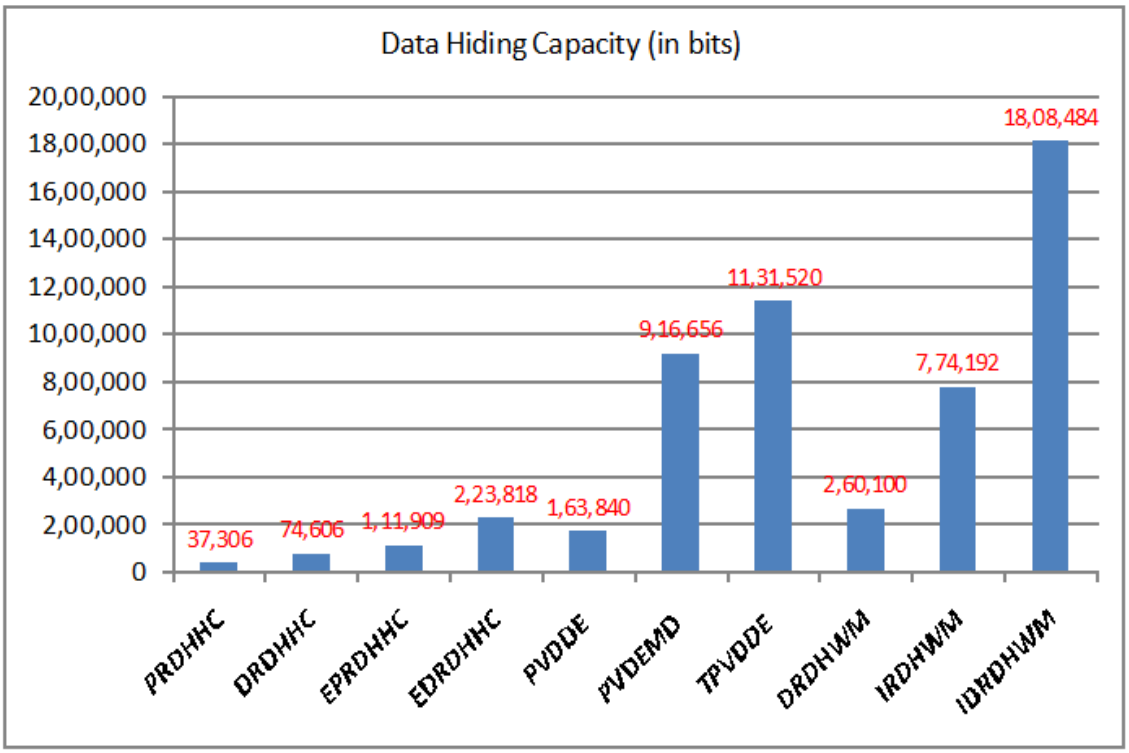


Figure 6.1: Comparison graph of proposed schemes in terms of capacity (bits)

Further, in order to improve the data embedding capacity while maintaining good visual quality some innovative weighted matrix based data hiding schemes have been formulated and

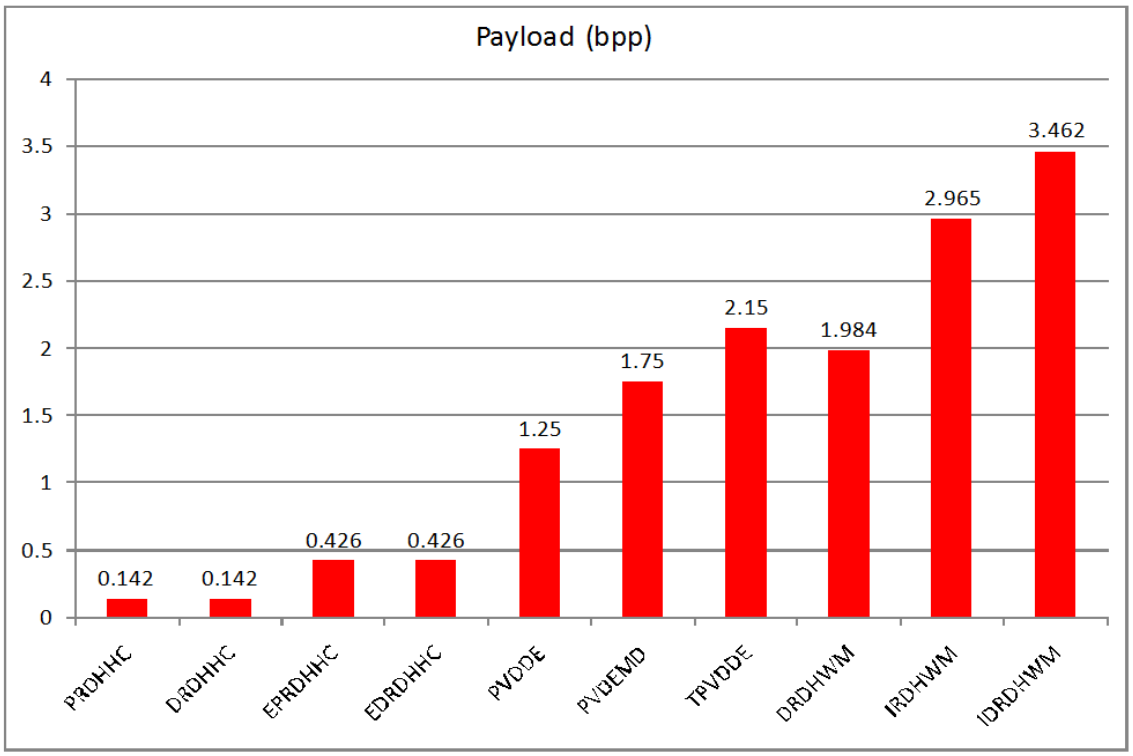


Figure 6.2: Comparison graph of proposed schemes in terms of payload (bpp)

solved. The developed DRDHWM, IRDHWM and IDRDHWM schemes have PSNR 39.73, 37.96 and 35.39 (dB) and the corresponding payloads are 1.984, 2.965 and 3.462 (bpp) respectively. So, far weighted matrix based data hiding scheme was irreversible with limited payload. Here dual image and image interpolation have been introduced to achieve reversibility and repeated entry-wise multiplication operation has been performed to increase payload. To enhance the security, modification or alteration of weighted matrix has been introduced for every new block using shared secret key.

All the data hiding schemes in this thesis have been implemented through MATLAB Version 7.6.0.324 (R2008a). The standard stegenalysis has been applied through RS analysis, Statistical analysis and some steganographic attacks are performed through Histogram attack and Brute Force Attacks. In has been observed that the designed schemes are robust against these attacks.

All the experimental results of steganalysis and steganographic attack have been shown in the Table 6.2. From this table, it is observed that the results of RS analysis and relative entropy are nearer to zero and correlation coefficients (ρ) are nearer to one which indicates developed

Table 6.2: Comparison of proposed RDH schemes with steganalysis values

Proposed Model	Capacity (bits)	PSNR (dB)	RS value	Relative Entropy	ρ	Payload (bpp)
PRDHHC	37,306	50.13	0.0357	0.0069	0.9864	0.142
DRDHHC	74,606	51.75	0.0578	0.0086	0.9834	0.142
EPRDHHC	1,11,909	32.14	0.667	0.0212	0.9786	0.426
EDRDHHC	2,23,818	38.23	0.1120	0.145	0.9752	0.426
PVDDE	1,63,840	38.95	0.0123	0.0131	0.9840	1.250
PVDEMD	9,16,656	40.43	0.0447	0.0131	0.9820	1.750
TPVDDE	11,31,520	26.18	0.531	0.139	0.9682	2.150
DRDHWM	2,60,100	39.73	0.0094	0.041	0.9901	1.984
IRDHWM	7,74,192	37.96	0.0065	0.0193	0.9823	2.965
IDRDHWM	18,08,484	35.39	0.0049	0.0387	0.9986	3.462

schemes are maintained good perceptibility and preserve good visual quality. Fig. 6.1, 6.2 and 6.3 represent the graphical comparison of proposed data hiding schemes with respect to data hiding capacity, payload (bpp) and PSNR (dB) respectively.

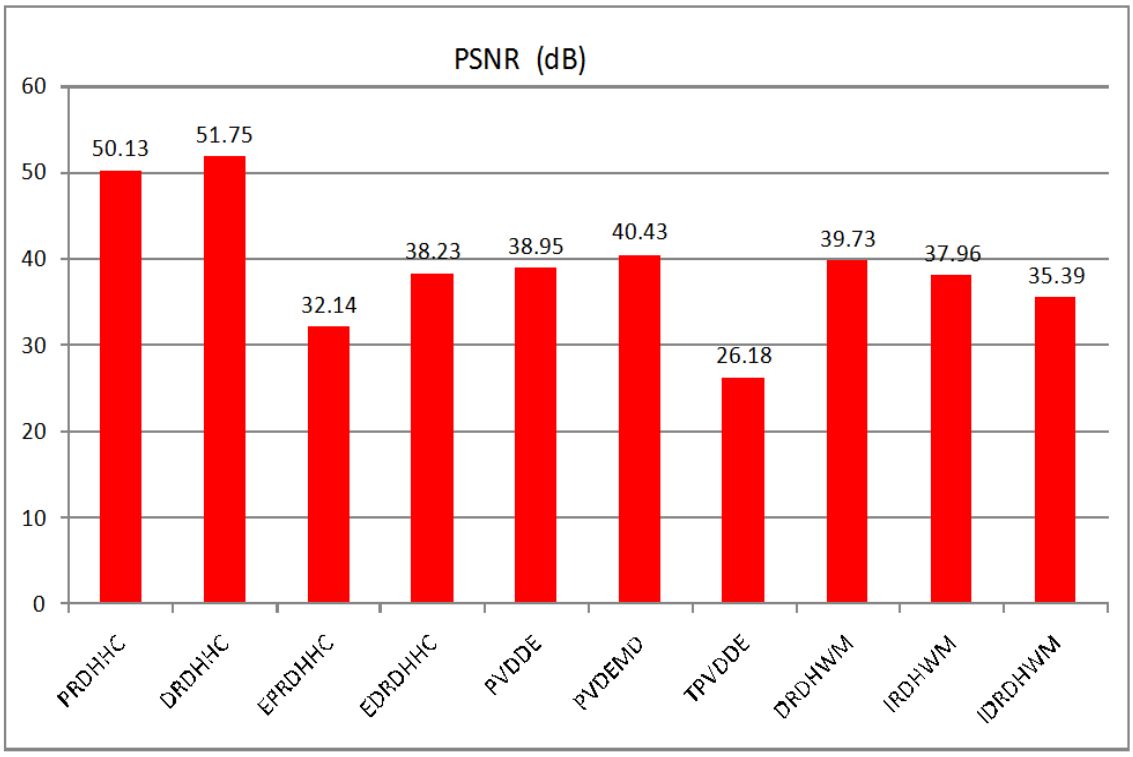


Figure 6.3: Comparison graph of proposed schemes in terms of PSNR (dB)

Chapter 7

Conclusion and Future Research Work

7.1 Conclusion

In this dissertation, some new data hiding techniques are designed and solved. All these developed techniques are validated by corresponding experimental results. The salient features of the developed techniques are as follows:

- Data hiding through Hamming code was not reversible. Some new data hiding techniques through Hamming code has been designed in which original cover image can be recovered successfully without any distortion. That means reversibility has been achieved using Hamming code based data hiding schemes through dual image.
- The Hamming code based data hiding schemes are designed in such a manner that receiver can retrieve the entire secret message without knowing the length of the secret message. It is possible only when receiver finds an error in a secret position of the stego image block, because data is embedded through error creation in two different locations, one at secret position and another at any suitable location except secret position.
- Shared secret key and dual image has been considered in data hiding problems which are used to enhance security, achieve good visual quality and improve data embedding capacity.
- Data hiding through PVD was not reversible. Some dual image based new data hiding techniques have been designed through PVD, DE, EMD and TPVD in which reversibility has been achieved.
- Designing high payload secure data hiding techniques with good visual quality through weighted matrix is a challenging task. To solve these problems some new weighted matrix based data hiding schemes have been developed.
- Data hiding through weighted matrix was not reversible. Some novel data hiding schemes are designed through weighted matrix using image interpolation and dual image in which reversibility has been achieved.
- To enhance security through weighted matrix based data hiding schemes, a new idea has been introduced in which the weighted matrix has been modified for each new block using shared secret key.

- To improve the payload through weighted matrix based data hiding schemes, dual image and image interpolation techniques have been considered with repeated entry-wise multiplication operations. The payload in IDRDHWM is 3.46 (bpp) with PSNR 35.39 (dB) where both dual image and image interpolation have been taken into account.

- The visual quality of stego images are preserved in developed weighted matrix based data hiding techniques because only one bit modification can embed a fixed length (four bits) secret data bits within a pixel of cover image.

- Overflow and underflow situations have been controlled in all of these developed schemes where it may possible to occur.

- Steganalysis have been applied using RS analysis and Statistical analysis. Some steganographic attacks are also have been performed and it has been observed that the proposed schemes are robust against various steganographic analysis and attacks.

7.2 Future Research Work

Now-a-days, due to the rapid development of digital technology, multimedia application and online transaction, much attention has been paid to secure hidden data communication. Among these present security schemes, data hiding technology has been widely used for copyright protection, content authentication and secret communication, etc.

A good data hiding algorithm should have high accuracy, utmost inserting capacity and satisfactory level of security. At the same time, simplicity of algorithm and worldwide relevance should also be considered. Thus new type of data hiding problems are required to be designed in the different hiding approaches.

There are various domains of image processing like spatial domain, transform domain and compress domain. The level of security in transform domain is much better as compared to spatial domain, because secret messages are embedded in the coefficients. Several data hiding problems can be formulated and solved in transform domain and compress domain with image interpolation and dual image. Data hiding in other than special domain have not been

considered in this dissertation. Now-a-days, due to development of digital technology and use of social media, people are more vulnerable. Hence future investigation may use in frequency, transform and compress domain to enhance security.

Though the design, formulation and assumption in the proposed methods presented in this dissertation are innovative, yet there are some limitations and hence there is a huge scope of extension of these schemes for future research work.

Although the concept of data hiding through PVD, DE, EMD, TPVD and weighted matrix are old to some extent but the idea of dual image and image interpolation are quite new. In the literature, there are several representation of image interpolation and dual image are explained. In future, researchers may give attention to this area and the application of dual image and image interpolation may be used for data hiding in real life problems. In this dissertation, dual image and image interpolation have been used separately in different schemes and also combined together in some schemes.

To improve the data hiding algorithm and measure the quality of the stego image, a variety of quantities can be considered like Region Of Interest (ROI), use of optimization algorithms like ant colony optimization, genetic algorithm, particle swarm optimization etc., neural networks, fuzzy logic and hybrid network may also help to embed secret data within cover image in such a way that it improves embedding capacity, stego image quality and innocuousness.

Regardless of these limitations and future scopes, the thesis will be useful for the personal those are perusing research in the field of Computer Science and Engineering, Computer Application and Information Technology. The schemes are superior over the existing schemes in terms of security, payload with acceptable visual clarity and reversibility. Practising engineers would also find this them to be an excellent reference resources.

Chapter 8

Bibliography

Bibliography

[1] Alattar, A. M. (2004), *Reversible watermark using the difference expansion of a generalized integer transform*, IEEE Transactions on Image Processing, 13(8), 1147-1156.

[2] Bai, J. and Chang, C.C. (2016), *A High Payload Steganographic Scheme for Compressed Images with Hamming Code*, International Journal of Network Security, 18(6), 1122-1129.

[3] Barton, J. M. (1997), *Method and apparatus for embedding authentication information within digital data*, U.S. Patent No. 5,646,997.

[4] Cao, Z., Yin, Z., Hu, H., Gao, X., and Wang, L. (2016), *High capacity data hiding scheme based on (7, 4) Hamming code*, Springer Plus, 5(1), 1-13.

[5] Chang, C. C., Kieu, T. D., and Chou, Y. C. (2007), *Reversible data hiding scheme using two steganographic images*, TENCON 2007-2007, IEEE Region 10 Conference, 1-4.

[6] Chang, C. C. and Chou, Y. C. (2009), *Information hiding in dual images with reversibility*, Third International Conference on Multimedia and Ubiquitous Engineering, 145-152.

[7] Chang, K. C., Chang, C. P., Huang, P. S. and Tu, T. M. (2008), *A novel image steganographic method using tri-way pixel-value differencing*, Journal of multimedia, 3(2), 37-44.

[8] Chang, C. C. and Chou, Y. C. (2008), *Using nearest covering codes to embed secret information in grayscale images*, Proceedings of the 2nd international conference on ubiquitous information management and communication, ACM, 315320.

[9] Chang, C. C., Kieu, T. D. and Chou, Y. C. (2008), *A high payload steganographic scheme based on (7, 4) hamming code for digital images*, International Symposium on Electronic Commerce and Security, IEEE, 16-21.

[10] Chang, C. C., Lu, T. C., Horng, G., Huang, Y. H. and Hsu, Y. M. (2013), *A high payload data embedding scheme using dual stego-images with reversibility*, 9th International Conference on Information Communications and Signal Processing (ICICS), IEEE, 1-5.

[11] Chen, J. (2014), *A PVD-based data hiding method with histogram preserving using pixel pair matching*, Signal Processing: Image Communication, Elsevier, 29(3), 375-384.

[12] Crandall, R. (1998), *Some notes on steganography*, Posted on steganography mailing list, http://os.inf.tudresden.de/ westfeld/crandall.pdf

[13] Fan, L., Gao, T., Yang, Q. and Cao, Y. (2011), *An extended matrix encoding algorithm for steganography of high embedding efficiency*, Computers & Electrical Engineering, Elsevier, 37(6), 973-981.

[14] Fan, L., Gao, T. and Cao, Y. (2013), *Improving the embedding efficiency of weight matrix-based steganography for grayscale images*, Computers & Electrical Engineering, Elsevier, 39(3), 873-881.

[15] Fridrich, J., Goljan, M. and Hogea, D. (2002), *Attacking the outguess*, Proceedings of the ACM Workshop on Multimedia and Security, Juan-les-Pins, France.

[16] Fridrich, J., Goljan, M. and Du, R. (2001), *Reliable detection of LSB steganography in color and grayscale images*, Proceedings of the 2001 workshop on Multimedia and security: new challenges, ACM, 27-30.

[17] Fridrich, J., Goljan, M. and Du, R. (2002), *Lossless data embedding: new paradigm in digital watermarking*, EURASIP Journal on Applied Signal Processing, 2002(1), 185-196.

[18] Fridrich, J., Goljan, M. and Du, R. (2001), *Invertible authentication*, Photonics West 2001-Electronic Imaging, International Society for Optics and Photonics, 197-208.

[19] Fridrich, J. and Soukal, D. (2006), *Matrix embedding for large payloads*, Electronic Imaging 2006, International Society for Optics and Photonics, 60721W-60721W.

[20] Fridrich, J., Lisonek, P. and Soukal, D. (2006), *On steganographic embedding efficiency*, Information Hiding, Springer Berlin Heidelberg, 282-296.

[21] Hamming, R. W. (1950), *Error detecting and error correcting codes*, Bell System Technical Journal, Wiley Online Library, 29(2), 147-160.

[22] Harmsen, J. J. and Pearlman, W. A. (2003), *Steganalysis of additive-noise modelable information hiding*, Electronic Imaging, International Society for Optics and Photonics, 131-142.

[23] Hong, W. (2013), *Adaptive image data hiding in edges using patched reference table and pair-wise embedding technique*, Information Sciences, Elsevier, 221, 473-489.

[24] Hong, W. and Chen, T. S. (2012), *A novel data embedding method using adaptive pixel pair matching*, IEEE Transactions on Information Forensics and Security, IEEE, 7(1), 176-184.

[25] Joo, J. C., Lee, H. Y. and Lee, H. K. (2010), *Improved steganographic method preserving pixel-value differencing histogram with modulus function*, EURASIP Journal on Advances in Signal Processing, 2010, 26.

[26] Jung, K. H. and Yoo, K. Y. (2009), *Data hiding method using image interpolation*, Computer Standards & Interfaces, Elsevier, 31(2), 465-470.

[27] Ker, A. D. (2005), *Steganalysis of LSB matching in grayscale images*, Signal Processing Letters, IEEE, 12(6), 441-444.

[28] Kim, C., Shin, D. and Shin, D. (2011), *Data hiding in a halftone image using hamming code (15, 11)*, Intelligent Information and Database Systems, Springer Berlin Heidelberg, 372-381.

[29] Kim, C. and Yang, C. N. (2014), *Improving data hiding capacity based on hamming code*, Frontier and Innovation in Future Computing and Communications, Springer Netherlands, 697-706.

[30] Kim, C. and Yang, C. N. (2014), *Data hiding based on overlapped pixels using hamming code*, Multimedia Tools and Applications, Springer, 1-13.

[31] Kieu, T. D. and Chang, C. C. (2011), *A steganographic scheme by fully exploiting modification directions*, Expert systems with Applications, Elsevier, 38(8), 10648-10657.

[32] Kuo, W. C., Kuo, S. H., Wang, C. C., and Wuu, L. C. (2016), *High capacity data hiding scheme based on multi-bit encoding function*, Optik-International Journal for Light and Electron Optics, 127(4), 1762-1769.

[33] Lee, C. F., Chen, H. L. and Tso, H. K. (2010), *Embedding capacity raising in reversible data hiding based on prediction of difference expansion*, Journal of Systems and Software, Elsevier, 83(10), 1864-1872.

[34] Lee, C. F. and Huang, Y. L. (2013), *Reversible data hiding scheme based on dual stegano-images using orientation combinations,* Telecommunication Systems, Springer, 52(4), 2237-2247.

[35] Lee, C. C., Wu, H. C., Tsai, C. S. and Chu, Y. P. (2008), *Adaptive lossless steganographic scheme with centralized difference expansion,* Pattern Recognition, Elsevier, 41(6), 2097-2106.

[36] Lee, C. F., Wang, K. H., Chang, C. C. and Huang, Y. L. (2009), *A reversible data hiding scheme based on dual steganographic images*, Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, ACM, 228-237.

[37] Lee, C. F., Weng, C. Y., and Chen, K. C. (2016), *An efficient reversible data hiding with reduplicated exploiting modification direction using image interpolation and edge detection,* Multimedia Tools and Applications, Springer, 1-24.

[38] Lee, C. F. and Huang, Y. L. (2012), *An efficient image interpolation increasing payload in reversible data hiding*, Expert Systems with Applications, Elsevier, 39(8), 6712-6719.

[39] Lee, S. K., Suh, Y. H. and Ho, Y. S. (2004), *Lossless data hiding based on histogram modification of difference images,* Advances in Multimedia Information Processing-PCM 2004, Springer Berlin Heidelberg, 340-347.

[40] Li, X., Li, B., Luo, X., Yang, B. and Zhu, R. (2013), *Steganalysis of a PVD-based content adaptive image steganography*, Signal Processing, Elsevier, 93(9), 2529-2538.

[41] Lien, B. K., Chen, S. K., Wang, W. S. and King, K. P. (2015), *Dispersed Data Hiding Using Hamming Code with Recovery Capability*, Genetic and Evolutionary Computing, Springer International Publishing, 179-187.

[42] Lin, C. C., Tai, W. L. and Chang, C. C. (2008), *Multilevel reversible data hiding based on histogram modification of difference images*, Pattern Recognition, Elsevier, 41(12), 3582-3591.

[43] Lin, C. C. and Hsueh, N. L. (2008), *A lossless data hiding scheme based on three-pixel block differences*, Pattern Recognition, Elsevier, 41(4), 1415-1425.

[44] Lou, D. C., Hu, M. C. and Liu, J. L. (2009), *Multiple layer data hiding scheme for medical images*, Computer Standards & Interfaces, Elsvier, 31(2), 329-335.

[45] Lu, T. C., Tseng, C. Y. and Wu, J. H. (2015), *Dual imaging-based reversible hiding technique using LSB matching*, Signal Processing, Elsevier, 108, 77-89.

[46] Lu, T. C., Wu, J. H. and Huang, C. C. (2015), *Dual-image-based reversible data hiding method using center folding strategy*, Signal Processing, Elsevier, 115, 195-213.

[47] Luo, W., Huang, F. and Huang, J. (2010), *Edge adaptive image steganography based on LSB matching revisited*, IEEE Transactions on Information Forensics and Security, 5(2), 201-214.

[48] MacWilliams, F. J. and Sloane, N. J. A. (1977), *The theory of error correcting codes*, Elsevier.

[49] Ma Z., Li Y. and Zhang X. (2013), *Based on Hamming and subordinate pixel compensation halftone image information hiding*, Journal of Shanghai University (Natural Science), 19(2), 111-115.

[50] Mielikainen, J. (2006), *LSB matching revisited*, Signal Processing Letters, IEEE, 13(5), 285-287.

[51] Ni, Z., Shi, Y. Q., Ansari, N. and Su, W. (2006), *Reversible data hiding*, IEEE Transactions on Circuits and Systems for Video Technology, IEEE, 16(3), 354-362.

[52] Qin, C., Chang, C. C. and Hsu, T. J. (2015), *Reversible data hiding scheme based on exploiting modification direction with two steganographic images*, Multimedia Tools and Applications, Springer, 74(15), 5861-5872.

[53] Sharp, T. (2001), *An implementation of key-based digital signal steganography*, Information hiding, Springer Berlin Heidelberg, 13-26.

[54] Shen, S. Y. and Huang, L. H. (2015), *A data hiding scheme using pixel value differencing and improving exploiting modification directions*, Computers & Security, Elsevier, 48, 131-141.

[55] Simmons, G. J. (1984), *The prisoners problem and the subliminal channel*, Advances in Cryptology, Springer US, 51-67.

[56] Stanley, C. A. (2005), *Pairs of Values and the Chi-squared Attack*, Department of Mathematics, Iowa State University.

[57] Tang, M., Hu, J. and Song, W. (2014), *A high capacity image steganography using multi-layer embedding*, Optik-International Journal for Light and Electron Optics, Elsevier, 125(15), 3972-3976.

[58] Tian, J. (2003), *Reversible data embedding using a difference expansion*, IEEE Transactions on Circuits and Systems for Video Technology, 13(8),890-896.

[59] Thodi, D. M., and Rodríguez, J. J. (2007), *Expansion embedding techniques for reversible watermarking*, IEEE Transactions on Image Processing, IEEE, 16(3), 721-730.

[60] Turner, L. F. (1989), *Digital data security system*, Patent IPN wo, 89, 08915.

[61] Tsai, P., Hu, Y. C. and Yeh, H. L. (2009), *Reversible image hiding scheme using predictive coding and histogram shifting*, Signal Processing, Elsevier, 89(6), 1129-1143.

[62] Tsai, Y. Y., Huang, Y. H., Lin, R. J., and Chan, C. S. (2016), *An Adjustable Interpolation-based Data Hiding Algorithm Based on LSB Substitution and Histogram Shifting*, International Journal of Digital Crime and Forensics (IJDCF), 8(2), 48-61.

[63] Tsai, Y. Y., Tsai, D. S. and Liu, C. L. (2013), *Reversible data hiding scheme based on neighboring pixel differences*, Digital Signal Processing, Elsevier, 23(3), 919-927.

[64] Tseng, Y. C., Chen, Y. Y. and Pan, H. K. (2002), *A secure data hiding scheme for binary images*, IEEE Transactions on Communications, IEEE, 50(8), 1227-1231.

[65] University of Southern California, ”The USC-SIPI Image Database”, http://sipi.usc.edu/database/database.php.

[66] Wang, C. M., Wu, N. I., Tsai, C. S. and Hwang, M. S. (2008), *A high quality steganographic method with pixel-value differencing and modulus function*, Journal of Systems and Software, Elsevier, 81(1), 150-158.

[67] Westfeld, A. (2001), *F5 a steganographic algorithm*, Information hiding, Springer Berlin Heidelberg, 289-302.

[68] Willems, F. M. and Van Dijk, M. (2005), *Capacity and codes for embedding information in gray-scale signals*, IEEE Transactions on Information Theory, 51(3), 1209-1214.

[69] Wu, D. C. and Tsai, W. H. (2003), *A steganographic method for images by pixel-value differencing*, Pattern Recognition Letters, Elsevier, 24(9), 1613-1626.

[70] Yang, C. H., Weng, C. Y., Tso, H. K. and Wang, S. J. (2011), *A data hiding scheme using the varieties of pixel-value differencing in multimedia images*, Journal of Systems and Software, Elsevier, 84(4), 669-678.

[71] Zaker, N., and Hamzeh, A. (2012), *A novel steganalysis for TPVD steganographic method based on differences of pixel difference histogram*, Multimedia Tools and Applications, Springer, 58(1), 147-166.

[72] Zhang, X. and Wang, S. (2006), *Efficient steganographic embedding by exploiting modification direction*, Communications Letters, IEEE, 10(11),781-783.

[73] Zhang, W., Wang, S. and Zhang, X. (2007), *Improving embedding efficiency of covering codes for applications in steganography*, Communications Letters, IEEE, 11(8), 680-682.

[74] Zhang, X. and Wang, S. (2004), *Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security*, Pattern Recognition Letters, Elsevier, 25(3), 331-339.

[75] Zeng, X. T. and Li, Z. (2012), *Reversible data hiding scheme using reference pixel and multi-layer embedding*, AEU-International Journal of Electronics and Communications, 66(7), 532-539.

Reprints

Optik 127 (2016) 3347–3358

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

High payload reversible data hiding scheme using weighted matrix

Biswapati Jana*

Department of Computer Science, Vidyasagar University, West Midnapore, 721102, India

ARTICLE INFO

Article history:
Received 11 August 2015
Accepted 9 December 2015

Keywords:
Reversible data hiding
Weight matrix
Image interpolation
RS analysis
Relative entropy
Histogram attack

ABSTRACT

In this paper, we proposes high payload reversible data hiding scheme using weighted matrix. Generate cover image by enlarge the size of original image using interpolation. Partition the original image into (3×3) pixel blocks and cover image into (5×5) blocks. Perform modular sum of entry-wise multiplication of an (3×3) original pixel block with same size shared predefined weighted matrix W. At most twelve multiplication operations are performed to embed 48 bits secret data within an (5×5) overlapped pixel block. In each operation, the data embedding position are identified and stored the positional value by modifying three least significant bits of the interleaved pixel of cover image to hide large amount of secret data. For i-th block $i = 1, 2, \ldots$, we update the weighted matrix W_{i+1} as $W_{i+1} = (W_i \times \kappa - 1) \, mod \, 9$, where $gcd(\kappa, 9) = 1$. The original pixels are not effected during data embedding in our scheme which assure reversibility. The proposed scheme provides average embedding payload 2.97 bits per pixel (bpp) with good visual quality measured by peak signal to noise ratio (PSNR) guaranteed to be higher than 37.97 dB. Finally, we compare our scheme with other state-of-the-art methods and obtain reasonably better performance in terms of data embedding capacity.

1. Introduction

The modern secret writing are tweaked the cover work in such a way that a secret message can be encoded within them. The secret message insertion may change every bit of information in the cover data. There are a number of ways to conceal information within cover work; The most usual methodologies are based on the least significant bits (LSBs) substitution, masking, filtering, [2–4] and the modulus operation [5–7]. Reversible data hiding using block are commonly used to increase visual quality or to achieve reversibility [9,11,10,12,8]. Reversible Data Hiding (RDH) presented by Ni et al. [13] which are based on histogram shifting with zero or minimum change of the pixel gray values. Multilevel reversible data hiding based on histogram shifting are proposed by Lin et al. [14] and Tsai et al. [15]. Adaptive reversible data hiding method using integer transform has been presented Peng et al. [16]. The method payload is up to 2.17 bpp. Designing a novel data hiding system accomplishing good visual quality, high embedding capacity, robustness and steganographic protection is a technically challenging problem. Jung and Yoo [17] first proposed data hiding through image interpolation using neighbor mean interpolation with payload 2.28 bpp. Then Lee and Huang [18] presented higher capacity image hiding by interpolating with neighboring pixels. Tang et al. [1] proposed high capacity reversible steganography using multilayer embedding with average payload 1.79 bpp and average PSNR 33.85 db. A secure data hiding scheme for binary images using a key matrix and a weight matrix W has been proposed by Tseng et al. [19] which can hide only 2 bits in a (3×3) block of pixels. Li Fan et al. [20] proposed an improved efficient data hiding scheme using weight matrix for gray scale images which can hide 4 bits in a (3×3) block. Both the matrix based schemes performed only one modular sum of entry wise multiplication of weighted matrix W with a (3×3) block of pixel in the original image. Only one embedding operation is performed with a single block and only 4 bits data embed within the block. High-capacity is still one of important research parts in data hiding. After the confidential message embedded is extracted, the requirement for the image reversibility for the entire recovery of the original object without any distortion goes high. Here we proposed a high capacity reversible data hiding scheme with where twelve time multiplication operations are performed in each and every block to hide forty eight bits secret data within each block. Also the scheme is achieved good PSNR and payload.

1.1. Motivation

In this paper, we introduced a new reversible data hiding scheme through weighted matrix.

* Tel.: +91 9126661253.
E-mail address: biswapati.jana@mail.vidyasaar.ac.in

http://dx.doi.org/10.1016/j.ijleo.2015.12.055
0030-4026/

- Our motivation is to enhance the embedding capacity and achieve reversibility in steganography. Data embedding using weighted matrix was not reversible. We have applied weighted matrix based data embedding scheme on interpolated image to keep the position of data embedding in the interleaved pixel on cover image to get reversible data hiding.
- One of the important modification that we have proposed in our scheme is to update weighted matrix W for each new block of cover image. For i-th block $i = 1, 2, \ldots$, we update the weighted matrix W_{i+1} as $W_{i+1} = (W_i \times \kappa - 1)\, mod\, 9$, where $gcd(\kappa, 9) = 1$. In Li Fan et al.'s [20] weighted matrix based data hiding schemes only single sum of entry-wise multiplication is applied on a block of cover image. Here, we enhanced the weighted matrix based data hiding by applied twelve times sum of entry-wise multiplication on a single block. As a result, 48 bits secret data are embedded in a single block which achieve high capacity data embedding and preserved good qualities stego.
- To enhance the security, our scheme is used shared secret key κ. During data embedding, the weighted matrix is updated using κ. It enhanced security because the number of attempts to revels the secret in a $(M \times N)$ image are $(2^{r-1}+1)^{M/3\times N/3}$, where r is the number of secret bits those will be embedded into each block of cover image. It is robust against brute force attacks because this huge number of attempts are computationally impracticable for current computers.

The rest of the paper is organized as follows. Section 2 describes some preliminaries. Proposed data hiding scheme in detail how an improvement in the image interpolating method can increase the embedding capacity using weight matrix while preserving good image quality has been discussed in Section 3. Experimental results with comparisons are discussed in Section 4. Finally, steganagraphic security analysis techniques are shown in Section 5. Conclusions are given in Section 6.

2. Preliminaries

Reversible data hiding become a very concerning and difficult job in hidden data communication with security. Information can be embedded within image which contain ownership identification, authentication and copy right protection. In this section we reviews, weighted matrix-based data hiding scheme.

2.1. Data hiding using weighted matrix

An $(m \times n)$ integer weighted matrix W will be shared by sender and receiver before data communication. The criterion of preferring W is that each element of matrix is arbitrarily allotted a value from the combination $(0, 1, 2, \ldots 2^{r-1}+1)$ and each element appears at least once in W, where r denotes the number of secret bits those will be embedded into each block of cover image F_i. Next, it will embed r data bits, say $b_1 b_2 \ldots b_r$ into image block F_i using the following equation

$$d = (b_1 b_2 \ldots b_r)_2 - SUM(F_i \otimes W) (\mathrm{mod}\, 2^r), \tag{1}$$

where $\otimes$ denotes entry-wise multiplication operator and $i = 1, 2, \ldots$, number of blocks. The function $SUM(.)$ represents the modular summation of all the entries of matrix $(F_i \otimes W)$. If d is equal to zero modulo 2^r then F_i is intact; otherwise, modify F_i to F_i' to satisfy the following equation

$$SUM(F_i' \otimes W) = b_1 b_2 \ldots b_r (\mathrm{mod}\, 2^r) \tag{2}$$

The receiver can derive $b_1 b_2 \ldots b_r$ by computing $SUM(F_i' \oplus W)$ $(mod\, 2^r)$. There exists high-risk security vulnerability in special case, because an attacker will be able to estimate the form of weight matrix by using brute-force attack. In order to overcome the drawbacks of data embedding by matrix method, an improved embedding strategy are developed in this paper by modifying the weighted matrix W which are used for every block of (3×3) original pixels. The W is updated using the formula shown in below.

$$W_{i+1} = (W_i \times \kappa - 1) \mod 9, \tag{3}$$

where $i = 0, 1, 2, \ldots, 2^r$ and $gcd(\kappa, 9) = 1$. The sender will send a weighted matrix and κ to the receiver during data communication. Then sender can modified by increasing or decreasing the pixel value of the original image at the dth position of the weighted matrix in the pixel location of the cover image which means if F_i increases by one then the modular sum $SUM(F_i \otimes W)$ will increase by $W \in \{0, 1, 2, \ldots, 2^{r-1}+1\}$ and if F_i decreases by one then the modular sum $SUM(F_i \otimes W)$ will decrease by $W \in \{0, 1, 2, \ldots, 2^{r-1}+1\}$. In extraction phase, the receiver only needs to calculate $SUM(F_i' \otimes W) (\mathrm{mod}\, 2^r)$.

Corollary 2.1. *Maximum possible number of weight matrix W will be* $(2^{r-1}+1)!$*, where r number of secret bits those will be embedded into each block of cover image.*

Proof. Consider r bits secret message are embedded in a single block, the possible combinations of r bits is 2^r. As per the requirement of weighted matrix the range will be $(0, 1, 2, \ldots 2^{r-1}+1)$. Thus the number of element within W is $2^{r-1}+1$. So, the maximum possible combination of W will be $(2^{r-1}+1)!$.

□

3. Proposed scheme

Consider an original image I, with height M and width N. Increase the size in double by interpolation and it become cover image C of height $(2 \times M - 1)$ and width $(2 \times N - 1)$ as follows.

$$\begin{cases} C(i,j) = I(p,q) \\ \qquad \{\text{where } p = 1 \ldots M,\ q = 1 \ldots N, \\ \qquad i = 1, 3, \ldots (2 \times M - 1),\ j = 1, 3, \ldots (2 \times N - 1)\} \\ C(i,j) = (C(i, j-1) + C(i, j+1))/2 \\ \qquad \{\text{where } (i \bmod 2) \neq 0,\ (j \bmod 2 = 0)\} \\ C(i,j) = (C(i-1, j) + C(i+1, j))/2 \\ \qquad \{\text{where } (i \bmod 2) = 0,\ (j \bmod 2) \neq , \\ \qquad 0\, i = 1 \ldots (2 \times M - 1), j = 1 \ldots (2 \times N - 1)\} \\ C(i,j) = (C(i-1, j-1) + C(i-1, j+1) + C(i+1, j-1) \\ \qquad + C(i+1, j+1))/4 \\ \qquad \{\text{where } (i \bmod 2) = 0,\ (j \bmod 2) = 0\} \end{cases} \tag{4}$$

Partitioned original image into (3×3) pixel block $BLOCK_{(3\times3)}$ and cover image into (5×5) pixel block $C_{(5\times5)}$.

$$\begin{cases} inpo = (S(i, j-1) + S(i, j+1))/2; \\ \qquad \text{where } (i \bmod 2) \neq 0 \text{ and } (j \bmod 2) = 0; \\ inpo = (S(i-1, j) + S(i+1, j))/2; \\ \qquad \text{where } (i \bmod 2) = 0 \text{ and } (j \bmod 2) \neq 0; \\ \qquad i = 1 \ldots (2 \times M - 1) \text{ and } j = 1 \ldots (2 \times N - 1); \\ inpo = (S(i-1, j-1) + S(i-1, j+1) + \\ \qquad S(i+1, j-1) + S(i+1, j+1))/4; \\ \qquad \text{where } (i \bmod 2) = 0 \text{ and } (j \bmod 2) = 0; \end{cases} \tag{5}$$

Consider the weighted matrix W of size (3×3). Then performed modular sum of entry wise multiplication (val) of original image

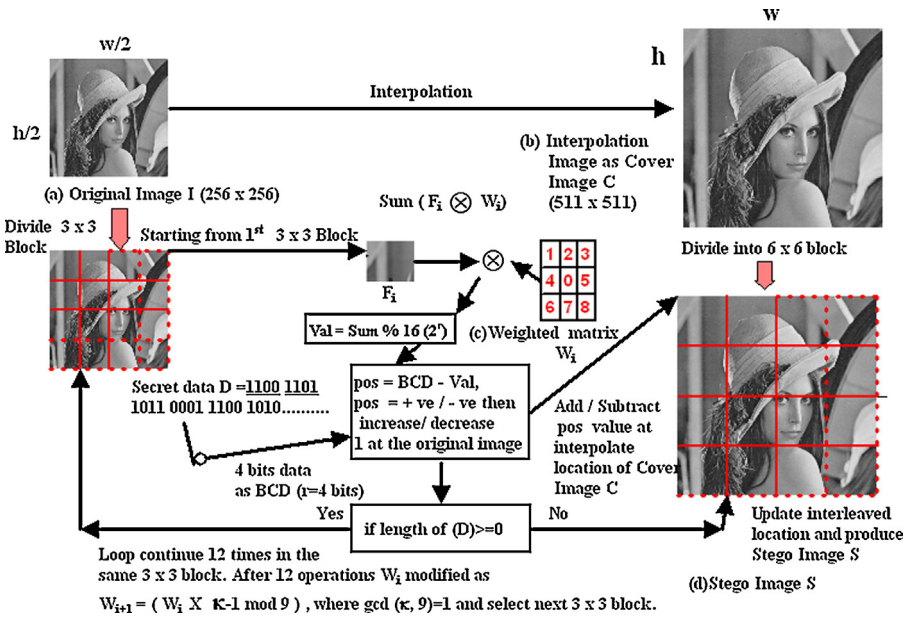


Fig. 1. Schematic diagram of embedding process.

block $BLOCK_{3\times3}$ with weighted matrix W. Calculated data embedding position by subtraction the modular sum (val) from secret data unit (D) that is $pos = D - val$. We check the sign of calculated position value (pos). If the sign of pos is positive/negative then increased/decreased the desired pixel value by one unit at the desired position of $BLOCK_{(3\times3)}$ pixel block. At the same time, stored the embedding position of original image that is pos in the interleaved pixel of the cover image block $C_{(5\times5)}$. We applied twelve time sum of entry-wise multiplication operations and each time we increased/decreased pixel value at $BLOCK_{(3\times3)}$ and keep the pos in the interleaved pixel $C_{(5\times5)}$ to acquire high data embedding capacity. Every operation embed four bits of secret data without any modification of original pixel at the cover image. As a result, without modification of original pixel value, our proposed scheme can hide $(12 \times 4) = 48$ bits secret information within a single block. After completion of data embedding in a particular $BLOCK_{(3\times3)}$ block we update weighted matrix for next block using $W_{i+1} = (W_i \times \kappa - 1)$ mod 9, where $gcd(\kappa, 9) = 1$. The Fig. 1 shows the schematic diagram of the data embedding stage. The numerical illustration of embedding process are described in Fig. 3 and corresponding algorithm are described in Algorithm 1.

During data extraction at the receiver end, we first extract the original image from stego image simply collect the pixel from odd row and odd column from stego image shown in Fig. 4. The scheme is reversible because the original image are successfully extracted from the stego image without any distortion. Now to extract the secret message, consider $BI_{(3\times3)}$ of size (3×3) from original image and $BC_{(5\times5)}$ of size (5×5) from stego image. Then we calculate interpolation value ($inpo$) using Eq. (5) and calculate position value (pos) by subtracting the stego pixel value from ($inpo$) that is $pos = inpo - current\ value$. Now we check the sign of pos for realizing increment or decrement. If $(pos \leq 0)$ then $d = 1$ else $d = -1$. We update $BI_{(3\times3)}$ using $BI_{(3\times3)} = BI_{(3\times3)} + d$ at the desired position pos of weighted matrix. Finally, we extracted 4 bits secret data in each entry wise multiplication operation using $d_i = (BI_{(3\times3)} \otimes W_i) (\text{mod } 16)$, where $i = 0, 1, 2 \ldots, 12$. The entry wise multiplication operation are performed on the same block for twelve time and corresponding 48 bits secret data are extracted. The schematic diagram for data extraction are shown in Fig. 2 and the numerical illustration are shown in Fig. 4. Also the extraction process are listed in Algorithm 2.

3.1. Overflow and underflow

Original image are enlarged by interpolation and become a cover image. The weighted matrix positions are embedded within interpolated location of enlarged image. The interleaved pixels are calculated using the pixel value of original image by Eq. (4). Overflow situation may occur when we update some pixel by pos value that may exceed the maximum pixel intensity value. For example, consider the pixel pair (254, 255). After interpolation at middle it become (254, 254, 255). If the pos value 8 is added with 254 it become 262 which is greater then 255, this situation is called overflow situation and it may occur during data embedding. Therefore, in our scheme if the d is 1 means data value is positive and need to addition pos value with the pixel value then for any pos value from 2 to 8, there may be a chance to occur overflow. When a pos subtracted from a pixel then there may be a change to occur underflow. For example, consider the pixel pair (2, 4) then after interpolation it becomes (2, 3, 4)). Consider a pos is equal to 4. If you subtract 4 from 3 then it is negative which is not a valid pixel value This is a underflow situation.

To overcome the overflow and underflow situation we adjust the pixel value as follows: Since, maximum pos value is 8 so, at the time of interpolation we adjust the pixel 247 when average cross the value 255 that is $(255 - 8) = 247$. In case of above example, (254, 254, 255) is set to (254, 247, 255). Now, if $d = 1$ and pos value is any one value from $0 - 8$ then set the interpolate pixel which is never crossed the limit 255.

$$IP = \begin{cases} 247 & \text{if } IP > 247 \\ 8 & \text{if } IP < 8 \end{cases} \quad \text{where IP is interpolated pixel.} \qquad (6)$$

To handle the underflow situation, fix the value of interpolated pixel is 8 when average value lies less than $(0 + 8) = 8$. For example, consider the pixel (2, 8, 4). If $d = -1$ and pos is any one within $0 - 8$ then the interpolated pixel never laying under the limit 0. At the time of extraction interpolated pixel value are calculated and follow the condition for overflow and underflow control mechanism that means when calculated interpolate value is greater then 247 than set it as 247 and if it is less then 8 than set it is 8.

Algorithm 1. Data embedding

Input: Original Image $I[M][N]$, Weighted Matrix $W[3][3]$, Secret Data $D = \{d_1, d_2, d_3, \ldots,$ where $d_i = 4$ bits each$\}$,and shared secret key κ , count=1;
Output: Stego Image $S[2 \times M - 1][2 \times N - 1]$;
Step-1: Create Cover Image $C[2 \times M - 1][2 \times N - 1]$ using equation (4) from $I[M][N]$, where $M = 256$ and $N = 256$ and make a copy S form C;
Step-2: Partition $I[M][N]$ into (3×3) overlapping blocks ;
Step-3: $Brow = \lfloor \frac{M+1}{sq-1} \rfloor - 1$; $Bcol = \lfloor \frac{N+1}{sq-1} \rfloor - 1$; **for** $p = 1$ *to* $Brow$ **do**
  for $q = 1$ *to* $Bcol$ **do**
    $BLOCK_{pq} 3 \times 3 \Leftarrow I_{MN}$; where sq=3;
    if $(p \neq Brow \ \& \ q \neq Bcol)$ **then**
      $sqr = (2 \times (sq-1) \times p)$; $sqc = (2 \times (sq-1) \times q)$;
    end
    if $(p \neq Brow \ \& \ q = Bcol)$ **then**
      $sqr = (2 \times (sq-1) \times p)$; $sqc = (2 \times (sq-1) \times q) + 1$;
    end
    if $(p = Brow \ \& \ q \neq Bcol)$ **then**
      $sqr = (2 \times (sq-1) \times p) + 1$; $sqc = (2 \times (sq-1) \times q)$;
    end
    if $(p = Brow \ \& \ q = Bcol)$ **then**
      $sqr = (2 \times (sq-1) \times p) + 1$; $sqc = (2 \times (sq-1) \times q) + 1$;
    end
    for $i = (2 \times (sq-1) \times (p-1))$ *to* sqr **do**
      for $j = (2 \times (sq-1) \times (q-1))$ *to* sqc **do**
        if *(i mod 2=0 or j mod 2=0)* **then**
          if *(count≤length(D))* **then**
            $SUM = BLOCK_{pq} \otimes W_r$;
            dec=BCD(d_{count}); val = SUM (mod16); pos=dec-val;
            if *(pos> 0)* **then**
              if *(pos> 8)* **then**
                pos=(16-pos); d=-1;
              else
                d=1;
              end
            end
            if *(pos< 0)* **then**
              if *(pos<-8)* **then**
                pos=abs(16+pos); d=1;
              else
                pos=abs(pos); d=-1;
              end
            end
            $BLOCK_{pq}(x, y) = BLOCK_{pq}(x, y) + d$ if W_r(x ,y)=pos, where
            x=1,2,3 and y=1,2,3;
            $S_{pq}(i, j) = C_{pq}(i, j) + (pos \times d)$;
            count=count+1;
          else
            goto **Step-4**;
          end
        end
      end
    end
    $W_{i+1} = (W_i \times \kappa - 1) \ \ mod \ \ 9$, where $gcd\ (\kappa, 9) = 1$;
  end
end
Step-4:Produced stego image $S_{2M-1 \times 2N-1}$

Step-5: End.

4. Experimental results and comparison

Our proposed method scheme is verified and tested using gray scale image Lena, Moon Surface, Aerial, Airplane, Clock, Resolution Chart and Chemical Plant which are collected from the USC-SIPI Image Database, University of Southern California [21] shown in Fig. 5. After interpolation and embedding the secret messages the stego image are generated as shown in Fig. 6. It shows original image I with (256×256) = 65, 536 bytes. After interpolation the size of cover image C as well as stego image S are increased with (511×511) = 2, 62, 121 bytes. The secret data, here we considered (382×254) image shown in Fig. 7 with $(382 \times 254 \times 8)$ equals to 7, 76, 224 bits.

Our developed algorithms: data embedding and extraction are implemented in MATLAB Version 7.6.0.324 (R2008a). Here, the impairment is assessed by means of two factors namely, Mean Square Error (*MSE*) and Peak Signal to Noise Ratio (*PSNR*). The *MSE* is calculated as follows:

$$MSE = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} [X(i, j) - Y(i, j)]^2}{(M \times N)}, \tag{7}$$

where M and N denote the total number of pixels in the horizontal and the vertical dimensions of the image respectively. $X(i, j)$ represents the pixels in the cover image and $Y(i, j)$ represents the pixels of the stego image. The difference between the original and stego images were assessed by the Peak Signal to Noise Ratio (*PSNR*). The analysis in terms of PSNR of cover image and stego image has given

Algorithm 2. Data extraction

Input: Stego Image $S[511][511]$; Weighted matrix $W[3][3]$; shared secret key κ; count=1;length of hidden data(stlen);$M = \frac{(511+1)}{2} = 256$;$N = \frac{(511+1)}{2} = 256$;
Output: Original Image $I[256][256]$; Secret Data D ;
Step-1: % Extract Original Image $I[256][256]$ from Stego Image $S[511][511]$;
i=1; p=1;
while *(*$i \leq 511$*)* **do**
  j=1; q=1;
  while *(*$j \leq 511$*)* **do**
    $I[p][q] = S[i][j]$; j+2; q=q+1;
  end
  i=i+2; p=p+1;
end
Step-2:
$Brow = \lfloor \frac{M+1}{sq-1} \rfloor - 1$; $Bcol = \lfloor \frac{N+1}{sq-1} \rfloor - 1$;
for $p = 1$ *to* $Brow$ **do**
  for $q = 1$ *to* $Bcol$ **do**
    $BLOCK_{pq}3 \times 3 \Leftarrow I_{MN}$; where sq=3;
    if *(*$p \neq Brow$ & $q \neq Bcol$*)* **then**
      $sqr = (2 \times (sq-1) \times p)$; $sqc = (2 \times (sq-1) \times q)$;
    end
    if *(*$p \neq Brow$ & $q = Bcol$*)* **then**
      $sqr = (2 \times (sq-1) \times p)$; $sqc = (2 \times (sq-1) \times q) + 1$;
    end
    if *(*$p = Brow$ & $q \neq Bcol$*)* **then**
      $sqr = (2 \times (sq-1) \times p) + 1$; $sqc = (2 \times (sq-1) \times q)$;
    end
    if *(*$p = Brow$ & $q = Bcol$*)* **then**
      $sqr = (2 \times (sq-1) \times p) + 1$; $sqc = (2 \times (sq-1) \times q) + 1$;
    end
    for i=*(*$2 \times (sq-1) \times (p-1)$*) to* sqr **do**
      for j=*(*$2 \times (sq-1) \times (q-1)$*) to* sqc **do**
        if *(*i *mod* 2=0 *or* j *mod* 2=0*)* **then**
          if *(*$count \leq$ $stlen$*)* **then**
            calculate interpolate value **(inpo)** by equation(5) at location (i,j)
            with the help of $S_{511\times511}$
            if $(inpo < S(i,j))$ **then**
              pos=S(i,j)-inpo; d=1;
            else
              pos=inpo-S(i,j); d=-1;
            end
            if $pos \neq 0$ **then**
              $BLOCK_{pq}(x,y) = BLOCK_{pq}(x,y) + d$ if W_r(x ,y)=pos,
              where x=1,2,3 and y=1,2,3;
            end
            $SUM = BLOCK_{pq} \otimes W_r$;
            d_{count} = SUM (mod16);
            count=count+1;
          else
            goto **Step-3**
          end
        end
      end
    end
    $W_{i+1} = (W_i \times \kappa - 1) \ \ mod \ \ 9$, where $gcd\ (\kappa, 9) = 1$;
  end
end
Step-3: Produce Original image $I_{256\times256}$ and Data D=$(d_1, d_2, \ldots)$ where d_i is the BCD 4 bits data

Step-4: End.

good results which are shown in Table 1. The formula of PSNR is as follows:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}, \tag{8}$$

Higher the values of PSNR between two images indicates better the quality of the stego image and very similar to the cover image where as low PSNR demonstrates the opposite.

To calculate payload in terms of bits per pixel (bpp) the following equation are used.

$$B = \frac{\left(\lfloor \frac{M+1}{2} \rfloor - 1\right) \times \left(\lfloor \frac{N+1}{2} \rfloor - 1\right) \times 48}{(2M-1)(2N-1)} \tag{9}$$

Here, $M = 256$, $N = 256$, and payload (bpp) $B = 2.96$.

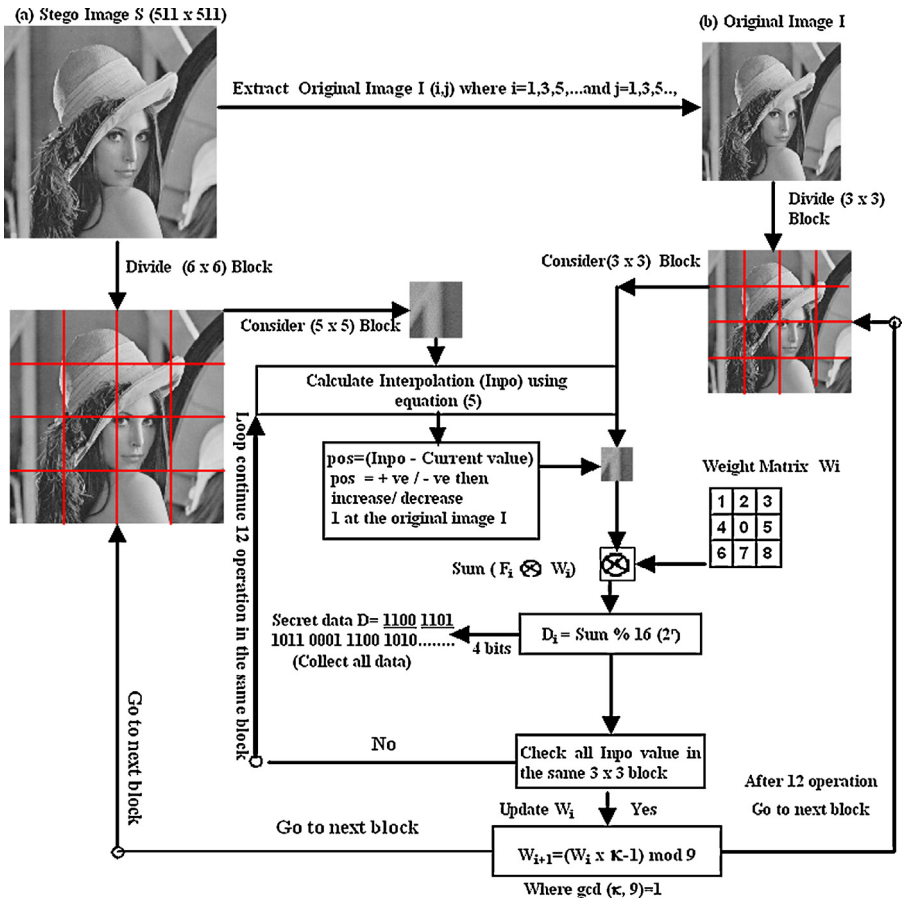


Fig. 2. Schematic diagram of extraction process.

We compared our experimental result with other existing scheme listed in Table 2. The PSNR of Ni et al.'s [13] scheme is 30.88 dB which is 7.09 dB less than our scheme and payload (bpp) 1.11 unit which are 1.85 unit less that our scheme. The both PSNR and payload of our proposed scheme is higher than other existing schemes shown in Table 2. Average PSNR of our proposed scheme is 37.97 dB which is more than the existing matrix based and interpolation based schemes which guarantees good visual quality. The payload is very high compared with other existing schemes and it is 2.96 (bpp). In terms of security on matrix embedding method a shared secret key κ is used to update weighted matrix for each new block. The security is increased due to application of different weighted matrix in different blocks.

5. Steganalysis

Steganalysis is the art of discovering whether or not a secret message is exist in a suspected image. Steganalysis does not however consider the successful extraction of the message. Now a days, steganographic systems does not achieve perfect security. So, they all leave hints of embedding in the stegogramme. This gives the steganalyst a useful way in to identifying whether a secret message exists or not. Steganalyst perform this work in various ways. The way is divided into two main categories-Targeted and Blind steganalysis. Some of the targeted steganalysis are visual attack, statistical attack and structural attack and one of the famous blind steganalysis method is RS analysis. Here we analyze our scheme by RS analysis and find result of relative entropy.

5.1. RS analysis

We analyzed our stego image by the RS analysis. Let us assume that we have a cover image of size ($M \times N$). In RS analysis method, first divide the stego image into disjoint groups G of n adjacent pixels ($x_1, \ldots, x_n$). Each pixel value is in a set P that is $p = \{0, 1, \ldots, 255\}$. Here, each group consists of 4 consecutive pixels in a row. Define a discrimination function f that returns a real number $f(x_1, \ldots, x_n) \in R$ to each pixel group $G = ((x_1, \ldots, x_n))$. The main goal of use the discrimination function is to identify the "Smoothness" or "Regularity" of each group of pixels G. The discrimination function f is define as:

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \tag{10}$$

An invertible function F is define which is operates on P, called "flipping". Flipping consists of two-cycles which permutes the pixels value. So, F^2 = Identity or $F(F(x)) = x$ for all x belongs to P. Flipping the LSB of each pixel value and the corresponding permutation F1 is: $0 \leftrightarrow 1, 2 \leftrightarrow 3, \ldots, 254 \leftrightarrow 255$. Define another function, named shift LSB flipping and treated as $F-1$. So the permutation $F-1$: $-1 \leftrightarrow 0$, $1 \leftrightarrow 2, \ldots, 255 \leftrightarrow 256$. In the other word, F_{-1} flipping can be define as:

$$F_{-1}(x) = F_1(x+1) - 1, \quad \text{for all } x. \tag{11}$$

There are three types of groups Regular groups (R), Singular groups (S) and Unusable groups (U) which are define depend on the

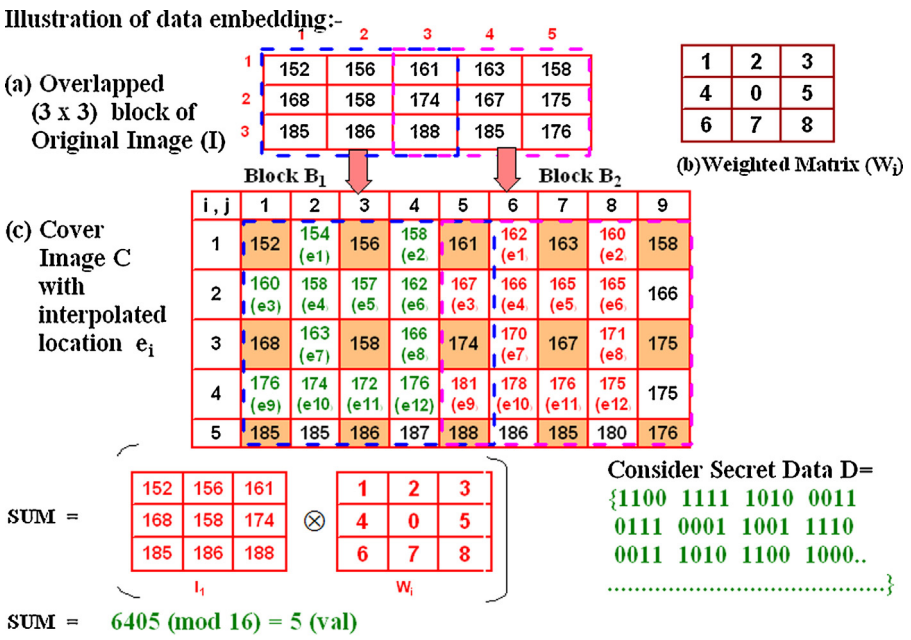


Fig. 3. Example of data embedding.

discrimination function f and the flipping operation F. Depending on the condition groups are define below.

$$\begin{cases} G \in R & \text{if} f(F(G)) > f(G) \\ G \in S & \text{if} f(F(G)) < f(G) \\ G \in U & \text{if} f(F(G)) = f(G) \end{cases} \tag{12}$$

where $F(G) = (F(x_1), \ldots, F(x_n))$. The flipping operation will be executed with the help of a mask value M, which is a n tuples with values -1, 0, and 1. The flipped group $F_M(G)$ is defined as $(F_M(1)(x1), F_M(2)(x2), \ldots, F_M(n)(xn))$. The RS analysis based on analyzing how the number of regular and singular groups changes with the increased message length embedded in the LSB plane. Then calculate the value of RS analysis using the following equation.

$$\frac{(|R_M - R_{-M}| + |S_M - S_{-M}|)}{(R_M + S_M)} \tag{13}$$

where R_M and R_{-M} is the total number of regular group with mask M and $-M$ respectively. S_M and S_{-M} is the total number of singular group with mask M and $-M$ respectively. When the value of RS analysis is closed to zero means the scheme is secure. It is observed from Table 4 that the values of R_M and R_{-M}, S_M and S_{-M} are nearly equal. Thus rule $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ is satisfied for the stego image in our scheme. So, the proposed method is secure against RS attack. In our experiment, the ratio of R and S lies between 0.0034 and 0.0065 for the lena stego image. Other values are shown in the Table 4. The RS value of original image are listed in Table 3.

5.2. Relative entropy

To measure the security in our proposed method, the relative entropy (R) between the probability distributions of the original image (P) and the stego image (Q) is calculated by

$$R(Q||P) = \sum q(x) \log \frac{q(x)}{p(x)} \tag{14}$$

When relative entropy between two probability distribution functions is zero then the system is perfectly secure. $R(Q||P)$ is a nonnegative continuous function and equals to zero if and only

Example of data extraction:

(a) Stego Image S =

i,j	1	2	3	4	5	6	7	8	9
1	152	161	156	161	161	162	163	160	158
2	155	151	161	156	167	166	165	165	166
3	168	171	158	171	174	170	167	171	175
4	171	181	174	172	181	178	176	175	175
5	185	185	186	187	188	186	185	180	176

The receiver first create the Original Image I from the Stego Image S by taking row wise 1,3,5, 7, 9 location and column wise 1,3,5 location..

(b) Original Image I =

	1	2	3	4	5
1	152	156	161	163	158
2	168	158	174	167	175
3	185	186	188	185	176

Block I_1

i,j	1	2	3
1	152	156	161
2	168	158	174
3	185	186	188

Block I_2

i,j	1	2	3
1	161	163	158
2	174	167	175
3	188	185	176

Calculate the interpolation value inpo as per using equation-10 ; The interpolate value inpo= 154 at e1 location then find the difference d= S[1,3]-inpo = 161-154=7; So, Increase 1 at I_1[3,1] i.e. 186+1=187 then performed entry wise multiplication with weighted matrix Wi

SUM (S) =

	1	2	3
1	152	156	161
2	168	158	174
3	185	187	188

I_1

$\otimes$

1	2	3
4	0	5
6	7	8

W_i

$= 6412 \pmod{16} = 12 = 1100.$

Repeat this process and extract the secret data D. Finally , one can get data as well as original image I.

Fig. 4. Example of data extraction.

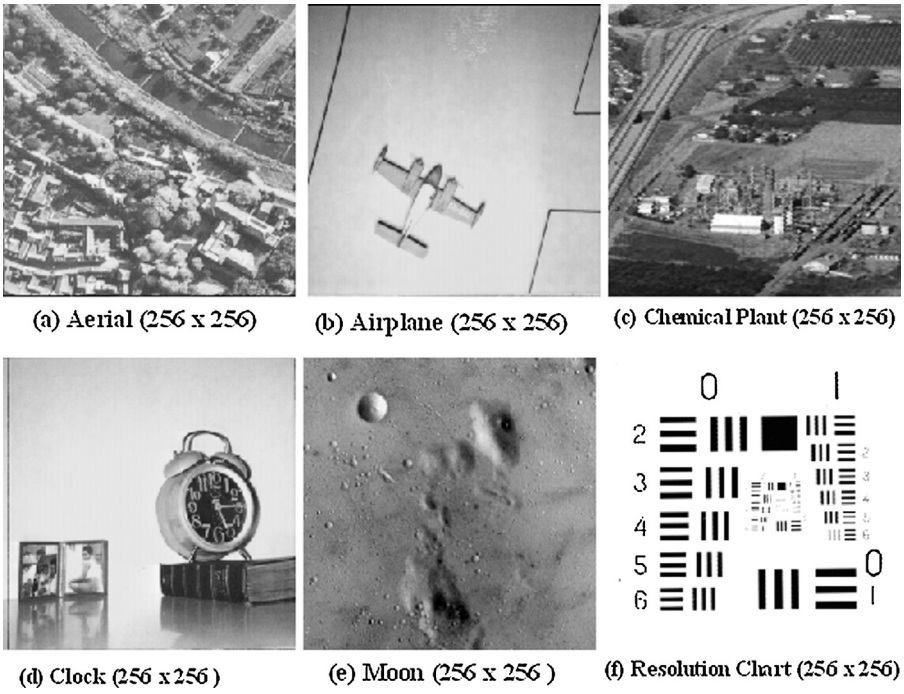
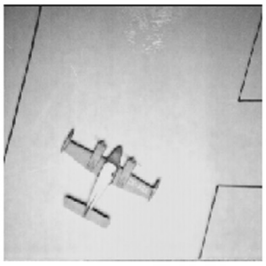


Fig. 5. Standard test images (256 × 256) pixel.

(a) Aerial (511 x 511)

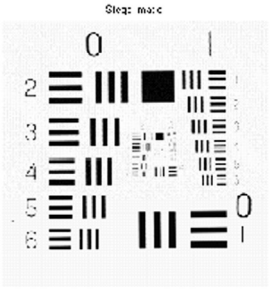


(b) Airplane (511 x 511)

(c) Chemical Plant (511 x 511)

(d) Clock (511 x 511)

(e) Moon (511 x 511)



(f) Resolution Chart (511 x 511)

Fig. 6. Six stego image with (511 × 511) pixel after embedding maximum secret data (776,224) bits.

(a) Camera Man 256 x 256 pixel

Fig. 7. Input image as secret data with (382 × 254) pixel (776,224 bits).

if p and q are coincide. Thus $R(Q||P)$ can be normally considered as a distance between the measures p and q. Relative entropy of the probability distribution of the original image and the stego image varies depending upon number of bits of secret message. In our experiment, it is shown that when the number of characters in the secret message increases, the relative entropy in stego image is also increases. The relative entropy in our experiment is varies between very small which implies the proposed scheme provides secure hidden communication. Relative entropy values are listed in Table 5.

Table 1
PSNR of cover image and stego image.

PSNR (dB) with capacity (bits) and payload			
Image I	Capacity (bits)	PSNR	Avg. PSNR
Lena	260,096	40.85	37.97
	560,000	37.24	
	776,224	35.80	
Moon surface	260,096	40.87	37.97
	560,000	37.24	
	776,224	35.81	
Arial	260,096	40.83	37.96
	560,000	37.24	
	776,224	35.8	
Airplane	260,096	40.82	37.96
	560,000	37.24	
	7,76,224	35.81	
Clock	260,096	40.85	37.96
	560,000	37.23	
	776,224	35.8	
Resolution	260,096	40.85	38.02
	560,000	37.41	
	776,224	35.81	
Chemical	260,096	40.86	37.97
	560,000	37.24	
	776,224	35.81	

Table 2
Comparison with existing scheme.

Scheme	Average PSNR (dB)	Payload (bpp)
Ni et al.'s [13]	30.88	1.11
Jung et al.'s [17]	33.24	0.96
Lee et al.'s [18]	33.79	1.59
Tang et al. [1]	33.85	1.79
Proposed	37.97	2.96

Table 3
RS analysis for original image.

Image	R_M	R_{-M}	S_M	S_{-M}	RS value
Lena	52,649	52,651	0	0	0.0003
Moon Surface	24,864	24,926	12,873	12,892	0.0022
Arial	51,628	51,089	1	12	0.0107
Airplane	36,778	37,061	4604	4473	0.0100
Clock	57,827	57,828	0	0	0.00001

6. Attacks

6.1. Histogram attack

Fig. 8 described the histogram of the cover and stego image and their difference histogram. The stego image are produced from cover image employing the maximum data hiding capacity. It is observed that the shape of the histogram is preserved after embedding the secret data. Histogram of cover image is represented as h whereas histogram of stego image is represented as h'. The change of histogram can be measured by

$$D_h = \sum_{m=1}^{255} |h'_m - h_m| \tag{15}$$

Table 4
RS analysis for stego image.

Image	Secret data (bits)	RS values of stego image				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Lena	260,096	22,838	22,763	11,743	11,701	0.0034
	560,000	22,356	22,119	15,041	15,058	0.0068
	776,224	21,788	21,641	17,258	17,366	0.0065
Moon surface	260,096	24,051	24,169	14,628	14,503	0.0063
	560,000	23,503	23,543	16,833	16,883	0.0022
	776,224	22,934	22,942	18,381	18,376	0.0010
Arial	260,096	19,801	19,465	11,506	11,618	0.0043
	560,000	20,192	19,788	14,115	14,337	0.0182
	776,224	19,978	20,124	15,995	16,097	0.0069
Airplane	260,096	33,190	33,666	8359	8190	0.0155
	560,000	28,070	28,319	14,078	13,951	0.0089
	776,224	23,202	23,765	18,698	16,231	0.0246
Clock	260,096	29,089	29,541	9901	9843	0.0131
	560,000	24,499	24,522	15,302	15,323	0.0011
	776,224	22,998	22,738	18,032	18,298	0.0128

Table 5
Relative entropy between original image and stego image.

Image	Data(bits)	Entropy (I)	Entropy (S)	Entropy difference
Lena	260,096	7.4429	7.4380	0.0049
	560,000		7.4524	0.0009
	776,224		7.4622	0.0193
Moon surface	260,096	6.6752	6.6866	0.0114
	560,000		6.6967	0.0215
	776,224		6.7044	0.0292
Arial	260,096	7.2988	7.2985	0.0003
	560,000		7.3091	0.0103
	776,224		7.3185	0.0197
Airplane	260,096	6.4568	6.4624	0.0056
	560,000		6.5419	0.0851
	776,224		6.6587	0.2019
Clock	260,096	6.7004	6.7566	0.0562
	560,000		6.9253	0.2249
	776,224		6.9484	0.248

The difference of the histogram is very small. It is observed that, bins close to zero are more in numbers and the bins which are away from zero are less in numbers. This confirm the quality of stego image. There is no step pattern observed which ensure the proposed method is robust against histogram analysis.

6.2. Statistical attack

The proposed scheme is also assessed based on statistical distortion analysis by some image parameters like standard deviation (*SD*) and correlation coefficient (*CC*) to check the impact on image after data embedding. The standard deviation (*SD*) before and after data embedding and correlation coefficient (*CC*) of cover and stego images are summarized in Table 6. Minimizing parameters difference is one of the primary aims in order to get rid of statistical attacks. From the Table 6 it is seen that there is no substantial divergence between the standard deviation of the cover-image and the stego-image. This study shows that the magnitude of change in stego-image based on image parameters is small from a cover image. Since the image parameters have not changed much, the method offers a good concealment of data and reduces the chance of the secret data being detected. Thus, it indicates a perfectly secure steganographic system.

6.3. Attacks with unknown weighted matrix and secret key

The proposed scheme constructs stego images which protect original information by hiding secret information using weighted

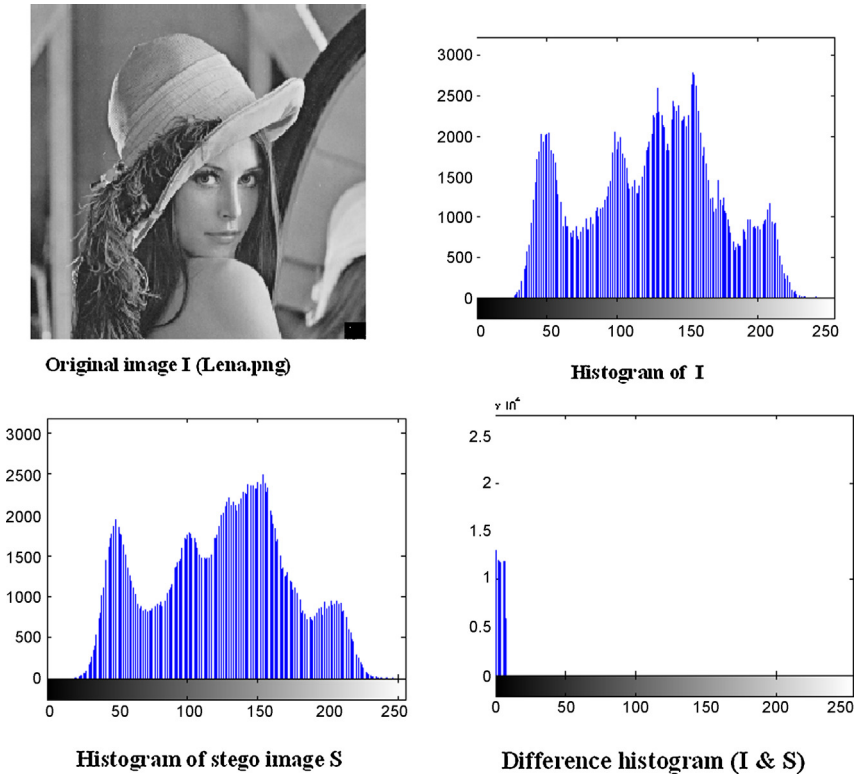


Fig. 8. Histogram of original and stego image and their difference.

Table 6
Standard deviation (SD) and correlation coefficient.

Image	Standard deviation(SD)		Correlation coefficient
	Image I	Stego Image S	I &S
Lena	47.8255	47.3553	0.9823
Moon surface	27.1998	27.4773	0.9895
Arial	45.0844	44.1284	0.9686
Airplane	32.0451	32.3063	0.9924
Clock	57.3003	57.4202	0.9976

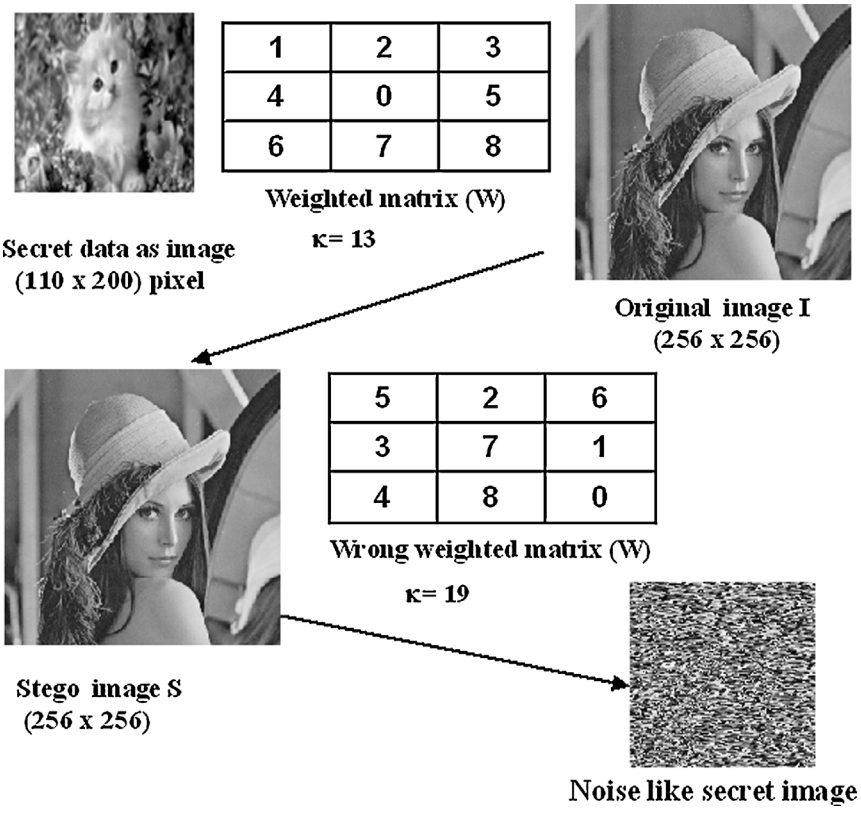


Fig. 9. Noise like secret data with wrong secret key.

matrix. We embed the data embedding position (*pos*), not the original information within stego image. We used secret key κ to update weighted matrix for each new block. The scheme is secure to prevent possible malicious attacks. The Fig. 9 shows the revelation example where with wrong key and wrong weighted matrix are used to revel the hidden message. If the malicious attacker holds the original image and stego image and is fully aware of the proposed scheme, the hidden message still cannot be correctly revealed without knowing the correct secret key and correct weighted matrix. For example, Fig. 9 shows stego image derived from lena image using correct weighted matrix and secret key which are different from that used to construct without knowing the weighted matrix and secret key. The result indicate that the attacker only acquires noise-like images when applying incorrect weighted matrix and secret key to reveal the hidden message. Furthermore, the attacker may employ the brute force attack that tries all possible permutation to reveal the hidden message. Maximum possible weighted matrix to embed r bits data length in each block are $(2^{r-1}+1)!$. We are using $(M \times N)$ original matrix and partitioned (3×3) blocks. Total number of blocks are $(M/3 \times N/3)$ and each block used a modified weighted matrix. So, the number of required trials to reveal the hidden message are $((2^{r-1}+1)!)^{(M/3\times N/3)}$. In our scheme, for (256×256) image with $r=4$, number of trails will be $(362, 880)^{7225}$ which are computationally infeasible for current computers. The proposed scheme achieve stronger robustness against several attacks when compared with existing data hiding scheme. Furthermore, the secret information can be retrieved without encountering any loss of data and recovered original image successfully from stego image.

7. Conclusion

A very high capacity reversible data hiding method using weighted matrix are proposed in this paper. We modified the interpolation technique where average value are inserted between zooming and incorporate weighted matrix for data embedding. We simple modify weighted matrix using secret key κ to enhanced security in data hiding. Also weighted matrix based data hiding was not reversible, here, in this scheme, we introduced reversible data hiding in case of weight-matrix based data hiding scheme. In this scheme we achieved good PSNR and high capacity data embedding as 2.97 bpp. Also we tested our scheme using RS analysis, calculated some statistical analysis data on stego image like Relative entropy, standard deviation and correlation coefficient which gives promising result. We have tested our scheme by several steganographic attacks like histogram attacks and brute force attacks with wrong

weighted matrix and wrong shared secret key, we observed that the scheme is secure and robust against such attacks.

References

[1] M. Tang, J. Hu, W. Song, A high capacity image steganography using multi-layer embedding, Elsevier Science, Optik 125 (15) (2014) 3972–3976.
[2] C. Chan, L. Cheng, Hiding data in images by simple LSB substitution, Pattern Recogn. 37 (3) (2004) 474–496.
[3] C. Chang, J. Hsiao, C. Chan, Finding optimal least-significant-bits substitution in image hiding by dynamic programming strategy, Pattern Recogn. 36 (7) (2003) 1583–1595.
[4] R. Wang, C. Lin, J. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern Recogn. 34 (3) (2001) 671–683.
[5] C. Chang, C. Chan, Y. Fan, Image hiding scheme with modulus function and dynamic programming, Pattern Recogn. 39 (6) (2006) 1155–1167.
[6] C. Thien, J. Lin, A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function, Pattern Recogn. 36 (12) (2003) 2875–2881.
[7] S.J. Wang, Steganography of capacity required using modulo operator for embedding secret image, Appl. Math. Comput. 164 (1) (2005) 99–116.
[8] F.X. Yu, H. Luo, S.C. Chu, Loss less data hiding for halftone images, in: J.-S. Pan, H.-C. Huang, L.C. Jain (Eds.), in: Information Hiding and Applications, vol. 227, SCI, Springer, Heidelberg, 2009, pp. 181–203.
[9] Z.M. Lu, H. Luo, J.-S. Pan, Reversible watermarking for error diffused halftone image using statistical features, in: Y.Q. Shi, B. Jeon (Eds.), IWDW 2006. LNCS, Springer, Heidelberg, 2006, pp. 71–81, Vol. 4283.
[10] B.K. Lien, Y. Lin, High-capacity reversible data hiding by maximum-span pairing, Multimed. Tools Appl. 52 (2011) 499–511.
[11] J.S. Pan, H. Luo, Z.M. Lu, A loss less watermarking scheme for halftone image authentication, Int. J. Comput. Sci. Netw. Secur. 6 (2b) (2006) 147–151.
[12] P.-S. Liao, J.-S. Pan, Y.-H. Chen, B.Y. Liao, A loss less watermarking technique for halftone images, in: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), KES 2005. LNCS (LNAI), Springer, Heidelberg, 2005, pp. 593–599, Vol. 3682.
[13] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol. 16 (3) (2006) 354–362.
[14] C.C. Lin, W.L. Tai, C.C. Chang, Multilevel reversible data hiding based on histogram modification of difference images, Pattern Recogn. 41 (35) (2008) 82–91.
[15] P.Y. Tsai, Y.C. Hu, H.L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, Signal Process. 89 (11) (2009) 29–43.
[16] F. Peng, X. Li, B. Yang, Adaptive reversible data hiding scheme based on integer transform, Signal Process. 92 (2012) 54–62.
[17] K. Jung, K. Yoo, Data hiding method using image interpolation, Comput. Stand. Interfaces 31 (2009) 465–470.
[18] C. Lee, Y. Huang, An efficient image interpolation increasing payload inreversible data hiding, Expert Syst. Appl. 39 (2012) 6712–6719.
[19] Y.-C. Tseng, Y.-Y. Chen, H.-K. Pan, A secure data hiding scheme for binary images, IEEE Trans. Commun. 50 (8) (2002) 1227–1231.
[20] L. Fan, T. Gao, Y. Cao, Improving the embedding efficiency of weight matrix-based steganography for grayscale images, Comput. Electr. Eng. 39 (2013) 873–881.
[21] University of Southern California, The USC-SIPI Image Database, 2015 http://sipi.usc.edu/database/database.php.

Dual-Image Based Reversible Data Hiding Scheme Using Pixel Value Difference Expansion

Biswapati Jana[1], Debasis Giri[2] and Shyamal Kumar Mondal[3]
(Corresponding author: Biswapati Jana)
Department of Computer Science, Vidyasagar University[1]
Midnapore, Pin-721102, India
(Email: biswapatijana@gmail.com)
Department of Computer Science and Engineering, Haldia Institute of Technology[2]
Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University[3]
(Received June 22, 2015; revised and accepted Aug. 12 & Aug. 22, 2015)

Abstract

In this paper, we propose a dual-image based reversible data hiding scheme. Here, we divide a secret message into sub-stream of size n bits, where $n-1$ bits are embedded using Pixel Value Differencing (PVD) and 1 bit is embedded using Difference Expansion (DE). We consider two consecutive pixels from cover image, calculate the difference between them and then embed $n-1$ bits secret message by modifying the pixel pair. Again, we consider that modified pixel pair to embed 1 bit secret message using embedding function. After that, we distribute these two stego pixel pairs among dual image depending on a shared secret key bit stream. At the receiver end, we extract the secret message successfully and recover original cover image from dual stego image without any distortion. Finally, we compare our scheme with other state-of-the-art methods and obtain reasonably better performance in terms of data embedding capacity.

Keywords: Difference expansion, dual image, pixel value differencing, reversible data hiding

1 Introduction

Steganography is one of the most commonly used protective method for information security. Steganography can be classified into two categories: irreversible and reversible. In irreversible technique, the secret data can be embedded and extracted successfully, but the original image might not be recovered [1, 5, 6, 8, 23]. On the other hand, reversible data hiding schemes [9, 10, 16, 17, 19, 20, 22, 24] are capable of embedding the secret message as well as can extract the secret message and recover the original image. Two important measures of reversible data hiding are embedding capacity and distortion of cover work. In recent years, a bunch of research [7, 13, 14, 15, 21, 27] have been performed to improve the embedding capacity and to minimize the distortion which is the objective of data hiding schemes. Wu and Tsai [26] proposed a data embedding method based on PVD, where, the difference of two adjacent pixels in the cover image is calculated. The number of bits to be embedded into these two pixels are determined by their absolute difference and a pre-defined reference table. By modifying these two pixel values, data bits can be embedded. Because the same range in the reference table will be referred before and after data embedding, the same number of secret data bits can be determined and thus the embedded secret data bits can be exactly extracted. Tian [22] proposed a difference expansion data hiding approach to conceal the secret data into the difference of a pair consecutive pixel values with high payload size. Lee et al. [13] utilized the histogram of the difference of pixel values to embed the secret data in host image for improving the quality of marked-images. Ni et al. [16] proposed reversible data hiding technique which is based on histogram shifting with zero or minimum change of the pixel gray values. Being reversible, both the original and the embedded data can be completely restored. Thodi et al. [21] presented a method that combines histogram-shifting and difference expansion reversible data hiding.

Chang et al. [2] proposed dual-image based data hiding technique using exploiting modification direction (EMD) method. They first established a (256×256) modulus function magic matrix. In their scheme, a binary secret message is first converted into secret digits in the base-5 numeral system. Then, two secret digits are taken to embed into a pixel pair at a time by embedding each secret digit into each steganographic image. Lee et al. [7] introduced a lossless steganographic technique that utilized centralized difference expansion to hide more secret data into smoother areas of host image. Later, Lee et al. [12] embed secret data using the four directions of the center point of pixels to obtain the stego-pixels of the two images. Lee and Huang [11] converted secret data into

quinary-based secret symbols and combined every two secret symbols as a set for embedding. Qin et al. [18] embedded the first image using EMD, and the second image through three rules which were dependent on the first image. Lu et al. [14] used the least-significant-bit (LSB) matching method for embedding. They obtained the stego-pixels of two images through the modulus function and the LSB, checked whether the stego-pixels are reversible via an averaging method, and then modified the non-reversible stego-pixels based on a rule table to successfully restored the image. Lee et al. [12] embedded secret data using directions to achieve high image quality, but the embedding capacity could only reach 0.75 bits per pixel (bpp). Chang et al. [2] embed secret data through the modulus function matrix to achieve a higher capacity that is 1.00 bpp, but image quality was inferior to that using the method by Lee et al. Thus, the challenge to enhance embedding capacity while maintaining high image quality through the use of dual-image techniques is still an important issue.

In this paper, we introduced a new dual-image based reversible data hiding scheme through Pixel Value Difference Expansion (PVDE).

- Our motivation is to enhance the embedding capacity and achieve reversibility in data hiding. Data embedding using PVD was not reversible. We have applied DE data embedding scheme to keep the distance parameter of sub range of reference table within the pixel pair. The lower bound of sub range of reference table help us to achieve reversibility in PVDE. The proposed scheme also enhance embedding capacity.

- One of the important modification that we have propose in our scheme is uniform sub range in the reference table. In PVD, the width of sub range varies and the number of embedding bits depends on the pixel value difference. More number of data bits are embedded in the complex area of an image which will effect more. To maintain the uniform effect after data embedding in all area, we propose uniform width of sub range in the reference table. Although data could be embedded without reference table, we use reference table to make PVD as reversible. The lower label of sub range in each embedding pair is essential for PVD to recover original image.

- Another motivation is to enhance security in data hiding. We distribute modified pixel pair among dual stego image, stego major (SM) and stego auxiliary (SA) based on shared secret key bit stream. The secret message bits are distributed among dual image. The receiver applies extraction technique using either PVD or DE that depends on the share secret key. Without key none can extract secret message. Finally, we recover original image using our extraction algorithm from dual image without any distortion.

The rest of the paper is organized as follows. Section 2 describes some preliminary techniques of data hiding scheme. Proposed data hiding scheme PVDE in detail is discussed in Section 3. The issue regarding overflow and underflow situation are described in Section 4. Experimental results with comparisons are discussed in Section 5. Section 6 present security analysis. Finally, we conclude our paper with some interesting insights and possible future directions in Section 7.

2 Preliminaries

Reversible data hiding become a very important and challenging task in hidden data communication specially in medical and military application for ownership identification, authentication and copy right protection. We propose dual-image based reversible data hiding scheme called PVDE. In this section, Wu and Tsai's PVD and Tian's DE techniques are discussed briefly.

2.1 Wu and Tsai's Scheme

Pixel Value Differencing (PVD), proposed by Wu and Tsai [26] is one of the popular data hiding techniques in spatial domain. Consider a two consecutive pixels P_x and P_{x+1} from cover image C of size $(M \times N)$. The difference value d of P_x and P_{x+1} can be derived by

$$d = |P_x - P_{x+1}|.$$

A reference table R is used which consists of n contiguous sub-blocks with fixed interval. The main function of the reference table is to provide data hiding information. Each sub-range has its lower bound (lb) and upper bound (ub) values and the width w of each sub-range is selected to be a power of 2. The hiding capacity of two consecutive pixels can be obtained by

$$t = \lfloor log_2\ w \rfloor. \tag{1}$$

Here, t is the number of bits that is hidden within pixel pair. A new parameter $d^{'}$ is generated using

$$d^{'} = m_1 + lb.$$

Now the secret data is embedded into pixel pair (P_x, P_{x+1}) by modifying it such that d and $d^{'}$ belongs to the same range in the reference table. The details of the embedding criteria are as follows:

$$(P_x^{'}, P_{x+1}^{'}) = \begin{cases} (P_x + \lceil d^{''}/2 \rceil, P_{x+1} - \lfloor d^{''}/2 \rfloor), & \\ \text{if } P_x \geq P_{x+1} \text{ and} & d^{'} > d; \\ (P_x - \lceil d^{''}/2 \rceil, P_{x+1} + \lfloor d^{''}/2 \rfloor), & \\ \text{if } P_x < P_{x+1} \text{ and} & d^{'} > d; \\ (P_x - \lceil d^{''}/2 \rceil, P_{x+1} + \lfloor d^{''}/2 \rfloor), & \\ \text{if } P_x \geq P_{x+1} \text{ and} & d^{'} \leq d; \\ (P_x + \lceil d^{''}/2 \rceil, P_{x+1} - \lfloor d^{''}/2 \rfloor), & \\ \text{if } P_x < P_{x+1} \text{ and} & d^{'} \leq d; \end{cases}$$

where $d^{''} = |d^{'} - d|$. An illustration of how $P_x^{'}$ and $P_{x+1}^{'}$ can be adjusted by Wu and Tsai 's scheme for the purpose of hiding secret data is shown in Figure 1. The recovery

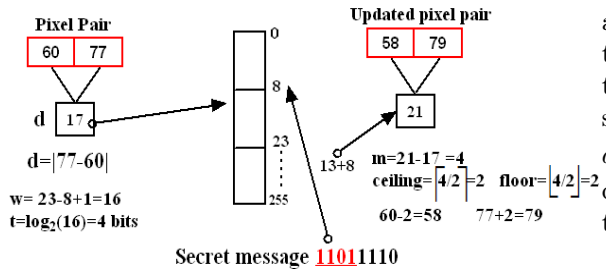


Figure 1: Data embedding through PVD with example

process of Wu and Tsai 's method is quite simple and easy. Given two consecutive pixels $P_x^{'}$ and $P_{x+1}^{'}$ of the stego image, we compute their difference value $d^{'}$ and obtain $d^{'}$=$|P_{x+1}^{'}$ - $P_x^{'}|$. Then we use the original reference table R in the embedding phase to obtain the same sub range. The length t of the hiding capacity can also be gained by using Equation (1). Then we extract message $m_1 = d^{'}$ - lb and convert the decimal value m_1 into a binary string whose length is t bits. For example, in Figure 1, m_1= $21-8=(13)_{10}$ and $t=4$, and then secret data $(1101)_2$ is extracted.

2.2 Tian 's Scheme

Tian [22] presented a reversible data hiding technique based on a difference expansion for gray-scale images. Consider a pixel pair of cover image P_x and P_{x+1}. After embedding 4 bits secret data using PVD, we obtained modified pixel $P_x^{'}$ and $P_{x+1}^{'}$. For embedding secret data within consecutive pixel pair $P_x^{'}$ and $P_{x+1}^{'}$, where $0\leq$ ($P_x^{'}$, $P_{x+1}^{'}$) ≤ 255 the following process is discussed. The average value A and the difference value d is computed by

$$A=\lfloor\frac{P_x^{'}+P_{x+1}^{'}}{2}\rfloor,\ d=|P_x^{'}-P_{x+1}^{'}|. \quad (2)$$

The inverse integer transform of Equation (2) is

$$P_x^{'}=A+\lfloor\frac{d+1}{2}\rfloor, P_{x+1}^{'}=A-\lfloor\frac{d}{2}\rfloor. \quad (3)$$

Such a transform in Equation (2) and Equation (3) are called integer Haar wavelet transform or S transform. Obviously, the transform is a one-to-one correspondence between ($P_x^{'}$, $P_{x+1}^{'}$) and (A,d). That means, it meets the requirement of reversibility. Tian expands the difference twice for vacate a space and embed a secret bit s, where $s\in\{0,1\}$ is the binary secret and generates a new difference value $d^{'}$ by

$$d^{'}=2\times d+s.$$

The new pixel values $P_x^{''}$ and $P_{x+1}^{''}$ are obtained by

$$(P_x^{''},P_{x+1}^{''})=(A+\lfloor\frac{d^{'}+1}{2}\rfloor, A-\lfloor\frac{d^{'}}{2}\rfloor).$$

Finally, the embedding operation is completed, and it produces a stego-image pixel pair by modifying ($P_x^{'}$ and $P_{x+1}^{'}$) to ($P_x^{''}$ and $P_{x+1}^{''}$). Figure 2 is the illustration of Tian's difference expansion scheme. During extraction the secret message, the difference value of consecutive pixel pair ($P_x^{''}$, $P_{x+1}^{''}$) is obtained by calculating $d^{'}=|P_x^{''}-P_{x+1}^{''}|$. The secret bit s can be extracted by computing $s=d^{'}\ mod\ 2$. Then, the average value A and the original difference value d are obtained by

$$A^{'} = \lfloor\frac{P_x^{''}+P_{x+1}^{''}}{2}\rfloor$$
$$d = \lfloor\frac{d^{'}}{2}\rfloor.$$

Now, the original pixel values are recovered using

$$(P_x^{'},P_{x+1}^{'})=(A^{'}+\lfloor\frac{d+1}{2}\rfloor,\ A^{'}-\lfloor\frac{d}{2}\rfloor).$$

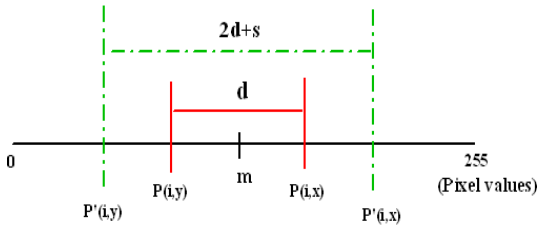


Figure 2: Difference expansion

3 Proposed PVDE Scheme

In this paper, we propose a new reversible data hiding scheme by combining Pixel Value Difference (PVD) and Difference Expansion (DE) on dual image called PVDE. According to this approach, first we have to select two consecutive pixels x_i and x_{i+1} from cover image C. Then we calculate the pixel value difference d between x_i and x_{i+1} that is

$$d=|x_i-x_{i+1}|.$$

The number of secret bits which will be embedded in the cover image is determined with the help of a reference table R. The reference table have equal sub range $[lb,ub]$ having length w that is $w=ub-lb+1$. In our proposed PVDE scheme, w is taken as 16. Hence forth the contiguous sub-ranges are $\{0-15,\ 16-31,\ 32-47,\ \ldots,\ 240-255\}$ which have capability to embed 4 secret bits within each pixel pair through PVDE in cover image. Now to embed 4 bits, two new parameters $d^{'}$ and $d^{''}$ are introduced as follows:

$$d^{'} = lb+m_1$$
$$d^{''} = d^{'}-d$$

where m_1 is decimal value of the secret message of size 4 bits. After that the pixel values x_i and x_{i+1} are adjusted into two new pixel values $x_i^{'}$ and $x_{i+1}^{'}$ by following

modifications.

$$\begin{aligned} x_i^{'} &= x_i - \delta \\ x_{i+1}^{'} &= x_{i+1} + \gamma \end{aligned}$$

where $\delta = \lceil \frac{d^{''}}{2} \rceil$ and $\gamma = \lfloor \frac{d^{''}}{2} \rfloor$. Then we apply DE on the pixels $x_i^{'}$ and $x_{i+1}^{'}$ to embed one bit. Now, we determine the lower range from the reference table R where the difference d belongs to. Then we calculate the parameters h, A and $h^{'}$ as follows

$$\begin{aligned} h &= (d - lb) \\ A &= (x_i^{'} + x_{i+1}^{'})/2 \\ h^{'} &= (2 \times h + m_2) \end{aligned}$$

where m_2 is one bit secret message. After this the pixel pair $x_i^{'}$ and $x_{i+1}^{'}$ are again modified by

$$\begin{aligned} x_i^{''} &= A + \delta_1 \\ x_{i+1}^{''} &= A - \gamma_1 \end{aligned}$$

where $\delta_1 = \lceil (h^{'}/2) \rceil$ and $\gamma_1 = \lfloor (h^{'}/2) \rfloor$. Finally, the stego pixel pairs $(x_i^{'}, x_{i+1}^{'})$ and $(x_i^{''}, x_{i+1}^{''})$ are distributed among dual stego image, Stego Major (SM) and Stego Auxiliary (SA) based on shared secret key K. If $K = 1$, then the pixel pair $(x_i^{'}, x_{i+1}^{'})$ is stored within the stego image SM and the pixel pair $(x_i^{''}, x_{i+1}^{''})$ is stored within the stego image SA. Again if $K = 0$ then the pixel pair $(x_i^{'}, x_{i+1}^{'})$ is stored within the stego image SA and the pixel pair $(x_i^{''}, x_{i+1}^{''})$ is stored within the stego image SM. The detailed schematic diagram of our proposed PVDE method for embedding process are shown in Figure 3 and the corresponding algorithm is shown in Algorithm 1.

Algorithm 1: Data embedding of PVDE

Input: Original image I ($M \times N$), Secret message M, Shared secret key K.
Output: Two stego images, Stego Major (SM) and Stego Auxiliary (SA) of size ($M \times N$).

1: Select pixel pair (x_i, x_{i+1}) from I in raster scan order;
2: Calculate difference $d = |x_i - x_{i+1}|$;
3: Select 4 bits secret message from M and convert into decimal value m_1 and 1 bit as m_2;
4: Calculate $d^{'} = m_1 + lb$; where, lb is the lower bound of the sub range of reference table R in which d belongs to;
5: Calculate $d^{''} = d^{'} - d$;
6: Compute $\delta = \lceil \frac{d^{''}}{2} \rceil$ and $\gamma = \lfloor \frac{d^{''}}{2} \rfloor$;
7: **if** $(x_i > x_{i+1})$ **then**
8: $x_i^{'} = x_i + \gamma$; $x_{i+1}^{'} = x_{i+1} - \delta$;
9: **else**
10: $x_i^{'} = x_i - \delta$; $x_{i+1}^{'} = x_{i+1} + \gamma$;
11: **end if**
12: Calculate $h = (d - lb)$;

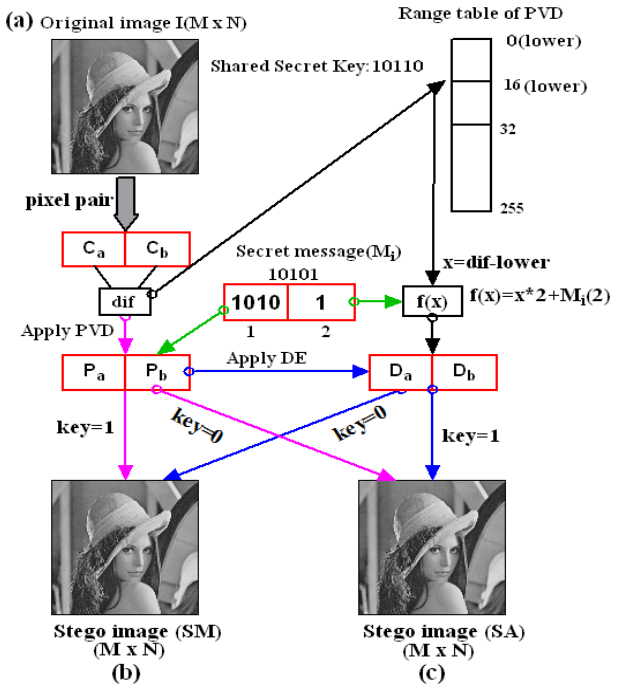


Figure 3: Schematic diagram of PVDE for data embedding process

13: Calculate $h^{'} = 2 \times h + m_2$; where, m_2 is 1 bit secret message;
14: Calculate Average $A = \lfloor \frac{(x_i^{'} + x_{i+1}^{'})}{2} \rfloor$;
15: Calculate $\delta_1 = \lceil \frac{h^{'}}{2} \rceil$; and $\gamma_1 = \lfloor \frac{h^{'}}{2} \rfloor$;
16: **if** $(x_i^{'} > x_{i+1}^{'})$ **then**
17: $x_i^{''} = A + \delta_1$; $x_{i+1}^{''} = A - \gamma_1$;
18: **else**
19: $x_i^{''} = A - \gamma_1$; $x_{i+1}^{''} = A + \delta_1$;
20: **end if**
21: **if** $(K = 1)$ **then**
22: Store $(x_i^{'}, x_{i+1}^{'})$ within stego image SM and store $(x_i^{''}, x_{i+1}^{''})$ within stego image SA;
23: **else**
24: Store $(x_i^{'}, x_{i+1}^{'})$ within stego image SA and store $(x_i^{''}, x_{i+1}^{''})$ within stego image SM;
25: Repeat **Line-1** through **Line-24** until $length(M) = 0$;
26: Dual stego image SM and SA are generated;
27: **end if**
28: End

At the receiver end, both the data extraction and original image reconstruction are performed by taking pixel from both the stego images SM and SA based on K. If $K = 1$, then select pixel pair $(x_i^{'}, x_{i+1}^{'})$ from SM and apply data extraction using PVD and at the same time select pixel pair $(x_i^{''}, x_{i+1}^{''})$ from SA and apply data extraction using DE. If $K = 0$, then apply the pixel pair selection process opposite manner, that means select pixel pair $(x_i^{'}, x_{i+1}^{'})$ from stego image SA and $(x_i^{''}, x_{i+1}^{''})$ from

stego image SM. Now the data extraction and original image reconstruction process are described as follows:

$$\begin{aligned} d &= |x_i^{'} - x_{i+1}^{'}| \\ m_1 &= d - lb \end{aligned}$$

where lb is the lower bound of the sub range of the reference table R to which d belongs to and m_1 is the 4 bits secret data. To recover another secret bit, we perform

$$h^{'} = x_i^{''} - x_{i+1}^{''}$$

and collect one bit secret message (m_2) from LSB of $h^{'}$. To recover the original image, we perform the following calculations

$$\begin{aligned} h &= \lfloor \frac{h^{'}}{2} \rfloor \\ d^{'} &= (h + lb) \\ d^{''} &= d^{'} - d \\ \delta &= \lceil \frac{d^{''}}{2} \rceil \\ \gamma &= \lfloor \frac{d^{''}}{2} \rfloor. \end{aligned}$$

Now, the original image pixel (x_i, x_{i+1}) is recovered by

$$(x_i, x_{i+1}) = \begin{cases} x_i^{'} - \gamma,\ x_{i+1}^{'} + \delta & \text{if } x_i^{'} > x_{i+1}^{'} \\ x_i^{'} + \delta,\ x_{i+1}^{'} - \gamma & \text{otherwise} \end{cases}$$

The extraction process of our proposed PVDE scheme is explained using a schematic diagram in Figure 4. The corresponding algorithm for data extraction and original image reconstruction is explained in Algorithm 2.

Algorithm 2: Data extraction of PVDE

Input: Two stego images SM and SA, Shared secret key K.

Output: Original Image $I(M \times N)$; Secret Message M;

1: Select pixel pair from SM and SA in raster scan order;
2: **if** $(K = 1)$ **then**
3: Collect $(x_i^{'}, x_{i+1}^{'})$ from SM and collect $(x_i^{''}, x_{i+1}^{''})$ from SA;
4: **else**
5: Collect $(x_i^{'}, x_{i+1}^{'})$ from SA and collect$(x_i^{''}, x_{i+1}^{''})$ from SM;
6: **end if**
7: Calculate $d^{'} = |x_i^{'} - x_{i+1}^{'}|$;
8: Secret message $m_1 = d^{'} - lb$, where lb is the lower bound of the sub range of range table R;
9: Calculate $h^{'} = (x_i^{''} - x_{i+1}^{''})$; (Extract secret message bit m_2 from LSB of $h^{'}$);
10: Calculate $h = \lfloor \frac{h^{'}}{2} \rfloor$;
11: Calculate $d = (h + lb)$; where lb is the lower bound of the sub range of the reference table R in which d belongs;

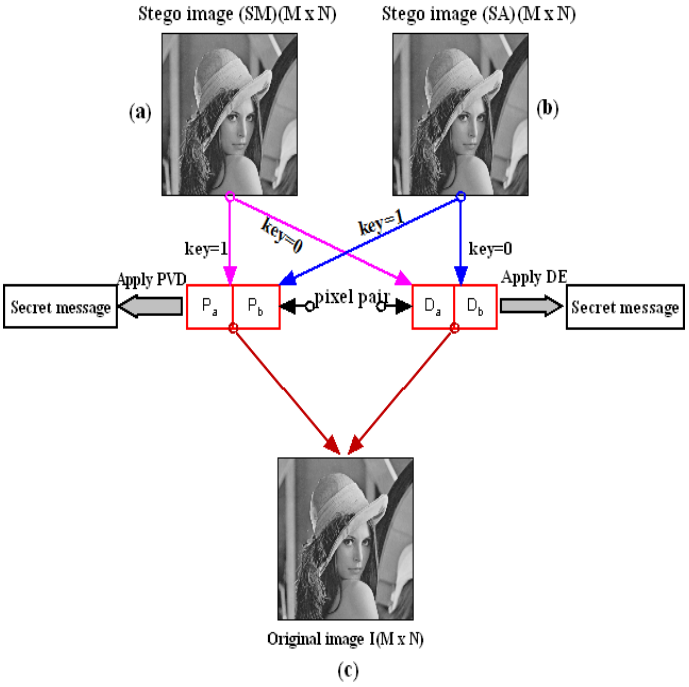


Figure 4: Schematic diagram of PVDE for data extraction process

12: Calculate $d^{''} = d^{'} - d$;
13: Calculate $\delta = \lceil \frac{d^{''}}{2} \rceil$;
14: Calculate $\gamma = \lfloor \frac{d^{''}}{2} \rfloor$;
15: **if** $(x_i^{'} > x_{i+1}^{'})$ **then**
16: $x_i = x_i^{'} - \gamma$; $x_{i+1} = x_{i+1}^{'} + \delta$;
17: **else**
18: $x_i = x_i^{'} + \delta$; $x_{i+1} = x_{i+1}^{'} - \gamma$;
19: **end if**
20: Repeat **Line-1** through **Step-19** until all data are extracted;
21: End

4 Overflow and Underflow

When the stego pixel value cross the upper range of gray scale then overflow occur and cross the lower limit of gray scale then underflow occur. We have use 8 bit image where gray scale is [0-255]. Suppose we have a pixel pair (C_a, C_b) with pixel values $C_a = 250$ and $C_b = 255$ and 4 bits secret data is $(1101)_2$ that is $(13)_{10}$. The difference between two pixels d is $|\ 250 - 255\ |$=5 and the new difference $d^{'}$ is 13+0=13. Therefore, m =13-5=8, c=4 and f=4. After embedding, the stego pixel pair becomes $P_a = 246$ and $P_b = 259$ which cross the upper limit that means $P_b > 255$ which shows overflow problem.

For underflow, suppose C_a=0 and C_b=7 and 4 bits secret data is $(1010)_2$ that is $(10)_{10}$. The difference between two pixels d is $|\ 0 - 7\ |$=7 and the new difference $d^{'}$ is 10+0=10. Therefore, m =10-7=3, c=2 and f=1. The

stego pixel pair becomes $P_a = -2$ and $P_b = 8$. We observe that $P_a < 0$ which shows underflow problem.

To overcome this problem, we do not embed any secret data within those specified pixel pair. We observed that after data embedding, the difference between two pixels is not much more than 31. To overcome the overflow problem, we use difference expansion method and set the difference 32 when data hiding by difference expansion is 0 and subtracting 32 from the average of two pixels. So, the modified pixel pair becomes $(D_a = avg-32, D_b = P_b)$ and set the difference 33 when data is 1 by subtracting 33 from the average of two pixels. So, the modified pixel pair becomes $(D_a = avg - 33, D_b = P_b)$.

To overcome the underflow problem, we set the difference 32 when data is 0 by adding 32 with the average of two pixels. So, the modified pixel pair will be $(D_a = avg + 32, D_b = P_b)$ and set the difference 33 when data is 1 by adding 33 with the average of two pixels. So, the modified pixel pair will be $(D_a = avg + 33, D_b = P_b)$.

In the receiver side, when difference between the pixels D_a and D_b is 32 or 33 the receiver understand that secret message is not embedded within that pair (P_a, P_b) corresponding to (D_a, D_b).

Figure 5: Standard test images with (256×256) pixel

5 Experimental Results and Comparison

In this section, our proposed method (PVDE) is verified and tested using gray scale image of size (256×256) pixels collected from [25] shown in Figure 5. After embedding the secret messages, dual stego image, Stego Major (SM) and Stego Auxiliary (SA) are generated as shown in Figure 6. Our developed algorithms: PVDE embedding and extraction are implemented in MATLAB Version 7.6.0.324 (R2008a). Here, the distortion is measured by means of two parameters namely, Mean Square Error (MSE) and Peak Signal to Noise Ratio $(PSNR)$. The MSE is calculated as follows:

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[X(i,j) - Y(i,j)]^2}{(M \times N)}$$

where M and N denote the total number of pixels in the horizontal and the vertical dimensions of the image respectively.

Figure 6: Dual stego images of (256×256) pixels after data embedding

$X(i,j)$ represents the pixels in the cover image and $Y(i,j)$ represents the pixels of the stego image. The difference between the original and stego images were assessed by the Peak Signal to Noise Ratio $(PSNR)$. The formula of PSNR is as follows:

$$PSNR = 10\ log_{10}\frac{255^2}{MSE}$$

Table 1: Data embedding capacity with PSNR

Image	Data(bits)	PSNR(SM)	PSNR(SA)	Avg. PSNR
cameraman	40,000	43.40	42.72	36.77
	80,000	35.75	38.84	
	1,60,000	30.77	36.19	
	1,63,592	30.35	36.14	
house	40,000	47.00	41.88	38.95
	80,000	40.59	38.53	
	1,60,000	35.84	36.01	
	1,63,592	35.79	35.97	
F16	40,000	47.07	43.29	37.88
	80,000	37.18	39.36	
	1,60,000	31.65	36.48	
	1,63,592	31.62	36.41	
lake	40,000	36.95	43.15	36.08
	80,000	33.47	39.70	
	1,60,000	30.82	37.03	
	1,63,592	30.63	36.93	
Lena	40,000	40.31	43.78	36.93
	80,000	35.31	40.19	
	1,60,000	30.77	37.28	
	1,63,592	30.67	37.18	
livingroom	40,000	38.93	43.47	36.69
	80,000	34.18	40.02	
	1,60,000	31.37	37.19	
	1,63,592	31.31	37.11	
peppers	40,000	39.67	43.47	37.27
	80,000	35.45	39.93	
	1,60,000	32.92	36.98	
	1,63,592	32.86	36.89	
pirate	40,000	39.79	43.75	37.05
	80,000	35.29	40.28	
	1,60,000	31.58	37.15	
	1,63,592	31.48	37.09	
bridge	40,000	34.57	43.54	35.85
	80,000	32.39	40.47	
	1,60,000	30.50	37.51	
	1,63,592	30.44	37.42	
Tiffany	40,000	40.44	43.75	37.36
	80,000	36.32	40.20	
	1,60,000	32.00	37.19	
	1,63,592	31.92	37.12	
Zelda	40,000	42.20	43.76	38.87
	80,000	39.10	40.09	
	1,60,000	36.08	36.98	
	1,63,592	35.86	36.90	
Goldhill	40,000	45.84	42.85	38.66
	80,000	39.77	39.48	
	1,60,000	34.09	36.80	
	1,63,592	34.06	36.76	

Higher the values of PSNR between two images indicates better the quality of the stego image and very similar to the cover image where as low PSNR demonstrates the opposit. Table 1 shows the experimental result upon Cameraman, House, Jet Plane, Lake, Lena, Living Room, Peppers, Pirate, Walk bridge and Woman images. Table 1 shows the average $PSNR$ of SM and SA with cover image. To asses the embedding capacity, we calculate payload (B) in terms of bits per pixel (bpp) using the following expression.

$$B = \frac{(\lfloor \frac{M}{2} \rfloor - 1) \times N \times 5}{(2M \times 2N)}$$

For example, if $M = 512$ and $N = 512$ then $B = \frac{255 \times 512 \times 5}{2 \times (512 \times 512)} = 1.25$. The bpp B of our dual image based PVDE scheme is 1.25.

To measure the complexity, we assume that the size of the cover image is $(M \times N)$ and the data embedding process embed five secret bits within a pixel pair. Two copies of cover image is used to distribute the stego pixel and each pixel pair from cover image produce two copies of pixel pair. So, the time complexity is O(MN). On the other hand, during data extraction, we need to scan the pixel pair from dual image depending on key. So, the time complexity is O($2MN$).

Table 2 lists the average PSNR values with payload of different existing dual image based data hiding scheme. The average PSNR of the stego images of the proposed scheme is lower than the method proposed by Qin et al.'s [18], Lu et al.'s [14, 15], Chang et al.'s [2, 3] and Lee et al.'s [11, 12] schemes. But the average PSNR is higher than the method proposed by Lee et al.'s [10] and Zeng et al.'s [27] schemes. The embedding payload of our scheme is 1.25 bpp which is higher than the other existing dual image based schemes. The embedding payload of the methods proposed by Qin et al. [18] is approximately 0.09 bpp less than that of our proposed PVDE method. The payload of Lu et al. [15] and Chang et al. [2, 3] is approximately 0.25 bpp less than our PVDE method. It is observed that our PVDE is superior than the other dual image based schemes in terms of embedding payload (bpp). From the above discussion, one can conclude that PVDE is better than other existing scheme in terms of payload, and the PSNR is also reasonable which implies the quality of the stego image is good.

Table 2: Comparison of average PSNR and payload (bpp) with existing schemes

Scheme	Avg. PSNR (dB)	Capacity (bpp)
Chang et al.(2007)	45.1225	1.00
Chang et al.(2009)	48.14	1.00
Lee et al. (2009)	52.3098	0.74
Lee et al. (2010)	34.38	0.91
Zeng et al. (2012)	32.74	1.04
Lee and Huang (2013)	49.6110	1.07
Qin et. al. (2014)	52.11	1.16
Lu et al. (2015)	49.20	1.00
Proposed PVDE	38.95	1.25

6 Steganalysis

Steganalysis is the art of discovering whether or not a secret message is exist in a suspected image. Steganalysis does not however consider the successful extraction of the message. Now a days, steganographic systems does not achieve perfect security. So, they all leave hints of embedding in the stegogramme. This gives the steganalyst a useful way in to identifying whether a secret message exists or not. Steganalyst perform this work in various ways. The way is divided into two main categories-Targeted and Blind steganalysis. Some of the targeted steganalysis are visual attack, statistical attack and structural attack and one of the famous blind steganalysis method is RS analysis.

6.1 RS Analysis

We analyzes our stego images by RS analysis [4]. Let us assume that we have a cover image of size (M × N). In RS analysis method, first the stego image is divide into disjoint groups G of n adjacent pixels $(x_1, \ldots, x_n)$. Each pixel value is in a set P that is $p = \{0, 1, \ldots, 255\}$. Here, each group consist of 4 consecutive pixels in a row. Define a discrimination function f that returns a real number $f(x_1, \ldots, x_n) \in R$ to each pixel group $G = (x_1, \ldots, x_n)$. The main goal of use the discrimination function is to identify the "Smoothness" or "Regularity" of each group of pixels G. The discrimination function f is define as:

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$$

An invertible function F is define which is operates on P, called "flipping". Flipping consists of two-cycles which permutes the pixels value. So, F^2= Identity or $F(F(x)) = x$ for all x belongs to P. Flipping the LSB of each pixel value and the corresponding permutation F1 is: $0 \leftrightarrow 1, 2 \leftrightarrow 3, \ldots, 254 \leftrightarrow 255$. Define another function, named shift LSB flipping and treated as F_{-1}. So the permutation F_{-1}: $-1 \leftrightarrow 0, 1 \leftrightarrow 2, \ldots, 255 \leftrightarrow 256$. In the other word, F_{-1} flipping can be define as:

$$F_{-1}(x) = F_1(x + 1) - 1, \quad \text{for all } x.$$

There are three types of groups Regular groups (R), Singular groups (S) and Unusable groups (U) which are define depend on the discrimination function f and the flipping operation F. Depending on the condition groups are define below.

$$\begin{cases} G \in R & \text{if } f(F(G)) > f(G) \\ G \in S & \text{if } f(F(G)) < f(G) \\ G \in U & \text{if } f(F(G)) = f(G) \end{cases}$$

where $F(G) = F(x_1), \ldots, F(x_n)$.
The flipping operation will be execute with the help of a mask value M, which is a n tuples with values -1, 0, and 1. The flipped group $F_M(G)$ is defined as $(F_M(1)(x1)$, $F_M(2)(x2)$, ..., $F_M(n)(xn))$. The RS analysis based on analyzing how the number of regular and singular groups changes with the increased message length embedded in the LSB plane.

Then calculate the value of RS analysis using the following equation.

$$((|R_M - R_{-M}| + |S_M - S_{-M}|)/(R_M + S_M))$$

where R_M and R_{-M} is the total number of regular group with mask M and -M respectively. S_M and S_{-M} is the total number of singular group with mask M and -M respectively. When the value of RS analysis is closed to zero means the scheme is secure. The stego images are tested under the RS analysis. It is observed from Tables 3 and 4 that the values of R_M and R_{-M}, S_M and S_{-M} are nearly equal for stego image SM and SA. Thus rule $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$ is satisfied for the stego image in our scheme. So, the proposed method is secure against RS attack. In our experiment, the ration of R and S lies between 0.0051 to 0.0616 for SM and 0.0043 to 0.0267 for SA of Cameraman image.

Table 3: RS analysis of PVDE method (Stego image SM)

Image	Data	SM				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Cameraman	20000	7118	7107	3551	3594	0.0051
	50000	6768	6851	3944	3895	0.0123
	75000	6304	5947	4943	5279	0.0616
	114582	6207	6035	4997	5173	0.0311
Lena	20000	5617	5607	4067	4068	0.0011
	50000	5563	5476	4291	4337	0.0135
	75000	5636	5539	4517	4589	0.0166
	114582	5641	5387	4509	4709	0.0447
Baboon	20000	5893	5815	4960	5105	0.0205
	50000	5897	5875	5076	5131	0.0070
	75000	6018	5813	5107	5313	0.0369
	114582	5844	5986	5256	5123	0.0248

Table 4: RS analysis of PVDE method (Stego image SA)

Image	Data	SA				
		R_M	R_{-M}	S_M	S_{-M}	RS value
Cameraman	20000	6945	7078	3877	3721	0.0267
	50000	6506	6535	4490	4472	0.0043
	75000	6514	6528	4287	4224	0.0071
	114582	6538	6647	4283	4225	0.0154
Lena	20000	5575	5565	4139	4133	0.0016
	50000	5590	5514	4239	4299	0.0138
	75000	5587	5442	4579	4665	0.0227
	114582	5652	5621	4592	4553	0.0123
Baboon	20000	5876	5881	4995	5092	0.0094
	50000	5821	5878	5121	5147	0.0076
	75000	5895	5827	5196	5283	0.0140
	114582	5874	5830	5194	5206	0.0051

6.2 Relative Entropy

To measure the security in our proposed method, the relative entropy (D) between the probability distributions of the original image (P) and the stego image (Q) is calculated by

$$D(Q||P) = \sum q(x) log \frac{q(x)}{p(x)}.$$

When relative entropy between two probability distribution functions is zero then the system is perfectly secure. D($Q||P$) is a nonnegative continuous function and equals to zero if and only if p and q are coincide. Thus D($Q||P$) can be normally considered as a distance between the measures p and q. Relative entropy of the probability distribution of the original image and the stego image varies depending upon number of bits of secret message. In our experiment, it is shown that when the number of characters in the secret message increases, the relative entropy in stego image is also increases. The relative entropy in our experiment is varies between 0.0027 to 0.0131 for lena image which implies the proposed scheme provides

Table 5: Relative entropy between I and SM

Image	Data(Bytes)	Entropy I	Entropy SM	Difference
Lena	5000	7.4451	7.4451	0.0027
	10000	7.4451	7.4452	0.0058
	20000	7.4451	7.4452	0.0105
	20249	7.4451	7.4453	0.0131
Barbara	5000	7.0480	7.0480	0.0031
	10000	7.0480	7.0482	0.0064
	20000	7.0480	7.0485	0.0112
	20249	7.0480	7.0486	0.0134
Tiffany	5000	7.2925	7.2925	0.0029
	10000	7.2925	7.2925	0.0057
	20000	7.2925	7.2926	0.0122
	20249	7.2925	7.2926	0.0129
Pepper	5000	7.2767	7.2767	0.0039
	10000	7.2767	7.2768	0.0077
	20000	7.2767	7.2770	0.0142
	20249	7.2767	7.2771	0.0169
Gold hill	5000	7.2367	7.2367	0.0034
	10000	7.2367	7.2371	0.0056
	20000	7.2367	7.2375	0.0112
	20249	7.2367	7.2379	0.0143

secure hidden communication. Other relative entropy values with SM are depicted in Table 5.

6.3 Histogram Attack

Figure 7 depicted the histogram of the cover and stego image and their difference histogram are obtained. The stego image are produced from cover image employing the maximum data hiding capacity. It is observed that the shape of the histogram is preserved after embedding the secret data. Histogram of cover image is represented as h whereas histogram of stego image is represented as $h^{'}$. The change of histogram can be measured by

$$D_h = \sum_{m=1}^{255} |h_m^{'} - h_m|.$$

The difference of the histogram is very small. It is observed that, bins close to zero are more in numbers and the bins which are away from zero are less in numbers. This confirm the quality of stego image. There is no step pattern observed which ensure the proposed method is robust against histogram analysis.

6.4 Statistical Attack

The proposed scheme is also assessed based on statistical distortion analysis by some image parameters like Standard Deviation (SD) and Correlation Coefficient (CC) to check the impact on image after data embedding. The SD before and after data embedding and CC of cover and stego images are summarized in Table 6. Minimizing parameters difference is one of the primary aims in order to get rid of statistical attacks. From the Table 6 it is seen that there is no substantial divergence between the SD of the cover-image and the stego-image. This study shows that the magnitude of change in stego-image based on image parameters is small from a cover image. Since the image parameters have not changed much, the method

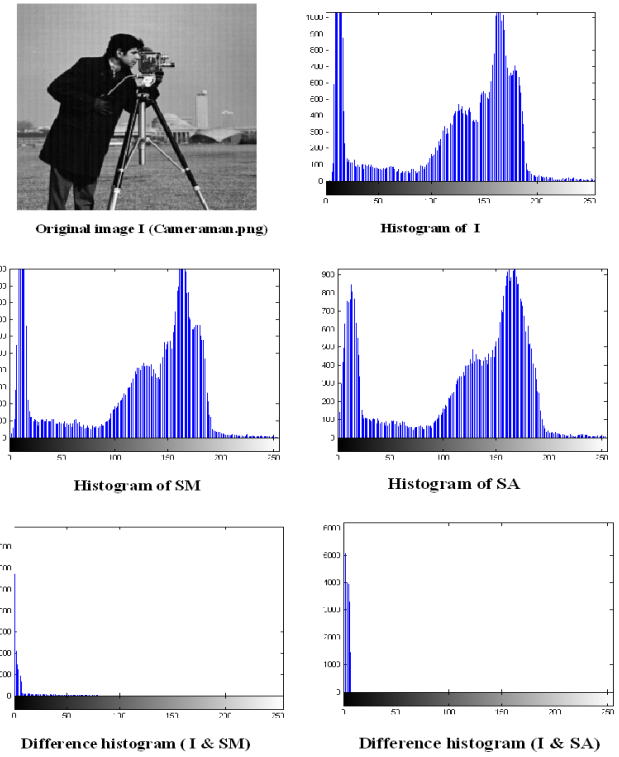


Figure 7: Histogram of Original, SM, SA and difference

Table 6: Standard Deviation (SD) and Correlation Coefficient (CC)

Image	SD			CC		
	I	SM	SA	I&SM	I&SA	SM & SA
Baboon	38.37	37.85	38.54	0.98	0.99	0.97
Cameraman	61.59	61.12	61.73	0.99	0.99	0.99
Lena	47.83	47.43	47.97	0.98	0.99	0.98

offers a good concealment of data and reduces the chance of the secret data being detected. Thus, it indicates a perfectly secure steganographic system.

6.5 Attacks with Unknown Secret Key

We have used 128 bits shared secret key K to distribute pixel among dual images. The scheme is secure to prevent possible malicious attacks. The proposed scheme constructs two stego images which protect original information by hiding secret information in both images SM and SA. The Figure 8 shows the revelation example where with key and without key stego images are used to revel the hidden message. If the malicious attacker holds the original image and dual images and is fully aware of the proposed scheme, the hidden message still cannot be correctly revealed without knowing the correct secret key. The result indicate that the attacker only acquires noise -like images when applying incorrect secret key to reveal the hidden message. Furthermore, the attacker may employ the brute force attack that tries all possible permu-

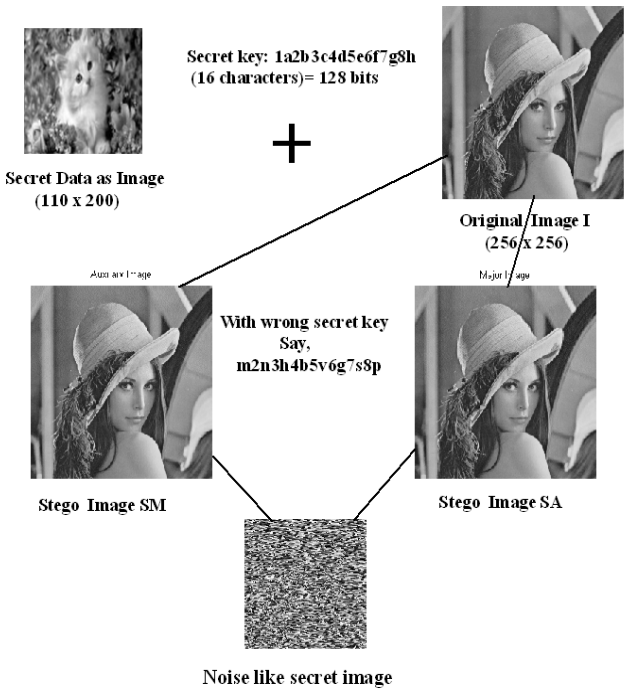


Figure 8: Noise like secret data with wrong secret key

tation to reveal the hidden message. The secret key are 128 bits length, so, the number of required trials to reveal the hidden message are 2^{128} which are computationally infeasible for current computers. The proposed scheme achieve stronger robustness against several attacks when compared with existing data hiding. Furthermore, the secret information can be retrieved without encountering any loss of data and recovered original image successfully from dual image.

7 Conclusion

In this paper, on the basis of pixel value difference and difference expansion a dual image based reversible data hiding scheme (PVDE) is introduced. Here, the reference table is modified allowing fix size four bits data embedding capacity. During difference expansion we keep the difference value of a subrange from the reference table which helps to recover the original image from stego images. In our proposed PVDE method, PVD achieved reversibility which demands the originality of our method. Also PVDE achieves security using the shared secret key by which stego pixels are distributed among two stego images. A shared secret key K has been used which guarantees security. The RS analysis provide low value which fulfilled the art of steganography. The visual attacks are analyzed by histogram analysis and statistical attacks are performed by SD and CC which provide robustness against several attacks. Also, the scheme maintains low relative entropy. In addition, it gains good PSNRs and higher payload than other existing methods of dual image based data hiding.

References

[1] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution", *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, 2004.

[2] C. C. Chang, Y. C. Chou. "Reversible data hiding scheme using two steganographic images" in *IEEE Region 10 Conference on TENCON*, pp. 1–4, 2007.

[3] C. C. Chang, Y. C. Chou, and T. D. Kieu, "Information Hiding in Dual Images with Reversibility", *Proceedings of Third International Conference on Multimedia and Ubiquitous Engineering*, pp. 145–152, 2009.

[4] J. Fridrich, J. Goljan, R. Du, "Invertible authentication", in *Proceedings of the SPIE, Security and Watermarking of Multimedia Contents*, vol. 4314, pp.197208, SanJose, CA, Jan. 2001.

[5] T. D. Kieu and C. C. Chang, "A steganographic scheme by fully exploiting modification directions", *Expert Systems with Applications*, vol. 38, pp. 10648–10657, 2011.

[6] H. J. Kim, C. Kim, Y. Choi, S. Wang, and X. Zhang, "Improved modification direction methods", *Computers and Mathematics with Applications*, vol. 60, no. 2, pp. 319–325, 2010.

[7] C. C. Lee, H. C. Wu, C. S. Tsai, Y. P. Chu, "Lossless Steganographic scheme with Centralized Difference Expansion", *Pattern Recognition*, vol. 41, pp. 2097–2106, 2008.

[8] C. F. Lee, C. C. Chang, P. Y. Pai, and C. M. Liu, "Adjustment hiding method based on exploiting modification direction", *International Journal of Network Security*, vol. 17, no. 5, pp. 607–618, 2015.

[9] C. F. Lee and H. L. Chen, "Adjustable prediction-based reversible data hiding", *Digital Signal Processing*, vol. 22, no. 6, pp. 941–953, 2012.

[10] C. F. Lee, H. L. Chen, and H. K. Tso, "Embedding capacity raising in reversible data hiding based on prediction of difference expansion", *Journal of Systems and Software*, vol. 83, no. 10, pp. 1864–1872, 2010.

[11] C. F. Lee, Yu L. Huang, "Reversible data hiding scheme based on dual stegano-images using orientation combinations", *Telecommunication Systems*, vol. 52, pp. 2237–2247, 2013.

[12] C. F. Lee, K. H. Wang, C. C. Chang, Y. L. Huang, "A reversible data hiding scheme based on dual steganographic images", in *Proceedings of the Third International Conference on Ubiquitous Information Management and Communication*, pp. 228–237, 2009.

[13] S. K. Lee, Y. H. Suh, Y. S. Ho, "Lossless data hiding based on histogram moification of difference images", in *Pacific Rim Conference on Multimedia*, LNCS 3333, pp. 340–347, Springer-Verlag, 2004.

[14] T. C. Lu, C. Y. Tseng, and J. H. Wu, "Dual imaging-based reversible hiding technique using LSB matching", *Signal Processing*, vol. 108, pp. 77–89, 2015.

[15] T. C. Lu, J. H. Wu, and C. C. Huang, "Dual-image-based reversible data hiding method using center folding strategy", *Signal Processing*, vol. 115, pp. 195–213, 2015.

[16] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[17] C. Qin, C. C. Chang, and Y. C. Chen, "Efficient reversible data hiding for VQ-compressed images based on index mapping mechanism" *Signal Processing*, vol. 93, no. 9, pp. 2687–2695, 2013.

[18] C. Qin, C. C. Chang, and T. J. Hsu, "Reversible data hiding scheme based on exploiting modification direction with two steganographic images" *Multimedia Tools and Applications*, pp. 1–12, 2014.

[19] C. Qin, C. C. Chang, Y. H. Huang, and Li T. Liao, "An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 7, pp. 1109–1118, 2013.

[20] C. Qin, C. C. Chang, and Li T. Liao, "An adaptive prediction-error expansion oriented reversible information hiding scheme", *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2166–2172, 2012.

[21] D. M. Thodi, J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking", *IEEE Translations on Image Processing*, vol. 16, no. 3, pp. 1057–7149, Mar. 2007.

[22] J. Tian, "Reversible data embedding using a difference expansion", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[23] Y. Yu Tsai, J. T. Chen, and C. S. Chan, "Exploring LSB substitution and pixel-value differencing for block-based adaptive data hiding", *International Journal of Network Security*, vol. 16, no. 5, pp. 359–364, 2014.

[24] H. W. Tseng and C. P. Hsieh, "Prediction-based reversible data hiding", *Information Sciences*, vol. 179, no. 14, pp. 2460–2469, 2009.

[25] University of Southern California, *The USC-SIPI Image Database*, Sept. 15, 2015. (`http://sipi.usc.edu/database/database.php`)

[26] D. Wu, W. Tsai, "A steganographic method for images by pixel-value differencing", *Pattern Recognition Letters*, vol. 24, pp. 1613–1626, 2003.

[27] X. T. Zeng, Z. Li, L. D. Ping, "Reversible data hiding scheme using reference pixel and multi-layer embedding", *AEU International Journal of Electron Communication*, vol. 66, no. 7, pp. 532–539, 2012.

Biswapati Jana is currently working as an Assistant Professor in the Department of Computer Science, Vidyasagar University, Paschim Medinipur, India. He received his B. Tech. and M. Tech. degrees in Computer Science and Engineering from University of Calcutta in 1999 and 2002 respectively. His research interests include Image Processing, Data Hiding and Steganography. He has published more than ten papers in National and International Conferences.

Dr. Debasis Giri did his masters (M.Tech and M.Sc) both from IIT Kharagpur, India and also completed Doctorate from IIT Kharagpur, India. He is ten-th all India rank holder in Graduate Aptitude Test in Engineering in 1999. He has published more than 25 papers in international journal/ conference. His current research interests include Cryptography, Information Security, E-commerce security and Design & Analysis of Algorithms. He is Editorial Board Member and Reviewer of many International Journals. He is also Program Committee Member of International Conferences. He is a life member of Cryptology Research Society of India.

Dr. Shyamal Kumar Mondal is currently Associate Professor in the Department of Applied Mathematics With Oceanology And Computer Programming, Vidyasagar University. He did his Ph.D. from Vidyasagar University in 2004, M.Tech from ISM, Dhanbad in 1999 and M.Sc from Vidyasagar University in 1994. His research interest include Operations Research, Meteorology, Fuzzy Set Theory, Soft Set Theory, Soft Computing and Data Hiding. He has published more than 50 papers in National and International Journals/Conferences.