EXPLOITING SIMILARITIES BETWEEN SECRET AND COVER IMAGES FOR IMPROVED EMBEDDING EFFICIENCY AND SECURITY IN DIGITAL STEGANOGRAPHY

By

ALAN ANWER ABDULLA

Department of Applied Computing

The University of Buckingham

United Kingdom

A Thesis

Submitted for the degree of Doctor of Philosophy in Computer Science to the School of Science in the University of Buckingham

October 2015

Abstract

The rapid advancements in digital communication technology and huge increase in computer power have generated an exponential growth in the use of the Internet for various commercial, governmental and social interactions that involve transmission of a variety of complex data and multimedia objects. Securing the content of sensitive as well as personal transactions over open networks while ensuring the privacy of information has become essential but increasingly challenging. Therefore, information and multimedia security research area attracts more and more interest, and its scope of applications expands significantly. Communication security mechanisms have been investigated and developed to protect information privacy with Encryption and Steganography providing the two most obvious solutions. Encrypting a secret message transforms it to a noise-like data which is observable but meaningless, while Steganography conceals the very existence of secret information by hiding in mundane communication that does not attract unwelcome snooping. Digital steganography is concerned with using images, videos and audio signals as cover objects for hiding secret bit-streams. Suitability of media files for such purposes is due to the high degree of redundancy as well as being the most widely exchanged digital data. Over the last two decades, there has been a plethora of research that aim to develop new hiding schemes to overcome the variety of challenges relating to imperceptibility of the hidden secrets, payload capacity, efficiency of embedding and robustness against steganalysis attacks. Most existing techniques treat secrets as random bit-streams even when dealing with non-random signals such as images that may add to the toughness of the challenges. This thesis is devoted to investigate and develop steganography schemes for embedding secret images in image files. While many existing schemes have been developed to perform well with respect to one or more of the above objectives, we aim to achieve optimal performance in terms of all these objectives. We shall only be concerned with embedding secret images in the spatial domain of cover images.

The main difficulty in addressing the different challenges stems from the fact that the act of embedding results in changing cover image pixel values that cannot be avoided, although these changes may not be easy to detect by the human eye. These pixel changes is a consequence of dissimilarity between the cover LSB plane and the secret

image bit-stream, and result in changes to the statistical parameters of stego-image bit-planes as well as to local image features. Steganalysis tools exploit these effects to model targeted as well as blind attacks. These challenges are usually dealt with by randomising the changes to the LSB, using different/multiple bit-planes to embed one or more secret bits using elaborate schemes, or embedding in certain regions that are noise-tolerant. Our innovative approach to deal with these challenges is first to develop some image procedures and models that result in increasing similarity between the cover image LSB plane and the secret image bit-stream. This will be achieved in two novel steps involving manipulation of both the secret image and the cover image, prior to embedding, that result a higher 0:1 ratio in both the secret bit-stream and the cover pixels' LSB plane.

For the secret images, we exploit the fact that image pixel values are in general neither uniformly distributed, as is the case of random secrets, nor spatially stationary. We shall develop three secret image pre-processing algorithms to transform the secret image bit-stream for increased 0:1 ratio. Two of these are similar, but one in the spatial domain and the other in the Wavelet domain. In both cases, the most frequent pixels are mapped onto bytes with more 0s. The third method, process blocks by subtracting their means from their pixel values and hence reducing the require number of bits to represent these blocks. In other words, this third algorithm also reduces the length of the secret image bit-stream without loss of information. We shall demonstrate that these algorithms yield a significant increase in the secret image bit-stream 0:1 ratio, the one that based on the Wavelet domain is the best-performing with 80% ratio.

For the cover images, we exploit the fact that pixel value decomposition schemes, based on Fibonacci or other defining sequences that differ from the usual binary scheme, expand the number of bit-planes and thereby may help increase the 0:1 ratio in cover image LSB plane. We investigate some such existing techniques and demonstrate that these schemes indeed lead to increased 0:1 ratio in the corresponding cover image LSB plane. We also develop a new extension of the binary decomposition scheme that is the best-performing one with 77% ratio.

We exploit the above two steps strategy to propose a bit-plane(s) mapping embedding technique, instead of bit-plane(s) replacement to make each cover pixel usable for secret embedding. This is motivated by the observation that non-binary pixel decomposition schemes also result in decreasing the number of possible patterns for the three first bit-planes to 4 or 5 instead of 8. We shall demonstrate that the combination of the mapping-based embedding scheme and the two steps strategy produces stego-images that have minimal distortion, i.e. reducing the number of the cover pixels changes after message embedding and increasing embedding efficiency. We shall also demonstrate that these schemes result in reasonable stego-image quality and are robust against all the targeted steganalysis tools but not against the blind SRM tool.

We shall finally identify possible future work to achieve robustness against SRM at some payload rates and further improve stego-image quality.

Acknowledgments

ALLAH THE MOST GRACIOUS AND MERCIFUL: Who gave me the energy, health, nerves and provided me with all the people to whom I am dedicating this hard work, which took a lot of determination and time until it came to light.

My parents: I would like to dedicate the fruitful outcome of this work to my father and mother who have been waiting for so long to see the result of their son's work and who haven't stopped praying for this work to be done. I hope I made them proud of me as I am very proud of them. My gratitude, love, respect, and thanks go to my dearest mother. Without her blessings, prayers, and unconditional love, I would have achieved nothing in the life.

My brother: My thankfulness and appreciation to my brother **Miran** for his support, encouragement, and patience during the time of my study.

My wife: My warm thanks go to my wife **Shaida** for her support, patience, and understanding throughout the duration of my study time.

My daughters: I would like to dedicate this thesis and all my success in my life to my lovely twin daughters **Aya** and **Anna**, who have been my continuous source of hope and determination to go on despite the difficult times I have been through. Their smiles have often pushed me to put in my best possible efforts to be a better person.

My supervisors: I would like to express my sincerest gratitude towards my supervisor Prof. **Sabah Jassim** for providing me the opportunity to work in the challenging areas of multimedia security. Prof. **Jassim** has been unrelenting source of motivation and support throughout my PhD life. Without his guidance, motivation, and support, this thesis would never come to light. I would also like to thank my supervisor Dr. **Harin Sellahewa** for his valuable comments, suggestions, and discussions.

My sponsor: I would like to express my sincere appreciation and gratitude to the Kurdistan Regional Government (KRG), and Ministry of Higher Education and Scientific Research in Kurdistan for sponsoring my postgraduate programme of study. Also my special thanks go to the University of Sulaimani.

My friends and Colleagues: Last but not least, I wish to thanks all friends, fellow students, and staff in the Department of Applied Computing, in the University of

Buckingham, for their encouragement and support during the time of this programme. I would like to express my special thanks to Mr. **Hongbo Du**, the lecturer in the department, for his encouragement and useful discussion. Also my special thanks go to Dr. **Nasser Al-Jawad,** the lecturer in the department, for his assistance and encouragement. I would like to thank Dr. **Stuart Hall,** the lecturer in the department, for his valuable advices and recommendations that significantly improves clarity as well as the soundness of the thesis. Finally, huge thanks and appreciates are also go to Dr. **Faraidun K. Hamasalh**, lecturer at the University of Sulaimani, for his help and encouragement.

Abbreviations

2D-DWT	Two Dimensional Discrete Wavelet Transform
dB	Decibel
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DIH	Difference Image Histogram
DIP	Digital Image Processing
DWT	Discrete Wavelet Transform
FLDs	Fisher Linear Discriminants
IWSIM	Integer Wavelet Based Secret Image Manipulation
IWT	Integer Wavelet Transform
LSB	Least Significant Bit
LSBM	Least Significant Bit Matching
LSBR	Least Significant Bit Replacement
MSB	Most Significant Bit
MSE	Mean Square Error
OPAP	Optimal Pixel Adjustment Process
PoV	Pair of Values
PRNG	Pseudo Random Number Generator
PSNR	Peak Signal-to-Noise Ratio
PVD	Pixel Value Differencing
RR	Reduction Ratio
RS	Regular and Singular
RWS	Revised Weighted Stego
SIM	Secret Image Manipulation
SISR	Secret Image Size Reduction
SRM	Spatial Rich Model
SVM	Support Vector Machine
VER	Variable Embedding Ratio
WS	Weighted Stego
WT	Wavelet Transform

Contents

Abstract I

Acknowledgments IV

Abbreviations VI

Contents VII

List of Figures X

List of Tables XV

Declaration XVII

Chapter 1 Introduction 1

1.1 From Ancient to Digital Steganography 3

1.2 Overview of the Research Problem 7

1.3 Challenges and Success Criteria for Digital Steganography 9

1.4 Digital Steganography– Some Emerging Applications 10

1.5 Motivation 11

1.6 Objectives 12

1.7 Contributions 12

1.8 Structure of the Thesis 13

Chapter 2 Steganography: Background, Objectives and Approaches 15

2.1 Information Security Mechanisms 15

2.1.1 Information Security – Objectives and Governing Principles 16

2.2 Digital Steganography – Categorisation and Hiding Methods 21

2.3 Steganalysis and Steganography Attacks 28

2.4 Performance Evaluations of Image Steganography Techniques 32

2.4.1 Data Payload or Capacity 32

2.4.2 Stego-image Quality 33

2.4.3 Un-detectability of Hidden Secrets 35

2.4.4 Embedding System Efficiency 35

2.5 Summary 37

Chapter 3 Image-based Steganography and Steganalysis: Literature Review 38
3.1 Image-based Steganography Approaches 38
3.1.1 LSB\higher LSBs (Bit-Planes) based embedding Approaches.....................42
3.1.2 Pixel value decomposition based embedding Approaches 47
3.1.3 Location\Region based embedding Approaches..51
3.1.4 High Embedding Efficiency Approaches 57
3.2 Image-based Steganalysis Approaches 63
3.3 An Overview of our Approach 71
3.4 Summary 73
Chapter 4 Multi Bit-planes Image-based Steganography..74
4.1 Bit-plane Indexing-based Embedding Scheme 74
4.1.1 Embedding and Extracting Procedures...75
4.1.2 Experimental Setup and Results 77
4.2 Fibonacci-Mapping based Embedding Scheme 82
4.2.1 Embedding and Extracting Procedures...82
4.2.2 Experimental Results 84
4.3 Discussion 89
Chapter 5 Secret Image Pre-Processing 90
5.1 Secret Image Manipulation (SIM)...91
5.1.1 SIM Forward Procedure...91
5.1.2 SIM Backward Procedure...94
5.1.3 Performance of SIM...94
5.2 Integer Wavelet based Secret Image Manipulation (IWSIM)..........................96
5.2.1 IWSIM Forward Procedure 98
5.2.2 IWSIM Backward Procedure...100
5.2.3 Performance of IWSIM 101
5.3 Secret Image Size Reduction (SISR) algorithm 103
5.3.1 SISR Encoding Procedure 104

5.3.2 SISR Decoding Procedure 106

5.3.3 Example Application of SISR Algorithm 106

5.3.4 Performance of SISR 109

5.4 Performance of Fibonacci-Mapping based scheme post SIM, IWSIM, and SISR………. 112

5.5 Discussion 118

Chapter 6 Cover Pixel Value Decomposition Schemes 120

6.1 Background 121

6.2 Simple Sequence based cover pixel value decomposition scheme (SS) 123

6.3 Extended-Binary cover pixel value decomposition scheme 126

6.3.1 Performance of Extended-Binary 130

6.4 Experimental Results 134

6.5 Discussion 140

Chapter 7 Mapping based Steganography for Hiding Secret Images in Cover Images ……… 142

7.1 Single bit Mapping Tables for pixel value decomposition schemes 143

7.1.1 The 5-rows Mapping Tables (Fibonacci, prime, natural, and CF) 143

7.1.2 The 4-rows Mapping Tables (Lucas, and Extended-Binary) 145

7.2 Efficient Secure image-based steganography schemes 146

7.3 Experimental Setup and Results 149

7.4 Summary 173

Chapter 8 Conclusions and Future Research Directions 175

8.1 Conclusions 175

8.2 Future Research Directions 180

References 182

List of Publications 191

Appendix 192

List of Figures

Figure 1-1: General structure of the image-based steganography process……………......6

Figure 2-1: Diagram of classification of security systems. 21
Figure 2-2: Bit-planes of Lenna image. 23
Figure 2-3: DFT and DCT Frequency domains: (a) original image, (b) spectrum of the DFT domain, and (c) DCT domain. 26
Figure 2-4: DWT (a) Original image, (b) Level 1, (c) Level 2, (d) Level 3. 27
Figure 2-5: General framework of universal steganalysis. 31
Figure 2-6: Classification of the steganalysis techniques. 31

Figure 3-1: Pseudo-Code of the LSBMR embedding technique. 59
Figure 3-2: Chan's approach (a) The decision tree of the data embedding procedure (b) An example. 61
Figure 3-3: Illustration of the (Iranpour & Farokhian, 2013) for the eight cases. 62
Figure 3-4:Example of PoV plot for cover image Lenna (without embedding). 65
Figure 3-5:Example of PoV plot for stego image Lenna (50% embedding). 65
Figure 3-6:Example of PoV plot for stego image Lenna (100% embedding). 65
Figure 3-7: RS diagram for Lenna image. The x- axis is a ratio of flipped LSBs; the y-axis is the (RM, RM-, SM, SM-). 67

Figure 4-1: Ratio of modified pixels for the LSBR and Indexing-based embedding technique. 78
Figure 4-2: Embedding efficiency for the LSBR and Indexing-based embedding technique. 78
Figure 4-3: The PSNR for the LSBR and Indexing-based embedding technique. 79
Figure 4-4: RS diagram for LSBR technique. 80
Figure 4-5: RS diagram for the Indexing-based embedding technique. 80
Figure 4-6: DIH steganalysis for LSBR and Indexing-based embedding technique. 81
Figure 4-7: RWS steganalysis for LSBR and Indexing-based embedding technique. ...81
Figure 4-8: The ratio of the modified pixels for the LSBR and Fibonacci-Mapping based embedding technique. 85
Figure 4-9: The embedding efficiency for the LSBR and Fibonacci-mapping based embedding technique. 85

Figure 4-10: The PSNR for the LSBR and Fibonacci-Mapping based embedding technique....86
Figure 4-11: RS diagram for Fibonacci-Mapping scheme....87
Figure 4-12: DIH steganalysis for LSBR and Fibonacci-Mapping based embedding technique....87
Figure 4-13: RWS steganalysis for LSBR and Fibonacci-Mapping based embedding technique....88

Figure 5-1: Lenna image and its modified version using SIM algorithm....92
Figure 5-2: Level one IWT sub-bands of Lenna image and histograms....97
Figure 5-3: Ratio of zero-bits of SIM and IWSIM....103
Figure 5-4: Ratio of side information bits of SIM and IWSIM....103
Figure 5-5: Ratio of modified pixels for the Fibonacci-Mapping based techniques....113
Figure 5-6: Embedding Efficiency for the Fibonacci-Mapping based techniques....114
Figure 5-7:The PSNR for the Fibonacci-Mapping based techniques....115
Figure 5-8: RS diagram for Mapping-based-SIM....116
Figure 5-9: RS diagram for Mapping-based-IWSIM....116
Figure 5-10: RS diagram for Mapping-based-SISR....116
Figure 5-11: DIH steganalysis for the Fibonacci-Mapping based techniques....117
Figure 5-12:RWS steganalysis for the Fibonacci-Mapping based techniques....117

Figure 6-1: Ratio of cover pixels' LSB = 0 for the different sequences of numbers....132
Figure 6-2: Ratio of LSB = 0 for different decomposition techniques....133
Figure 6-3: Ratio of capacity for different decomposition techniques....133
Figure 6-4: The ratio of the modified pixels for the Original_EB, SISR_EB, SIM_EB, and IWSIM_EB schemes....136
Figure 6-5: The embedding efficiency for the Original_EB, SISR_EB, SIM_EB, and IWSIM_EB schemes....136
Figure 6-6: The PSNR for the Original_EB, SISR_EB, SIM_EB, and IWSIM_EB schemes....137
Figure 6-7: RS diagram for the Original_EB, SISR_EB, SIM_EB, and IWSIM_EB schemes....138
Figure 6-8: DIH steganalysis for the Original_EB, SISR_EB, SIM_EB, and IWSIM_EB schemes....139

Figure 6-9: RWS steganalysis for the Original_EB, SISR_EB, SIM_EB, and IWSIM_EB schemes. 139

Figure 7-1: Embedding procedure for our image-based steganography schemes. 147
Figure 7-2: Extracting procedure for our image-based steganography schemes. 148
Figure 7-3: Secret images: Lenna and Jet. 150
Figure 7-4: Ratio of modified pixels for the SIPI experimental images. 152
Figure 7-5: Ratio of modified pixels of the cover BOSSBase image when Lenna is the secret image. 152
Figure 7-6: Ratio of modified pixels of the cover BOSSBase image when Jet is the secret image. 152
Figure 7-7: Embedding efficiency for the SIPI database. 153
Figure 7-8: Embedding efficiency for the BOSSBase database when the secret image Lenna is embedded. 153
Figure 7-9: Embedding efficiency for the BOSSBase database when the secret image Jet is embedded. 154
Figure 7-10: PSNR for the tested steganography schemes for the SIPI database. 154
Figure 7-11: PSNR for the BOSSBase stego images when the secret image Lenna is embedded. 155
Figure 7-12: PSNR for the BOSSBase stego images when the secret image Jet is embedded. 155
Figure 7-13: RS diagram for all tested steganography schemes for SIPI database. 157
Figure 7-14: RS diagram for all tested schemes for the BOSSBase database when Lenna image is embedded. 158
Figure 7-15: RS diagram for all tested schemes for the BOSSBase database when Jet image is embedded. 159
Figure 7-16: PoV diagram for sample stego-image from SIPI database. 161
Figure 7-17: PoV diagram for sample stego-image from BOSSBase when the Lenna image was embedded. 163
Figure 7-18: PoV diagram for sample stego-image from BOSSBase when the Jet image was embedded. 164
Figure 7-19: DIH steganalysis for all tested steganography schemes for SIPI database. 165
Figure 7-20: DIH steganalysis for BOSSBase database when Lenna image was embedded. 165

Figure 7-21: DIH steganalysis for BOSSBase database when Jet image was embedded. 166
Figure 7-22: WS steganalysis for stego-images in SIPI database. 167
Figure 7-23: WS steganalysis for BOSSBase stego-images when Lenna image was embedded. 167
Figure 7-24: WS steganalysis for BOSSBase stego-images when Jet image was embedded. 167
Figure 7-25: RWS steganalysis for all tested steganography schemes for SIPI database. 168
Figure 7-26: RWS steganalysis for BOSSBase database when Lenna image was embedded. 169
Figure 7-27: RWS steganalysis for BOSSBase database when Jet image was embedded. 169
Figure 7-28: LSBMS steganalysis for SIPI database. 170
Figure 7-29: LSBMS steganalysis for BOSSBase database when Lenna image was embedded. 170
Figure 7-30: LSBMS steganalysis for BOSSBase database when Jet image was embedded. 171
Figure 7-31: SRM steganalysis of SIPI database. 172
Figure 7-32: SRM steganalysis of BOSSBase database when Lenna image was embedded. 172
Figure 7-33: SRM steganalysis of BOSSBase database when Jet image was embedded. 173

Figure A-1: PoV diagram for stego-image number 330 from SIPI database. 196
Figure A-2: PoV diagram for stego-image number 965 from SIPI database. 197
Figure A-3:PoV diagram for stego-image number 1023 from SIPI database. 199
Figure A-4: PoV diagram for stego-image number 1417 from SIPI database. 200
Figure A-5: PoV diagram for stego-image number 1832 from SIPI database. 202
Figure A-6: PoV diagram for stego-image number 122 from BOSSBase when the Lenna image was embedded. 203
Figure A-7: PoV diagram for stego-image number 489 from BOSSBase when the Lenna image was embedded. 205
Figure A-8: PoV diagram for stego-image number 664 from BOSSBase when the Lenna image was embedded. 206

Figure A-9: PoV diagram for stego-image number 855 from BOSSBase when the Lenna image was embedded. 208
Figure A-10: PoV diagram for stego-image number 970 from BOSSBase when the Lenna image was embedded. 209
Figure A-11: PoV diagram for stego-image number 122 from BOSSBase when the Jet image was embedded. 211
Figure A-12: PoV diagram for stego-image number 489 from BOSSBase when the Jet image was embedded. 212
Figure A-13: PoV diagram for stego-image number 664 from BOSSBase when the Jet image was embedded. 214
Figure A-14: PoV diagram for stego-image number 855 from BOSSBase when the Jet image was embedded. 215
Figure A-15: PoV diagram for stego-image number 970 from BOSSBase when the Jet image was embedded. 217

List of Tables

Table 3-1: RS steganalysis for Lenna image. 66

Table 4-1: Fibonacci-Mapping Table............83

Table 5-1: Grayscale values (0-255) in descending order of number of 1s in its binary representation............93
Table 5-2: SIPI database - Ratio of 0:1 in the secret images and SIM modified secret images I'. 95
Table 5-3: BOSSBase database - Ratio of 0:1 in the secret images and SIM modified secret images I'.............95
Table 5-4: Ratio of bits of the SIM side information.............96
Table 5-5: SIPI database - Ratio of 0:1 in the secret images and IWSIM modified secret images I'. 101
Table 5-6: BOSSBase database - Ratio of 0:1 in the secret images and IWSIM modified secret images I'.............101
Table 5-7: Ratio of bits of the side information using IWSIM.............102
Table 5-8:Ratio of increased bits to represent the modified sub-bands.............102
Table 5-9: Number of obtained bits from proposed SISR algorithm for block size 4x4. 106
Table 5-10: Block of 16 pixels............106
Table 5-11: Differences between pixels value and minimum pixel value.............107
Table 5-12: Producing original pixels value from the recovered Dij. 108
Table 5-13: Average of 0:1 ratio before and after applying the SISR for 4x4 block size. 109
Table 5-14: Ratio 0:1 SISR algorithm for different block sizes. 110
Table 5-15: Average RRs for SISR algorithm for different image and block sizes. 111
Table 5-16: Average RRs for SISR, RLE, Huffman, and LZW for different image sizes. 111
Table 5-17: Average time cost for SISR, RLE, Huffman, and LZW for different image sizes. 112

Table 6-1: Number of bit-planes and their corresponding weights for different pixel value decomposition techniques............125

Table 6-2: Pixel values and their decomposition using Extended-Binary scheme. 129
Table 6-3: Ratio of the cover pixels' LSB zero value of the Extended-Binary decomposition technique for SIPI database. 130
Table 6-4: Ratio of the cover pixels' LSB zero value of the Extended-Binary decomposition technique for BOSSBase database. 130
Table 6-5: Ratio of 0:1 LSB for different decomposition techniques..........................134

Table 7-1: Mapping for Fibonacci, prime, natural, and CF. 144
Table 7-2: Mapping for Lucas and Extended-Binary. 145
Table 7-3: Capacity of the tested steganography techniques. 150
Table 7-4: Ratio of 0:1 in the binary representation of the tested secret images. 151

Table A-1: Grayscale values (0-511) in descending order of number of 1s in its binary representation...192

Declaration

I hereby declare that all the work in my thesis entitled (*EXPLOITING SIMILARITIES BETWEEN SECRET AND COVER IMAGES FOR IMPROVED EMBEDDING EFFICIENCY AND SECURITY IN DIGITAL STEGANOGRAPHY*) is my own work except where due reference is made within the text of the thesis.

I also declare that, to the best of my knowledge, none of the material has ever previously been submitted for a degree in the University of Buckingham or any other University.

Alan Anwer Abdulla

Chapter 1

Introduction

Digital steganography is an information security mechanism that is general concerned with concealing the presence of a secret data/object during mundane communication sessions by embedding the secret data in another innocuous data/object in such a way that only the sender and intended recipient are aware of the secret's existence. It is an alternative to cryptography in protecting sensitive secrets where the adversary is aware of the presences of the secret but cannot extract it. Thus, digital steganography is the art and science of making the act of communication itself a secret.

In recent years, interest in steganography has shifted from traditional and ancient practices into hiding secret data and media objects, especially secret image files, in image files. This area of steganography, for example, is becoming a common technique in protecting sensitive communications by intelligence and law enforcing agencies to crime prevention by exchanging facial images of suspects to be compared with databases of known criminal faces. Moreover, forensic investigators often need to take and transmit photos of the scene of the crime, or left fingerprints, for later comparison without undermining the integrity of the evidence. Armed forces have a variety of similar needs such as exchanging military maps or surveillance video in hostile environment/situations. Modern health care systems required by law to maintain the privacy of critical information when storing or exchanging patient's medical images such as X-ray. Furthermore, financial as well as commercial organizations such as banks can benefit from such technology to prevent customers' account information/identification from being accessed illegally by unauthorised users. Therefore, those mentioned communication systems become more and more dependent on digital steganography.

This thesis is concerned with the design, the development and the testing the performance of secure embedding and transmission of secret messages in image objects. Throughout the recent history of digital communication, many steganography techniques have been developed for embedding secrets into digital images primarily by manipulating their least significant bit-planes (LSBs). Although, the effect of these changes may not be visible to human eye, but the presences of the secret can become more detectable, by a determined and digitally skilled adversary, the longer the secret message is. Steganographers must address the problem of embedding capacity of the cover image while protecting against detectability. Embedding longer secrets, though desirable, definitely result in some form of cover image distortion or even degraded image quality. Hence, the robustness of message embedding against adversary attacks is closely linked to maintaining image quality. Embedding efficiency is the most important requirement for digital steganography that employs all the above addressed problems (i.e. payload capacity, message detectability or security, and stego-image quality). Embedding efficiency means minimising the changes made to the cover image pixels, as a result of embedding a secret message, while maintaining capacity.

We shall investigate and test techniques to improve security and efficiency of message embedding techniques in digital images. Most existing steganography techniques focus on the embedding strategy and give no consideration to pre-processing the secret image except encrypting or compressing the secret. Here encryption is aimed at protected the secret even if it was detected while compression is used to improve the quality of the resulted stego-image. One of the premises of this thesis is that applying carefully selected pre-processing techniques could help enhance the embedding efficiency and security of the steganography systems. The objective of our approach, in relation to pre-processing, is to increase the probability of similarity between the secret bits value and the cover pixels' least significant bit (LSB) value. Consequently, designing a new pixel value decomposition technique to decompose cover pixels value with aim of making the cover pixels' LSB value similar, as much as possible, to the secret bits value could support our objective.

This chapter provides a general introduction to the research area and the investigations carried out in this thesis by first starting with some background knowledge and examples of ancient and digital steganography, then an overview of the research problem is explained. Moreover, the challenges and success criteria for digital steganography are discussed followed by listing some recent and potential applications of digital steganography. Furthermore, the main motivation of this study is discussed, and the research objectives are identified based on the established definition of the research project followed by an overview of our contributions. We close the chapter by highlighting the structure of the thesis.

1.1 From Ancient to Digital Steganography

Linguistically, steganography means secret writing since the word *steganography* originally derives from two Greek words, *steganos* means covert or secret, and *graphy* means writing (Cole & Krutz, 2003). Practically, it means the art and science of hiding secret data in an innocent looking dummy container in such a way that the existence of the embedded data is imperceptible and un-detectable (Kahn, 1996). Thus, steganography is the process of hiding secret data within the publicly accessible information.

In physical (i.e. non-digital) steganography, the cover object may be basically anything, for example a physical text document, a painting, or a piece of wood, as long as it can be used to convey a hidden message to the intended recipient without raising suspicion of untrusted parties. Interestingly, the first documented cover object used for the purposes of steganography was the human body. Greek historian Herodotus detailed that steganography's ancient origin can be traced back to 440 BC (Macaulay & others, 1904). It was started by the Greek fellow named Histiaeus, the ruler of the ancient Greek city of Miletus, who shaved the hair of his most trusted slave and wrote/tattooed the message on his head. Once the hair had grown, the message was hidden and he was sent to their allies to communicate with them without the enemies' knowledge. The purpose was to instigate a revolt against Persians (Macaulay & others, 1904). Another example of physical steganography was again ancient Greeks technique by writing secret messages on wax-covered tables. To pass a hidden message, a person would scrape the wax off a table, write a message on underlying wood and again cover the table with wax to make it appear blank and unused. The recipient would simply remove the wax from the table to see the message (Johnson & Jajodia, 1998). Also, invisible ink was used for writing secret messages by the American revolutionaries during the USA revolution on pieces of paper so that the paper appeared to the average person as just being blank pieces of paper. Liquids such as urine, milk, vinegar and fruit juices were used as ink. When these substances are heated, they darken and become visible to the human eye (Mangarae, 2006). Also, invisible ink was used in both World Wars by the German army. In World War II, Germany also used microdots to hide large amounts of data on printed documents, masqueraded as dots of punctuation (Cole & Krutz, 2003).

The advent of electronic and computer technology as well as advances in communication technology triggered an interest in developing steganography techniques to fit the new medium of communication. Although, the focus of the early days of the new technology era was on cryptography as the main security mechanism for the protection of sensitive information. This may have been a result of the fact that access to computer technology in the early days of main frame computers and minicomputers was limited to governmental and corporate organisations besides the scientific community. The advent of space exploration in the early sixties led to the emergence of Digital Image Processing (DIP) and nuclear medicine. The convergence of communication and computer technology has triggered the digital revolution that has escalated over the last two decades and pushed mobile technology into the front to finally widen access to this technology beyond any expectation. This has led to an emergence of huge interest in digital image processing for a variety of applications with new security concerns that is very difficult to address by cryptography alone. The rise of terrorism has finally rekindled the interest in digital steganography. It is often claimed that the 11th of September bombers were using steganography for hiding their secret plans in innocuous communications of digital media objects. Whether it is true or not, this story and similar more recent cases seem to be generating more incentives to research various aspects of steganography and steganalysis.

Digital steganography exploits properties of digital media files such as images, audios, and videos to hide a variety of secrets that could remain undetected. Although some digital/computer based steganography references can be found before 1995, most of the interest and action in the field has occurred since 2000 (Cole & Krutz, 2003). In image-based steganography, a secret message is often hidden within an image in such a way that others cannot discern the presence or contents of the secret. It is important that the stego-image does not contain any easily detectable artefacts due to message embedding that could be detected by electronic surveillance. For example, a message might be embedded in an image by changing the pixels' LSB to be the message bits.

The Prisoner’s problem

Steganography systems have one general principle, described by (Katzenbeisser & Petitolas, 2004) based on a simple scenario formulated by Simmons known as the prisoners’ problem (Simmons, 1984) as follows: Two criminals, *Alice* and *Bob*, have been arrested and locked in separate cells. The warden *Wendy* allows them to exchange messages but the communication has to be completely open to her. As *Alice* and *Bob* need to coordinate their escape plan, they need to find a different way to communicate secretly without being caught. Since any suspicion of a secret information exchange would result in an immediate communication cut off, the prisoners cannot protect their message exchange by encryption. *Alice* and *Bob* resort to using steganography to avoid detection. The job of the warden *Wendy* is steganalysis, she needs to find out whether or not *Alice* and *Bob*’s communications include secretly embedded messages. *Alice*, who wants to send a secret message to the recipient *Bob*, randomly chooses a harmless cover file and embeds the secret message in the cover file and probably uses a stego key. *Alice’s* constructed stego file must be as much as possible undistinguishable from the cover file neither by a human eye nor by a computer system. *Alice* transmits the stego file to *Bob* over the open communication channel allowed by the prison authority. The purpose of the system is to prevent *Wendy*, from observing or noticing the presence of a hidden message. On the other side, *Bob* extracts the embedded message since he knows the embedding method and the stego key used in the embedding process. Only the transmitter and the intended recipient should have the stego key. Most steganography systems prompt users to provide a stego key when they try to embed information in a cover file.

Sometimes, attackers like *Wendy* can detect a hidden message in a stego file and determine how the message was embedded, but they are unable to extract the hidden message. This system is considered to be a secure steganography system because the secret message is unreadable unless one has the related stego key. Therefore, stego keys must be chosen as strong as possible in order to prevent attackers from breaking the steganography systems using all possible stego keys (Cox, et al., 2005).

The security of steganography systems must be based on the assumption that attackers have full knowledge of the steganography system design, the embedding and extracting algorithm. However, attackers only miss the stego key to suspect that a secret communication is taking place. Therefore, most of steganography systems available nowadays meet this principle (Rabah, 2004). Therefore, it is assumed that *Wendy* has a complete knowledge of the steganography algorithm that the prisoners may use. This is a very important assumption and it is one of the most important security principles that have been accepted and practiced since the middle of the 19th century and known as Kerckhoff's principles (Rabah, 2004). Figure 1-1 illustrates the general structure of image steganography systems.

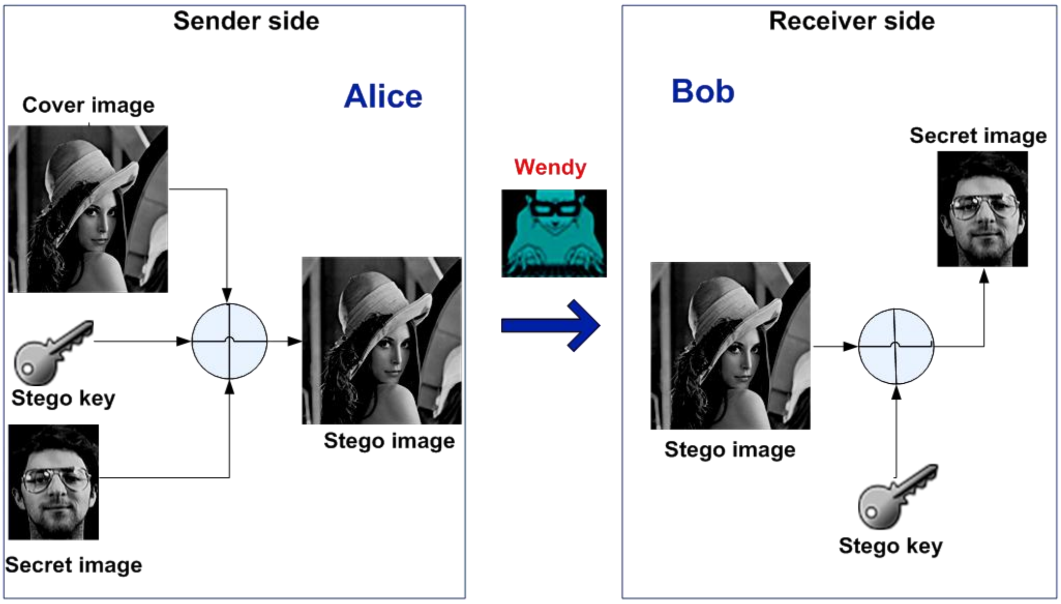


Figure 1-1: General structure of the image-based steganography process.

The main terminologies used in the steganography are:

1. The cover object is the carrier of the secret message.

2. The secret message is the information that is to be hidden in a suitable cover object producing stego object. Note that in this thesis the terms message, hidden message, and secret message are used interchangeably.

3. The process of hiding information is called embedding algorithm which is the way or the idea that usually used to embed the secret message in the cover object (Swanson, et al., 1996) (Petitcolas, et al., 1999) and it is also called steganography system, steganography technique, or steganography scheme.

4. The stego key is a random key agreed between the participants, and it is usually used to control the hiding process/algorithm so as to restrict detection or recovery of the embedded data to authorised parties only.

In Section 1.3, steganography systems requirements are presented.

1.2 Overview of the Research Problem

With advancements in digital communication technology, the exponential growth in the use of the Internet, and a huge increase in computer power; the difficulties in ensuring the privacy of information become increasingly challenging. Therefore, information and multimedia security research area attract more and more interest, and its scope of applications expand significantly. Communication security mechanisms have been investigated and developed to protect information privacy with Encryption and Steganography providing the most obvious solutions. Encrypting a secret message transforms it to a meaningless data which looks more like random noise and is generally observed during transmission, while steganography is not observable. Steganography aims to make the secret communication itself undistinguishable from mundane communication, i.e. hiding the presence of the secret message. It modifies the carrier/cover in an imperceptible way only so that it reveals nothing neither the embedding of a message nor the embedded message itself.

Although steganography is an old field, the recent developments in digital communication technology and the emergence of social media networks have brought new attention to steganography. With the arrival of the digital era, advances in mobile devices and technologies, and the widespread availability of efficient multimedia manipulation tools, exchanging secrets and sensitive information between different groups of users has become a much easier task. Digital steganography is a much easy to use alternative to encryption and creates new opportunities for crime and abuse through hiding secrets and illegal material in digital carriers/covers such as audio, image and videos files. The phenomenal volume of exchanged messages and media files greatly reduce the chance of being caught by legitimate authorities. The growth in the use of multimedia files for steganography is due a high degree of redundancy in the media data, which makes them suitable to embed information without degrading their visual quality. Steganography is made much easier if the cover file includes a lot of redundancies. Images and videos are examples of files that contain a high degree of spatial and temporal redundancies that are often exploited for compression. For example, it is well known that changing the least significant bits of some/all image pixels do not have a noticeable impact on image quality/content. Consequently, the LSB plane may be seen as redundant and those bits can be altered to hide any binary secret. Note that, such type of data hiding certainly result in some distortion that could be detectable by image statistical analysis tools, but criminals bet on the difficulty experienced by authorities in processing huge volumes of daily file exchanges.

Most hiding schemes are designed, and their performances are tested with a random bit-stream secret. However, this thesis focuses on the scenario whereby both the embedded message and the cover are grayscale images. The reason for choosing images as a cover is that images usually have a high degree of redundancy and also images are widely exchanged over the Internet than other digital media and they attract little suspicion. Moreover, the reason for choosing images as secret messages is due to their frequently used in many applications as mentioned earlier.

There are different ways of categorising the different techniques of steganography due to the variety of media file types that can be used as cover files as well as the fact that there is more than one way of representing cover files. For example, audio and image files can be represented in the spatial domain as well as frequency domain and each of these can provide different ways of embedding secrets with advantages as well as limitations. In Chapter 2, we shall give an account of categorisation of embedding techniques.

While steganographers aim to design difficult to detect, and efficient steganography techniques, steganalysers attempt to defeat the goal of steganography by detecting the presence of a hidden message, even if they cannot retrieve it. Steganalysis schemes attempt to exploit the fact that any embedding scheme will result in some kind of local random distortions, albeit difficult to detect by the naked eye, or may violate in a small way, but computable, some statistical/correlation models that are known/expected to hold among the different spatial/gray-level components of cover images. There are numbers of existing image steganalysis tools that are widely used to detect the presence/absence of a hidden message and estimate the size of the embedded secret message. These tools are classified in different ways, whereby some are targeting specific embedding schemes while others are designed to detect the presences of hidden messages without knowledge of the embedding algorithm. We will discuss the classification of these tools in Chapter 2. In this thesis, we aim to investigate steganography schemes that have the ability of withstanding against most reliable and well-known steganalysis techniques, and therefore we shall give details about the theory and working of certain tools in Chapter 3.

1.3 Challenges and Success Criteria for Digital Steganography

The most important and obvious success criteria for steganographers is the ability to avoid attracting suspicion of the presence of a hidden message in otherwise innocuous looking communication. General factors that influence this overarching objective and must be addressed by image-based steganography systems are: 1) the quality of the stego-image (i.e. minimising the perceptual difference between the stego and the cover image); 2) the payload capacity of the cover image (i.e. the amount of secret data that can be embedded in the cover image); 3) detectability of the message (i.e. prevent detection/recovery by a third party); and 4) the robustness of the stego-image (i.e. protection against distortion attacks). However, the first two requirements are at odds with each other, and it is quite difficult to increase the payload capacity and simultaneously maintain the imperceptibility of a stego-image. Consequently, a compromise may have to be found that is application dependent. The third requirement is relevant to the first one; in other words, by improving the stego-image quality the steganography system becomes less detectable. Currently, most existing steganography systems deal with first three requirements without taking the robustness against distortion attacks into account. This is most likely due to the fact that robustness is application dependent (Cox, et al., 2007), and most steganography systems consider the passive warden scenario in which the warden does not interfere with the stego file in any way (Cox, et al., 2005). It is a challenge for steganographers to achieve a good balance among all these different steganography requirements.

The above first three requirements, are affected directly by the number of changing pixels of the cover image after embedding the secrets, and, therefore, in the literature minimising this change has been stated as the most important requirement. The amount of change must be considered relative to the payload capacity, and, hence, it is natural to model this requirement by the ratio of changed pixels to the size of the secret message. In recent proposed steganography techniques, less ratio of changing of cover image pixels' value after message embedding, while maintaining payload capacity, has been used as an indicator of higher stego-image quality and lower message detectability. The evaluation of the ratio of changing cover image pixels' value is called *embedding efficiency* in the literature, which can be defined as the number of secret bits embedded per one embedding change. When the embedding efficiency increases, the less detectable traces will be introduced in the stego-image, and the more robust against steganalysis techniques. Embedding efficiency is the main objective in this thesis.

All these requirements can be associated with quantitative measures that can be modelled and determined in terms of the stego-images that can provide objective tools to test the performance of any embedding scheme. Performance evaluations for image steganography techniques are presented in Section 2.4.

1.4 Digital Steganography– Some Emerging Applications

Digital steganography has various useful applications. However, like any other science it can be used for good as well as ill intentions. Government organisations and business communities rely heavily on exchanging, sharing and processing information to assist them in making a variety of strategic decisions and steganography is one of the security infrastructures that are established to help protect and preserve the integrity of information flowing across different channels. Digital steganography is useful in protecting sensitive communications for many applications such as intelligence and law enforcing agencies to prevent crime (Petitcolas, et al., 1999) (Mercuri, 2004); military purposes such as exchanging military maps (Jenifer, et al., 2014) (Wayner, 2002) (Wu & Tsai, 2003); in health care systems to maintain the privacy of critical information such as medical records (Liu, et al., 2013) (Cheddad, et al., 2008) (Li, et al., 2007) (Raul, et al., 2007); and in financial and business organizations (Juarez-Sandoval, et al., 2013) such as banks to prevent customers' account information from being accessed illegally by unauthorised users, or identity cards; where individuals' details are embedded in their photographs (Jain & Uludag, 2002).

On the other hand, digital steganography is also used by malicious users, organised crime, and international terrorism to hide their ill intentions. Here two examples of steganography threat are highlighted out of many examples. The most dangerous usage of steganography was when it was used by a terrorist group on the 11th of September 2001. In his article (Lau, 2003), Stephen states that *"News stories began appearing in mainstream United States media in the days following September 11th reporting that Osama bin Laden and al-Qaeda were using the internet to covertly communicate between various terrorist cells to plan and relay information. One interesting aspect of the media reports was that the al-Qaeda was supposedly using a technique known as steganography to covertly communicate."*

Terrorists are not the only criminals who may employ steganography techniques for illegal purposes. Steganography was reported to be used by South American drug dealers to communicate photographs of transit routes and cocaine shipment information

(Kodovsk, 2012). The mentioned examples of illegal usage of steganography have all relied on the use of digital images as cover files because images have a high degree of redundancy and are suitable to embed information without degrading their visual quality. Moreover, images are widely exchanged over the Internet than other digital media and they attract little suspicion.

1.5 Motivation

There has been an explosive growth in multimedia technology and applications in the past several years. Efficient representation for storage, transmission, retrieval and security of information are some of the biggest challenges faced. With growing need of information security, digital image steganography has established itself as an important discipline in signal processing and multimedia security. That is due in part to the strong interest from the research community. The motivation behind developing image steganography methods its growing use by various organizations to communicate securely, which include the military or intelligence operatives (in the field of espionage and crime prevention) as well as a variety of companies and organisations that provide public services to protect customers information. The main goal of using the image steganography is to avoid drawing attention to the transmission of hidden information.

I am particularly motivated to help in reducing the huge digital gap that exists between the developed world and my own nation Kurdistan (and may other third world countries that are in the process of building its institutions and suffer from terrorisms). For that, I try to study and design efficient and secure image-based steganography techniques in order to be used in my country which by the many organizations that need to maintain security and privacy of its information such as hospitals needing to establish and benefit a medical information records system, intelligence and law enforcing agencies for knowledge-based crime and terrorism prevention, and financial organizations and banks to protect customers account information against illegal access by unauthorised users/actions. Despite the fact that there are many existing practical image-based steganography systems, but still the trade-off between steganography requirements is a problem in this research area. This is a motivation to design image-based steganography techniques that minimises the embedding impact on the stego-image while maintaining payload capacity. Security considerations form another incentive to assess robustness against a plethora of steganalysis tools.

1.6 Objectives

Section 1.3 has highlighted the main challenges of the steganography systems as increasing the payload capacity while maintaining the stego-image quality and message detectability; or improving message detectability while maintaining payload capacity. The main objectives of the investigations conducted in this thesis are focused on the design image-based steganography schemes that have the property of improving embedding efficiency as well as message un-detectability while maintaining payload capacity. In this thesis, we confine our investigations into the steganography schemes that work in the spatial domain of the cover images and manipulate/modify image bit-planes. To meet the above objectives, our approach strategy can be summarised in this research question: *Can the probability of similarity between the secret image bit-streams and the cover images LSB plane be increased without compromising the payload capacity?* In achieving this, i.e. optimising similarity between the secret image bit-streams and cover image LSB plane, the following sub-objectives are identified:

- To manipulate the secret image prior to embedding it in a way that its binary representation contains a higher ratio of 0 bits to the 1 bits (0:1) and possibly reduce the size of the secret image before embedding.
- To study and investigate existing pixel value decomposition techniques used for image steganography, and design a new pixel value decomposition scheme that achieves the best ratio of 0:1 in the LSB plane of the cover image.
- To exploit the possible increase in similarity between the pre-processed secret image bit-stream and the LSB plane of a decomposed cover image, and develop new steganography schemes that achieve high embedding efficiency, acceptable stego-image quality, high payload capacity and robustness against the most common steganalysis attacks.

1.7 Contributions

The main contributions of this thesis are related to the above stated objectives and can be enumerated as follows:

1. We developed three pre-processing algorithms that encode the secret images, prior to embedding, into bit-streams with significantly increased 0:1 ratio. The first two algorithms are based on the similar strategy adopted in statistical coding by exploiting the structure of histograms of the secret image spatial

domain and Integer Wavelet sub-bands, respectively. One of these algorithms provides highest 0:1 ratio (80% on average). The third algorithm is directly applied on the spatial domain of the secret image, and it reduces the secret image bit-stream size by 30% of the original secret image size. This algorithm not only reduce the size of the secret image, but produces a similar property to that of the first two algorithms in that the reduced size also have approximately 57% ratio of 0:1 (on average).

2. A new pixel value decomposition scheme is proposed that has a property that on average, approximately 77% of cover pixels have 0 LSB value. This would increase the probability of similarity between the cover images LSB plane and the secret image bit-streams obtained in 1. This results in reducing the ratio of the pixels change of the cover image after embedding the secrets.
3. A bit-plane mapping technique is proposed for Fibonacci based message embedding. This mapping based embedding is solved the problem of skipping some cover pixels, due to Zeckendorf's theorem, to use for embedding the secrets. In other words, by using mapping based embedding instead of bit replacing based embedding, every cover pixel can be used for embedding the secret bits. This proposed mapping technique is not only applicable on Fibonacci based embedding technique, but also applicable on some other pixel value decomposition schemes including our proposed in 2.
4. As a combination of steps 1, 2, and 3, efficient and secure image-based steganography approaches are designed that increase embedding efficiency and improve message un-detectability due to minimise the ratio of cover pixels change after message embedding, while maintaining the payload capacity. Minimising the ratio of changed of the cover image pixels reflects less detectability and withstands to existing well-known steganalysis tools.

1.8 Structure of the Thesis

The rest of the thesis is organised into seven chapters:

Chapter two introduces background information about steganography that is relevant to the objectives and challenges of digital steganography. It also describes common secure communication mechanisms, namely steganography and cryptography, highlighting their objectives and relevance to information security. In addition, it explains various categories of steganography techniques, the main performance evaluations criteria, and different kinds of attacks that undermine their effectiveness.

Chapter three gives a literature review of the most relevant image-based steganography approaches as well as different steganalysis approaches. Also, it gives a brief overview of our proposed approaches.

Chapter four presents two initial simple image-based steganography schemes that manipulate more than one bit-plane including the LSB to embed one or two secret bits. The first scheme embeds one secret bit in a way that depends on the first two LSBs of the cover image pixels, and the second scheme attempts to double capacity by embedding two secret bits in each cover pixel value; and improve un-detectability. It also discusses the related experimental results.

Chapter five presents the first step of our embedding strategy by showing three proposed algorithms as a pre-processing on the secret image prior to embedding. It also shows the related experimental results.

Chapter six presents the second step of our embedding strategy by studying and investigating different pixel value decomposition techniques, and presenting the proposed new pixel value decomposition technique. Furthermore, it discusses the related experimental results.

Chapter seven presents the last step of our embedding strategy by showing the proposed mapping based embedding schemes. Experimental results are provided to show the efficiency and security of these proposed embedding schemes.

Chapter eight presents the conclusions and potential directions for future research.

Chapter 2

Steganography: Background, Objectives and Approaches

This chapter aims to present a reasonable account of background information about steganography that is relevant to the objectives and challenges of digital steganography. We first describe common secure communication mechanisms, namely steganography and cryptography, highlighting their objectives and relevance to information security. Understanding similarities and differences between steganography and cryptography helps in developing appropriate security tools that meet the multi-faceted and dynamically changing requirements of highly connected society for robust and efficient tools. We shall describe various categories of steganography techniques, the main performance evaluations criteria, and different kinds of attacks that undermine their effectiveness. The main focus in these discussions will be on image-based steganography systems, i.e. hiding sensitive data in innocuous image file covers.

2.1 Information Security Mechanisms

Over the last few decades, the phenomenal advances in, and convergence of, computing and communication technologies has led to an exponential growth in their deployment in all aspects of societal, health, crime fighting and economic activities. The emergence of smart and mobile technologies, the social networking, the rise in terrorism, and cloud computing have led to an explosion in the amount, type, and sensitivity of exchanged information all the time and have raised serious concerns about the security of information and infrastructure. Safeguarding the secrecy of sensitive and valuable information assets is not new and predates the digital age. Cryptography is the more commonly used mechanism of information security and attracted the focused research efforts and matured throughout the centuries. The advents of digital technology in the last few decades re-energised research interest in the other long practiced security mechanisms of information hiding and steganography, which has led to the development of a plethora of dual use tools that could be used to undermine or to protect sensitive digital information. In this section, we shall describe the main objectives of the two security mechanisms of cryptography and steganography, the principles that govern their development, evaluation and evolutions, and their main developed techniques.

2.1.1 Information Security – Objectives and Governing Principles

Any computer and communication system have several security related requirements that should be addressed if the system is to be accepted and work reliably. Overall, five key objectives/services have been identified for securing a variety of information systems: confidentiality, integrity, availability, authentication, and non-repudiation. Cole identifies these concepts as follows (Cole & Krutz, 2003):

1. *"Confidentiality* deals with protecting, detecting, and deterring the unauthorized disclosure of information".
2. *"Integrity* deals with preventing, detecting, and deterring the unauthorised modification of information".
3. *"Availability* relates to preventing, detecting, or deterring the denial of access to critical information".
4. *"Authentication* in most transaction you need to be able to authenticate or validate that the people you are dealing with are who they say they are".
5. *"Non-repudiation* deals with the ability to prove in a court of law that someone sent something or signed something digitally".

The above individual objectives do not have the same priority or importance in all information systems. Cryptography and steganography have emerged as the two well-suited mechanisms to protect sensitive information, but both mechanisms are fraught with serious challenges. It is widely accepted that no one security method can address all the above objectives, but together steganography and cryptography can provide the tools that cover most of these services (Cole & Krutz, 2003). However, securing computing and communication systems is not a collection of procedures and tools that could be put together once and be assured that nothing could go wrong. The landscape of security is dynamically changing due to many factors such as rapid changes in technology and emergence of new services and scenarios.

Confidentiality is the most fundamental security service offered by cryptography since the secret message is scrambled in such a way that only the intended recipient can unscramble it. By using hash functions combined with cryptographic keys, integrity and authentication services are provided. Cryptography hash function thus used to ensure the integrity of data. Digital signature also offers data authentication as well as support non-repudiation. Digital signature schemes encrypt the message with a private key. The encrypted message acts as a signature since only a specific private key could have produced the specific result (Cole & Krutz, 2003). To summarise, of the five security services, cryptography offers confidentiality, integrity, authentication, and non-repudiation.

On the other hand, since steganography ensures the privacy of sensitive information by concealing it in other information objects, then confidentiality is also offered by steganography. Since the embedded information could have been altered intentionally or not, and the alteration will not be noticed by the receiver, therefore the integrity of the steganography cannot be checked. Authentication and non-repudiation are not offered automatically by steganography, since steganography does not have the functionality of knowing the origin of embedded information and someone can later deny having embedded the information. However, authentication can be offered if the steganographic key is used, since knowledge of the key can identify a person to be the one who sends the secret message. To summarise, steganography only offers confidentiality and authentication out of five security services. Thus, cryptography and steganography have two security services in common, namely confidentiality and authentication/identification. However, cryptography can offer two additional security services that are not offered by steganography at the moment, namely data integrity and non-repudiation. Although both cryptography and steganography are offered confidentiality service, steganography provides more confidentiality and information security than cryptography since it conceals the existence of secret message rather than only protecting the message contents.

Although both cryptography and steganography provide the two well researched secret communication techniques, they have different ways of achieving their intended objectives. Cryptography conceals only the meaning or contents of a secret message from an attacker by scrambling it, whereas steganography even conceals the existence of the secret message. In other words, use of cryptography would not stop a third party knowing that some secret communication is going on, while in steganography, the message to be sent is concealed in such a way that an intruder would not normally know whether any secret communication is going on or not. Cryptography and steganography they have a different definition in terms of system breaking (Zollner, et al., 1998). A cryptography system is considered broken if an attacker can read the secret message. On the other hand, steganography system is considered broken if an attacker can detect the existence or read the contents of the embedded secret message. Intuitively, the security of the steganography system depends on the inability of an attacker to distinguish a cover object from stego object (Katzenbeisser & Petitcolas, 2002). Moreover, steganography system will be considered to have failed or be insecure if an attacker detects the presence of secret message even without decoding it (Zollner, et al., 1998) (Katzenbeisser & Petitcolas, 2002). As a result, this consideration makes steganography systems more fragile than cryptography systems in terms of system security failure. Therefore, steganography systems must avoid detection in order to achieve security and not considered failed systems.

Since steganography adds an extra layer of protection to cryptography, it is recommended that they be used together for achieving a higher level of security. For example, one straightforward approach in securing a sensitive message may be based on first encrypting it and then hide it in a cover object.

Due to the fact that both cryptography and steganography tools are no longer a private enterprise, but are regularly used by the public to protect their information assets and their privacy, these tools must adhere to certain principles in order to be accepted and used. As early as 1883 the Dutch cryptographer Auguste Kerkhoff has laid down six principles that are now referred to as the Kerkhoff's Principles, for the design of secure ciphers. The most important interpretation of these principles stipulates that security of ciphers is not served in any way by using a secret cipher algorithm but rather on the secrecy of the key used for encryption. This is why research efforts in cryptography are dominated by the security of key management systems and protocols. As mentioned in the last chapter, this principle extends naturally to steganography in that one should always credit the warden with knowledge of the embedding algorithm while ensuring the secrecy of the steganographic key parameter of the algorithm.

Finally, it is essential to recognise that steganography is only one class of information hiding (Petitcolas, et al., 1999), and information hiding has a wider remit and objectives beyond the security of communication. Over the last few decades, the concept of hiding information has provided solutions to other non-security-oriented applications such as copyright protection, detecting breaches of licences/agreements, protection against fraud, abuse of power and falsification of evidences. The three classes of information hiding share some common characteristics with steganography while having important differences in their requirements. We now briefly describe the other two classes.

Digital Watermarking

Watermarking is an old technique of embedding a mark into documents such as paper currency and traveller cheques as a protection against forgery. Digital watermarking is similarly concerned with embedding marks into digital documents to protect against the removal of copyright. It is aimed to protect the right of the owners of digital media such as images, music, and videos. Even if people copy or make a minor change to the watermarked file, the owner should still be able to prove it is his or her file. There are two kinds of watermarks, visible and invisible. In the visible case, the watermark, typically a text or logo, is visibly embedded in the image or video. Invisible watermarking is similar to steganography in that the mark is made imperceptible to maintain document quality. Often these invisible marks are textual messages embedded in audio or image files for authentication of the digital file to protect against fraud and illegal distribution.

The similarity between watermarking and steganography in terms of the operational objective of embedding a message may give the impression that these are two different names for the same concept. On the contrary, there are many settled differences. For example, unlike the case of steganography the embedded message/mark in watermarking is not a secret. Moreover, the two hiding concepts differ significantly in terms of system breakability. A watermarking system, whether visible or not, is considered broken if an attacker can remove or distort the mark perhaps by embedding another mark to undermine copyright ownership. On the other hand, a steganography system is considered broken if an attacker can detect the existence of a secret been communicated even if the embedded secret is not retrieved. In other words, the two concepts differ in terms of robustness which in watermarking is a measure of the ability to remove/distort the mark while in steganography, robustness refers to the ability of the embedding scheme to avoid detection by steganalysers.

Here we note that it may be difficult to achieve absolute robustness of watermarking schemes, and, therefore, it is more realistic to aim at practical robustness, i.e. it is either infeasible to remove the mark or the amount of work needed to remove the mark results in useless output document. In this respect, Stirmark is an example of attack on invisible watermark which in reality does not remove the mark but render it undetectable (Petitcolas, et al., 1998). Depending on the application and watermarking requirements, the list of distortions and attacks to be considered includes, but is not limited to: Signal enhancement (sharpening, contrast enhancement, colour correction, gamma correction); additive and multiplicative noise (Gaussian, uniform, speckle, mosquito); linear filtering (low-pass, high-pass filtering); non-linear filtering (median filtering, morphological filtering); and lossy compression (Katzenbeisser & Petitcolas, 2002).

Another difference between watermarking and steganography is that the first is used to hide a small amount of information and therefore unlike steganography, embedding capacity is not an issue for watermarking.

Fingerprinting

This third kind of information hiding is aimed at detecting any break of licensing agreement or copyright infringement. This would be necessary for the music and film industry as well as software industry when selling multiple copies of a digital product/release to prevent secondary copying and illegal re-selling to the third party. A different fingerprint, i.e. a small serial number, would be embedded in every copy of the digital file. In this way, the fingerprint conveys information about the legal recipient of the copy rather than the source of digital data, as in the case of watermarking, in order to identify legally distributed copies of the data. In this way, the presence of individual fingerprint is useful for monitoring or tracing back the source of illegal action. Invisible hidden fingerprint requires a high robustness against standard data processing as well as malicious attacks. To some extent, differences and similarities between fingerprinting and steganography are very much like those between watermarking and steganography.

Although watermarking and fingerprinting are not strictly designed as security tools, and as such are not concerned with the secrecy of a message/mark, they share many common underpinning protection oriented concepts and objectives. With this wider interpretation of security in mind, the classification of security systems are often depicted as follows:

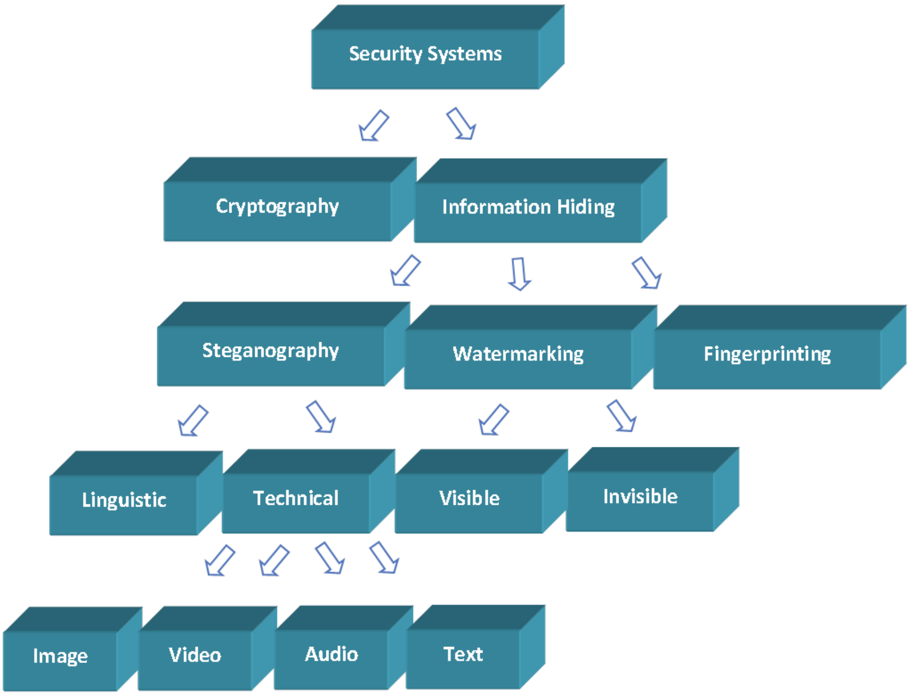


Figure 2-1: Diagram of classification of security systems.

2.2 Digital Steganography – Categorisation and Hiding Methods

Digital steganography has been categorised in the literature in different ways by different research reviews. Here we should confine our discussion to categorisations of digital steganography relating to the use of media files for cover (i.e. carrier) and how the different representation of such files can be exploited for hiding secrets. This would be more relevant to our research objectives and the stated scope of this thesis.

In steganography, file format with a high degree of redundancy is preferable since redundant bits can be replaced with secret information without the embedded information being perceivable. The data/information content of most types of digital media files are well-known for the presence of high level of redundancy, and a variety of media files can accommodate sufficient capacity for embedding large secrets. Moreover, media files are widely exchanged over the Internet than other digital files without attracting much suspicion. Therefore, digital media files are probably the richest source of cover files for steganography. Here we are concerned with the categorisation of digital steganography according to the carrier media file type and the embedding domain representation of the media file.

Carrier type based categorisation

Different type of digital media are often used as cover files, due to the fact that such files involve sufficiently large amount of redundancies, for hiding secrets without having significant impact on the information content, or quality of the stego file. The first approach to categorise steganography techniques is therefore based on the choice of the cover file type. Different types of digital media cover files have different properties and structure that would most likely dictate how the secret data can be hidden according to these properties. Understanding the common properties and structure of the type of cover file can give us an indication or idea on how and where the secret data might be hidden (Cole & Krutz, 2003). Accordingly, different steganography types can be classified as to whether the cover file is an image, audio, video, or text file. For example, the steganography system that uses digital images as cover files benefits from the different bit-planes decomposition of the images, knowledge about the statistical properties of these bit-planes, the nature of local and global natural image texture, colour distribution, as well as the properties of different frequency domain of images. For audio files, understanding the frequency of delays, pitch structure, as well as frequency decomposition can be exploited to hide secrets without being audible or effecting the quality of the signal. Hiding secrets in video files would be based on hiding the secret using the sequence of the video frames as well as the audio signal. Therefore, steganography schemes for video files can benefit from properties of the audio and visual data while providing much more payload capacity. For digital text files, steganographers exploit the formatting of the documents of variable spacing between characters/words. Added spaces before certain words may be linked to the hidden secret and interpreted in different ways including associating an importance to the following word or its first character.

Media file representation domain based categorisation

Approaches to digital steganography can be classified into two groups in terms of embedding domains: spatial domain and transformed/frequency domain methods. In what follows, we shall focus on the case where the cover file is a grayscale image.

1. Spatial Domain Techniques:

The spatial domain of an $M \times N$ image refers to the image data modelled as an $M \times N$ matrix of integers representing the gray-level intensities of image pixels each being represented as an unsigned 8-bit byte. For any such image, one can identify 8 bit-planes binary images where the i^{th} bit-plane image of an $M \times N$ grayscale image f, is the $M \times N$ binary image f_i which is defined for each pixel (x,y) as:

$$f_i\ (\mathrm{x,y})\ =\ i^{th}\ \text{bit of}\ f\ (\mathrm{x,y}) \qquad (2.\ 1)$$

From Figure 2-2, one can see the 8 bit-planes in the binary representation of a grayscale image of Lenna and how these bit-planes are significant. It is noticeable that the most significant bit-plane (MSB) contains most significant information comparing to other bit-planes and the 1^{st} LSB bit-plane contains the least significant information. In other words, when we look at each bit-plane, it does appear as though the 1^{st} LSB plane is more random than that of bit-planes of higher scale (e.g. 5^{th} or more). In fact, the first three bit-planes, from right-down, contain redundant information, and these redundancies are suitable to be exploited to embed secret bits without degrading the cover image visual quality.

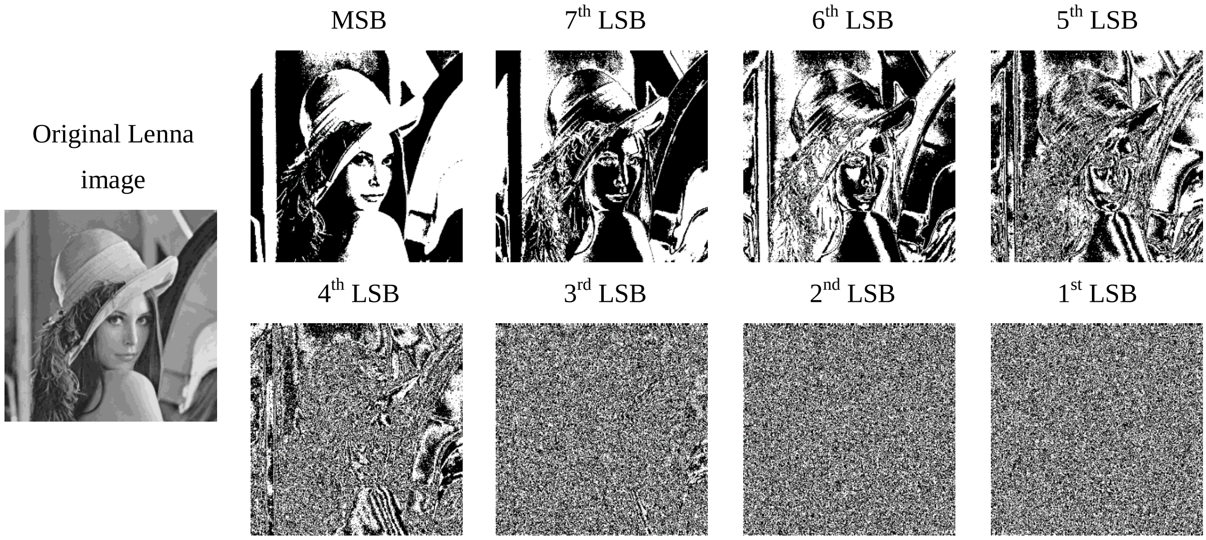


Figure 2-2: Bit-planes of Lenna image.

The least significant bit replacement/substitution (LSBR) steganography scheme and its variants are the most common embedding techniques developed over the last decades in the spatial domain. These spatial domain substitution techniques, simply replace the bits of the secret message in the LSB of the cover image pixels, perhaps using some agreed order of the selected pixels. In short, these schemes produce a stego-image which only differ from the cover image in their LSB plane and, therefore, causing little or no drastic/visible distortion to the cover image. These are relatively efficient and easy to use, and therefore, are the most common techniques used for digital steganography and especially with digital images. However, the information embedded in the LSB plane of an image could easily be destroyed by applying a slight change to the stego-image such as compression (Rabah, 2004).

Looking back at Figure 2-2, one can see that the randomness of pixels in the 2nd bit-plane, and to some extent the 3rd bit-plane, provide some opportunities for hiding secrets without being noticeable and embedding techniques have been developed that exploit randomness in these and higher bit-planes than the LSB but not without consequences. Some more details on steganography techniques based on a variation of these substitution approaches are presented and reviewed in Chapter 3 highlighting advantages and disadvantages.

Here, an example of secret bits embedding in the cover pixels' LSB is illustrated. Let the three integer numbers 16, 197, 243 be three cover pixels' value. In order to embed three secret bits 0,1,0, the cover pixels' value need to convert in binary form each of 8 bits length. The following bit-streams are binary representation of the cover pixels' value:

$$(16)_{decimal} = (00010000)_{binary}$$
$$(197)_{decimal} = (11000101)_{binary}$$
$$(243)_{decimal} = (11110011)_{binary}$$

The left-most bit in the stream is called the Most Significant Bit (MSB), and the right-most bit is called Least Significant Bit (LSB). Furthermore, the second bit from the right is called 2nd LSB, i.e. generally, the ith bit from the right is called ith LSB, where $1< i <7$. Most of steganography schemes based on the substitution techniques are replacing the secret bits with the LSB, since modifying the LSB has less effect on the cover pixel value, either it changes by 1 or remain as it is. Thus, the following bit-streams are binary representation of the stego pixels after embedding the secret bits 0, 1, 0, each bit in one pixel:

$$(00010000)_{binary} = (16)_{decimal}$$
$$(11000101)_{binary} = (197)_{decimal}$$
$$(11110010)_{binary} = (242)_{decimal}$$

As a result, cover pixel values 16, 197, 243 become 16, 197, 242 after embedding the secret bits 0, 1, 0 in each pixel respectively.

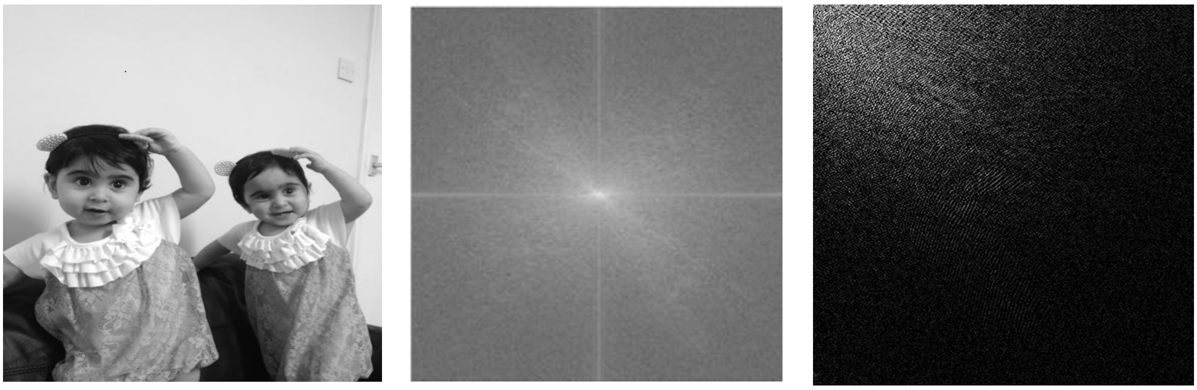
Besides the binary representation of the pixel values of grayscale images in 8-bit bytes, in recent years, other kinds of representation of pixel values have been investigated to use in steganography. The idea is that, instead of using the sequence of powers of 2, $\{2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7\}$ to represent the pixel values $\{0,1,2, \ldots 255\}$, one can use different sequences, such as the Fibonacci sequence $\{1,2,3,5, 8,\ldots, 233\}$, to express grayscale image values in other than 8-bits. These could be useful in reducing the effect on image quality when higher bit-planes are used for message embedding. In the next chapter, we shall review such schemes as well as schemes based on the use of the Lucas, Catalan, prime, and natural sequences for pixel value representation. We shall also review other schemes that manipulate/use bit-planes in ways that cannot be literally described as substitutions, although make some substitutions.

2. Frequency Domain Techniques:

The frequency domain of an image usually refers to the representation of the image (or signal) in terms of waveforms, and a variety of such waveforms have been used to decompose/transform an image signal in terms of sub-bands of the frequencies of the waveforms that generate the given image. In 1822, Jean B. Fourier the French mathematician has shown that certain types of functions (which include audio and image data files) can be represented (i.e. decomposed/analysed) by linear combination of the periodic trigonometric sinusoidal wave functions (e.g. $\sin(x, y)$, $\cos(x, y)$) of different frequencies (Gonzalez & Woods, 2002). The coefficients of the waveforms that a signal can be expressed in terms of their linear sum, is known as the frequency domain of the signal/image as compared to the original spatial domain representing the image pixel intensities. The Fourier transform can be inverted without loss, and thus become a useful tool to process/manipulate a signal/image in the spatial domain by processing/analysing the frequency content of the signal. The Fourier transform has been developed further, and its discrete version DFT has become the main tool for analysing and processing images as well as audio signals.

The Discrete Cosine Transform (DCT) is based on the real part version of the DFT and provides an efficient alternative for image compression and other image processing techniques (Gonzalez & Woods, 2002). While both DCT and DFT provide information about the frequencies of the waveforms that contribute to a signal/image, there is no information about the location of such frequencies within the signal, i.e. DFT and DCT provide frequency support but not spatial/time support. This is due to the fact that the trigonometric waveforms periodic functions whose support is infinite and covers the entire real line. For images of sufficiently large size, this makes the process of decomposing them by DFT or DCT inefficient. To overcome this shortcoming, it is customary to apply these transforms on blocks of small fixed size (usually 8×8 pixels). However, in image compression, and other processing tasks, this approach results in creating blocky effects and image artefacts.

Below is an example of an image and its transformed domains using DFT, and DCT. In the case of DFT, the displayed image is the Fourier spectrum. Since the Fourier coefficients are complex numbers, then we cannot display the corresponding frequency domain. The DCT coefficients are real numbers and can be displayed. However, like the DFT, each DCT coefficient depends on every pixel in its area of definition, and image c is the scaled DCT image computed on the whole image, not in blocks.

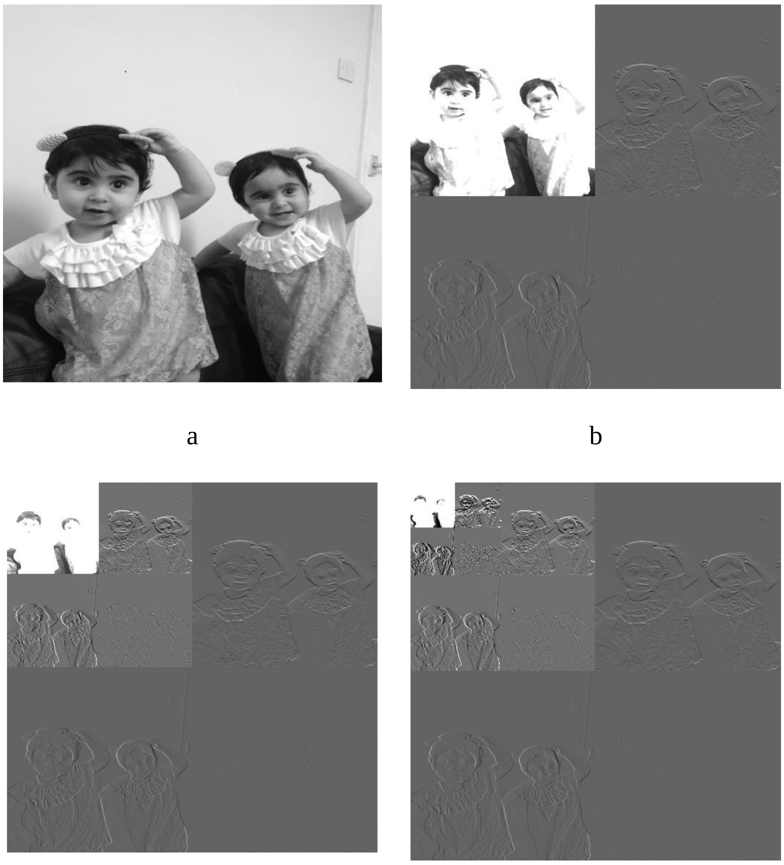


a b c

Figure 2-3: DFT and DCT Frequency domains: (a) original image, (b) spectrum of the DFT domain, and (c) DCT domain.

A Wavelet Transform (WT) is another frequency domain signal processing/analysis function that unlike the DFT and DCT simultaneously provides information about the frequencies present in the signal and their spatial location. The Discrete Wavelet Transform (DWT) is a special case of WT that decomposes a signal into at multiple scales low- and high- frequency sub-bands allowing one to extract and analyse the regular patterns as well as anomalies that may be present in the signal. It provides a compact representation of a signal/image in time and in terms frequency subranges and is efficiently computed (Gonzalez & Woods, 2002). There are a number of different ways of decomposing an image by a wavelet transform. The most commonly used

DWT image decomposition is the pyramid scheme (also referred to as the non-standard decomposition). The 2D-DWT is a multi-resolution decomposition of an image by the successive application of the DWT on the rows of the image followed by application on its columns, and is equivalent to filtering the input image with a bank of band-pass filters whose impulse responses are modelled by different scales of the same mother wavelet. Consequently, a wavelet-transformed image is decomposed into a set of sub-bands with different resolutions each represented by a different frequency band. At a resolution level of k, the pyramidal scheme decomposes an image I into 3k + 1 sub-bands (LLk, HLk, LHk, HHk, . . . , HL1, LH1, HH1). LLk represents the k-level approximation of the image, while HL1, LH1, and HH1 contain vertical, horizontal, and diagonal features of the image I (see Figure 2-4).



a b

c d

Figure 2-4: DWT (a) Original image, (b) Level 1, (c) Level 2, (d) Level 3.

For these various frequency domain representation of images, changing the coefficients slightly or the quantisation is expected have comparably slight effect on the visual appearance of the images when the transforms are inverted. Therefore, all these provide opportunities to manipulate the frequency domain coefficients to embed a secret stream without raising suspicion.

Embedding in the transformed domain is performed on the coefficients of the transformed domain of the image. The three main types of transforms used for image-based steganography are (Codr, 2009): Discrete Fourier Transform DFT (Bhattacharyya, et al., 2009), Discrete Wavelet Transform DWT (Chen, et al., 2006), and Discrete Cosine Transform DCT (Westfeld, 2001). More details with an example of message embedding in the frequency domain are given in the next chapter.

Although, in this thesis, we will only develop spatial domain steganography schemes, we shall be using wavelet to manipulate secret images for improved embedding efficiency and message detectability (see Chapter 5).

2.3 Steganalysis and Steganography Attacks

Steganalysis is the study of detecting the presence of suspect communication transaction that carries a steganographically hidden secret, i.e. the art of seeing the unseen. The two fields, therefore, operate in a 'cat and mouse' style strategy, and steganalysers attempt to defeat the goal of steganographers by detecting the presence of a hidden message. Attacks on general information hiding can be classified as active or passive attacks. Active attacks aim to destroy the embedded secret message while passive attacks aim to determine the presence/absence of a hidden message and estimate its size.

Active attacks assume that the attacker can capture the stego file and change it by introducing distortion before passing it on in order to prevent secret communication (Cox, et al., 2005). Examples of active attack are linear and non-linear filters (e.g. blurring, sharpening, median filtering), lossy compression, gamma correction, recolouring, resampling, scaling, rotation, noise adding, cropping, etc. (Fridrich, 1999). These kinds of attacks are most likely to be used for watermarking and authentication applications rather than attacking steganography files.

The passive attack, also known as steganalysis, do not attempt to interference by altering the suspect stego file, but can either prevent or permit the message delivery. The communication between parties will be blocked if the warden suspects that a secret is being communicated. Currently, most steganography research is concerned with such kind of scenarios, (Cox, et al., 2005). In general, neighbouring pixels in natural images (i.e. images without hidden secrets) are known to be highly correlated and there is a certain level of statistical dependence between the LSB-plane and the other bit-planes beside statistical properties that exists between pairs of consecutive gray values ($0 \leftrightarrow 1$, $2 \leftrightarrow 3$ …, $254 \leftrightarrow 255$). The act of embedding a secret in a natural image will result in changing these known correlations and statistics in a manner that is influenced by the payload. Steganalysis tools have designed to exploit these facts by analysing images to discover whether the image contains a secret message or not. Here, we describe two classes of steganalysis:

1- *Structural steganalysis*: Aim to detect specific modifications due to the parity structure of the LSB replacement using local pixels' correlation. While efficient, such detectors rely on empirical pixel correlation models and do not exploit global statistical methods (Cogranne, et al., 2014). This class includes the *regular and singular group* RS tool (Fridrich, et al., 2001), the *weighted stego* WS tool (Fridrich & Goljan, 2004), the *revised WS* tool (Ker & Bohme, 2008), and LSBM steganalyser (Ker, 2005).

2- *Statistical steganalysis*: Analyses the underlying statistics of an image in order to detect modifications due to statistical property of the stego-image (Kaur & Kaur, 2014). This class includes the *pairs of value* (PoV) tool (Westfeld & Pfitzmann, 2000) and the *difference image histogram* (DIH) tool (Zhang & Ping, 2003).

A detailed description of these steganalysis tools will be given in Chapter 3. It is clear that these tools target specific steganography embedding schemes and are therefore referred to in the literature as targeted (specific) steganalysis tools. For example, the RS, PoV, the two versions of the WS, and DIH tools are designed to break the steganography embedding techniques that are based on LSB replacement, while Ker's LSBM steganalysis is designed to break the LSB matching embedding techniques (Sharp, 2001).

In recent years, interest has increased in non-targeted steganalysis tools, also known as *blind* steganalysis, whereby no knowledge of the algorithm or its effect is assumed. While, the targeted tools are designed to defeat certain steganography embedding algorithms that operate on the LSB, blind steganalysis tools are designed to detect the existence of secret messages embedded in digital media irrespective of the steganography embedding algorithm (Luo, et al., 2008). This type of steganalysis is referred to as universal in that it attempts to detect different types of steganography embedding techniques. For example, the spatial rich model (SRM) developed by (Fridrich & Kodovsky, 2012) is designed to break different steganography systems and tested on three different steganography techniques such as LSB matching (Sharp, 2001), edge adaptive (EA) (Luo, et al., 2010) and Highly Un-detectable steGO (HUGO) (Pevn, et al., 2010). These attacks are based on the fact that any embedding method creates different minor local distortions (referred to as features) throughout the cover image and modelling such features (i.e. quantifying the relationship between a pixel and its neighbours) could help reveal the presence of secrets. However, these methods cannot get any information about the amount of embedded messages (Zhang & Ping, 2003). Universal steganalysis can be considered as a two-class pattern recognition problem and consists of two parts, feature extraction and pattern classification. Universal detection aims at classifying given images into two categories: cover and stego images. Some existing universal image steganalysis methods first extract some features from images, then select or design a classifier, and train the classifier using the features extracted from training image sets, and lastly, classify the features (Luo, et al., 2008). Generally, classifiers like a Fisher Linear Discriminants (FLDs) or Support Vector Machine (SVM) are used. The general framework of blind steganalysis is illustrated in Figure 2-5. Such steganalysis techniques are less accurate compared to targeted steganalysis since they can detect a wider class of steganography techniques. Since such kinds of steganalysis are feature-based steganalysis, where a set of effective statistical/distortion features is extracted to differentiate cover images from stego-images; therefore take longer time and are not considered as real-time tools (Ker, et al., 2013) (Holub, et al., 2014).

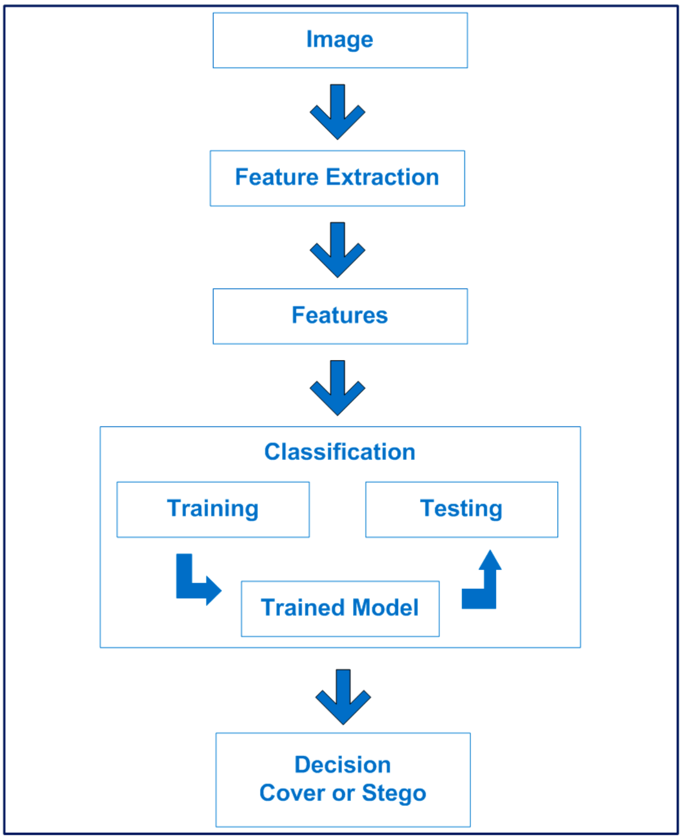


Figure 2-5: General framework of universal steganalysis.

Moreover, some of these steganalysis techniques only determine the presence/absence of the embedded message while some others go further attempting to estimate the size of the embedded message such as WS (Fridrich & Goljan, 2004), revisiting WS (Ker & Bohme, 2008) and DIH (Zhang & Ping, 2003).

Most of the steganalysis mentioned in this section are used in our experiments to evaluate the un-detectability performance of the proposed steganography schemes and will, therefore, be described in more details and reviewed in the next chapter. The diagram below illustrates the above classification of steganalysis techniques.

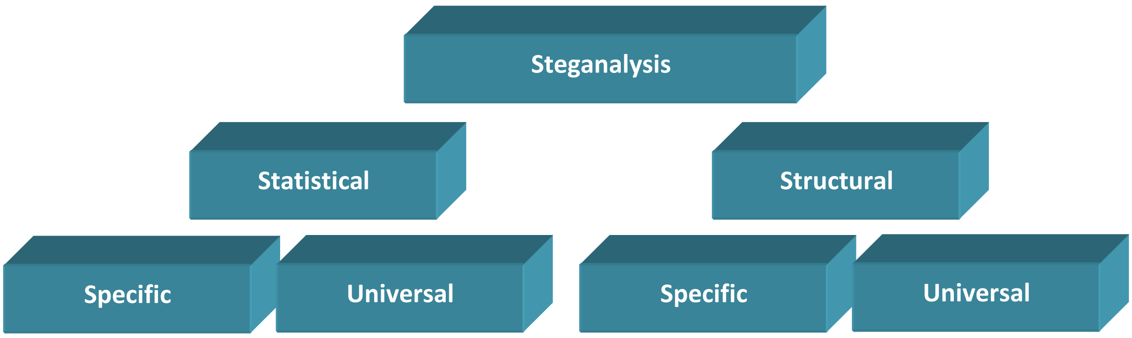


Figure 2-6: Classification of the steganalysis techniques.

2.4 Performance Evaluations of Image Steganography Techniques

In order to evaluate and compare the performance of a steganography technique, we need some criteria that could be quantitatively measured directly from the stego file. Currently, no standard test or measure is available in order to evaluate the performance or the effectiveness of steganography systems, and benchmarking approaches for steganography algorithms or applications are uncommon (Kraetzer, 2007).

In Section 1.3, five commonly used evaluation criteria for image steganography techniques were identified: payload capacity, stego-image quality, detectability, robustness against active attacks, and embedding efficiency. Since we have discussed before that steganography techniques do not need to be robust against active attacks (Cox, et al., 2005) (Cox, et al., 2007); but desirable steganography techniques should satisfy high embedding capacity and imperceptibility. In this section, we present and discuss the four remaining criteria.

2.4.1 Data Payload or Capacity

This defines the maximum length of secret binary string that can be embedded in the cover image while all other requirements are met. In the case of image spatial domain based steganography techniques, the payload may be stated in units of measurements such as the data embedding rate in terms bits per pixel (bpp), or the ratio of the secret message to number of cover pixels. When bpp = 1.0, then the number of embedded secret bits is equal to the number of cover pixels and also means that the embedding rate is 100 % or full capacity. In this thesis, the capacity is measured by using embedding ratio, i.e. if a cover image I is of size $M \times N$ pixels, and the length of the embedded secret is L bits, then the embedding ratio p is given by equation (2. 2):

$$p = \frac{L}{M \times N} \qquad (2.\ 2)$$

As discussed before, there is a trade-off between the payload capacity and imperceptibility. Nevertheless, steganography techniques that embed messages for which $L > (M \times N)$ and introduce distortions to stego files are considered as worthless systems. On the other hand, increasing the steganography capacity while maintaining an acceptable level of stego-image quality is considered a positive contribution. Additionally, improving the stego-image quality while maintaining the steganography capacity is also considered a significant contribution (Wu & Hwang, 2007).

2.4.2 Stego-image Quality

In steganography, stego-image quality is an indicator that there is no visual difference between the cover image and the stego-image. Evaluating the quality of stego-images is a significant indicator of the performance of the embedding algorithm (Wu & Hwang, 2007). Generally, there are two ways to measure stego-image quality: objective quality methods and subjective quality methods (Stoica, et al., 2003). The objective methods are based on measurements automatically computed using the image data, while subjective methods are based on human observer judgement. In practice, subjective evaluation is usually too inconvenient, time consuming and expensive. The goal of objective image quality assessment is to develop quantitative measures that can automatically predict perceived image quality. Objective image quality evaluation metrics are classified into three categories according to the availability of the original image (reference): full reference (FR), no-reference (NR), and reduced reference (RR) image quality assessment (Wang, et al., 2003). The full reference means that the original image and the test image are available, while the no reference means that only the test image is available. The reduced reference means that the test image and some information about the original image are available (Ponomarenko, et al., 2008).

For objective quality methods, two types of perceptibility can be distinguished and evaluated in signal processing systems, namely fidelity and quality. Fidelity means the perceptual similarity between signals before and after processing. However, quality is an absolute measure of the goodness of an image as perceived by the human eye. For example, a distorted, blurred and low-resolution grayscale image is naturally considered to be of low quality. A stego-image is expected to look identical to the cover image but it may have a slightly lower quality, but because it is indistinguishable from the cover image, then it would have high fidelity. For image-based steganography, the fidelity is defined as the perceptual similarity between the original cover image and the stego-image. Therefore, the fidelity evaluation requires both images before and after embedding. However, attackers and perhaps recipients do not have access to the original cover image. Additionally, steganography systems must avoid attracting the attention of anyone not involved in the secret communication process and therefore stego-images must have a reasonably good quality. Therefore, quality is the major perceptual concern for most steganography techniques in order to avoid any suspension and therefore detection (Cox, et al., 2005).

There are two measurements, the peak signal-to-noise ratio (PSNR) and the mean square error (MSE) that are widely used as image quality measures. Both represent perceptual distance metrics and quantify the distortion amount between an image and a processed version of it. By definition, these two are measures of similarity between two images (Wang, et al., 2003) and, therefore, are fidelity metrics and not as quality measures. Significantly, fidelity is defined as the perceptual quality of stego files and therefore PSNR, and MSE describe how imperceptible the secret message is (Cox, et al., 2005). Although MSE and PSNR can result in poor performance, and they are not very well matched to perceived visual quality, they are still applicable in several image processing applications for their simplicity in computation and independence of viewing conditions and individual observers (Wang & Bovik, 2002) (Wang, et al., 2004). Thus, in this thesis, we are adopting the use of these two measures as indicators of perceptibility of the secret message in the stego-image. Accordingly, a high imperceptible secret in a stego-image can be discerned from a high PSNR value, and, therefore, both cover image and stego-image are perceived to be very similar.

In our experiments, the quality of the stego-image is examined using the PSNR to test the performance of the various embedding schemes developed in this thesis in terms of this criteria. For self-containment, we shall now formally state the definition of MSE and PSNR measures.

1. Mean Square Error (MSE):

   MSE is a full reference (FR) metrics used to measure the difference between two images, (Wang, et al., 2003) (Stoica, et al., 2003). It is the average of square of differences between the pixel values in the two images, i.e.

$$\text{MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{ij} - X'_{ij})^2 \qquad (2.\ 3)$$

   Where:
   X_{ij} is the i^{th} row and the j^{th}column pixel in the original image (cover image).
   X'_{ij} is the i^{th} row and the j^{th}column pixel in the reconstructed image (stego-image).
   M and N are the height and width of the image.

2. Peak Signal-to-Noise Ratio (PSNR):

   PSNR is another full reference (FR) metrics used for objective image quality evaluation. Like MSE, it is a measurement of similarity between two images. It is used in many image processing applications and considered as a reference model to evaluate the efficiency of other objective image quality evaluation methods (Wang, et al., 2002). It is widely used and very popular, since the computation of these two metrics is very easy and fast (Ponomarenko, et al., 2008). PSNR is a logarithmic function of MSE and is measured in decibels (dB) units, (Wang, et al., 2003) (Stoica, et al., 2003):

$$\text{PSNR} = 10.log_{10}\frac{I^2}{MSE} \qquad (2.\ 4)$$

Where, I is the maximum pixel value. For the 8-bit grayscale image, I = 255. The resultant PSNR is a decimal value between 0 and infinity (∞). In the case of two identical images, the PSNR value is ∞. Moreover, the higher the value of PSNR indicates the higher similarity between the cover and stego-image.

2.4.3 Un-detectability of Hidden Secrets

Un-detectability of a hidden secret in an otherwise mundane communication transaction is the main and most important requirement for any steganography system. By un-detectability, we understand the inability of an attacker/steganalyser to distinguish between cover and stego-image. For modern digital communication, it is somewhat impractical to be concerned with detectability by a human observer. Hence, the un-detectability requirement is concerned with the ability of the embedding algorithm to withstand against steganalysis attacks that aim to decide whether an input image has a secret embedded in it or not. This basically means that the produced stego-images should be statistically undistinguishable from cover images (Fridrich & Goljan, 2004). As discussed in Section 2.3, there are a variety of steganalysis techniques for determining whether or not an image contains a secret message.

2.4.4 Embedding System Efficiency

The embedding efficiency is an important attribute of steganography techniques directly influencing their security and is defined by (Fridrich & Soukal, 2006) as the number of message bits embedded per one change as a result of embedding. Thus, in image-based steganography, high embedding efficiency refers to reducing the number of necessary changes of cover pixels for a given embedding rate. The concept of embedding efficiency has been first introduced by (Crandall, 1998), and was first adopted by (Westfeld, 2001) for embedding in DCT domain. It has since been accepted as an important attribute of steganography schemes (Fridrich, et al., 2007). Because a smaller number of embedding changes is less likely to disrupt statistical properties of the cover image, schemes that employ high embedding efficiency generally have better steganography security. In other words, steganography techniques that employ high embedding efficiency, they produce stego-images with minimal distortion.

A formal definition of steganography security was given by (Cachin, 1998), and the concept of embedding efficiency is an essential indicator of steganography security. The detectability of a data hidden in a stego-image is influenced by many factors, such as the choice of the cover object, the selection rule used to identify individual elements of the cover that could be modified during embedding, the type of the embedding operation that modifies the cover elements, and the number of embedding changes, directly related to the secret message length. Assuming two embedding techniques share the same source of cover object, the same selection rule and embedding operation, the one that introduces fewer embedding changes will be less detectable as it decreases the chance that any statistics used by the warden will be sufficiently disturbed to mount a successful steganalysis attack (Fridrich, et al., 2007). Since our concern and contribution in this thesis is embedding efficiency, some of image-based steganography techniques that have high embedding efficiency are reviewed in the next chapter. The embedding efficiency can be calculated such as:

$$\text{embedding efficiency} = \frac{1}{ratio\ of\ modified\ pixels} \tag{2. 5}$$

The embedding efficiency also can be calculated by the ratio of necessary cover pixel change, the number of cover pixels that need to be changed after embedding the secrets proportion to the length of the secret message. The higher is the value of embedding efficiency; the lower is the embedding change of the cover pixels.

2.5 Summary

In this chapter, we have covered the necessary background related to the research area of the thesis by introducing the reader to the main information concerning the relationship between steganography and information security. The general discussion covered steganography within the wider security mechanism of information hiding and elaborated on the distinction between steganography and the other hiding schemes (watermarking and fingerprinting). We also discussed and identified the various security services provided by the two secret communication mechanisms, namely steganography and cryptography. Moreover, a brief insight into steganography categorisation based on carrier type and image domain is given. Furthermore, we have investigated issues concerning steganalysis and steganography attacks. Finally, different criteria used to evaluate the performance of steganography techniques and to make a decision of which steganography technique is better than another is presented.

In the next chapter, Chapter 3, we review the literature for research on image-based steganography approaches, highlighting the strengths and limitations of each of them. In addition, some well-known steganalysis tools for detecting the secret message that embedded in the image spatial domain are described and reviewed.

Chapter 3

Image-based Steganography and Steganalysis: Literature Review

In the previous chapter, general background information about digital steganography was presented and discussed to provide the reader with sufficient knowledge of the research area of interest in this thesis. The focus was on the link between information hiding and cryptography as security mechanisms, general categorisation of different known digital steganography schemes, basic attacks on steganography systems, and different measurements that are used to evaluate the performance of the image-based steganography schemes. In this chapter, we conduct a literature review of the most relevant image-based steganography schemes in Section 3.1, and different steganalysis tools in Section 3.2. We shall also give Section 3.3, a brief overview of our proposed approaches in this thesis.

3.1 Image-based Steganography Approaches

We have already pointed out that digital carriers/covers such as audio, image and video files have become the most obvious choices to use in digital steganography in order to conceal a secret message into it. In this thesis, we are using images as cover files for carrying secret messages, due to the fact that images usually have a high degree of redundancy, widely exchanged over the Internet than other digital media and do not attract suspicion. Therefore, the literature review, in this chapter, will be limited to digital steganography approaches that have been developed for grayscale images. This review will cover embedding methods for the spatial domain and transformed/frequency domain image representation. We shall first review the frequency domain techniques.

Frequency Domain Steganography

These techniques work first by using a frequency domain transform of the input spatial domain cover image, and then exploiting redundancies in the transformed coefficients, or other properties usually used for compression or other frequency domain image processing such as quantisation, to embed the secret message such that inverting the frequency transform produces that stego-image have little or no effect on the visual appearance of the cover image.

The three main types of frequency domain image transforms, described in Chapter 2, have been used for image-based steganography. The following references describe each of these three frequency domain steganography techniques.

The DCT-based F5 steganography technique was developed by Andreas Westfield with the aim of preserving the statistical properties of a stego-image (Westfield, 2001). The DCT transforms each block 8 x 8 of cover pixels into 8 x 8 matrix of frequency coefficients that are real numbers. The entries of the matrix appear in the order of their absolute values along its zig-zag entries, with the AC coefficient in the top left corner being the most significant low frequency content representing the block energy. This property has been exploited for compression whereby as many as possible insignificant DC coefficients along the zig-zag path are ignored, from a selected position onwards, after which a quantisation step is used to reduce the number of symbols to be coded. During secret embedding, instead of changing the LSBs of the quantised DCT coefficients; F5 algorithm decrements the absolute value of the quantised coefficients by one (Westfield, 2001). The F5 adopted quantisation suitably rounds the selected DCT coefficients to integers in the range -2048 to 2047. In order to minimise the necessary number of changes when embedding a message, the F5 algorithm employs the matrix embedding algorithm proposed by (Crandall, 1998). It does not embed the secret bits sequentially into the DCT coefficients but into randomly chosen DCT coefficients. The F5 is one of the most popular embedding schemes in DCT domain steganography for its robustness against statistical steganalysis attacks (e.g. the PoV) though it has been successfully broken in (Fridrich et. al, 2003).

An image embedding technique based on DFT is proposed by (Bhattacharyya et. al, 2009) with the aim to be resistant against the statistical attack of PoV. The technique first divides the cover image into non-overlapping blocks of size 2 x 2 and transforms the blocks spatial domain into the frequency domain using DFT. The bits of the secret message are then embedded in the LSB within the real part of the DFT coefficients excluding the first one. Unfortunately, the authors do not pay any attention to any of the other requirements of steganography such as stego-image quality and payload capacity.

The main disadvantage of using DFT or DCT transforms for embedding is that these transforms do not provide spatial support that basically implies that every coefficient in the frequency domain depends on and is affected by, every image pixel. Hence, distortion as a result of embedding one bit will be spread over the entire image or the block. Inevitably this will have an impact on stego-image quality which can only be dealt with by limiting the payload capacity. On the other hand, DWT-based steganography do not have a similar disadvantage. And there has been a great interest in DWT-based steganography as well as watermarking.

In 2006, Chen et al. proposed a steganography technique which embeds the secret message in DWT domain with the aim of keeping the message safe from being destroyed by unintended users on the Internet. The secret message embeds in the high frequency coefficients of the DWT domain by substituting the secret bits with the LSB of the DWT high frequency coefficients while coefficients in the low frequency sub-band are preserved unaltered to improve image quality (Chen, et al., 2006).

Frequency domain steganography techniques are expected to be more robust against active attacks. However, embedding secrets in the frequency domain of images are known to have several limitations including the limited payload capacity. Even the embedding of a small message is known to have a significant effect on the cover image quality (Chen, et al., 2006). Furthermore, embedding a secret bit in the frequency domain may have an effect on more than one pixel. These known disadvantages of frequency based steganography are the main reason for our interest in spatial domain steganography. In particular, we would be aiming to improve security/un-detectability of the steganography technique, enhance embedding efficiency while maintaining the secret message quantity. In the next section, we shall have a more extensive review of spatial domain steganography for grayscale images.

Spatial Domain Steganography

Many steganography approaches for embedding secret messages in images' spatial domain have been proposed, and it is evident that spatial domain based steganography is probably the most dominant approach in the literature. There are many aspects of this area of research that we need to review, and the next few introductory paragraphs are meant to highlight briefly these aspects that have greatly influenced the research we conducted in this thesis. The content of these paragraphs will be expanded in the following subsections for more substantive review of the literature.

In general, image-based steganography approaches are often classified into adaptive and non-adaptive approaches (Agaian, et al., 2007). In adaptive approaches, the embedding capacity and positions depend on the statistical characteristics of the cover image (Westfeld, 2001), and this means that some of the regions are avoided for secret embedding. Whereas, in non-adaptive approaches, data embedding does not depend on the cover image content and every pixel is used. Thus, in non-adaptive based steganography approaches, the embedding rate is higher than in adaptive based approaches. However, adaptive steganography approaches are more robust against steganalysis techniques, since the message is embedded in *noisy* regions, but it has a limitation of capacity.

Existing spatial domain embedding schemes are mostly designed to embed secret bit-streams in places where there would be the least effect on stego-image quality and make least impact on perceptibility. The LSB plane of an image is the most obvious source of such places. However, there are also techniques that embed the secret message in higher bit-planes than LSB plane. There are some schemes that do not directly replace bit-planes with the secret bits, but by modifying pixel values according to a particular (reversible) function (Picione, et al., 2006). An extensive review of these schemes will be presented in Section 3.1.1 where we also highlight advantages and disadvantages.

Yet, other schemes apply similar techniques but using different decompositions of pixel value integers other than usual binary sequences. In Section 3.1.2, we will review the literation relating to steganography based on different decomposition schemes for cover pixel values.

Security or message detectability is considered as the most important requirement for image-based steganography schemes. Chao Wang et al. identified two common ways to enhance steganography security: 1) reduce the embedding changes at a given embedding rate, i.e., to increase the embedding efficiency; and 2) embed the secret bit into the cover pixels only in inconspicuous parts, e.g., the noisy regions of an image (Wang, et al., 2010). In 2013, Ker et al. discussed and highlighted the problems of steganography and steganalysis that are important to be addressed in the future research. The two main important problems that related to the spatial domain based image steganography are: 1) design efficient embedding schemes – sender hides the message while minimising an embedding distortion, and 2) design distortion functions relating to the statistical detectability – sender hides the message in the regions of the cover image that determined by the defined distortion function, e.g. noisy or textured regions (Ker, et al., 2013). In Section 3.1.3 and 3.1.4, we will review the literation relating to the two security-related issues of region based embedding and embedding efficiency. In this thesis, high embedding efficiency and message un-detectability will be of major concern to us, and to some extent, we focus on addressing the first problem mentioned in (Ker, et al., 2013).

3.1.1 LSB\higher LSBs (Bit-Planes) based embedding Approaches

The common theme in these schemes is to embed the secret message bits into, a priory selected and agreed with the receiver, bit-plane(s) of the cover image. The embedding could take the form of replacing the bits of the chosen bit-plane with the secret bit-stream according to agreed order of the image pixels. The most common algorithm belonging to this class is the scheme that selects and uses the LSB of the binary representation of the cover pixels to represent the message bit was first suggested by (Bender, et al., 1996) and explained by (Chan & Cheng, 2004) (Thien & Lin, 2003). In the literature, this scheme is referred to as the Least Significant Bit Replacement (LSBR) and its popularity is due to its simplicity, ease of implementation, and visual imperceptibility. Moreover, LSBR supports full payload in the sense that every cover pixel can be used to carry a secret bit, and it is difficult to notice a change in the value of the pixel by the naked eye. LSBR was first used by embedding the secret bits into cover pixels in sequential order, and it is referred to as LSBR sequentially. This scheme is not secure, since attackers can simply retrieve the LSB plane to quickly recover the hidden information (Hempstalk, 2006).

The security problem of the LSBR sequentially can be partially mitigated by the use of pseudorandom number generator (PRNG) to randomly distribute the hidden message across the cover image according to a seed that is specified by the sender instead of embedding the message in sequential order (Hempstalk, 2006). This is called LSBR randomly embedding technique (Provos & Honeyman, 2003). Using the same PRNG at the receiver part, the secret bits can be extracted from the stego pixels' LSB.

Both LSBR sequentially and LSBR randomly, increase (decrease) even (odd) pixel values either by one or leave unchanged. This creates an imbalance in the embedding distortion in the stego-image as a result of distorting the statistical distribution in the pixel values (0, 1); (2, 3); . . . (254, 255) (Luo, et al., 2010). This imbalance is called *asymmetry* problem, which can be exploited to detect the existence of a hidden message using certain targeted steganalysis techniques, even at a low embedding rate.

To overcome the undesirable asymmetry problem of LSBR schemes, the decision of changing the least significant bit is randomized, i.e. if the message bit does not match the cover pixel's LSB, then cover pixel value is randomly either increased or decreased by 1. This technique is popularly known as LSB Matching (LSBM), also called ± embedding, and was proposed by (Sharp, 2001). After embedding the secret message, LSB of the stego pixel represents a secret bit and by extracting it at the receiver part, the message can be obtained. LSBM based embedding technique does not suffer from the asymmetry problem and has the same payload capacity of the LSBR scheme with good visual imperceptibility property, i.e. not noticeable by the naked eye. Andrew Ker in (Ker, 2005), has pointed out that the LSBM approach is dealing with the asymmetric problem by randomizing the change, unfortunately, result in creating another problem; designed a steganalysis tool to defeat it. The reported disadvantage is concerned with changes to the DFT of the histogram when the image is down-sampled. In theory, down-sampling images should not have changes to their histograms significant enough to affect the DFT of the histograms (see Section 3.2).

In order to avoid the above mentioned vulnerabilities of the LSBR and LSBM schemes and possibly increase payload capacity, new steganography approaches emerged whereby the secret bits are not only embedded in the cover image LSB plane but also embed in higher bit-planes. However, such kinds of techniques are expected to have degrading effects on the quality of the stego-image compared to the LSB-only schemes. We shall now review few such schemes.

A digital steganography scheme that embeds two secret bits into the first two LSBs of the pixels of a cover image, called 2LSB, was developed in (Ker, 2007). The advantage of the 2LSB techniques is the doubling of payload capacity compared to LSBR and LSBM schemes. In the worst case, pixel values could change by 3, and this leads to distorting the stego-image quality more comparing to LSBR and LSBM based embedding techniques. Although Ker proposed a steganalysis technique to detect the secrets embedded in 2LSB (Ker, 2007), but 2LSB embedding techniques is still harder to be detected by steganalysis techniques that are designed to detect the embedded secrets in LSB-only schemes (Ker, 2007).

LSB-Witness embedding technique is proposed by Rashid et al. in which the secret message is already present in the LSB plane but instead of changing the cover image LSB values, the second LSB plane will be changed as a witness/informer to the receiver during message recovery. For the extraction purpose, only second bit-plane needs to be checked, if the value of the second bit-plane is 0 then the secret bit is equal to the LSB plane bit value, otherwise the secret bit is inverse of LSB plane value. Although this approach may affect the stego-image quality, it eliminates the weakness of the LSBR schemes that exploited by steganalysis techniques that designed to detect the secret bits embedded in LSB (Rashid, et al., 2013).

Wang et al. proposed an embedding algorithm that embeds the secret bit in the 4^{th} LSB by applying bit-plane substitution method and a local pixel adjustment process to reduce the cover pixel degradation (Wang, et al., 2000). If the secret bit replaced directly the 4^{th} LSB, then the cover pixel value either not change or it will change by ± 8, and this significant change leads to degrading stego-image quality. The local pixel adjustment procedure proposed by Wang et al, is applied when the secret bit does not match the 4^{th} LSB of the cover pixel by modifying the other bit-planes (from 1^{st} to 3^{rd}) according to some assumptions/cases reported in (Wang, et al., 2000). The receiver can retrieve the secret bits only by extracting from the 4^{th} LSB of the stego-image pixels. The following example illustrates this adjustment:

Let the cover pixel value P_i = 8 = 00001000_2 and the secret bit be 0. Replacing the 4^{th} LSB by the secret bit makes the corresponding stego pixel value P_i' = 0 = 00000000_2, i.e. an error e_i= P_i' - P_i = -8. Instead, the pixel value is adjusted to P_i'' =7 = 00000111_2, which reduces the error to e_i= P_i'' - P_i = -1.

The authors claimed that this embedding technique improve the stego-image quality comparing to the directly replace the secret bit with 4^{th} LSB, and illustrated this by embedding a secret in all pixels of Lenna image using the direct replacement of the 4^{th} LSB and the proposed embedding technique and showing that the value of PSNR has increased from 33.02 to 38.75. However, if P_i = 31 = 00011111_2 and the secret bit be 0, then the error becomes -8.

In 2001, Chan and Chen improved the above scheme by embedding the secret bit in the 4th LSB but based on special look-up table reported in (Chan & Cheng, 2001), that instead of only modifying the 1st to 3rd LSBs it modifies all of the bit-planes except 4th LSB. It is claimed this scheme improved the PSNR of the stego Lenna image to 42.352. The following example illustrates this approach:

> If the cover pixel value P_i = 31 = 00011111_2 and the secret bit be 0, then the scheme first replaces the 4th bit with the secret makes the corresponding stego pixel value P_i' = 23 = 00010111_2 and calculates the error e_i= P_i' - P_i = -8 which is high, then the scheme changes as many bits as necessary as long as the error is reduced. Hence, in this case, the corresponding stego pixel value P_i'' =32=00100000_2, i.e. an error of 1.

In (Chan & Cheng, 2004), a new embedding scheme has proposed, called optimal pixel adjustment process (OPAP), that embeds 3 bit secrets in the first 3 LSBs of a single cover pixel and then uses a modification of the above local adjustment. It is aimed to enhance the stego-image quality obtained by simple bit-plane substitution method. The main idea of OPAP is to minimise the error between the cover and stego-image based on three cases determined by a partition of the e_i between P_i and P_i' into 3 subsets and adjusting the bits beyond the 3rd LSB in a way that depends on the subset that the error e_i belongs to. The following example illustrates the working of the OPAP:

> Let P_i = 8 = 00001000_2 and the secret bits be 111_2. The corresponding stego pixel value obtained by conventional substitution method P_i' =15 = 00001111_2, i.e. an error of 7 and it falls into case 1 out of the cases reported in (Chan & Cheng, 2004). The OPAP embedding scheme makes the corresponding stego pixel value P_i''= 7 = 00000111_2, i.e. an error e_i= P_i'' - P_i = -1.

As a result, the authors claim that the stego-image quality can be improved while the number of secret bits that can be embedded has increased three times of that in LSBR and LSBM based embedding techniques.

Daneshkhah et al. proposed an embedding technique that embeds two bits of information in a cover pixel in a way that not only the LSB of the cover pixel is allowed to change but also the second and fourth LSBs are allowed to be manipulated (Daneshkhah, et al., 2011). In this technique, for embedding two secret bits in a cover pixel, only one alteration in one bit-plane happens. To guarantee retrieval of the secret bits from the stego pixels, the authors have designed a (3,1) convolution decoder circuit, which outputs three bits for every input of first four LSBs of the cover pixel. The highest output bit is ignored, and the other two output bits are replaced with the two secret bits which will then be used to represent the two secret bits. As the authors claimed, this proposed embedding technique has the advantages of capacity (two secret bits embedded in one cover pixel), and the detection of the embedded message became much harder for steganalysis compared to LSBR or LSBM techniques (Daneshkhah, et al., 2011). However, stego quality is degraded compared to LSBR and LSBM embedding techniques.

In 2012, Janakiraman et al. proposed a new embedding technique by extending the idea of (Daneshkhah, et al., 2011). In this technique, a maximum of 1 or 2 bit-planes has been altered to embed four secret bits. This technique would not just embed the secret bit in LSB of the cover pixel, but also it might be embedding the secrets in the second, third, and fifth bit-planes or any one of the 15 possible combinations. Besides improved capacity, the authors claim improved un-detectability. However, the stego-image quality can degrade since the fifth bit-plane might also be altered.

In all the above schemes that embed in higher bit-planes, the changes are made regardless of knowledge of the surrounding of the next pixel to which a secret bit is to be embedded. However, the visible effect of such actions depends on whether the surrounding region is dark or very light. This idea was exploited by a Buckingham MSc. student in his project by designing an illumination-adaptive higher bit-plane embedding scheme (Abdullah, et al., 2014). It is based on determining the recorded lighting condition and computed quality of the cover image prior to embedding. It divides the cover image into blocks and identifies blocks according to their lighting conditions. The most useful blocks for embedding are based on their entropy and average values. According to this, the scheme selects the right bit-plane for embedding. This kind of block selection made the embedding process scatters the secret messages randomly around the cover image. Different tests have been performed for selecting a proper block size, and this is related to the nature of the used cover image. Experimental results reported in (Abdullah, et al., 2014) demonstrate that different image quality used for the cover images will have an effect when the stego-image is attacked by different active attacks. Although the secret bits are embedded in higher bit-plane, they cannot be recognised visually within the stego-images.

In summary, hiding approaches reviewed in this section that embed secrets in higher bit-planes than LSB are aimed to increase the payload capacity by embedding more than one secret bit in each selected cover pixel and/or increase the un-detectability. However, if we only focus on increasing the data hiding capacity, the PSNR decreases, and the stego-image appears distorted which hampers the main aim of image steganography, i.e. stealth hiding. In the next section, different steganography approaches that are based on representing cover pixel values in other than the usual binary system will discuss.

3.1.2 Pixel value decomposition based embedding Approaches

In image processing, it is customary to represent pixel values of grayscale images as an 8-bit byte. Each greyscale integers in the range {0,1, …, 255} is decomposed uniquely in terms of its partition as the sum of powers of 2 in the sequence $\{2^0, 2^1, 2^2, \ldots, 2^7\}$. This is also influenced by the way computers process data, but as we saw in the last few examples embedding in higher bit-planes could result in significant changes in pixel values, compared to embedding in LSB, unless special mechanisms are used to avoid this such as changing different bits as in the case of (Chan & Cheng, 2001) scheme. However, in recent years, many steganography researchers recognised to the possibility of using other sequences of integers to decompose pixel values while adhering to use of binary strings but in other than 8-bits. In particular, these researchers were interested in providing more bit-plane but with smaller changes in their actual values so that embedding in higher bit-planes do not lead to big changes in pixel values and thus has less impact on visibility in comparison to the binary decomposition. In this section, we will review steganography approaches based on different decomposition techniques such as Fibonacci, prime, Lucas, Catalan-Fibonacci, and the natural.

In 2006, Picione et al. proposed the first decomposition technique used for embedding purposes over binary decomposition technique based on representing the grayscale values in terms of set {1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233} of Fibonacci sequence (Picione, et al., 2006). This scheme, referred to as the *Fibonacci* integer decomposition and each grayscale image is represented by 12 bit-planes. The extra bit-planes are called virtual bit-planes. Embedding techniques based on the Fibonacci decomposition can benefit from embedding in higher bit-planes with less stego quality distortion compared to the binary based embedding techniques. However, unlike the binary decomposition, the Fibonacci representation is not unique, i.e., more than one bit-stream can represent the same pixel value (Picione, et al., 2006). For example, the number 5 can be coded in 4-bit Fibonacci number system as 1000 or 0110. The non-uniqueness of Fibonacci representation, however, can be avoided by applying the following theorem:

Zeckendorf theorem *Each positive integer can be represented as the sum of distinct numbers in the sequence of Fibonacci numbers using no two consecutive Fibonacci numbers*.

Accordingly, 0110 is not valid Zeckendorf code. While the uniqueness of representation is solved by the above theorem, Fibonacci based embedding techniques faces another problem in that the act of embedding could result in violating the theorem. The following example illustrates this problem:

Let the cover pixel value $P_i = 7 = (000000001010)_{Fib}$ and the secret bit be 1. Replacing the 1st LSB by the secret bit makes the Fibonacci representation of the corresponding stego pixel value becomes $(00000000101\underline{1})_{Fib}$ and by returning this stream of bits back into decimal the corresponding stego pixel value becomes 8. At the receiver, the Fibonacci representation of the stego pixel value $8 = (00000001000\underline{0})_{Fib}$. Extracting from the stego pixel's LSB, the bit 0 is obtained, which is not the original embedded secret bit.

Hence, cover pixels for which embedding certain secret bits cause a violation of the Zeckendorf theorem cannot be used for embedding and are skipped. To retrieve the secret data, the selected stego pixel value is first decomposed into Fibonacci representation, and then it needs to be checked whether it is a good candidate or not, if it is, then the secret bit is extracted from the agreed bit-plane.

The skipping of bad pixel candidates for embedding result in reduced capacity. Although, some authors have proposed to overcome the capacity limitation by embedding in other than the LSB plane, but the usual binary based embedding techniques, reviewed in the previous section, that embed in higher bit-planes do not face this problem. Note that embedding in the higher bit-planes still has the limitation of payload capacity. In Chapter 4, this problem will be considered, and an innovative solution will be proposed.

Battisti et al. improved the above scheme by using generalized Fibonacci decompositions instead the classical Fibonacci (Battisti, et al., 2006). The most common generalization of Fibonacci is the *p*-number (also called p-code) Fibonacci sequences, where *p* is the distance between the i^{th} element in the Fibonacci sequence and the previous element $(i\text{-}p)^{th}$. Such decomposition schemes provide more places for embedding, by increasing the number of bit-planes and thereby reducing the amount of changes in integer values of consecutive bit-planes. In Battista et al. scheme, the randomly selected pixel value is first decomposed into bit-planes using *p*-number Fibonacci, and then the selected bit-plane is chosen for embedding as long as the Zeckendorf theorem is valid. In their experiments, a comparison between this proposed scheme and classical binary embedding is done in term of quality and capacity. When embedding in bit-planes higher than the LSB, the proposed scheme has less effect on stego-image quality when compared to classical binary embedding, but not in terms of capacity.

Battista et al. scheme has then been modified by adding a key made up of two parameters p and r (Mammi, et al., 2008) to increase the security of the whole system; without their knowledge it is not possible to perform the same decomposition used in the embedding process and to extract the embedded information. The decomposition is based on adding the previous r elements starting from a distance p, and for the sake of uniqueness of representation the following constraints must be satisfied (Mammi, et al., 2008):

1. A valid (p,r) Fibonacci coefficient vector c must contain less than p-1 zeros between two ones.
2. A valid (p,r) Fibonacci coefficient vector c cannot contain more than r consecutive groups, being constituted by one symbol equal to 1 followed by p-1 symbols equal to 0.

Obviously, when p=0, we obtain the classical binary sequence, and when p=1, we obtain the classical Fibonacci sequence. Apart from the security strength, this version of Fibonacci sequence has the same advantages and limitations of the above scheme.

The prime decomposition of integers in terms of sequence {1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43} provides another decomposition based embedding technique, that was proposed by (Dey, et al., 2007). Here, each cover pixel value is decomposed into 15 bit-planes. In this embedding scheme, it is possible to embed secret bits in higher bit-planes, than possible with the binary and Fibonacci schemes, without making big changes to actual pixel values. Again representation is not unique, but a unique prime representation can be obtained by selecting the string with lexicographical highest value, discard all other representations. For example, pixel value 5 can be encoded in 4-

bit prime number system as 1000 or 0110 so we should select 1000 being lexicographically higher than 0110.

The example, below, illustrates that the prime-based scheme has similar problem suffered by the Fibonacci schemes in relation to capacity as a result of having unsuitable pixel values which would violate the uniqueness condition post embedding. Therefore, to retrieve the secret data, the selected stego pixel value is first decomposed into prime representation, and then it needs to be checked whether it is a good candidate or not, if it is, then the secret bit is extracted from the agreed bit-plane (Dey, et al., 2007).

> Let the cover pixel value P_i = 7 = $(000000000001010)_{Pr}$ and the secret bit be 1. Replacing the 1st LSB by the secret bit makes the prime representation of the corresponding stego pixel value becomes $(00000000000101\underline{1})_{Pr}$ and by returning this stream of bits back into decimal the corresponding stego pixel value becomes 8. At the receiver, the prime representation of the stego pixel value 8 = $(00000000001000\underline{0})_{Pr}$. Extracting from the stego pixel's LSB, the bit 0 is obtained, which is not the original embedded secret bit.

The authors of the prime scheme developed a similar scheme using the sequence {1, 2, 3, 4, … , 23} of natural (Dey, et al., 2007). However, this scheme has the same structure, aims, advantages as well as disadvantages in terms of capacity and stego-image quality.

The Catalan-Fibonacci (CF) pixel value decomposition was proposed by (Aroukatos, et al., 2012) to improve the Fibonacci scheme by using a sequence of numbers formed by the union of subset of Fibonacci numbers and subset of Catalan numbers. Catalan numbers are defined in terms of the combinatorial formula for randomly selecting n objects out of 2n ones. For n>0, it is defined as $Cn = \frac{\binom{2n}{n}}{n+1}$, and the set {1, 2, 5, 14, 42, 132} are the first few Catalan numbers. The CF sequence used Aroukatos et al. for pixel representation and embedding is {1, 2, 3, 5, 8, 13, 14, 21, 34, 42, 55, 89, 132, 144, 233} and again to ensure uniqueness among different CF codes the scheme is based on selecting the lexicographically highest code. Any grayscale image has 15 bit-planes CF-decomposition. Unfortunately, this scheme is only different from the above decomposition techniques in the sequence, but otherwise it has the same advantages and disadvantages, discussed earlier in terms of capacity and stego-image quality.

Yet another pixel value decomposition technique has been proposed called Lucas decomposition which decomposes a grayscale image into 12 bit-planes (Alharbi, 2013).

The Lucas sequence is defined using the same Fibonacci recurrence formula but is initiated by the L_0=2 and L_1=1, i.e. the sequence is {2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199}. Unfortunately, this scheme is not different to the others in relation to objectives, structure, advantages and disadvantages in terms of capacity and stego-image quality.

All these different schemes, that share the same objectives and structure, have been based on using sequences that have been of interest in mathematics and number theory, and we noted that they share the same *theoretical* disadvantages in terms of capacity. One might ask whether the choices of mathematically interesting sequences have played any significance in these choices and whether there are any unforeseen advantages that could be exploited in steganography. One could also ask whether decomposition schemes can be exploited for different objectives in steganography.

Faced with these, I first investigated a 16-bit planes image decomposition sequence {1, 2, 4, 6, 8, 10, 12, 14, 16, 20, 22, 24, 26, 28, 30, 32}, which is not known for its mathematical significance beyond a trivial observation that all but the first one are even integers, and defined an embedding technique to embed secrets in high bit-planes (Abdulla, et al., 2014). Unlike all the above schemes, we can embed in all pixels if we only use the LSB, i.e. capacity is the same as the LSBR. Otherwise, it has most of the other disadvantages.

To test where there are unforeseen advantages in the decomposition scheme, in Chapter 6, we shall revisit these schemes and test their performance for various purposes that relate embedding efficiency and security, and we shall design a new decomposition scheme outperform all the existing decomposition schemes in terms of these objectives.

3.1.3 Location\Region based embedding Approaches

In the above two sections, no specific criteria were used to adapt the embedding of the secret bits in the sense that the pixel of the cover to have the next bit embedded into could be anywhere in the image. In fact, in most schemes, the choice of embedding positions within a cover image mainly depends on a PRNG without considering the relationship between the cover image content itself and the size of the secret message. Even when some pixels skipped in the schemes reviewed in Section 3.1.2, this was not done because of the cover image content but the pixel value decomposition.

For improved security in terms of un-detectability, embedding techniques have been developed to hide secret bits in textured regions and regions that could be confused with noise, but perhaps at the expense of limiting payload capacity. However, region-based schemes also include embedding in both edge regions as well as smooth regions with different proportions so that more secret bits are embedded in textured regions. In other words, the regions of edges present more complicated statistical features and are highly dependent on the image content; therefore, it is more difficult to observe changes at the edges than those in smooth regions. Images that have more edge areas can overcome the limitation of capacity. Embedding in smooth/flat regions of the cover images, results in poor visual quality and low security especially for those images with many smooth regions (Luo, et al., 2010).

The Sobel, Prewitt, Canny, and Laplacian are the most popular edge detection techniques that can help in identifying edge pixel regions to be used for embedding, but some researchers use other variants of gradient method for edge detection. The embedding scheme proposed in (Chen, et al., 2010), uses Canny edge detector, and the authors argue that this yields increased capacity because more edge pixels are detected compared to other edge detectors. Chen et al. scheme uses higher bit-planes for improved capacity, but their claim on security and robustness against statistical steganalysis tools is not substantiated by experimental evidence. However, edge based embedding techniques have a problem with determining the same edge area by the receiver because the act of embedding in an edge area could change the original edge pixels into a non-edge pixels. In other words, a pixel that is detected as an edge point before embedding the secret bit may not be detected as an edge point after message embedding. Thus, some parts of a secret message may be lost. In the literature, different approaches have been suggested to dealing with this problem, but in any case when an edge pixel is selected for embedding a secret bit one must make sure that the act of embedding will not make a non-edge pixel. It has been suggested that embedding more secrets in sharp edges than in faint/blur edges (Iranpour, 2013).

In 2007, (Singh, et al., 2007) proposed an embedding technique where each pixel is labelled as an edge pixel if the Laplacian operator applied to its 3x3 neighbours is larger than a fixed threshold θ. The scheme then embeds 1 bit in each edge pixel not by LSB replacement but using a probabilistic model to guarantee that the pixel remains an edge pixel after embedding. The authors calculate the maximum embedding capacity to be relatively low ($1/9 \approx 11.1\%$).

Hempstalk proposed the *FilterFirst* hiding scheme which aims to overcome the problem of extracting the secret bit from the correct edge pixels by first setting the LSB of every cover image pixel to 0, then extracting edges pixels using Sobel (or any edge detector) and use LSBR for embedding in the edge pixels (Hempstalk, 2006). They also extended this scheme whereby the edge pixels are determined after vanishing the first 2 or more LSB's. As a result, FilterFirst can guarantee to retrieve the secret information from the same edge pixels used for hiding, because the bit-planes used for filtering are not changed by the hiding process. Although this technique can embed most secret data along sharper edges and can achieve more visually imperceptible stego-images, but again it has lower capacity than LSBR.

Geetha and Giriprakash proposed an embedding technique that adopts a Variable Embedding Ratio (VER) approach to embed secrets with higher ratio in edge regions with the aim of improving capacity, increasing the secrecy and un-detectability of the embedded message (Geetha & Giriprakash, 2012). For increased capacity, the Canny edge-detector is repeated three times to detect more edge regions and embeds 4 secret bits in edge pixels and two secret bits in non-edge pixels. The constant VER ratio of 4:2 does not distinguish between sharp edges and not so sharp ones. However, this scheme assumes the receiver has the original cover to recover secrets that are lost during embedding.

The use of VER goes back to 2003, when Wu and Tsai proposed the pixel value differencing (PVD) steganography scheme that embeds more secret bits in edge areas than in smooth areas (Wu & Tsai, 2003). The authors claim that their technique provides an easy way to produce a more secure result than those yielded by simple LSB replacement methods. Instead of using edge detectors, PVD first partitions the cover image into non-overlapping blocks of two consecutive pixels, p_i and p_{i+1}, and process these pairs in a zigzag manner. For each block, the absolute difference $d = |\, p_{i+1} - p_i|$ is calculated, with $d \in [0,255]$. Split the interval of d values into a number of contiguous ranges, R_i $(i = 0,1, \dots n)$. A block with d close to 0 is considered to be an extremely smooth block, whereas a block with d close to 255 is considered as a sharply edged block. The number of bits to be embedded in each block varies and depends by the range that d belongs to. Less secret bits are embed in blocks that have a smaller index (i.e. smooth blocks) and more secret bits are embed in blocks that have higher index (i.e. edge blocks). Finally, the difference value is replaced by a new value to embed the value of a sub-stream of the secret bits, the sub-stream of secret message converts to decimal value then replaced with the value of d. To illustrate this rather complicated procedure that depends on certain equations, we present an example and readers interested in the details are referred to (Wu & Tsai, 2003)

> Assume the two-pixel block p_i and p_{i+1} are 50 and 65. The difference $d = |\, p_{i+1} - p_i|$ = 15, which is in the range of 8 through 23. Based on this range, the d can be used to embed 4 secret bits. If the 4 secret bits is 1010, then add its decimal value to the lower bound value of the range which becomes 18. The algorithm uses a special equation to change the two pixel values to the pixel values become 48 and 66. The receiver calculate the difference d =18 and uses another equation to recover the decimal 10 of the secret.

The method is designed in such a way that the modification is never out of the range interval. The secret bits can be retrieved at the receiver side by first segmenting the stego-image into non-overlapping blocks of two consecutive pixels, and then calculating the absolute difference between the two pixels in the block. The value of d, and its range, determines the number of secret bits to be extracted. The PVD steganography technique has higher capacity but lower stego-image quality comparing to the LSBR (cover pixels' value might change by more than 1) and has poor resistance to some statistical steganalysis tools (Luo, et al., 2010). The most important drawback of PVD based approaches is that only horizontal differences, i.e. vertical edges are used for embedding, while there are also many horizontal edges in the cover images which are not used in this approach, reported in (Iranpour, 2013).

Another elaborate steganography technique is presented by (Chang & Tseng, 2004) that associate with each pixel x, the difference d between value at x and the average value of its upper and left neighbours. The first row and the first column of the cover image are excluded for data embedding. Larger d indicates sharper edge pixel. The scheme embeds more bits in pixels whose d values are higher using a different formula than that used in the PVD scheme, to decide the number of bits to be embedded. The number of bits, say n, which can be embedded in the pixel x is calculated by $n = log_2|d|$ if $|d| > 1$, otherwise only one bit is embedded. This scheme has a higher capacity than LSBR but less capacity than the PVD. In addition, the stego quality is lower than that for LSBR, since cover pixels value might change by more than 1.

(Luo, et al., 2010) use a similar idea of thresholding neighbouring pixel value differences to select edge pixels. However, the threshold is dependent on the size of the secret and cover image content. Prior to determining the neighbouring pixel value differences, the cover image is divided into 4 blocks, and each is rotated by a random degree selected from the set of {0, 90, 180, 270}. The transformed cover image is divided into non-overlapping blocks of two pixels, and the absolute differences between adjacent pixels, are thresholded, to select the embedding regions. The secret bits are embedded in the pixels of the selected region/edges areas using LSB matching revisited (LSBMR) scheme. This scheme is adaptive: For lower embedding rate, only sharper edge regions are used for embedding, and as embedding rate increases more edge regions can be released adaptively for data hiding by adjusting just a few cover image content-based parameters. These parameters and the angles of rotation becomes a side information (i.e. a key) that need to be transmitted (by hiding) to the receiver. Using the side information, the receiver identifies the selected regions, and the secret bits are retrieved using the extraction process of LSBMR embedding algorithm. The embedding algorithm has been shown to be robust against statistical steganalysis techniques such as RS compared to LSBR based and PVD. However, the stego quality is more affected comparing to LSBR, since in some cases cover pixels value may change by more than 1.

Huang and Ouyang state that beside of smooth areas, some edge areas are also sensitive to be used for hiding data (Huang & Ouyang, 2010). Their algorithm avoids embedding in pixels belonging to *fragile regions* in a cover image (pixels for which embedding one bit results in changes to its differences with many of its neighbours). Regions, such as smooth or frequent figure patterns, a region with regular changes in pixel values are called fragile region. The algorithm extends the use of absolute difference to all the 8 neighbour of a candidate pixel. It counts the number of surrounding pixels for which differences with centre exceeds a given threshold T, and if the count is greater than a constant C, then a secret bit can be embedded. In other words, this algorithm tries to maintain local texture in the stego-image, and thereby it is secure because of less chance of detectability. After region selection, Huang and Ouyang use LSBMR for embedding in the non-fragile pixels. The receiver detects non-fragile pixels in the same way and extracts the secret from these pixels. The scheme is more robust against the steganalysis technique of (Ker, 2005) compared to usual LSBMR embedding technique, but it has a limitation of payload capacity. Moreover the thresholds T and C must be exchanged.

In 2013, Iranpour modified the FilterFirst scheme, by using a special method to determine the sharpness of edges that are extracted using the Sobel edge detector after they ignore the first p bit-planes (Iranpour, 2013). The other difference with FilterFirst is that the embed up to p-bits in the first p bit-planes depending on the level of sharpness of the edge pixels so that the number of bits embedding in the sharper edges should be more than the ones in the weaker edges. The sharpness threshold *T* depends on the length of the secret message, and embedding is first done in the sharper edges before embedding in the weaker edges and the smooth regions. They claim that this algorithm has significantly enhanced the security against RS steganalysis, increased capacity, and has almost the same stego-image quality as the LSBR.

Finally, in the recent years, Fridrich and her group has developed a strategy to constrain secret embedding to noisy or textured regions (determined by appropriately defined distortion functions) and avoiding smooth and clean edge regions (Holub & Fridrich, 2012) (Holub, et al., 2014). The idea is based on the fact that complex texture or noisy areas are difficult to model directly, but their distortion can be approximated by certain functions that relate a pixel to its surrounding region. This approach improves resistance to steganalysis techniques that use rich models such as (Fridrich & Kodovsky, 2012). In their latest approach, they proposed a steganography technique based on a defined distortion function called UNIWARD, which stands for universal wavelet relative distortion (Holub, et al., 2014), which is similar to their previous proposed approach in (Holub & Fridrich, 2012) but it is suitable for embedding in an arbitrary domain, namely spatial and frequency domain, and it is an extended version of (Holub & Fridrich, 2013). This proposed distortion function is defined as the sum of the relative changes of all wavelet coefficients with respect to the cover image. In other words, it is a sum of relative changes between the stego and cover images represented in the wavelet domain (Holub, et al., 2014). The UNIWARD function depends on a bank of wavelet multiple directional high-pass filters called filter bank to obtain the so called directional residuals, which are related to the predictability of the pixel in a certain direction. By measuring the impact of embedding on every directional residual, the predictable in at least one direction is considered as smooth or clean edge pixel, while unpredictable in every direction that is considered as textured or noisy pixel. Next, the Syndrome Trellis Codes embedding technique (Filler, et al., 2011) is used to embed the secret bits after textured, or noisy pixels are identified.

Steganography schemes based on designing distortion functions to identify the texture and noisy regions have a property of increasing the security for the steganography systems, but limit the capacity when the cover image contains high ratio of smooth regions. Furthermore, currently, all the steganography techniques based on defined distortion functions proposed by Fridrich and her group are un-detectable only when the amount of embedded payload not exceeds 0.5 bpp. To overcome these limitations, namely capacity and detectability, it would be ideal to design steganography techniques that produces fewer changes and has high embedding efficiency without the need to exclude smooth regions or clean edge pixels. Ker et al. highlighted this problem, i.e. design efficient embedding schemes, as an important open problem to address in future research (Ker, et al., 2013). In the next section, image-based steganography approaches that concern on improving embedding efficiency are reviewed.

3.1.4 High Embedding Efficiency Approaches

In image-based steganography, embedding efficiency is defined by (Fridrich & Soukal, 2006) as the ratio of number of cover image pixels whose value change as a result of embedding to the size of a secret message. The concept of embedding efficiency has been first introduced by (Crandall, 1998), and was first adopted by (Westfeld, 2001) for embedding in DCT domain. It has since been accepted as an important attribute of steganography schemes that directly influencing their security, because smaller number of embedding changes is less likely to disrupt statistic properties of the cover image (Fridrich, et al., 2007). Thus, schemes that employ high embedding efficiency generally have better security, and they produce stego-images with minimal distortion while maintaining payload capacity.

A formal definition of steganography security was given by (Cachin, 1998) in terms of detectability of the hidden data in a stego-image, and the concept of embedding efficiency is an essential indicator of steganography security. The detectability of a data hidden in a stego-image is influenced by many factors, such as the choice of the cover object, the selection rule used to identify individual elements of the cover that could be modified during embedding, the type of the embedding operation that modifies the cover elements, and the number of embedding changes relative to the secret message length. Assuming two embedding techniques share the same source of cover object, the same selection rule and embedding operation, the one that introduces fewer embedding changes will be less detectable as it decreases the chance that any statistic used by the warden will be sufficiently disturbed to mount a successful steganalysis attack (Fridrich, et al., 2007).

For the LSBR or LSBM schemes, the probability of pixel change is 0.5, i.e. on average, such algorithms add 0.5*p* of the noise in the cover image pixels, where *p* is the embedding rate in bits/pixel. In other words, the embedding efficiency of LSBR or

LSBM embedding based techniques is 2 (Westfeld, 2001). Ker et al. highlighted the open problems in steganography and steganalysis in future research, and design efficient embedding schemes is addressed as an important problem (Ker, et al., 2013). So far, steganography approaches that focused on designing a high embedding efficiency and minimising the noise due to message embedding are very limit. Therefore, achieving high embedding efficiency is a fundamental objective that we aspire to achieve in this thesis.

The *matrix encoding* technique proposed by (Crandall, 1998) was probably the first attempt to improve embedding efficiency. In matrix encoding, to embed k bits secret message, it needs to employ $2^k - 1$ pixels in the cover image and at most one pixel is changed by one from each group. The following example illustrates how the matrix encoding algorithm hides 2 bits secret message m_1 and m_2 into 3 cover pixels (Note that only one of three cover pixels is meant to change). Let $a=[a_1\ a_2\ a_3]$ be the LSB of the 3 cover pixels. Embedding works by changing one of the values as follows:

$$m_1 = a_1 \oplus a_3,\ m_2 = a_2 \oplus a_3 \quad \Rightarrow \text{ change nothing}$$

$$m_1 \neq a_1 \oplus a_3,\ m_2 = a_2 \oplus a_3 \quad \Rightarrow \text{ change } a_1$$

$$m_1 = a_1 \oplus a_3,\ m_2 \neq a_2 \oplus a_3 \quad \Rightarrow \text{ change } a_2$$

$$m_1 \neq a_1 \oplus a_3,\ m_2 \neq a_2 \oplus a_3 \quad \Rightarrow \text{ change } a_3$$

In all four cases, we do not change more than one bit. The most important advantage of using matrix encoding is that it decreases the number of necessary pixels which must be changed, 25% are changed when k =2, while it limits the payload capacity, 67% on average. In general, embedding k bits using this method, increases embedding efficiency to 2^k but limits the capacity to $k/(2^k - 1)$. Thus, such kinds of embedding techniques are not useful for those applications that require full capacity, i.e., embedding one secret bit per cover pixel.

To further improve efficiency while maintaining payload capacity, Mielikainen proposed a variant of LSBM, called LSB matching revisited (LSBMR), which employs the binary function in equation (3.1) to embed two secret bits, namely m_i and m_{i+1}, in a pair of pixels x_i and x_{i+1}.

$$f(x_i, x_{i+1}) = \mathrm{LSB}\left(\left\lfloor \frac{x_i}{2} \right\rfloor + x_{i+1}\right) \qquad (3.\ 1)$$

This results in two stego pixels, y_i and y_{i+1}, where at most one is different from the cover pair using the procedure in Figure 3.1 (Mielikainen, 2006). After embedding message, the LSB of the ith stego pixel y_i represents the ith secret bit m_i, and the LSB of the result of the binary function represents the (i+1)th secret bit m_{i+1}. Theoretically, this function reduces the probability of changing pixel values from 0.5 to 0.375, i.e. the embedding efficiency has been increased to 2.66 compared to LSBR and LSBM. However, these improvements come at the expense of limited payload capacity because LSBMR algorithm cannot be performed on saturated pixels, i.e. pixels that have either a minimal or maximal allowable value (0 or 255). But this limitation is negligible compared to the matrix encoding embedding technique. Moreover, LSBMR has better resistance to steganalysis techniques comparing to LSBM embedding technique. Furthermore, LSBMR does not have LSBR style imbalance. LSBMR also has a property of visual imperceptibility, since the cover pixel's value should change by one. Thus, it is difficult to notice by the naked eye.

input: a pair of cover image pixels x_i, x_{i+1}
       two message bits m_i, m_{i+1}
output: a pair of stego image pixels y_i, y_{i+1}

if $m_i = \mathrm{LSB}(x_i)$
  if $m_{i+1} \neq f(x_i, x_{i+1})$
    $y_{i+1} = x_{i+1} \pm 1$
  else
    $y_{i+1} = x_{i+1}$
  end
   $y_i = x_i$
else
  if $m_{i+1} = f(x_i - 1, x_{i+1})$
    $y_i = x_i - 1$
  else
    $y_i = x_i + 1$
  end
   $y_{i+1} = x_{i+1}$
end

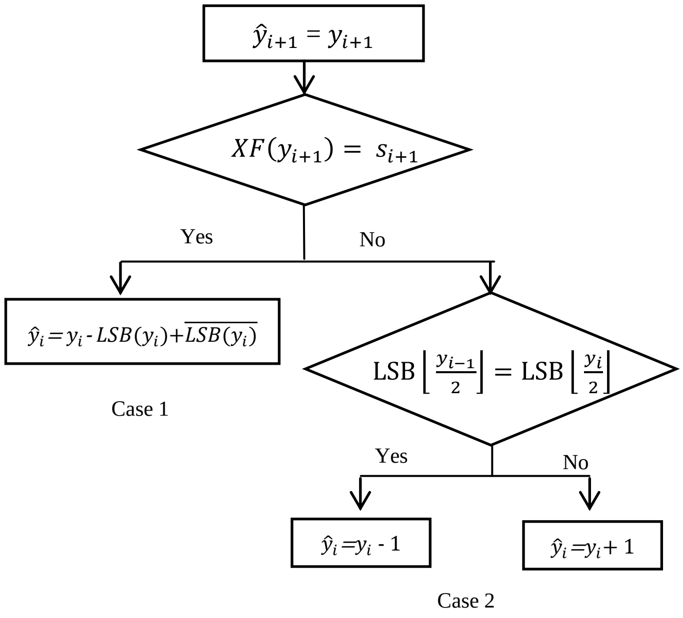
Figure 3-1: Pseudo-Code of the LSBMR embedding technique.

In 2009, Chan proposed another embedding scheme that aims to further reduce the number of modified cover pixels, and like above scheme uses a binary function defined consecutive pixels but it attempts to embed a number of secret bits by successive application of the function on a number of consecutive pixels until the output of the function is different to the secret bit aligned with the last pixel (Chan, 2009). Figure 3-2 illustrates this method. The function is defined in equation (3.2) by XORing the 2nd bit of the previous pixel with the LSB of the current pixel, if the result matches the next secret bit then continue to the next pixel without making any change otherwise either add 1 or -1 according to whether the outcome of the function applied to the next pixel is a match or not.

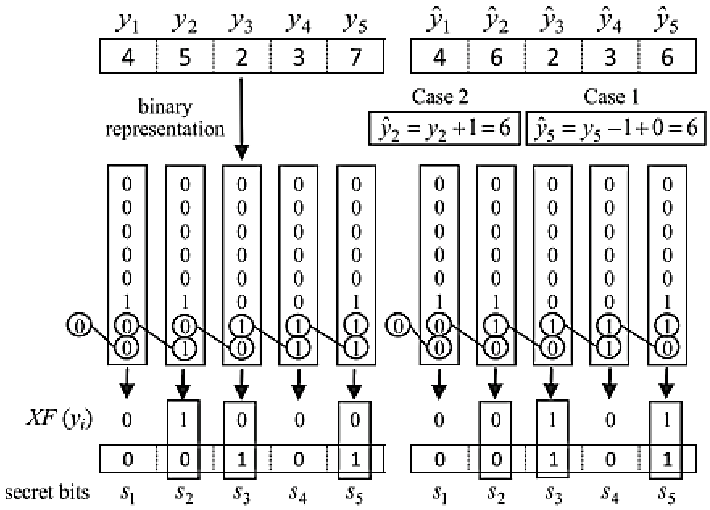
$$XF(y_i) = \mathrm{LSB}(\left\lfloor \frac{y_{i-1}}{2} \right\rfloor) \oplus \mathrm{LSB}(y_i) \qquad (3.\ 2)$$

where y_i represents the pixel value at the position i, and $\oplus$ is the exclusive OR operator.

This proposed approach is not only superior to Mielikainen's approach in terms of higher embedding efficiency but also it has higher capacity since every cover pixel can be used for embedding. Figure 3-2 presents the decision tree of the data embedding procedure and an example of data embedding. In this figure, i is the first position with different values, y_i is the original pixel at position i, s_i is the secret bit, $\hat{y_i}$ is the modified pixel value at position i , and the symbol $\overline{LSB\ (y_i)}$ indicates the complement of the least significant bit of y_i. In the data extraction procedure, the secret bits can be obtained by computing $XF(y_i)$ where y_i represents the pixel value at position i of the stego-image.



(a)



(b)

Figure 3-2: Chan's approach (a) The decision tree of the data embedding procedure (b) An example.

The experimental results reported in (Chan, 2009) demonstrated that this scheme achieves higher embedding efficiency than LSBMR. They report that embedding a secret Lenna image of size 256 x 128 (i.e. 262144 bits) in a Lenna cover image of size 512 x 512, only 87374 cover pixels are changed; while the LSBMR results in 98176 changed pixels. This improvement may be dependent on the secret and the cover images. Again, when a cover pixels change, its values either increase or decrease by 1, and hence it does not have the asymmetry problem as LSBR has, i.e. has better resistance to steganalysis techniques comparing to LSBR based embedding techniques.

In 2013, Iranpour and Farokhian generalise the last two schemes and increases the embedding efficiency by using three binary functions to embed three secret bits in three cover pixels in a similar way to the LSBMR schemes (Iranpour & Farokhian, 2013). Note that the secret bits themselves are not directly embedded/extracted into/from the cover/stego pixels' LSB, but they are embedding or extracting from the results of the following three defined functions:

$$f_1(\mathrm{x,y,z}) = \mathrm{LSB}(\left\lfloor\frac{x}{2}\right\rfloor + \left\lfloor\frac{y}{2}\right\rfloor + \left\lfloor\frac{z}{2}\right\rfloor) \qquad (3.\ 3)$$

$$f_2(\mathrm{x,y,z}) = \mathrm{LSB}(\left\lceil\frac{x}{2}\right\rceil + \left\lceil\frac{y}{2}\right\rceil + \left\lfloor\frac{z}{2}\right\rfloor) \qquad (3.\ 4)$$

$$f_3(\mathrm{x,y,z}) = \mathrm{LSB}(\left\lfloor\frac{x}{2}\right\rfloor + \left\lceil\frac{y}{2}\right\rceil + \left\lceil\frac{z}{2}\right\rceil) \qquad (3.\ 5)$$

For embedding three secret bits into three cover pixels, eight cases may occur with any combination of three secret bits with the results of three defined functions. When there is no match, the scheme either adds 1 or -1 based on equations (3.6), and (3.7). Figure 3-3 presents the eight occurred cases when three secret bits (m_i, m_{i+1}, and m_{i+2}) are embedded in three cover pixels (x_i, x_{i+1}, and x_{i+2}) based on the following two defined rules for modifying the value of a pixel:

$$r_1(\mathrm{t}) = \begin{cases} t+1 \ \ if\ t\ is\ even \\ t-1 \ \ if\ t\ is\ odd \end{cases} \qquad (3.\ 6)$$

$$r_2(\mathrm{t}) = \begin{cases} t-1 \ \ if\ t\ is\ even \\ t+1 \ \ if\ t\ is\ odd \end{cases} \qquad (3.\ 7)$$

$m_i == f_1(x_i, x_{i+1}, x_{i+2})$	$m_{i+1} == f_2(x_i, x_{i+1}, x_{i+2})$	$m_{i+2} == f_3(x_i, x_{i+1}, x_{i+2})$	**Action**
T	T	T	nothing
T	T	F	$x_{i+2} = r_1(x_{i+2})$
T	F	T	$x_{i+1} = r_1(x_{i+1})$
T	F	F	$x_i = r_2(x_i)$
F	T	T	$x_i = r_1(x_i)$
F	T	F	$x_{i+1} = r_2(x_{i+1})$
F	F	T	$x_{i+2} = r_2(x_{i+2})$
F	F	F	$x_i = r_1(x_i)$, $x_{i+1} = r_1(x_{i+1})$, $x_{i+2} = r_1(x_{i+2})$

Figure 3-3: Illustration of the (Iranpour & Farokhian, 2013) for the eight cases.

From Figure 3-3, you can notice that except for one case, in all other cases at most one pixel out of three pixels is modified, either increased or decreased by one. Although all three cover pixels should be modified in only one case, it is not sever drawback of this proposed technique, because the probability of this happening is estimated experimentally to be <4.2% (Iranpour & Farokhian, 2013). Furthermore, theoretically, in this embedding technique the probability of changing cover pixel value is 0.375, i.e. the embedding efficiency is 2.66, which is the same as LSBMR embedding technique (Mielikainen, 2006). However, the authors demonstrate higher efficiency in practice. Thus, this approach sets a new state of the art in terms of embedding efficiency. Additionally, this embedding technique has good visual imperceptibility. Finally, this proposed embedding technique does not have LSBR style imbalance, namely, asymmetry problem. The only limitation of this embedding technique is capacity, since pixels that have either a minimal or maximal allowable value cannot be used for message embedding. Moreover, this embedding technique allows the same amount of embedding as LSBMR (Mielikainen, 2006) but with fewer changes to the cover image pixels.

3.2 Image-based Steganalysis Approaches

Whilst steganographers aim to design steganography techniques; steganalysers attempt to defeat the goal of steganography by detecting the presence of a hidden message, but not necessarily to retrieve the secret. There are number of existing image-based steganalysis techniques to determine the presence/absence of a hidden message and estimate the size of the embedded secret message. In Chapter 2, we gave a brief description of the classifications of the steganalysis tools into targeted and universal, and statistical and structural. For robustness in terms of resisting steganalysis attacks, un-detectability is the main success criteria (i.e. stego-images should be statistically undistinguishable from cover images) (Fridrich & Goljan, 2004). In this section, we review some of the most common steganalysis techniques that we used to test and examine the un-detectability/security of our steganography schemes.

1. Pairs of Value (PoV)

The PoV, also known as *Chi-Square* steganalysis, uses the statistical Chi-square test (Westfeld & Pfitzmann, 2000) to test if the LSB plane of a suspect image is statistically different from that of natural images in terms of changes in the pairs of consecutive grayscale values (i.e. $0 \leftrightarrow 1$, $2 \leftrightarrow 3$, …, $254 \leftrightarrow 255$). This is based on the fact that LSBR schemes change the distributions of the pairs. Note that in LSBR schemes, pixel value 2 will never become 1 or vice versa after embedding the secret bit. Flipping the pair of values $n_{2i} \leftrightarrow n_{2i+1}$ (i = 0, 1, …, 127), as a result of embedding 1 bit may result in many pairs of pixels that have PoV (n_{2i}, n_{2i+1}) become of equal values and hence change the frequency distributions of these values. As the number of pixels for which LSB has been replaced increases, the frequencies of both values of each PoV tend to become equal. But the sum of them (n_{2i},n_{2i+1}) stays the same. Thus, the arithmetic mean of sum, as in equation (3. 8), can be taken as the theoretically expected frequency in the Chi-square test for the frequency of occurrence of n_{2i} or n_{2i+1}. Then the Chi-square statistic may be given as in equation (3. 9) and the probability of embedded payload (p) can be calculated by equation (3. 10).

$$y_i^* = \frac{n_{2i} + n_{2i+1}}{2} \qquad (3.\ 8)$$

$$\chi_{k-1}^2 = \sum_{i=1}^{k} \frac{(y_i - y_i^*)^2}{y_i^*} \qquad (3.\ 9)$$

Where $y_i = n_{2i}$, and $k - 1$ is a degree of freedom.

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}} \Gamma\left(\frac{k-1}{2}\right)} \int_0^{\chi_{k-1}^2} e^{-\frac{x}{2}} x^{\frac{k-1}{2} - 1} dx \qquad (3.\ 10)$$

Where, p is the probability of embedding the secret message, and Γ is the Euler Gamma function.

Testing any suspect image against the PoV tool generates a plot from which one can determine an estimate of the embedded secret. For example, if the PoV steganalysis output a plot similar to that in Figure 3-4 then the image is considered as a natural image. But an output plot similar to those in Figure 3-5 and Figure 3-6 indicate that with a high probability the image has been embedded with 50% and 100% capacity respectively.

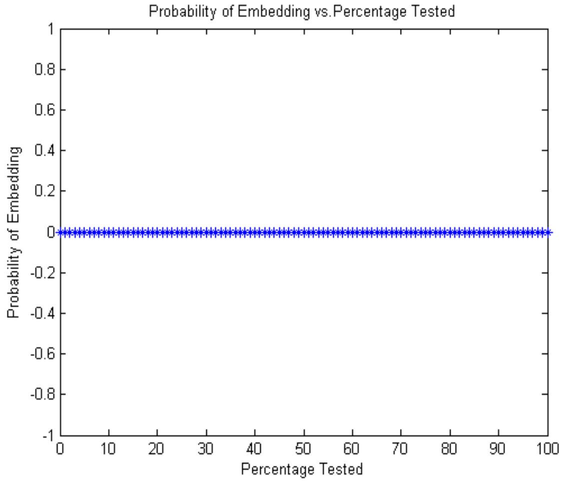


Figure 3-4: Example of PoV plot for cover image Lenna (without embedding).

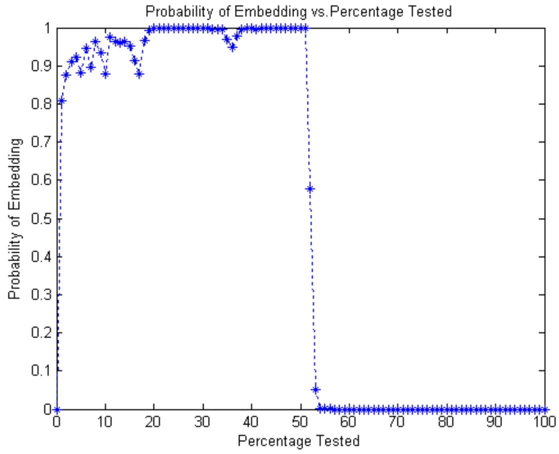


Figure 3-5: Example of PoV plot for stego image Lenna (50% embedding).

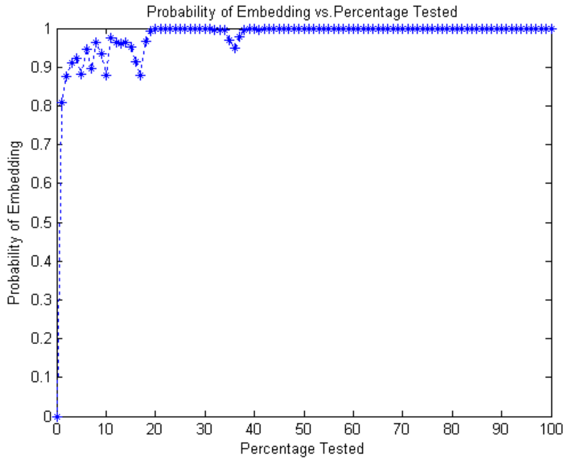


Figure 3-6: Example of PoV plot for stego image Lenna (100% embedding).

To test robustness of any embedding scheme against PoV, most researchers use a small number of stego-images, since each tested image it has own plot.

2. Regular and Singular groups (RS)

The RS steganalysis is a structural targeted tool that differs from the PoV in that it does not rely on the statistical analysis of the LSB plane. On its own, the LSB plane is a random variable but has no easily recognizable structure and its statistical parameters vary for different type of images, and cannot relied on to detect distortion of the LSB plane. However, the LSB plane has known correlation with other bit-planes, which are exploited by the RS steganalysis technique for the detection of LSB embedding in grayscale images (Fridrich, et al., 2001).

The RS tool is based on the analysis of the relative frequency between the so called Regular groups (R) and Singular groups (S) of image pixels depending upon some properties. These groups are defined in terms of the effect of random flipping the LSB values using two pixel functions: F_1 changes a pixel value so that 0 ↔ 1, 2 ↔ 3, 4 ↔ 5,..., 254 ↔ 255 and F_{-1} changes a pixel value so that $-1 \leftrightarrow$ 0, 1 ↔ 2, 3 ↔ 4, ..., 255 ↔ 256. RM is the ratio of the groups of pixels for which the total number fluctuations increases when F_1 is applied to the groups with the mask M = [0 1 1 0], and SM is the ratio of groups for which the total of fluctuations decreases when $\mathrm{F}_1$is applied to the blocks with the mask M. Similarly, RM- and SM- are defined but with F_{-1}, instead of F_1. Fridrich et al. found that the RS ratio of a natural image should satisfy the rule: RM ≅ RM- and SM ≅ SM- through a large number of experiments. If LSB of the cover pixel is changed, the difference between RM and RM- and the difference between SM and SM- increases and hence the above rule is violated; therefore, one could conclude that the tested image carries a secret message. Table 3-1 and Figure 3-7 respectively, illustrate an example of RS results and diagram of the cover and its stego-image carrying different payload ratios, using LSBR technique, for Lenna image. We note that when the payload capacity p = 0 %, i.e. cover without an embedded message, the value of RM is close to RM-, and the value of SM is close to SM-. By increasing the rate of the embedded payload, the difference between RM and RM- is increased, and also the difference between SM and SM- is increased.

Table 3-1: RS steganalysis for Lenna image.

p	0 %	25 %	50 %	75 %	100 %
RM	40.52	38.38	36.28	34.37	32.39
SM	24.09	25.89	27.87	29.74	31.71
RM-	40.06	42.13	44.24	46.29	48.29
SM-	24.42	22.86	21.27	20.04	18.86

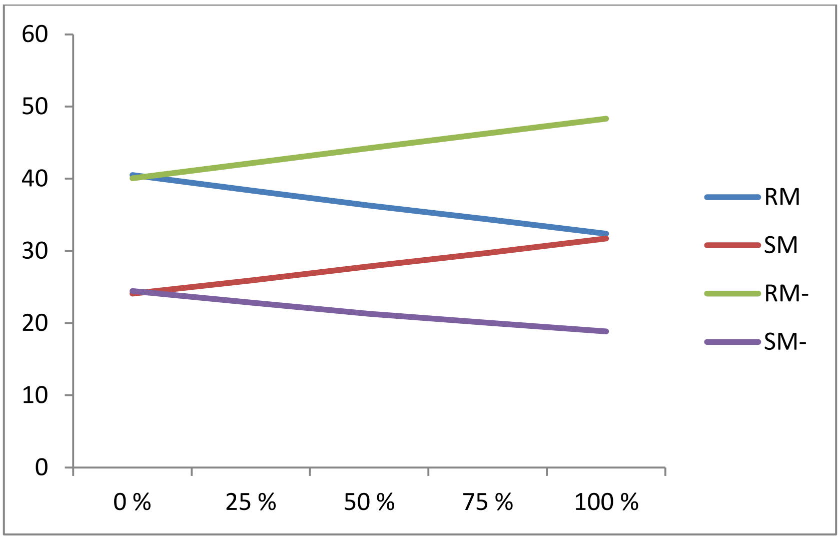


Figure 3-7: RS diagram for Lenna image. The x- axis is a ratio of flipped LSBs; the y-axis is the (RM, RM-, SM, SM-).

3. Difference Image Histogram (DIH)

This statistical steganalysis technique could not only detect the existence of the hidden messages in the cover image but also estimate the amount of hidden messages with extreme precision (Zhang & Ping, 2003). Difference image histogram (DIH) is defined as follows: For an image I define the difference image D as the horizontal gradient image:

$$D(i,j) = I(i,j) - I(i,j+1) \qquad (3.\ 11)$$

The DIH is defined as the histogram of the difference image D. This technique works first by flipping all the bits in the LSB plane of the tested image, and second by setting zeros value to all the bits in the LSB plane of the tested image, and then doing a difference comparison based on DIH with the original image. Zhang and Ping found that there exist the difference between the DIH for natural images and images obtained after flipping the LSB plane (Zhang & Ping, 2003). Translation coefficients between difference image histograms are defined as a measure of the weak correlation between the LSB plane and the remaining bit-planes (Zhang & Ping, 2003). These translation coefficients are relationships between DIH of the original image and images obtained after flipping the LSB plane. This correlation can be used to construct classifiers that discriminate between cover and stego-image. They claim that translation coefficients for natural images there exists a weak correlation between the LSB plane and the remained bit-planes. As more and more secret bits are embedded, such correlation becomes weaker and weaker, and finally the LSB plane becomes independent of remaining bit-planes.

When an image is submitted to this steganalysis technique, a real number between 0 to 1 is the output which should indicate the probability of having a secret hidden with the output ratio, 0 means the tested image is a cover and 1 it means the tested image is considered as stego with embedding rate = 100%. DIH values close to 0 indicates less suspicion of a stego and/or a very short secret is been estimated.

4. Weighted Stego (WS)

It is another targeted steganalysis technique proposed by (Fridrich & Goljan, 2004) and aims to estimate the secret message length embedded in a digital image using LSBR. This is done by defining an optimisation problem obtained by considering all possible pixel change ratios. WS works as follows:

Let $X = \{x_i\}_{i=1}^{n}$ be a column vector of integers in the range [0, 255] representing a grayscale cover image with $n = M_x \times N_x$ pixels. Let $\overline{x_i}$ be the value of x_i after flipping its LSB, i.e.

$$\overline{x_i} = x_i + 1 - 2(x_i \bmod 2) \qquad (3.\ 12)$$

Let $S = \{s_i\}$ denote the stego image after embedding *qn* bits, $0 \le q \le 1$, in *qn* pixels randomly selected from the cover image X and for $0 \le p \le 1$ define $S^p = \{s_i^{(p)}\}$ as the weighted stego image:

$$s_i^{(p)} = s_i + (\overline{s_i} - s_i)\frac{p}{2} \qquad (3.\ 13)$$

The $S^{(q)}$ is the closest weighted stego-image to X in the least square approximation among all weighted stego-images $S^{(p)}$. Here, only the stego-image is available and therefore, we need to use s_i instead of x_i and $\overline{s_i}$ instead of $\overline{x_i}$. Therefore, one can estimate the secret message length as the solution of the above optimisation problem. The least square estimation formula is derived as follows:

$$\bar{q} = -\frac{2}{n}\sum_{i=1}^{n}[s_i - F(N(s_i))]\ (\overline{s_i} - s_i) \qquad (3.\ 14)$$

Where, $(N(s_i))$ is the estimated pixel value of stego-image from the neighbourhood and the filter:

$$F\big(N(s_i)\big) = \frac{1}{4}(s_{i+1,j} + s_{i-1,j} + s_{i,j+1} + s_{i,j-1}) \qquad (3.\ 15)$$

When an image is submitted to this steganalysis technique, then again a real number between 0 to 1 is output which should indicate the ratio of having a secret message length proportion to the cover size, 0 it means the tested image does not carry a secret message and 1 it means the tested image is considered as stego with embedding rate = 100 %. Moreover, the WS output value close to 0 indicates the image is less suspecting to be a stego and/or the secret message length is estimated as a very short message.

5. Revisiting Weighted Stego (RWS)

This steganalysis technique is proposed by (Ker & Bohme, 2008) as an improvement version of the WS tool with improved accuracy. In this steganalysis technique, there is a consideration in which for the embedding rate 100%, there should 50% of the cover pixels' LSB are flipped. In other words, the proportion $M/2N$ of the cover pixels are flipped when embedding a payload of length M , where N is the number of cover pixels. They modified the WS by upgrading the method's three components: 1) cover pixel prediction, by using different filter from used in WS, 2) least square weighting, and 3) bias correction, either the new moderated weights should be used or no weights needed at all depending on the smoothness nature of the tested image. Based on their analysis, the new moderate weight detector is more accurate for the images that are flat with less noise and texture, and no weight (un-weighted detector) needed for the images that contain more noise or texture (Ker & Bohme, 2008). In 2013, Fridrich and Kodovsky demonstrated the benefits of using RWS for many applications. Moreover, the RWS is still be the better choice since it does not require any training phase and keeps the high accuracy (Kodovsky & Fridrich, 2013).

When an image is submitted to RWS steganalysis, the output is a real number between 0 to 0.5 indicating the ratio of flipped cover pixels, 0 it means the tested image does not carry a secret message and 0.5 it means the tested image is considered as stego with embedding rate = 100%.

6. LSB matching Steganalysis (LSBMS)

The steganalysis techniques discussed earlier are designed to attack LSBR based embedding techniques, but are unable to detect the LSBM techniques. The most reliable and well-known steganalysis technique that is designed to defeat LSBM was proposed by (Ker, 2005) called LSB matching steganalysis. This steganalysis technique strategy is based on the known information about the energy distribution $H[k]$ of the histogram characteristic function (HCF) which is the discrete Fourier transform (DFT) of the histogram of any tested image. The histogram characteristic function centre of mass (HCF-COM), denoted by $C(H[k]$, which is calculated by equation (3. **16**), gives a general information about the HCF, can be exploited to capture the effect of the additive noise.

$$C(H[k]) = \frac{\sum_{i=0}^{n} i\ |H[i]|}{\sum_{i=0}^{n}\ |H[i]|} \qquad (3.\ 16)$$

Where, n = N/2 to avoid the redundant parts of the DFT, and N = 256 (number of intensity values for 8-bits grayscale image from 0 to 255), and k=0,.., N/2.

The $C(H[k]$ can successfully detect the hiding schemes that act as additive noise. Ker's experimental results showed that the HCF-COM based steganalysis method performed quite well for colour images, but it turned out to have very poor performance for grayscale images due to the high variability of the cover images' HCF. Therefore, a down-sampled image by a factor of two in both dimensions and processed by a straightforward averaging filter was employed to calibrate the HCF-COM of the full-sized image. In view of the variation between the magnitudes of the HCF-COM of a tested image, denoted by $C(H[k])$, and that of the down-sampled image, denoted by $C(H'[k])$, the ratio $C(H[k] / C(H'[k])$ is then proposed as a dimensionless discriminator.

When $C(H[k] \approx C(H'[k])$, the tested image is considered cover image, while if the tested image is stego, there should $C(H[k] < C(H'[k])$. Another way of applying the HCF-COM is also introduced by computing the adjacency histogram. The author claimed that this designed steganalysis technique is also detect other types of steganography beside of LSBM based steganography techniques.

7. Spatial Rich Model (SRM)

This steganalysis technique differs from previous mentioned steganalysis techniques earlier in that it is a universal steganalysis while others were targeted steganalysis. Unlike the above steganalysis tools, SRM steganalysis technique was not designed for real life application since it needs a large training sets and high dimensional feature spaces (Ker, et al., 2013) (Holub, et al., 2014). Also, SRM steganalysis only detects whether the tested image is a cover or stego without estimating the embedded message length (Fridrich & Kodovsky, 2012). SRM is based on feature extraction, and the goal is to capture a large number of different types of features reflecting dependencies among neighbouring pixels to give the model the ability to detect a wide variety of embedding algorithms. The process starts with assembling a rich model of the noise component as a union of many diverse sub-models formed by joint distributions of neighbouring samples from quantized image noise residuals obtained using linear and non-linear high-pass filters. Multiple noise residuals can be defined as representing the image using a feature computed from image spatial domain noise components, and is called spatial rich model (SRM). The quantization makes the residual more sensitive to embedding changes at spatial discontinuities in the image (i.e. edges and textures). The sub-models will be constructed from computing the correlation between neighbouring pixels in the horizontal, vertical, and diagonal directions. In total, 34671 sub-models/ features are computed. Finally, the proposed machine learning ensemble classifier in (Kodovsky, et al., 2012) is used to classify whether the image is a cover or a stego. Ensemble classifier consists of multiple classifiers to predict more accurately the class labels of unknown examples by aggregating the predictions of multiple classifiers (Tan, et al., 2006). An ensemble classifier usually adopts a weighted/unweighted majority vote on the predictions of the base classifiers.

Security/detectability is quantified using the ensemble's "out-of-bag" (OOB) error $\mathrm{E}_{\mathrm{OOB}}$, which is an unbiased estimate of the testing error, averaged, over multiple bootstrap samples of the image source during training. The image database is randomly splitting into two equal size groups training and testing group. The SRM strategy is performed on a given cover source and its stego version embedded with a fixed payload. The final evaluation of steganography techniques, output by SRM, is based on how many stego-images are identified and how many pass through undetected.

3.3 An Overview of our Approach

This thesis is concerned with hiding secret image files in image files. The message embedding is done in the spatial domain by concealing the secret bits in the cover pixels LSB (and in some cases in the 2^{nd} LSB). Our main objective is to design image-based steganography scheme that has the advantage of high embedding efficiency, acceptable stego-image quality, and low secret message detectability without compromising payload capacity. Obviously, achieving a high embedding efficiency leads to achieving a high message un-detectability/security while maintaining capacity, because when the embedding efficiency increases, the less detectable traces will be introduced in the stego-image. Therefore, in order to enhance embedding efficiency, our main innovative approach is first to increase the probability of similarity between the secret bits value and the cover pixels' LSB value. This will be achieved in 2 novel steps involving manipulation of both the cover image and the secret image that results a higher ratio of both the secret bits and the cover pixels' LSB having a value of zero than one. For the first step three algorithms are proposed, in Chapter 5, to pre-process the secret image prior to embedding so that the resulting secret bit-stream contains a higher number of 0 bits than 1, but one of these algorithms also reduces the length of the secret bit-stream without losing information. For the second step in our increased similarity strategy, we investigate a number of pixel value decomposition techniques, in Chapter 6, and determine the best decomposition that achieves the highest number of 0 bits in the cover LSB plane. Finally, in Chapter 7, we exploit the above two steps strategy to propose a bit-plane(s) mapping embedding technique, instead of bit-plane(s) replacement in order to make each cover pixel can be used for secret embedding. We shall demonstrate that the combination of the mapping-based embedding scheme and the above two-steps strategy produces stego-images that have minimal distortion, i.e. reducing the number of the cover pixels changes after message embedding and increasing embedding efficiency. Finally, in order to evaluate our proposed image-based steganography techniques in terms of detectability/security, different kinds of common and well-known steganalysis tools are applied on the produced stego-images.

3.4 Summary

In this chapter, different image-based steganography approaches have been reviewed to conceal secrets. The spatial domain approaches included: approaches to hide secrets in cover pixels' LSB or other bit-planes; approaches based on different pixel value decomposition rather than the usual binary decomposition; approaches that embed in specific regions of cover images based on texture criteria; and approaches that they have high embedding efficiency. Embedding in the spatial domain has advantages over embedding in the frequency domain in terms of higher payload capacity and better stego visual quality. We also reviewed the most common steganalysis tools and these tools are used to evaluate the performance of our proposed embedding schemes in terms of embedded message detectability. Currently, the most successful image-based steganography approaches are those employ high embedding efficiency by producing a stego-image with minimal distortion, in order to resist steganalysis attacks. Finally, we presented an overview of the approaches adopted in this thesis and the main contributions planned to achieve high embedding efficiency, and robustness against steganalysis tools while maintaining capacity when the secrets are images.

Chapter 4

Multi Bit-planes Image-based Steganography

In this chapter, we initiate our research investigations into spatial domain image based steganography by developing and testing schemes that manipulate more than one bit-plane including the LSB plane to embed one or two secret bits. The main objectives are to improve un-detectability of the secret and/or capacity of embedding. First, in Section 4.1, we introduce an Indexing-based hiding scheme that embeds one secret bit in a way that depends on the first two LSBs of the cover image pixels, which will be shown experimentally to have improved un-detectability compared to LSBR while maintaining the same capacity of LSBR. In the second scheme, we shall attempt to double capacity and improve un-detectability of the Indexing-based scheme. This second scheme, introduced in Section 4.2, uses a Mapping-based embedding to embed two secret bits in three LSBs of the cover image pixels represented by the Fibonacci pixel value decomposition. We shall demonstrate that this Mapping-based does meet the stated objectives on capacity and un-detectability. We shall also test both schemes for robustness against the three well-known targeted steganalysis tools (RS, DIH and RWS) described in Chapter 3.

4.1 Bit-plane Indexing-based Embedding Scheme

In this section, we present the first proposal for hiding a secret image into the spatial domain of a cover image which works by embedding a single bit secret by manipulating multiple cover image bit-planes for increased security without undermining capacity.

The incentive for this approach comes from a desire to improve the visual quality of existing schemes that embed in more one bit-plane. This approach is based on bit-planes index manipulation confined to the first two LSBs of the cover image. We shall present the results of a sufficiently large experiment conducted to test the performance of this scheme in terms of un-detectability and robustness against targeted steganalysis tools. Experimental results demonstrate that the proposed technique is secure against steganalysis techniques such as DIH, and RWS, while RS detects it. The developed scheme has the same payload capacity of LSBR but at the expense lower stego-image quality, and it was published in (Abdulla, et al., 2013)

4.1.1 Embedding and Extracting Procedures

Like any steganography scheme, this algorithm consists of two components, the embedding procedure and the extracting procedure. Although, the main focus of this thesis is on embedding secret images, this algorithm is equally applicable to hide any type of secrets. However, in the presented experimental results, the secrets are images of size 128x256 resulting in a secret of length 262144 bits.

Embedding Procedure

1. The cover image is first pre-processed by modifying the 2LSBs of each pixel in the original image so that they are not equal, i.e.

$$2LSBs = \begin{cases} 01 & if\ 2LSBs = 11 \\ 10 & if\ 2LSBs = 00 \\ 2LSBs & otherwise \end{cases} \qquad (4.\ 1)$$

2. One secret bit is embedded in each pixel. The secret bit is first compared with the first LSB of the modified cover pixel. If they are equal, then record the index of the first LSB plane. Otherwise, record the index of the second LSB plane (i.e., record 0 if the secret bit matches the first LSB; record 1 if the secret bit matches the second LSB).

3. For the next secret bit, check the same similarity. This time the record value of the index must be different from the previous one because resulting vector of indices must be in form of 10s or 01s, i.e. if the previous index value was 1, the next index value must be 0, otherwise swap the first two LSBs of the cover pixel.

4. Finally, the vector of indices is either of form 1 0 1 0….1 0 or 0 1 0 1 …. 0 1, i.e., each index value differs from the previous one by a circular shift of size 1. This

vector must be sent to the receiver in a form n(10) or n(01), n is the number of repeating 10s or 01s in a vector. For example, if there are one thousand secret bits, then the receiver should get 500 (10) or 500 (01).

This algorithm makes two possible changes to the cover image and informs the receiver of the index sequence. The first change, eliminate the possibility of pixels having their 2LSB bit equal. This would mean that the 2LSB's of any pixel are different. Now index of the bit in the 2LSB of the cover pixel that matches the secret is recorded but the system first the 2LSB are swapped if they match the 2LSB of the previous pixel. We shall also give a specific example for embedding a 4-bit short secret in a 4-pixel image.

Example

If we have the secret bits 0 0 1 0, and the first two LSBs of the four pre-processed cover pixels are 01, 01, 10, 10. The first secret bit (which is 0 here) is compared with the first LSB (which is 1) of the first selected cover pixel. Because they are not equal then we compare the secret bit with second LSB (which is 0), now they are equal, and we record the index value 1 indicating that the secret bit is similar to the second LSB of the selected cover pixel. The next secret bit (which is 0) is compared with the first LSB of the next selected cover pixel (which is 1), because they are not equal then the secret bit must be compared with the second LSB (which is 0) and now they are equal but cannot record the index value 1 because the previous index value was 1, in this case do the swapping between the first and second LSB, i.e. change 01 to 10, and now the secret bit is similar to the first LSB then record index value 0. Continuing in this way we get a vector of indices such (1, 0, 1, 0). Now the sender should send 2(10) to the receiver indicating 2 pairs of 10s.

Extracting Procedure

1. Depending on the n(10) / n(01) the receiver creates the vector of indices.
2. If the element of the vector of indices is 0, it means the secret bit must be extracted from the first LSB of the selected stego pixel, otherwise (i.e. the element is 1) the secret bit must be extracted from the second LSB. All bits can be extracted by repeating this procedure.

The extraction is a fairly simple once you know the pattern. If the first pattern is received, then starting from the first stego pixel, the secret bits are retrieved in pairs either the 2nd LSB from the current pixel and the LSB of the next pixel, vice-versa if you receive n(10) or n(01), respectively.

4.1.2 Experimental Setup and Results

To evaluate the performance of the proposed Indexing-based steganography scheme, we need to use a sufficiently large set of different types of secret images to be embedded into different cover images and evaluate the various measures associated with some of the steganography success criteria (embedding efficacy, un-detectability, and stego-image quality). We shall do these for different embedding payloads.

Setup

In our experiments, the Miscellaneous Volume of Signal and Image Processing Institute (SIPI) database of University of Southern California (Viterbi, 1981) is used to evaluate our proposed steganography system. This database consists of 44 different size images of which 16 are colour, and 28 are monochrome images. This database includes some standard images such as Lenna, Baboon, Peppers, Jet, Tiffany, Couple, Bridge, Pirate, House and Lake. We created two versions of these 44 images by resizing to 512 x 512, and 128 x 256; and convert them into grayscale images with 8 bits per pixel. The reason of resizing these images to 128 x 256 is to make the number of bits that represent a secret image (262144 bits) equal to the number of cover image pixels which are of size 512 x 512.

Three sets of experiments are conducted to evaluate the performance of the proposed steganography system: The first is to measure the embedding efficiency, the second is to test the stego-image quality, and the third one is to measure the detectability/security of the embedded message.

Results

In each of the three experiments, we use each of the SIPI database 44 images size 128 x 256 as a secret, after transforming into a binary stream, which were embedded in each of the 44 images of size 512 x 512, once with our scheme and once using the LSBR scheme for comparison. In each experiment, the test is based on 5 different payload ratio (20%, 40%, 60%, 80%, and 100%) of the size of the secret stream. Note that for each tested steganography technique we tested a total of (44 x 44 = 1936) stego-images.

1. Embedding Efficiency Evaluation

The results in these experiments are presented in two equivalent ways: the ratio of changed pixels to the embedded secret size (in Figure 4-1), and the formal efficiency values (in Figure 4-2). In both cases, the results are the averaged values over all the 1936 stego-images.

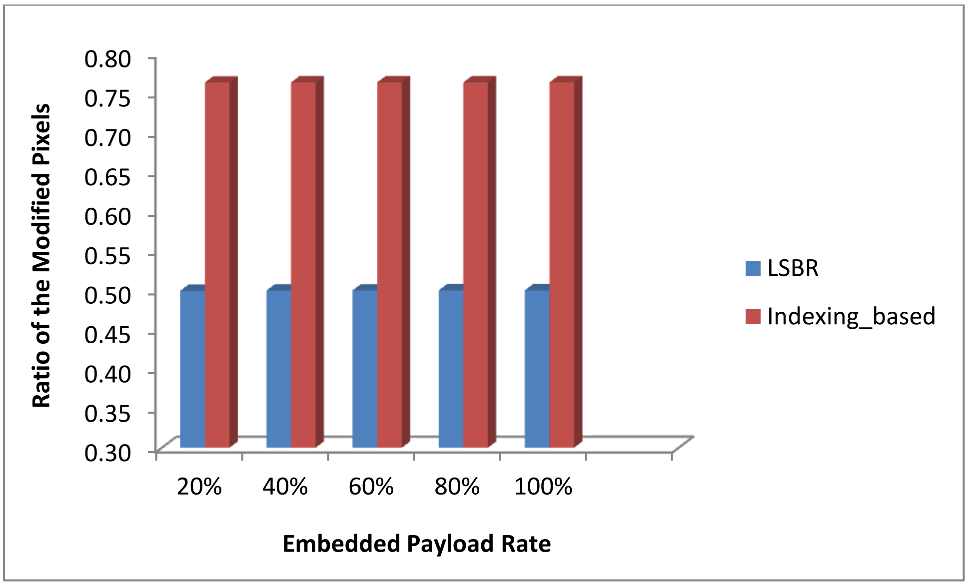


Figure 4-1: Ratio of modified pixels for the LSBR and Indexing-based embedding technique.

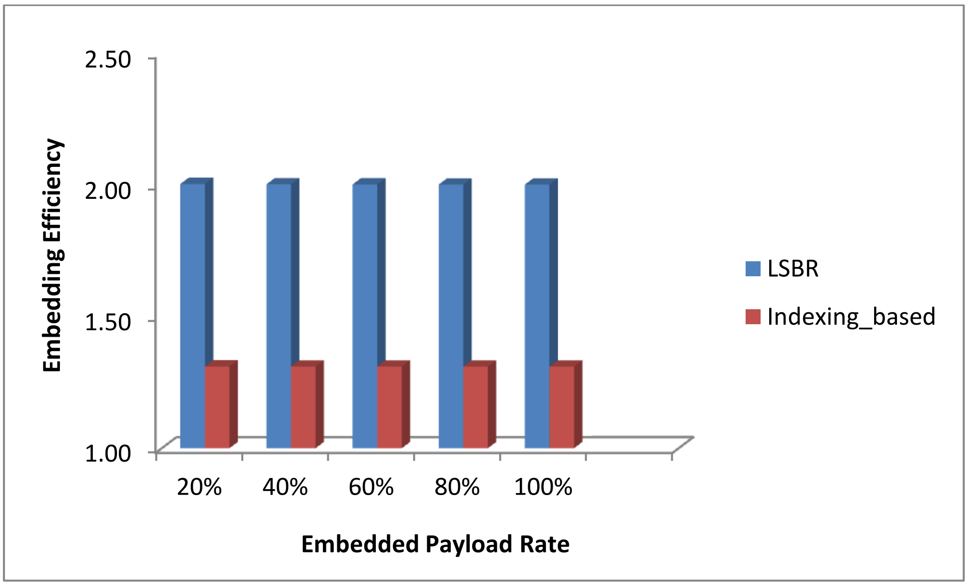


Figure 4-2: Embedding efficiency for the LSBR and Indexing-based embedding technique.

From Figure 4-1, it is noticeable that our Indexing-based embedding technique causes a higher number of cover pixels to be changed than the LSBR, and hence has lower embedding efficiency, as presented in Figure 4-2. This seems to reflect the pre-processing on the cover image and swapping the first two LSBs of the cover pixel. In order to improve efficiency, perhaps we need to think of how to avoid the effect of the pre-processing and/or the swapping.

2. Stego-Image Quality Evaluation

Figure 4-3 presents the average value of the PSNR of the two tested steganography techniques at different embedding payloads. It is noticeable that for all payloads, the PSNR of the proposed Indexing-based embedding technique is lower than PSNR of the LSBR. This is because, in the Indexing-based embedding technique, the 2nd LSB is also may change after message embedding. Also, these results reflect the low efficiency achieved by the Indexing-based scheme.

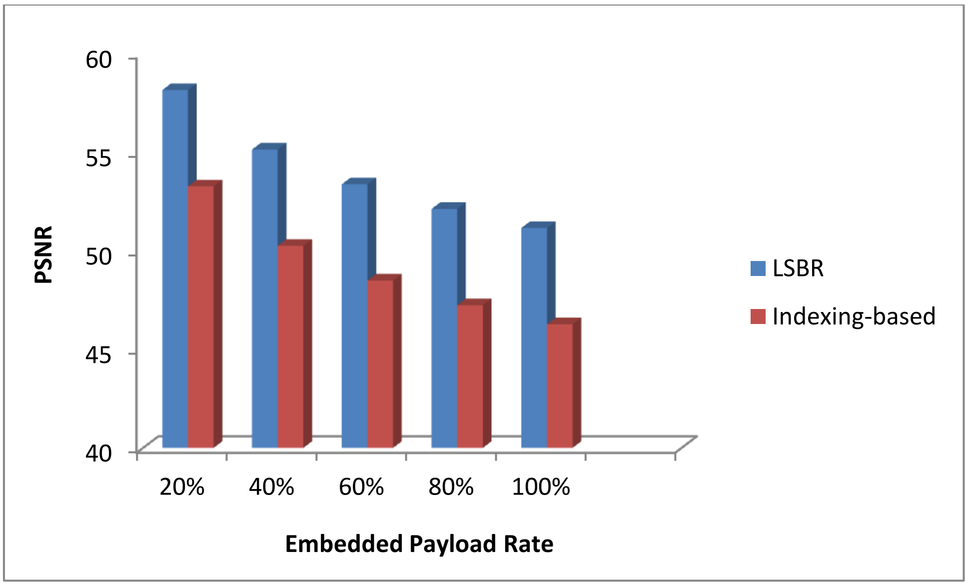


Figure 4-3: The PSNR for the LSBR and Indexing-based embedding technique.

3. Detectability Evaluation

Three well-known steganalysis techniques (detectors) have been used to evaluate the detectability of the proposed steganography technique. These steganalysis techniques are RS, DIH, and RWS. (The detailed descriptions of these steganalysis tools were given in Chapter 3).

Robustness Against RS Detector

Figure 4-4 and Figure 4-5 are presenting the RS diagram for the LSBR and our Indexing-based embedding techniques. It is noticeable that embedding higher rate of secret bits lead to an increase in the differences between RM and RM-, SM and SM-, indicating the presence of a secret message. Therefore, both the LSBR and the proposed Indexing-based technique are not robust against the RS detector. The reason is that pre-processing the cover image prior to secret embedding cause to change the cover pixel value, even if the secret bit is similar to the cover pixel's LSB.

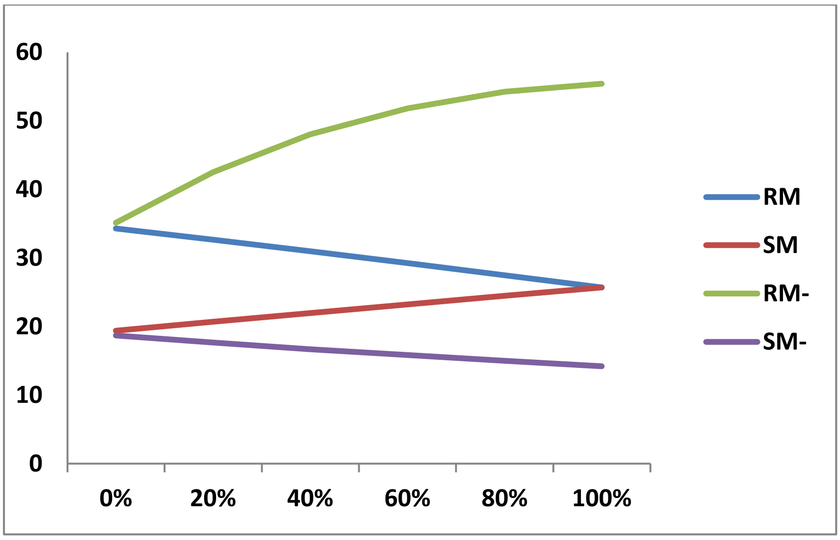


Figure 4-4: RS diagram for LSBR technique.

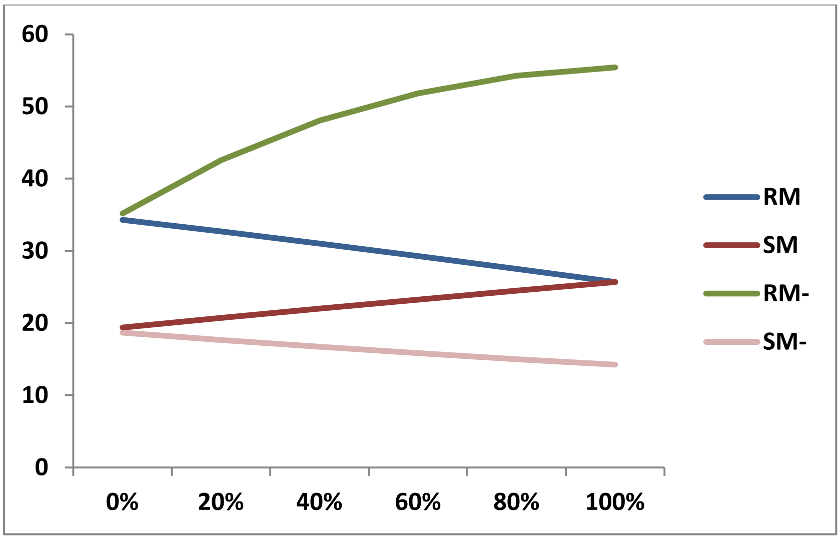


Figure 4-5: RS diagram for the Indexing-based embedding technique.

Robustness Against DIH Detector

For each embedding ratio, the chart of Figure 4-6 presents the average values representing the probability of having a secret hidden with the given embedding ratio. These probabilities are estimated by a classification of the differences between histograms obtained from image differences between the image and the two types of flipped LSB's.

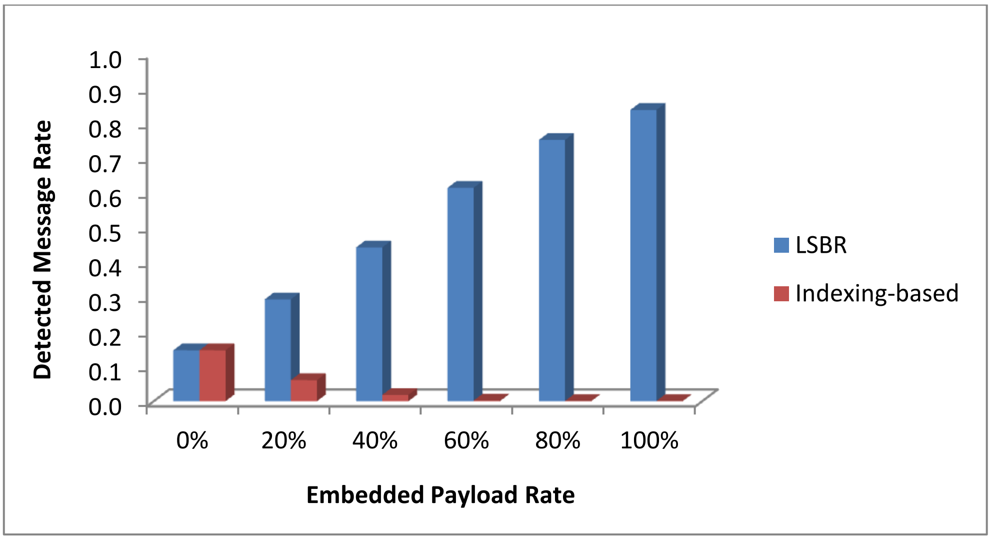


Figure 4-6: DIH steganalysis for LSBR and Indexing-based embedding technique.

From Figure 4-6, one can see that embedding the secret images using the Indexing-based technique is robust against the DIH. In fact, DIH ability to detect our scheme diminishes the more payload is embedded, while DIH predicts the presence of a secret embedded by the LSBR at higher than the actual payload for 20%, 40%, and 60%. The robustness of our scheme is that some of the secret bits are embedded in the 2nd LSB, while the DIH tool is designed to detect the secret message that embedded in the 1st LSB.

Robustness Against RWS Detector

Figure 4-7 presents the average values of the estimation results of the flipped cover pixels' LSB of each tested steganography techniques after message embedding.

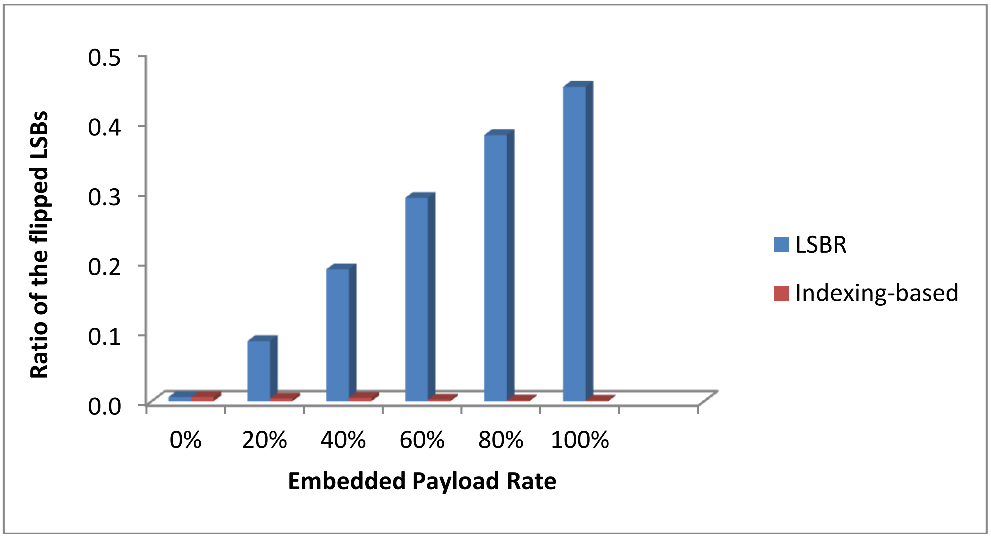


Figure 4-7: RWS steganalysis for LSBR and Indexing-based embedding technique.

Figure 4-7, demonstrates beyond any doubts the robustness of our Indexing-based scheme against the RWS tool. For all payloads, the RWS returns nearly 0% flipping of pixels LSB. In contrast, the LSBR detectable by the RWS by reporting the percentage of flipped is nearly 0.5 of the embedded message size. The same reasons that we mentioned above as to you our scheme is robust against the DIH can equally explain its robustness against RWS steganalysis tool.

In summary, we find that this rather primitive attempt to use 2 LSBs for embedding one bit secret succeeded in improving robustness against two targeted LSB steganalysis tools and not succeed against a third one. However, this success comes at the expense of poorer stego visual quality and lower embedding efficiency. This raises the question can we maintain this level of robustness against these steganalysis tools and yet improve efficiency and/or visual quality? The experience with this scheme, indicate that using other than the LSB of the cover image has helped in fooling the steganalysis tools, although changes were made.

4.2 Fibonacci-Mapping based Embedding Scheme

In this section, we present a second approach which investigates and designs to exploit the good properties of the Fibonacci decomposition of elongating the pixel bit representation and reducing the effect of changes to the first few LSB's. These benefits of Fibonacci decomposition encourage us to embed more than 1 bit into the first 3 LSBs and thereby increasing embedding capacity not increasing the chances of pixel changes. Existing Fibonacci embedding schemes still relies on bit replacement which will reduces the ability to embed in every pixel due to the fact that the process of replacing any bit could lead to violating the Zeckendorf theorem, see Chapter 3 for more detail. Our approach is designed to solve this problem and make all cover image pixels usable as candidates for embedding using an innovative idea that extends Fibonacci-like steganography by bit-plane(s) *mapping* instead of bit-plane(s) *replacement*. Experimental results will demonstrate that the ability to double the embedding capacity, compared to LSBR, and it is secure against steganalysis techniques such as RS, DIH, and RWS, (Abdulla, et al., 2013).

4.2.1 Embedding and Extracting Procedures

This scheme is based on a mapping table for embedding. First, the cover image pixels are to be decomposed by the Fibonacci sequence {1, 2, 3, 5, 8,…,233} into unique 12 bit strings that adhere to the Zeckendorf condition where no two consecutive 1's are allowed. According to the Zeckendorf theorem, the probabilities of first three LSBs of a cover pixel in Fibonacci representation are (000, 001, 010, 100, 101). The Fibonacci-Mapping steganography scheme embeds two secret bits at a time by changing the first 3 LSBs of the Fibonacci decomposed pixels according to Table 4-1, below.

Table 4-1: Fibonacci-Mapping Table.

Cover bits	**Secret bits**			
	00	**01**	**10**	**11**
000	000	001	100	101
001	000	001	100	101
010	010	001	100	101
100	010	001	100	101
101	010	001	100	101

Note that, mapping the two secret bits into the 3 LSB cover pixels will only result in changing the LSB in half of the cases. Having said that, the use of the table also results in changing the other bit-planes.

Secret Embedding:

1. Use the mapping table to change the 3LSB of the input Fibonacci code according to the two bits of the input secret.
2. Check: If the Zeckendorf theorem is violated (i.e. the resulting Fibonacci code = x…x11xx) then replace it with the Zeckendorf-compliant Fibonacci code (x…x01xx).

Example

Let P be a cover pixel with the Fibonacci code: 0 0 0 0 0 1 0 0 1 0 0 1 and the two secret bits 11. The mapping table changes P into the Fibonacci code:

0 0 0 0 0 1 0 0 **1 1** 0 1

which violates the Zeckendorf theorem, the checking step will output

0 0 0 0 0 1 0 0 **0 1** 0 1

Secret Extraction:

At the receiver end, the secret message can be simply extracted as the first and third LSBs of the Fibonacci representation of the selected stego pixel value.

This proposed Fibonacci-Mapping based embedding has the advantage of doubling capacity, since every cover pixel is used for message embedding and each pixel can carry two secret bits. However, this may result in more degradation and low stego quality, because the secret bits are embedded in higher bit-planes of the cover pixel. The extent of which this approach leads to degradation will be determined experimentally in the next section.

4.2.2 Experimental Results

In this subsection, we test the performance of the proposed scheme for embedding secret images using the same 44 images from the SIPI database according to the same experimental strategy used for the Indexing-based scheme. We created three versions of these 44 images by resizing to 512 x 512, 256 x 256 and 128 x 256. Each image in the last two versions is used as secret images to be embedded in each of the 44 images in the 512 x 512 version. The inclusion of the 256 x 256 version images as secrets is necessitated by the fact that our scheme has double the capacity of the LSBR scheme which means that we could not realise a full capacity embedding only by using the 128 x 256 images. Our experiments will be conducted by embedding 5 payloads (20%, 40%, 60%, 80%, 100%) for the LSBR and our scheme, but we will include an extra payload experiment for our Fibonacci-Mapping scheme at 200%. For each payload then we have a total of (44 x 44 = 1936) stego-images for each tested steganography technique.

As before, three sets of experiments are conducted to evaluate the performance of the proposed steganography system: The first is to measure the embedding efficiency, the second is to test the stego-image quality, and the third one is to measure the detectability/security of the embedded message.

1. Embedding Efficiency Evaluation

Figure 4-8 presents the average value of the ratio of modified pixels to the length of the secret bits, for both tested steganography techniques, and Figure 4-9 presents the average value of the embedding efficiency of the tested steganography techniques.

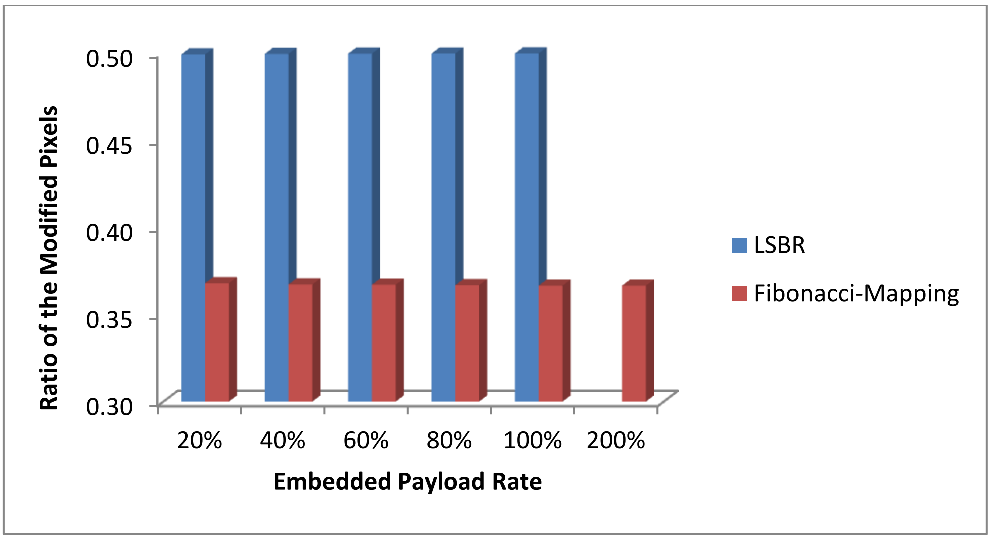


Figure 4-8: The ratio of the modified pixels for the LSBR and Fibonacci-Mapping based embedding technique.

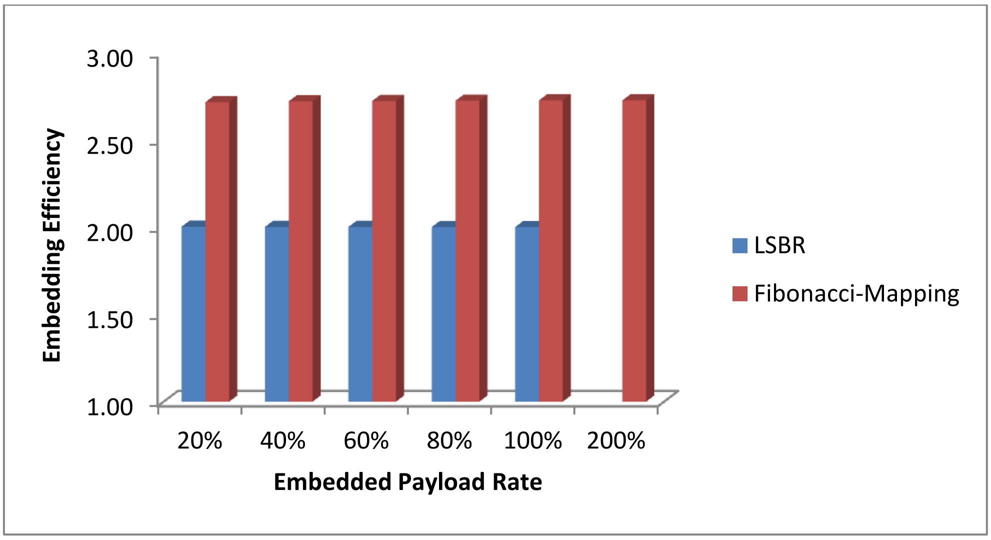


Figure 4-9: The embedding efficiency for the LSBR and Fibonacci-mapping based embedding technique.

From Figure 4-8, it is clear that our Fibonacci-Mapping based embedding technique causes lower number of changed cover pixels after secret embedding compared to the LSBR, and consequently it has higher embedding efficiency, as presented in Figure 4-9. These improved results, compared to the performance of our earlier Indexing-based scheme, are achieved due to embedding two secret bits in one cover pixel.

2. Stego-Image Quality Evaluation

Figure 4-10 presents the average of the PSNR values of the stego-images relative to the cover images computed for the tested steganography techniques. Unfortunately, for all embedded message rate, the PSNR of the Fibonacci-Mapping embedding technique is lower than that achieved by the LSBR scheme. This is because, in the Fibonacci-

Mapping scheme, the higher bit-planes may also change after message embedding. However, the PSNR achieved by this scheme is only marginally lower than achieved by the previous Indexing-based scheme, and yet we increased doubled the capacity.

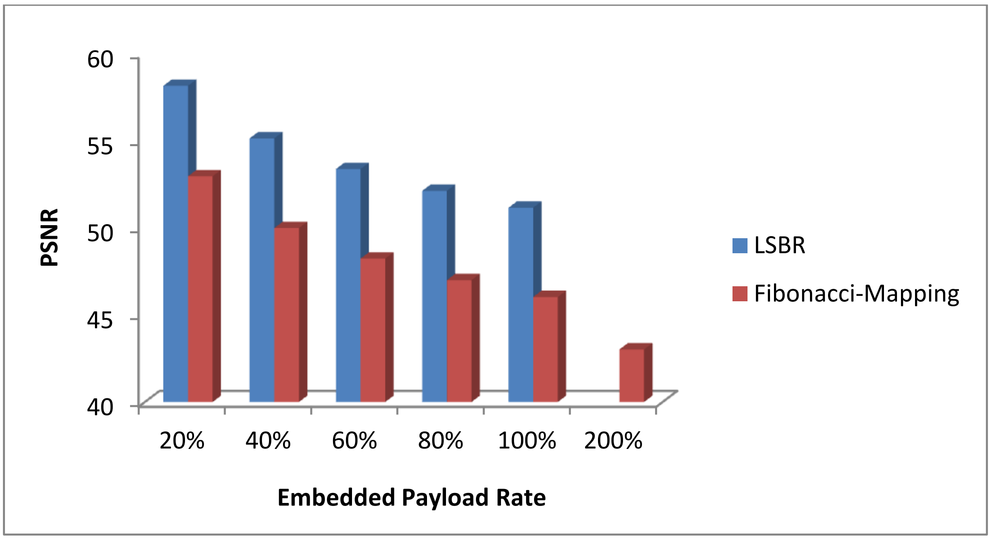


Figure 4-10: The PSNR for the LSBR and Fibonacci-Mapping based embedding technique.

3. Detectability Evaluation

Again, the three well-known steganalysis detectors (RS, DIH, and RWS) have been used to evaluate the detectability of the proposed steganography scheme.

Robustness Against the RS Detector

Figure 4-11 displays the RS diagram for the Fibonacci-Mapping scheme, from which it is clear that for all payloads, including the double capacity load, there are hardly any differences between RM and RM-, SM and SM-, demonstrating the robustness of the proposed scheme against the RS detector. The reason is that in this case, lower numbers of cover pixels are changed after secret embedding compared to LSBR and the Indexing-based scheme, since only one cover pixel may change by embedding two secret bits. Moreover, the LSB of most changed pixels remain unaffected and therefore RS is unable to detect significant changes. In fact, the combined of the effect of using the Fibonacci cover pixel decomposition and the mapping table show that only 10 out 20 combinations result in changed LSB, i.e. probability of changed LSB is $\leq$ 50%. However, this upper bound of the probability of LSB change reduces significantly to about 18.3%, because for all embedding payloads only 36.6% of pixels change after embedding.

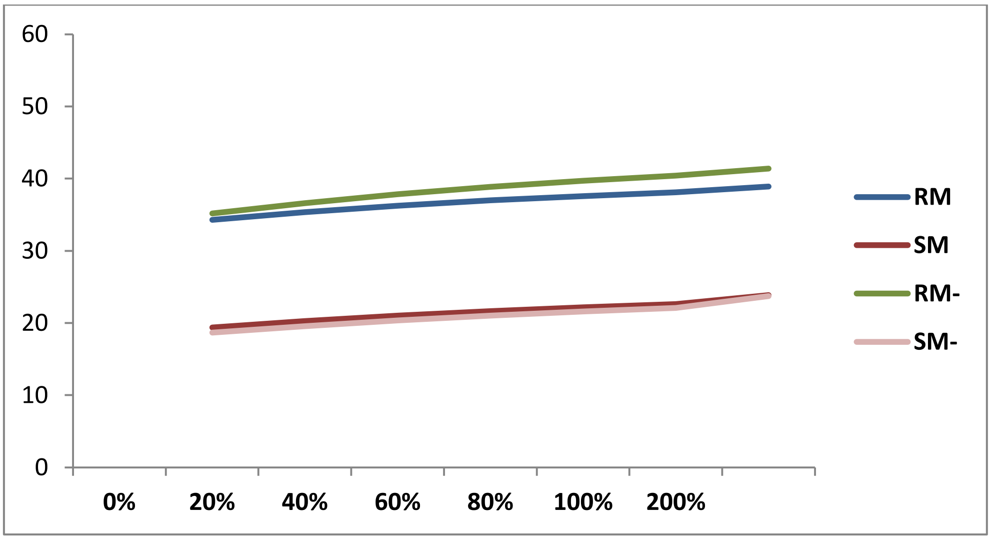


Figure 4-11: RS diagram for Fibonacci-Mapping scheme.

Robustness Against DIH Detector

For each embedding ratio, the chart of Figure 4-12 presents the average values representing the probability of having a secret hidden with the given embedding ratio.

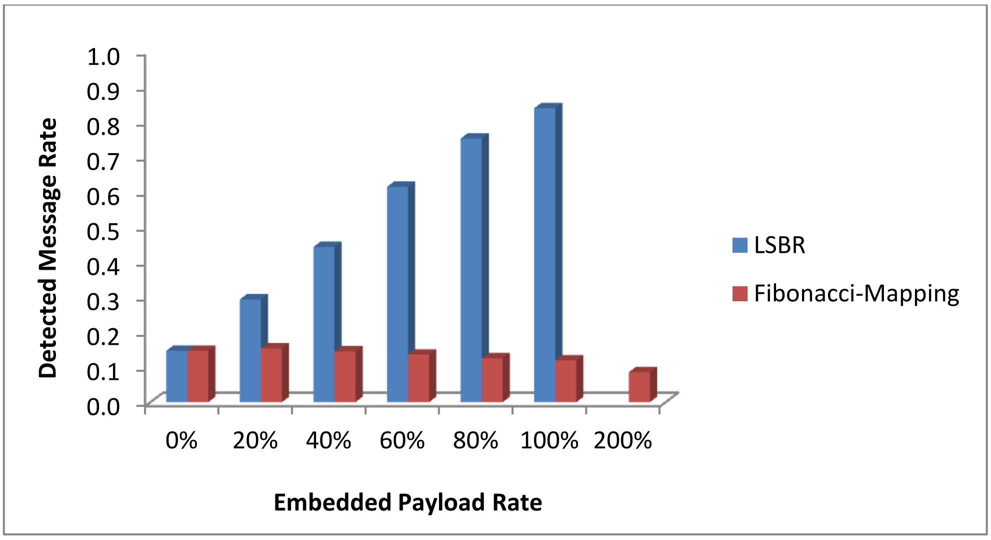


Figure 4-12: DIH steganalysis for LSBR and Fibonacci-Mapping based embedding technique.

From Figure 4-12, we see that embedding the secret images using the Fibonacci-Mapping technique is robust against the DIH. Similarly to the case of the Indexing-based scheme, DIH ability to detect the secrets embedded by the Fibonacci-Mapping scheme diminishes the more payload is embedded. However, the detection probabilities are slightly higher than that reported for the Indexing-based scheme. As discussed before, DIH predicts the presence of a secret embedded by the LSBR at higher than the actual payload for 20%, 40%, and 60%.

Robustness Against RWS Detector

Figure 4-13 presents the average values of the estimation ratios of the flipped cover pixels' LSB of our Fibonacci-Mapping scheme against the LSBR at different embedding ratios. Note that, the 200% payload here is only possible by our mapping scheme.

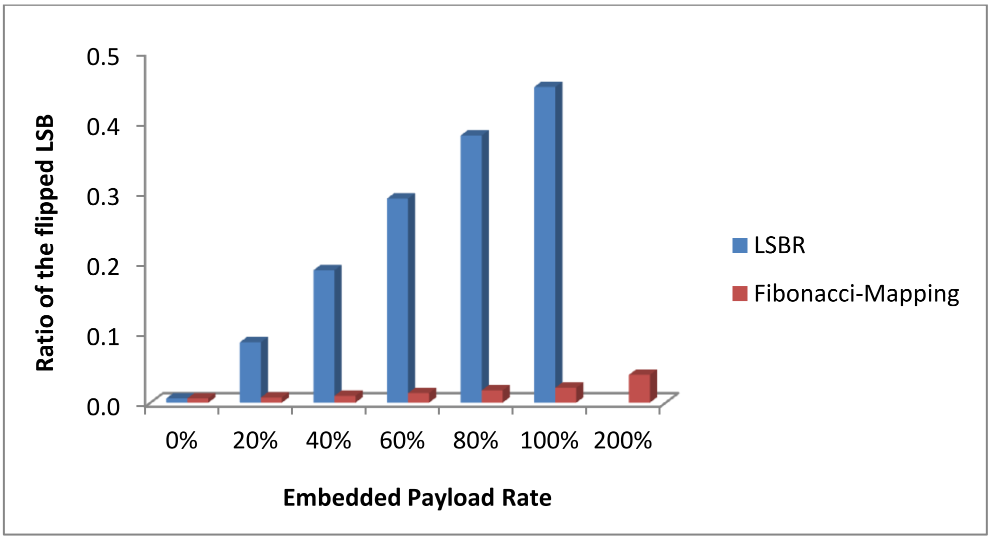


Figure 4-13: RWS steganalysis for LSBR and Fibonacci-Mapping based embedding technique.

Figure 4-13, demonstrates beyond any doubt the robustness of the Fibonacci-Mapping against this LSB-based steganalysis tool. As before, this is due to the fact that this scheme results in flipping the LSB of fewer cover pixels than the LSBR. However, the detected ratios are slightly higher than that reported for the Indexing-based scheme. This yet another evidence that our scheme achieves high robustness against all LSB-targeted steganalysis tools.

In summary, the use of Fibonacci decomposition of cover pixels resulted in reducing the number of different first 3 LSB pattern from the normal 8 to 5, which encouraged the use of these 3 bit-planes for embedding two secret bits and thereby doubling the payload capacity. The use of a mapping table helped reduce the number of possible LSB changes and increase the embedding efficiency compared to the LSBR and the Indexing-based scheme. Consequently, it improved robustness against the three well-known steganalysis tools.

4.3 Discussion

In this chapter, we designed two rather simple embedding approaches and tested their performances in terms of embedding efficiency, stego-image quality and robustness against steganalysis tools. In both cases, we attempted to include more than the LSB plane for hiding the secret. The first scheme, embeds only one bit in each pixel and uses a combination of pre-processing the image pixels to eliminate the possibility of having equal bits in the 2LSB planes, followed by a system that report the index of the bit that matches the secret bit. Compared to the LSBR, this scheme resulted in lower stego quality and embedding efficiency, but it is robust against two of the steganalysis tools. The second approach extends Fibonacci-like steganography by bit-plane(s) mapping instead of bit-plane(s) replacement to embed two secret bits in three Fibonacci bit-planes. Unlike the original Fibonacci scheme, no cover pixels are excluded from embedding because actions are taken to comply with Zeckendorf theorem. Consequently, this scheme has double the embedding capacity of LSBR. Furthermore, it is secure against steganalysis techniques such as RS, DIH, and RWS. The improved capacity and robustness seems to come at the expense of further reduction of stego-image quality compared to the Indexing-based scheme. Considering the structure of the mapping table, may help in finding ways of improving stego quality of the Fibonacci-Mapping scheme while preserving the gains made in robustness against steganalysis tools and the embedding efficiency. We note that 6 out of the 10 cases where LSB is changed as a result of embedding are coming from the case where the two embedded secret bits are 01 or 11. In the next chapter, we shall investigate secret image pre-processing to transform secret images to increase the number of 0 bits value in its bit-stream representation, which will reduce the number of occurrences of 01 and 11.

Chapter 5

Secret Image Pre-Processing

In the previous chapter, we demonstrated the benefits of using a combination of Fibonacci decomposition of cover image pixel values and a mapping table for embedding a secret bit-stream, in terms of doubling capacity, higher efficiency of embedding and improved robustness against steganalysis tools. However, the stego-image quality was less than desirable and slightly lower than what was achieved by the simple Indexing-based scheme as well as the LSBR. However, we noted that the structure of the mapping table may explain this degradation in quality, because changes of LSB in the cover image pixels occur 6 times out of 20 in the table entries under the secret columns labelled by 01 and 11. Therefore, a possible way of improving the stego quality is to pre-process the secret image with the aim of increasing the ratio of 0 to 1 (0:1) in its bit-stream. This is feasible due to the fact that the secret image bit-stream is not random and existing local spatial correlations. This would the main task for our investigations in this chapter, which could provide motivation for moving the focus of steganography research into content-based schemes.

Different from existing approaches for enhancing embedding efficiency and security of the steganography techniques, our approach's idea is to exploit our knowledge of secret image information content to increase the probability of similarity between the secret bits value and the cover pixels' LSB value. The rest of this thesis is devoted to investigate and develop image processing schemes that can be used to achieve high similarity between secret image bits and the cover pixels' LSB, and increase the ratio of

0:1 for both secret bits and cover image LSB plane. In this chapter, we shall focus on processing the secret image prior to embedding and propose three algorithms that result in bit-streams containing a higher number of 0s than 1s. The three algorithms differ slightly in the objectives, in that the third one doesn't only increase the ratio of 0:1, but will reduce the length of the secret bit-stream, which the first two algorithms don't compress the secret bit-stream but achieve higher ratio of 0:1. The first two algorithms are similar in their structure except that the first is a spatial domain based manipulation while the second is in the Integer Wavelet domain. These three algorithms are presented in Sections 5.1, 5.2, and 5.3. In Section 5.4, we shall test the performance of the Fibonacci-Mapping embedding scheme post each of these three secret image processing schemes to demonstrate their positive impact on the stego-image quality as well as embedding efficiency and message detectability.

5.1 Secret Image Manipulation (SIM)

The SIM algorithm, exploits the structure of the secret image histogram to define a grayscale transform that maps secret pixel values according to the descending order of their frequencies so that more frequent pixel values are mapped into bytes with lower number of 1s. When two or more pixel values have the same frequencies, then they are mapped in according to their appearance in the sorted frequency vector. It is simply a substitution function on the histogram of the secret image, which means no loss of information as long as the receiver applies the inverse grayscale transform. This approach has similarity with statistical coding, but instead of assigning shorter codes to most frequent pixel values we keep the length and assign bytes of the lower number of 1s to the more frequent pixel values.

To simplify the process, we create an ordered table of the grayscale values in the ascending order of the number of 1s in the binary representation, see Table 5-1. We shall now describe the SIM forward and backward steps.

5.1.1 SIM Forward Procedure

The SIM forward includes the following steps that could be used for any steganography scheme:

1- Load the secret image I, and let h is the histogram of I .

2- Let h' is the sorted version of h in descending order of pixel value frequency.

3- Based on h', do replace the first highest repeated pixel value in the image I with the first new value in the Table 5-1. This step is continued by replacing the next highest repeated pixel value by the next new value, till all pixel values in the image I are replaced. This results in producing a new image I'.

4- Covert I' into binary to create the secret bit-stream.

5- Construct a side information bit-stream of length $(9 + (8 \times N))$ bits, where N refers to the number of pixel values present in the image I, to inform the receiver about the start of the secret image data. The first 9 bits of the side information represent N. The next $8 \times N$ bits of the side information, lists the original pixel values in descending order of frequencies.

6- Append the bit-stream of the secret image I' to the side information bit-stream, and embed in the chosen cover image using the given hiding scheme.

Note that for the SIM algorithm, the maximum possible number of bits for the side information part is (9 + 256 * 8 = 2057). This will reduce the payload capacity, but only by a very negligible proportion. The second part of the bit-stream represents the modified secret image I' and each 8 bits represent a pixel value that need to be inverted using side information. Figure 5-1 below, displays a secret image I (Lenna) and its SIM modified version I'.

Figure 5-1: Lenna image and its modified version using SIM algorithm.

Table 5-1: Grayscale values (0-255) in descending order of number of 1s in its binary representation.

value	Binary rep.	value	Binary rep.	value	Binary rep.	value	Binary rep.
0	00000000	82	01010010	135	10000111	174	10101110
1	00000001	84	01010100	139	10001011	179	10110011
2	00000010	88	01011000	141	10001101	181	10110101
4	00000100	97	01100001	142	10001110	182	10110110
8	00001000	98	01100010	147	10010011	185	10111001
16	00010000	100	01100100	149	10010101	186	10111010
32	00100000	104	01101000	150	10010110	188	10111100
64	01000000	112	01110000	153	10011001	199	11000111
128	10000000	131	10000011	154	10011010	203	11001011
3	00000011	133	10000101	156	10011100	205	11001101
5	00000101	134	10000110	163	10100011	206	11001110
6	00000110	137	10001001	165	10100101	211	11010011
9	00001001	138	10001010	166	10100110	213	11010101
10	00001010	140	10001100	169	10101001	214	11010110
12	00001100	145	10010001	170	10101010	217	11011001
17	00010001	146	10010010	172	10101100	218	11011010
18	00010010	148	10010100	177	10110001	220	11011100
20	00010100	152	10011000	178	10110010	227	11100011
24	00011000	161	10100001	180	10110100	229	11100101
33	00100001	162	10100010	184	10111000	230	11100110
34	00100010	164	10100100	195	11000011	233	11101001
36	00100100	168	10101000	197	11000101	234	11101010
40	00101000	176	10110000	198	11000110	236	11101100
48	00110000	193	11000001	201	11001001	241	11110001
65	01000001	194	11000010	202	11001010	242	11110010
66	01000010	196	11000100	204	11001100	244	11110100
68	01000100	200	11001000	209	11010001	248	11111000
72	01001000	208	11010000	210	11010010	63	00111111
80	01010000	224	11100000	212	11010100	95	01011111
96	01100000	15	00001111	216	11011000	111	01101111
129	10000001	23	00010111	225	11100001	119	01110111
130	10000010	27	00011011	226	11100010	123	01111011
132	10000100	29	00011101	228	11100100	125	01111101
136	10001000	30	00011110	232	11101000	126	01111110
144	10010000	39	00100111	240	11110000	159	10011111
160	10100000	43	00101011	31	00011111	175	10101111
192	11000000	45	00101101	47	00101111	183	10110111
7	00000111	46	00101110	55	00110111	187	10111011
11	00001011	51	00110011	59	00111011	189	10111101
13	00001101	53	00110101	61	00111101	190	10111110
14	00001110	54	00110110	62	00111110	207	11001111
19	00010011	57	00111001	79	01001111	215	11010111
21	00010101	58	00111010	87	01010111	219	11011011
22	00010110	60	00111100	91	01011011	221	11011101
25	00011001	71	01000111	93	01011101	222	11011110
26	00011010	75	01001011	94	01011110	231	11100111
28	00011100	77	01001101	103	01100111	235	11101011
35	00100011	78	01001110	107	01101011	237	11101101
37	00100101	83	01010011	109	01101101	238	11101110
38	00100110	85	01010101	110	01101110	243	11110011
41	00101001	86	01010110	115	01110011	245	11110101
42	00101010	89	01011001	117	01110101	246	11110110
44	00101100	90	01011010	118	01110110	249	11111001
49	00110001	92	01011100	121	01111001	250	11111010
50	00110010	99	01100011	122	01111010	252	11111100
52	00110100	101	01100101	124	01111100	127	01111111
56	00111000	102	01100110	143	10001111	191	10111111
67	01000011	105	01101001	151	10010111	223	11011111
69	01000101	106	01101010	155	10011011	239	11101111
70	01000110	108	01101100	157	10011101	247	11110111
73	01001001	113	01110001	158	10011110	251	11111011
74	01001010	114	01110010	167	10100111	253	11111101
76	01001100	116	01110100	171	10101011	254	11111110
81	01010001	120	01111000	173	10101101	255	11111111

5.1.2 SIM Backward Procedure

The receiver receives a bit-stream which contains two parts, the first part is the side information and the second part is the SIM modified secret image I' bit-stream. Based on the side information, the receiver is able to reconstruct the original secret image from the extracted bit-stream from stego-image using the following steps:

1- Extract the side information and the SIM modified secret image I'.

2- Let h' is the histogram of I'.

3- The original image I can be reconstructed by replacing the pixel values in the image I' that has the i^{th} value in the h' with the i^{th} value of the reconstructed original pixel values from the side information.

Note that the histogram $h'$is already started from the highest to the lowest frequency in the same order of Table 5-1.

5.1.3 Performance of SIM

To test the performance of the SIM algorithm in terms of ratio of 0:1 bits in secret image bit-stream before and after modification, we conducted experiments on the following image databases:

- 44 images from the Miscellaneous volume of Signal and Image Processing Institute (SIPI) database of University of Southern California (Viterbi, 1981). This database consists of 16 colour images and 28 monochrome images. It includes some standard images such as Lenna, Baboon, Peppers, Jet, Tiffany, Couple, Bridge, Pirate, House and Lake. We resized these 44 images to 512 x 512, 256 x 256, and 128 x 256; and converted them into grayscale images with 8 bits per pixel.
- 1000 images from BOSSBase version 1.0 database of grayscale images with a size 512 x 512 with 8 bits per pixel (Bas, et al., 2011). This database including images of, but not limited to, landscapes, people, plants, and building. This database consists of 10000 images; in our experiments the first 1000 images are used. These images are also resized to 256 x 256, and 128 x 256.

Experimental Results

Results of the experiments conducted for three different image sizes are shown in Table 5-2 and Table 5-3 for the SIPI and BOSSBase databases, respectively. Tables include statistical parameters (mean μ, standard deviation σ, minimum M_n, and maximum M_x) of the ratios of 0:1 before and after SIM modification as well as length of the side information over all images in the respective database. Here R refers to the 0:1 ratio before SIM, R' refers to the 0:1 ratio post SIM, L refers to the length of the side information.

Table 5-2: SIPI database - Ratio of 0:1 in the secret images and SIM modified secret images $\boldsymbol{I'}$.

	Image size 128 x 256			**Image size 256 x 256**			**Image size 512 x 512**		
	$\boldsymbol{R}$	$\boldsymbol{R'}$	$\boldsymbol{L}$	$\boldsymbol{R}$	$\boldsymbol{R'}$	$\boldsymbol{L}$	$\boldsymbol{R}$	$\boldsymbol{R'}$	$\boldsymbol{L}$
μ	0.49	0.71	1639	0.49	0.71	1676	0.49	0.73	1565
σ	0.08	0.07	279	0.09	0.07	273	0.10	0.08	499
$\boldsymbol{M_n}$	0.12	0.61	993	0.12	0.60	993	0.11	0.60	25
$\boldsymbol{M_x}$	0.65	0.93	2057	0.66	0.94	2057	0.66	0.99	2057

Table 5-3: BOSSBase database - Ratio of 0:1 in the secret images and SIM modified secret images $\boldsymbol{I'}$.

	Image size 128 x 256			**Image size 256 x 256**			**Image size 512 x 512**		
	$\boldsymbol{R}$	$\boldsymbol{R'}$	$\boldsymbol{L}$	$\boldsymbol{R}$	$\boldsymbol{R'}$	$\boldsymbol{L}$	$\boldsymbol{R}$	$\boldsymbol{R'}$	$\boldsymbol{L}$
μ	0.54	0.68	1815	0.54	0.68	1850	0.54	0.68	1912
σ	0.07	0.05	278	0.07	0.05	260	0.07	0.05	228
$\boldsymbol{M_n}$	0.17	0.57	385	0.16	0.57	537	0.15	0.56	777
$\boldsymbol{M_x}$	0.85	0.93	2057	0.86	0.93	2057	0.86	0.92	2057

From Table 5-2, we note that on average the ratio of 0:1 of the SIM modified images is increased by about 45% of the corresponding ratio for the original images. Similarly, the results of Table 5-3 show an increase of 26% in the ratio of 0:1 post SIM. The difference between the percentages of changed ratio reflects the variation in the nature of images in the two databases. In fact, the statistical parameters (μ and σ) in Table 5-3 are constant and independent of the image sizes, while this is not the case in Table 5-2. The minor changes are unlikely to be due to SIM but due to the fact that resizing has some minor effects on the ratios of 0:1 in the original SIPI images. These results also demonstrate that the proposed SIM algorithm can be used for any secret image sizes. The only drawback of the SIM is the side information that needs to send to the receiver, and this results in slightly decreasing the capacity of embedding. Table 5-4 shows the average percentage of the length of the side information out of modified SIM secret images. The table demonstrates that the increase in the total embedded secret size, as a result of the side information, is negligible and diminishes for larger size secret images. Thus, the embedded capacity is reduced by minute percentages.

Table 5-4: Ratio of bits of the SIM side information.

Image sizes	SIPI	BOSSBase
128 x 256	0.006	0.007
256 x 256	0.003	0.004
512 x 512	0.001	0.001

Two questions arise about the performance of SIM. Is it not possible to do the same with non-image secret bit-streams? And if not, what could be done to improve this performance even further? The answer to the first question is that the distribution of image pixel values are usually non-uniform which is exploited by SIM by mapping pixel values according to their non-uniform frequencies while other secret bit-streams are highly unlikely (or for security reasons are expected) to have uniform when it partitioned into 8-bits bytes. In fact, this also explains the significant differences between SIM's performance for different images in the two databases. This also points to the way of improving SIM performance for any image and in the next section we will develop an Integer Wavelet domain version of SIM which would help creating patterns of non-uniform distributions of certain Wavelet coefficients to be exploited for mapping the image bit-streams with higher 0:1 ratio.

5.2 Integer Wavelet based Secret Image Manipulation (IWSIM)

In Chapter 2, we described Discrete Wavelet Transforms (DWT) as multi-resolution frequency domain tools that analyse/split images into sub-bands of different frequencies ranges at different scales. The most important properties, for our purpose, of the Wavelet transformed image is that the histogram of the LL sub-band is an approximation of that of the original image, while the coefficients in each of the other high frequency sub-bands are have a Laplacian distribution (also known as Generalised Gaussian distributions), see Figure 5-2. The Integer Wavelet Transform IWT is a special kind of Wavelet transforms for which all sub-bands coefficients are integers rather than real numbers (Calderbank, et al., 1997). An example of IWT sub-bands for one level of Lenna image is presented in Figure 5-2, together with the histogram for each of the Wavelet sub-bands. The computation of the Wavelet coefficients in the IWT sub-bands is based on the following formulae:

$$\begin{aligned} d &= x_{i+1} - x_i \\ s &= x_i + \left\lfloor \frac{d}{2} \right\rfloor \end{aligned} \qquad (5.\ 1)$$

Where x_i and x_{i+1} are two consecutive pixels.

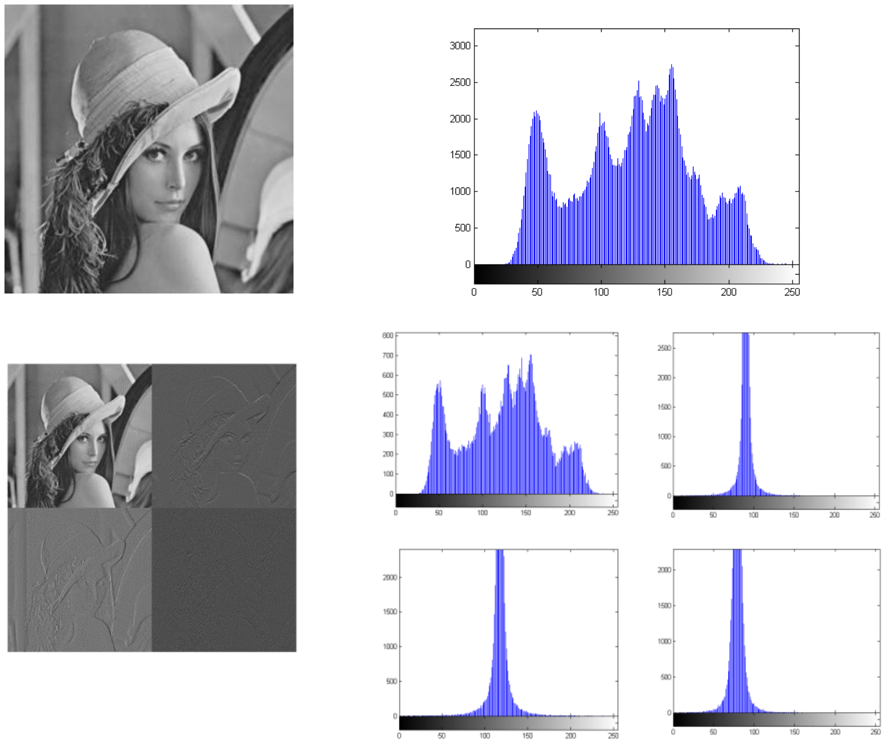


Figure 5-2: Level one IWT sub-bands of Lenna image and histograms.

IWSIM is an extended version of the SIM idea and objective. In contrast to SIM, IWSIM algorithm is not applied directly on the spatial domain of the secret image but in the Integer Wavelet domain. The IWSIM scheme first applies the IWT on the secret image only for one level, and then for each sub-band uses a SIM-like mapping depending on the range of coefficients in these sub-bands. Due to the fact, mentioned above, the coefficients in each of the three high frequency sub-bands have a Laplacian distribution. This implies that most of the coefficients in these sub-bands can be mapped using the similar approach to SIM to produce more 0s than 1s. However, the range and the number of present values differ from one sub-band to another. The IWT decomposed image will contain some coefficients whose values exceed 255 which require more than 8-bits to represent. At level 1, high frequency integer coefficients may require up to 10 bits to represent, and therefore for some high frequency sub-bands we need to expand the designed SIM mapping Table 5-1 and adjust the side information accordingly. At IWT decomposition level 2 or above, coefficients ranges usually expand and require even more than 10-bits to represent. This is why we apply the IWT only to level one, because at higher level decomposition requires much larger SIM-like mapping tables and increased size of side information that would reduce embedding capacity. Another reason for avoiding level 2 of IWT is the number of sub-bands increase to 7 sub-bands, and then each sub-band needs its own side information, and this reflects on increasing the total side information size for the secret image. We shall now describe the IWSIM forward and backward procedures.

5.2.1 IWSIM Forward Procedure

The IWSIM first uses the one level IWT to decompose the secret image into four sub-bands: approximate LL, horizontal HL, vertical LH, and diagonal HH. The number of different coefficients (C) present in the high frequency sub-bands are no longer guaranteed to be ≤ 256, and the range of coefficient values are no longer in the range [0..255]. For the A sub-band, IWSIM simply applies the SIM procedure, but for the other sub-bands, IWSIM requires two different procedures, IWSIM1 and IWSIM2. IWSIM1 is applicable when $C \leq 256$ and in this case the SIM Table 5-1 is used but with different side information to cater for coefficients > 255. The IWSIM2 is applicable to sub-bands for which $C > 256$. In this case, a new table is designed in the same way as in Table 5-1, but representing values in the range 0 to 511 arranged in ascending order of the number of 1s in their binary representation. This table is rather long to be included in this chapter but will be shown as an appendix. In IWSIM1, 8 bits are sufficient to represent each coefficient value even if >255, while in IWSIM2, 9 bits are sufficient. This compact representation of coefficients >255 is based on the fact that $C \leq 512$ and the two mapping tables facilitate this as described below.

The IWSIM algorithm

Apply the IWT on the secret image I, and for the LL sub-band call the SIM procedure. For each other sub-band S follow the steps below:

1. Calculate m = min (S).
2. Let S' be the modified of S after subtracting m from all coefficient values in the S.
3. Compute the histogram h of S'.
4. Sort h in descending order of coefficient value frequency, and denote the sorted version by h'.
5. Determine the number of different coefficient values C appearing in S'.
6. If ($C \leq 256$), then call IWSIM1 procedure

Else call IWSIM2 procedure.

Note that the side information needs to initiate with a 1 bit indicator, where 0 indicate IWSIM1 is used and 1 to indicate that IWSIM2 is used.

7. Based on h', replace the first highest repeated coefficient value in the S' with the first new grayscale value. This step is continued by replacing the next highest repeated coefficient value with the next new suggested value, till all coefficient values in the $S'$are replaced. This yields a new sub-band S''.
8. Covert S'' into binary to create the secret bit-stream.
9. Append the side information bit-stream constructed by the appropriate procedure (described below).
10. Append the secret bit-stream to the side information.

IWSIM1 side information construction:

1- Append 8 bits to represent m .

2- Append 2 bits as an indicator of how many bits are needed to represent each coefficient in S': 00, 01, or 10 indicate that 8, 9, or 10 bits are needed to represent each coefficient value.

The reason of having only three cases is that the maximum coefficient value in S' does not exceed 1023, but in some sub-bands it does not exceed 255 or 511.

3- Append 9 bits to represent C.

4- Append C * (8 or 9 or 10) bits to list the S' values in descending order of frequencies.

Note that, the maximum possible number of bits for the side information is (8 + 2 + 9 + (256 * 10) = 2579).

IWSIM2 side information construction:

1- Append 8 bits to represent m .

2- Append 1 bits as an indicator of how many bits are needed to represent each coefficient in S': 0 indicates that 9 bits are needed while 1 indicates that 10 bits are needed.

The reason of having only two cases is that the maximum coefficient value in S' does not exceed 1023 but in some sub-bands it does not exceed 511.

3- Append 10 bits to represent C.

4- Append C * (9 or 10) bits to list the S' values in descending order of frequencies.

Note that, the maximum possible number of bits for the side information is (8 + 1 + 10 + (512 * 10) = 5139).

5.2.2 IWSIM Backward Procedure

After extracting the secret, in according with the embedding scheme, the receiver knows that this secret is not the image but it is made of 4 parts each representing the sub-band of the IWT decomposed secret image. The first part represents the LL sub-band processed by SIM. Hence, the LL sub-band can retrieve by the procedure described in the last section. The rest represent the other sub-bands in the order HL, LH and then HH which could be reversed by the procedure below, one by one. Each bit-stream of these sub-bands contains the following three parts: The first bit is an indicator that indicates whether IWSIM1 or IWSIM2 is applied. The second part is the side information consisting of: 8 bits for value of m, 2 bits or 1 bit indicating the number of bits needed to represent coefficients in S', 9 or 10 bits to give the size of C, followed by the distinct values in S' in descending order of their frequencies. The size of the last part is determined depends on the previous sub-stream as well as the first bit. The third part is then the modified sub-band S''. Then, the original sub-band S can be reconstructed from the received sub-band S'' by the following steps:

1- Calculate the histogram h' of S''. Note that $h'$is already sorted in descending order.

2- Construct the sub-band S' by replacing the coefficient values in S'' that has the ith value in the h' with the ith value of the reconstructed original coefficient values from the side information.

3- Add m to all coefficient values of S', to retrieve the original sub-band S .

Finally, after extracting LL, HL, LH, and HH, apply the inverse of the IWT to reconstruct the original secret image I.

5.2.3 Performance of IWSIM

To test the performance of the IWSIM algorithm in terms of ratio 0:1 of bits in the secret image bit-stream before and after modification, the same databases and image sizes in which used for testing SIM are used.

Experimental Results

Results of the experiments conducted for three different image sizes are shown in Table 5-5 and Table 5-6 for the SIPI and BOSSBase databases, respectively. As in the case of SIM, these tables include statistical parameters (mean μ, standard deviation σ, minimum M_n, and maximum M_x) of the ratios of 0:1 before and after IWSIM modification as well as length of side information and resulted sub-band bit-stream over all images in the respective database. Here R refers to the ratio of 0:1 before IWSIM, R' refers to the ratio of 0:1 post IWSIM, Ls refers to the length of side information needed, and L refers to the number of bits of the bit-stream that represent the secret image after IWSIM is applied.

Table 5-5: SIPI database - Ratio of 0:1 in the secret images and IWSIM modified secret images $\boldsymbol{I'}$.

	Image size 128 x 256				Image size 256 x 256				Image size 512 x 512			
	$\boldsymbol{R}$	$\boldsymbol{R'}$	$\boldsymbol{Ls}$	$\boldsymbol{L}$	$\boldsymbol{R}$	$\boldsymbol{R'}$	$\boldsymbol{Ls}$	$\boldsymbol{L}$	$\boldsymbol{R}$	$\boldsymbol{R'}$	$\boldsymbol{Ls}$	$\boldsymbol{L}$
μ	0.49	0.81	5171	262703	0.49	0.81	5358	525405	0.49	0.84	5238	2113536
σ	0.08	0.06	1590	2736	0.09	0.06	1607	5471	0.10	0.06	2272	34980
$\boldsymbol{M_n}$	0.12	0.73	1188	262144	0.12	0.73	1268	524288	0.11	0.76	334	2097152
$\boldsymbol{M_x}$	0.65	0.99	11054	278528	0.66	1.00	10629	557056	0.66	0.99	11212	2228224

Table 5-6: BOSSBase database - Ratio of 0:1 in the secret images and IWSIM modified secret images $\boldsymbol{I'}$.

	Image size 128 x 256				Image size 256 x 256				Image size 512 x 512			
	$\boldsymbol{R}$	$\boldsymbol{R'}$	$\boldsymbol{Ls}$	$\boldsymbol{L}$	$\boldsymbol{R}$	$\boldsymbol{R'}$	$\boldsymbol{Ls}$	$\boldsymbol{L}$	$\boldsymbol{R}$	$\boldsymbol{R'}$	$\boldsymbol{Ls}$	$\boldsymbol{L}$
μ	0.54	0.80	5277	262341	0.54	0.81	5829	525418	0.54	0.83	6781	2124153
σ	0.07	0.03	1286	1307	0.07	0.03	1432	5032	0.07	0.04	1781	54159
$\boldsymbol{M_n}$	0.17	0.70	1220	262144	0.16	0.72	1644	524288	0.15	0.74	1756	2097152
$\boldsymbol{M_x}$	0.85	0.94	9081	278528	0.86	0.95	10039	573440	0.86	0.97	12015	2293760

From Table 5-5, we note that on average the ratio of 0:1 of the IWSIM modified images is increased by about 66% of the corresponding ratio for the original images. Similarly, the results of Table 5-6 show an increase of 49% in the ratio of 0:1 post IWSIM. Again, the difference between the percentages of changed ratio reflects the variation in the nature of images in the two databases. As in the case of SIM, the statistical parameters (μ and σ) in Table 5-6 are constant and independent of image size, while is not the case in Table 5-5. Resizing seems to explain the minor effects on the ratios of 0:1 in the original SIPI images. These results also demonstrate that the proposed IWSIM algorithm can be used for any secret image sizes.

Again, there are two drawbacks of the proposed IWSIM algorithm; the first one is that the side information that need to be sent to the receiver and this results in slightly decreasing the capacity of any adopted steganography technique. Table 5-7 shows the average percentage of the number of bits, for the side information in proportion to the number of bits that represent the secret images for both used SIPI and BOSSBase databases. Clearly bigger size images require a lower ratio of side information to the actual secret image size.

Table 5-7: Ratio of bits of the side information using IWSIM.

Image sizes	SIPI	BOSSBase
128 x 256	0.020	0.020
256 x 256	0.010	0.011
512 x 512	0.002	0.003

The second drawback is for those sub-bands for which IWSIM2 is applied; the number of bits that represent the sub-band is increased. This increment happened because, in the case of using IWSIM2, each coefficient value needs 9 bits to represent in binary form, and this leads to reducing the capacity slightly. Table 5-8 shows the average percentage of the number of increased bits to represent all sub-bands in proportion to the number of bits that represent the secret images for both used SIPI and BOSSBase databases.

Table 5-8: Ratio of increased bits to represent the modified sub-bands.

Image sizes	SIPI	BOSSBase
128 x 256	0.002	0.001
256 x 256	0.002	0.002
512 x 512	0.008	0.013

The combined effect of these two drawbacks of IWSIM is a negligible reduction in embedding capacity of any embedding scheme to an estimate of (1 – (0.020 + 0.002) = 0.978 for the SIPI database) and (1 – (0.020 + 0.001) = 0.979 for the BOSSBase database) of the actual capacity of the embedding scheme.

To compare the performance of the IWSIM with that of the SIM, we present in Figure 5-3 the ratio of 0:1 for images size 128 x 256 in both SIPI and BOSSBase databases. Figure 5-3, reveals that the both SIM and IWSIM algorithms increase the ratios by significant percentages, but the IWSIM procedure significantly outperforms the SIM. However, from Figure 5-4, it is clear that the capacity limitation of the IWSIM is slightly more than the capacity limitation of the SIM.

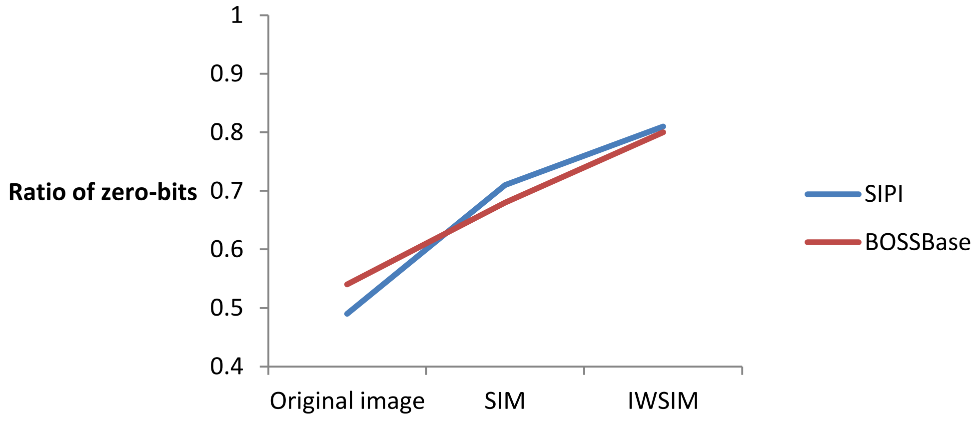


Figure 5-3: Ratio of zero-bits of SIM and IWSIM.

Figure 5-4: Ratio of side information bits of SIM and IWSIM.

5.3 Secret Image Size Reduction (SISR) algorithm

In the last two sections, we developed two schemes that increase the ratio of 0:1 in the secret image bit-stream without changing the size of the secret. As mentioned earlier, the SIM and IWSIM tables are organised and used seem to be fitting for use for compressing the size of the secret. Reducing the secret bit-stream length while increasing the ratio of 0:1 in the shorter bit-stream can provide opportunities for improving image quality. The SISR is our new spatial domain encoding algorithm designed to achieve reduced secret length without loss of information and high 0:1 ratio (Abdulla, et al., 2014). SISR aims primarily to improve stego-image quality but may also increase embedding efficiency. The SISR algorithm is a block based, and the image is first partitioned into non-overlapping blocks of equal sizes. In our experiments, the SISR algorithm is applied on blocks from both the original secret image I and its complement version I_C (i.e. the negative of I). The selected secret image bit-stream is the one that achieves highest 0:1 ratio. In this case, the side information is a single bit to inform the receiver about the source of the selected bit-stream (0 for I and 1 for I_C). The encoding and decoding steps for the SISR algorithm, applied to both I and I_C, are explained in the Section 5.3.1 and 5.3.2, respectively.

5.3.1 SISR Encoding Procedure

The encoding steps for the SISR algorithm work as follows:

A. Partition the secret image I into non-overlapping blocks of size $A \times A$. Here we take A=4, 8, or 16.

B. For each block B_{ij} , i, j ϵ {1,..., A}, do the following steps:

1. Let m = min (B_{ij}), and let i*, j* be the indices of the element in B_{ij} achieving m with smallest j, then smallest i.
2. Let $D_{ij} = B_{ij} - m$, be the difference between each pixel and m.
3. Set $D_{max} = \max_{i\,j} (D_{ij})$, be the maximum difference value.
4. Let T be a set of possible thresholds to determine the number of bits that represent D_{ij}.

$$T = \{2^n - 1 \mid 0 \leq n \leq 8\}$$

5. Let $t \equiv t_{ij} = \min(z)$, where $z \in T$ and $z \geq D_{max}$. Here, t is the smallest element in T which is $\geq D_{max}$.
6. Encode each block as follows:
   a) If t = 255, record 1 and then append the original 8-bit pixel values in the given order, i.e. total number of bits representing such a block is increased by 1 to (1 + 8 × ($A \times A$)) bits.
   b) Else record 0, append the 8-bit value of m, and do:

If t = 0, (i.e. if all pixel values in the block are equal), then append 3-bit representation of t and stop. In such a case, only 12 bits are needed.

Else

Append 3-bit representation of t.

Append $\log_2 (A \times A)$ bits for position of m

Append the bit representations of the pixels' differences $D_{i\,j}$ in the given order.

In this case, the block requires: $(1 + 8 + 3 + \log_2 (A \times A) + ((A \times A) - 1) \times \log_2 (t+1))$ bits.

Note that the 3-bit representations are arranged based on frequently occurring t values. The four most frequently occurring t values are represented by 3 bits with two or more 0s. While the three remaining 3-bit strings represent the three least frequent t values. For example, the 3-bit that has a higher number of 1s, namely 111, represent the t = 255 which is least frequent in the blocks. This arrangement of 3-bit representation is a factor for increasing the ratio of 0:1 in the SISR bit-stream. Finally, the sender sends a bit-stream that represent either the image I or I_C, the bit-stream is the concatenation of all blocks sub-streams.

Note that the 8 bits that represent m and the 3 bits that indicate the value of t is needed when the image pixel value is between 0 and 255. It is also possible to extend SISR algorithm for different pixel value ranges, for example if the image pixel value is between 0 to 511, we can extend the algorithm to be applicable by representing the m in 9 bits instead of 8 bits and the value of t in 4 bits instead 3 bits.

Table 5-9, illustrates the number of obtained bits and the number of reduced bits depending on the value of t for 4x4 block of pixels. It is clear that only in the case t= 255, the SISR procedure increases the number of bits by 1; otherwise, the algorithm reduces the number of bits to represent the block of 16 pixels. Although the main focus in this thesis of proposing SISR algorithm is to reduce the secret image size prior to embedding, but this can also use to reduce the required image storage in a lossless way.

Table 5-9: Number of obtained bits from proposed SISR algorithm for block size 4x4.

t	Number of bits		
	Obtained	**Original**	**Reducing**
0	12	128	116
1	31	128	97
3	46	128	82
7	61	128	67
15	76	128	52
31	91	128	37
63	106	128	22
127	121	128	7
255	129	128	-1

5.3.2 SISR Decoding Procedure

This procedure receives as input the blocks sub-streams b_i, all concatenated in one bit-stream. For each b_i, follow the steps below:

If $b_i[1] = 1$, convert the remaining bits in the sub-stream b_i to a decimal by taking 8 bits at a time.

Else:

1. m= decimal (b_i[2..9]) and t= decimal (b_i[10..12]).
2. s=$\log_2$(t+1).
3. Index of m = decimal (b_i[13 ..13+ $\log_2 (A \times A)$ -1]).
4. D_{ij}= be the matrix of decimals obtained by converting each s bits into decimals, starting with the first s-bits of the remainder of b_i until the last s-bits.
5. $B_{ij} = D_{ij} + m$.

Assembling all the reconstructed blocks will either produce the original secret image or its complement. This is determined by the 1-bit side information.

5.3.3 Example Application of SISR Algorithm

The example 4x4 block of pixel intensities in Table 5-10 will be used to illustrate the SISR encoding and decoding steps.

Table 5-10: Block of 16 pixels

30	25	26	35
35	22	29	28
31	24	22	29
30	34	32	30

SISR Encoding steps

1. The minimum pixel value is 22 (00010110 in 8 bits), and its index is 5 (0101 in 4 bits).

2. The differences between pixels value and the minimum pixel value are presented in Table 5-11.

3. The maximum value of the subtraction, see 3rd column of the Table 5-11, is 13.

4. The nearest value in set T that should be equal or greater than the maximum value of the subtraction, which is 13, is 15 (i.e. t= 15).

5. Now this stream of bits represents the block of Table 5-10:

The first bit 0 indicates that the algorithm has been done on the block, i.e. t is not equal to 255.

The 8-bits 00010110 represent the value of m, and the 3-bits 100 is the value of t.

The next 4-bits 0101 represent the index of the minimum pixel value.

The bits 1000, 0011, 0100, 1101, 1101, 0111, 0110, 1001, 0010, 0000, 0111, 1000, 1100, 1010, 1000 represent the difference values (see 3rd column of Table 5-11) 8, 3, 4, 13, 13, 7, 6, 9, 2, 0, 7, 8, 12, 10, 8, respectively.

Therefore, the 4x4 block of Table 5-10 is represented by the following 76-bits instead of the original 128-bits

b_i =(0 00010110 100 0101 1000 0011 0100 1101 1101 0111 0110 1001 0010 0000 0111 1000 1100 1010 1000).

Table 5-11: Differences between pixels value and minimum pixel value.

B_{ij}	m	D_{ij}	Binary of D_{ij}
30	22	8	1000
25	22	3	0011
26	22	4	0100
35	22	13	1101
35	22	13	1101
29	22	7	0111
28	22	6	0110
31	22	9	1001
24	22	2	0010
22	22	0	0000
29	22	7	0111
30	22	8	1000
34	22	12	1100
32	22	10	1010
30	22	8	1000

In Table 5-11, the 1st column B_{ij} displays the pixel values of Table 5-10 (excluding the minimum value 22). The 2nd column is the minimum pixel value m, and the 3rd column, D_{ij} is the subtraction of the minimum pixel value from each pixel value. The 4th column represents D_{ij} in binary form.

SISR Decoding steps

From the received bit-stream b_i, the original 4x4 block of pixels can be recovered as follows:

1. Take the first bit; which is 0, and then go to the next step.
2. Convert the next 8 bits into decimal, which is 22, that represents the m.
3. The next 3 bits, 100, represent the value of t, i.e. t=15.
4. The next 4 bits, 0101, represent the index of m.
5. Since t= 15, take each next 4 bits 15 times and convert them into decimal to represent D_{ij} as illustrated in Table 5-12.
6. Add m to D_{ij}, then original pixels B_{ij} are obtained (see 4th column of Table 5-12).
7. Sequentially insert each value in the 4th column of Table 5-12 to its position in the block; the block in Table 5-10 is recovered exactly as it is.

Table 5-12: Producing original pixels value from the recovered $\boldsymbol{D_{ij}}$.

b	D_{ij}	m	B_{ij}
1000	8	22	30
0011	3	22	25
0100	4	22	26
1101	13	22	35
1101	13	22	35
0111	7	22	29
0110	6	22	28
1001	9	22	31
0010	2	22	24
0000	0	22	22
0111	7	22	29
1000	8	22	30
1100	12	22	34
1010	10	22	32
1000	8	22	30

5.3.4 Performance of SISR

In this section, the same evaluation protocol used for SIM and IWSIM algorithms is used to evaluate the performance of the SISR algorithm. In other words, both SIPI and BOSSBase databases with three different image sizes, 128 x 256, 256 x 256, and 512 x 512 are used in the experiments.

First of all, the effect of using the image complement on the ratio of 0:1 in the SISR 4x4 block bit-stream are tested, and the results are presented in Table 5-13 for the SIPI and BOSSBase databases. Rz and Rz'refer to the average 0:1 ratio of SISR bits for the original secret image I and their complement I_C , respectively. Moreover, max($Rz,\ Rz'$) refers to the 0:1 ratio of the selected image version. During our experiments, we observed that the reduction ratio RR values for the SISR with original images and their complements are the same, but they have different 0:1 ratio. We note that for both databases, the SISR results in the same average of 0.57 for all image sizes, and this is more than the actual maximum of both which could be explained by the fact for each secret image the procedure selects the best individually. In fact, these results justify the processing of both versions of secret images. From previous sections, the original average 0:1 ratio was 0.49 for SIPI and 0.54 for BOSSBase which means that SISR perform better on the images in SIPI. This can be explained by the huge variations of structures in the BOSSBase database.

Table 5-13: Average of 0:1 ratio before and after applying the SISR for 4x4 block size.

SIPI database									
	Image size 128 x 256			**Image size 256 x 256**			**Image size 512 x 512**		
	Rz	Rz'	max(Rz,Rz')	Rz	Rz'	max(Rz,Rz')	Rz	Rz'	max(Rz,Rz')
μ	0.55	0.56	0.57	0.54	0.56	0.57	0.55	0.56	0.57
σ	0.04	0.03	0.03	0.04	0.03	0.03	0.05	0.04	0.03
M_n	0.35	0.52	0.55	0.34	0.47	0.53	0.25	0.45	0.51
M_x	0.60	0.71	0.71	0.59	0.71	0.71	0.64	0.72	0.72
BOSSBase database									
μ	0.57	0.55	0.57	0.56	0.55	0.57	0.56	0.55	0.57
σ	0.02	0.02	0.01	0.02	0.02	0.01	0.01	0.01	0.01
M_n	0.49	0.46	0.54	0.48	0.46	0.55	0.47	0.45	0.54
M_x	0.65	0.67	0.67	0.64	0.66	0.66	0.64	0.65	0.65

Table 5-14 show the results of the experiments conducted to test the performance of the SISR, on the images in SIPI and BOSSBase databases, in terms of the 0:1 ratio for the three different image sizes and the three different block sizes. Again, the results demonstrate that for both SIPI and BOSSBase databases and all block sizes, the SISR algorithm produces a higher 0:1 ratio than in the original secret image bit-streams. However, it is clear that the 4x4 SISR has better performance than other block sizes. This can be seen by comparing the standard deviations for the different block sizes and image sizes.

Table 5-14: Ratio 0:1 SISR algorithm for different block sizes.

	Image size 128 x 256			**Image size 256 x 256**			**Image size 512 x 512**		
SIPI database									
	4x4	**8x8**	**16x16**	**4x4**	**8x8**	**16x16**	**4x4**	**8x8**	**16x16**
μ	0.57	0.57	0.56	0.57	0.57	0.56	0.57	0.57	0.57
σ	0.03	0.04	0.06	0.03	0.04	0.05	0.03	0.05	0.06
M_n	0.55	0.52	0.51	0.53	0.53	0.51	0.51	0.53	0.52
M_x	0.71	0.77	0.82	0.71	0.75	0.79	0.72	0.84	0.88
BOSSBase database									
μ	0.57	0.57	0.56	0.57	0.57	0.56	0.57	0.56	0.56
σ	0.01	0.02	0.03	0.01	0.02	0.02	0.01	0.01	0.02
M_n	0.54	0.52	0.51	0.55	0.52	0.51	0.54	0.52	0.52
M_x	0.67	0.71	0.76	0.66	0.70	0.75	0.65	0.67	0.72

In comparison to SIM and IWSIM, the average 0:1 ratio achieved by the SISR is certainly lower than that achieved by the other two schemes (approximately 0.73 and 0.80). However, SISR also reduces the secret image size bit-streams, without losing information. Hence, the comparison needs to take into account the number of 0s relative to the size of the original secret image bit-stream. But first we need to determine the extent to which SISR compresses secret images.

To evaluate the reduction efficiency (i.e. compression) of SISR algorithm, we use the following reduction ratio measure:

$$\mathrm{RR} = \frac{Total\ size\ in\ bits\ of\ the\ obtained\ bitstream}{Total\ size\ in\ bits\ of\ the\ input\ image} \qquad (5.\ 2)$$

Table 5-15, shows the average RR after applying the SISR algorithm for three different block sizes (4 x 4, 8 x 8, and 16 x 16) on the databases SIPI and BOSSBase of images for three different sizes (128 x 256, 256 x 256, and 512 x 512). These results show that the best RR is achieved with 4x4 blocks. In other words, the smaller the block size used in the proposed SISR, the better RR is, and this is what we should expect because in small blocks pixels; values are more homogeneous, and this results in lower number of bits needed to represent such block of pixels. Furthermore, the bigger the image size, the better RR value is. Note that, the lower RR is, the better reduction in 0:1 ratio when the original image size is taken into account.

Table 5-15: Average RRs for SISR algorithm for different image and block sizes.

	Image size 128 x 256			Image size 256 x 256			Image size 512 x 512		
	4x4	**8x8**	**16x16**	**4x4**	**8x8**	**16x16**	**4x4**	**8x8**	**16x16**
SIPI database									
μ	0.70	0.75	0.84	0.67	0.73	0.81	0.63	0.67	0.76
σ	0.14	0.14	0.11	0.14	0.15	0.13	0.16	0.16	0.16
M_n	0.21	0.25	0.44	0.17	0.16	0.28	0.13	0.10	0.15
M_x	0.92	0.97	1.00	0.86	0.96	0.98	0.85	0.88	1.00
BOSSBase database									
μ	0.68	0.74	0.82	0.66	0.71	0.79	0.63	0.67	0.74
σ	0.09	0.09	0.09	0.10	0.10	0.09	0.10	0.10	0.10
M_n	0.29	0.27	0.31	0.27	0.27	0.30	0.27	0.25	0.29
M_x	0.92	0.96	1.00	0.90	0.96	0.99	0.86	0.93	0.99

Since SISR does not lead to loss of information, then it acts as a lossless compression. However, SISR aims differ from general lossless image compression systems, because SISR not only reduce the number of bits to represent the image but also results in bit-streams with higher 0:1 ratio. Nevertheless, we shall now compare its performance against three standard lossless image compression techniques: Run Length Encoding (RLE), Huffman, and Lempel-Ziv-Welch (LZW). For details see (Gonzalez & Woods, 2002). Table 5-16, shows the average RR for our block size 4x4 SISR against the RLE, Huffman, and LZW.

Table 5-16: Average RRs for SISR, RLE, Huffman, and LZW for different image sizes.

	Databases					
	SIPI			**BOSSBase**		
	Image size 128 x 256	**Image size 256 x 256**	**Image size 512 x 512**	**Image size 128 x 256**	**Image size 256 x 256**	**Image size 512 x 512**
SISR	0.70	0.67	0.63	0.68	0.66	0.63
RLE	1.22	1.25	1.22	1.29	1.31	1.33
Huffman	0.83	0.83	0.79	0.88	0.88	0.88
LZW	0.90	0.95	0.94	0.99	1.05	1.08

It is clear that the reduction ratio RR of the SISR is significantly improved compared to the RLE, Huffman, and LZW. Note that the RR values achieved by the RLE technique is very high, because some images may have less repetitions of neighbouring pixel values, and this could increase the size of the next value indicators. This also happens sometime in the case of Huffman and LZW techniques.

Earlier we have seen that SISR performance on the 0:1 ratio is much lower than what was achieved by SIM and IWSIM. However, such a comparison does not take into account the combined effect of reduction in bit-stream size as a result of achieving 0.7 RR and 0.57 of 0:1 ratio. When we take these two factors into account, then the number of 0s produced by SISR would be equivalent to getting 0.8125 (0.57/0.7) 0s out of the original image bit-stream size.

Although, computation time is not an issue in steganography, we shall compare the time cost of the SISR, RLE, Huffman, and LZW in Table 5-17. These averages are measured in seconds. It is clear that the only drawback of the proposed SISR is time consumption compared to Huffman, and LZW, but not with RLE.

Table 5-17: Average time cost for SISR, RLE, Huffman, and LZW for different image sizes.

	Databases					
	SIPI			**BOSSBase**		
	Image size 128 x 256	**Image size 256 x 256**	**Image size 512 x 512**	**Image size 128 x 256**	**Image size 256 x 256**	**Image size 512 x 512**
SISR	3.57	7.11	27.53	3.46	6.88	29.17
RLE	5.77	11.70	43.53	5.28	10.78	45.53
Huffman	3.42	6.70	24.71	2.73	5.24	20.60
LZW	3.15	6.51	25.13	3.18	6.56	27.85

5.4 Performance of Fibonacci-Mapping based scheme post SIM, IWSIM, and SISR

In this section, we test and evaluate the effectiveness of each of the proposed pre-processing algorithms SIM, IWSIM, and SISR, on the steganography requirements when the resulting secret image bit-streams are embedded into cover images using the Fibonacci-Mapping based embedding technique that was proposed in Chapter 4. In line with the experimental setup of Chapter 4, we use each of the 44 SIPI size 128 x 256 images as a secret image, and we use the same 44 images but of size 512 x 512 as cover images. First we apply the three pre-processing schemes on each secret image and use the output bit-streams, separately, for embedding into each cover image using the Fibonacci-Mapping based scheme at 5 payloads, namely (20%, 40%,60%, 80%, 100%) of the size of the resulting bit stream. Note that the SISR bit-stream is shorter than those output by SIM and IWSM. For each payload, then we have a total of (44 x 44 = 1936) stego-images. The various performance factors will be compared to that of Fibonacci-Mapping based when the original secret bit-stream are embedded at these 5 payloads. Note that, in the following experiments, *Mapping-based* refers to embedding the secret image (without pre-processing), while *Mapping-based-SIM, Mapping-based-IWSIM*, and *Mapping-based-SISR* refers to embedding of the resulting bit-stream after the SIM, IWSIM, and SISR algorithm is applied on the secret image, respectively.

Three performance factors are measured in these experiments. The first is to measure the embedding efficiency, the second is to test the stego-image quality, and the third measure is the detectability/security (i.e. robustness against the targeted steganalysis tools) of the embedded message. Note that in the case of embedding the SIM bit-stream, the average capacity is 0.994 (since 0.006 of bits are needed for the side information), and in the case of embedding IWSIM bit-stream, the average capacity is 0.978.

1. *Embedding Efficiency Evaluation*

Figure 5-5 and Figure 5-6 presents the average value of the ratio of modified pixels to the size of the secret message and the embedding efficiency for Fibonacci-Mapping based embedding techniques.

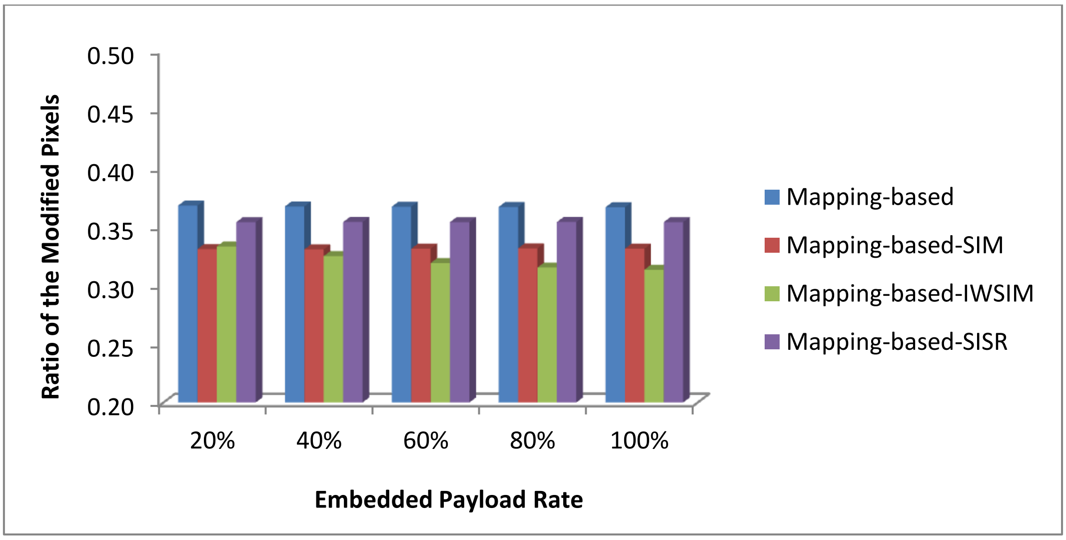


Figure 5-5: Ratio of modified pixels for the Fibonacci-Mapping based techniques.

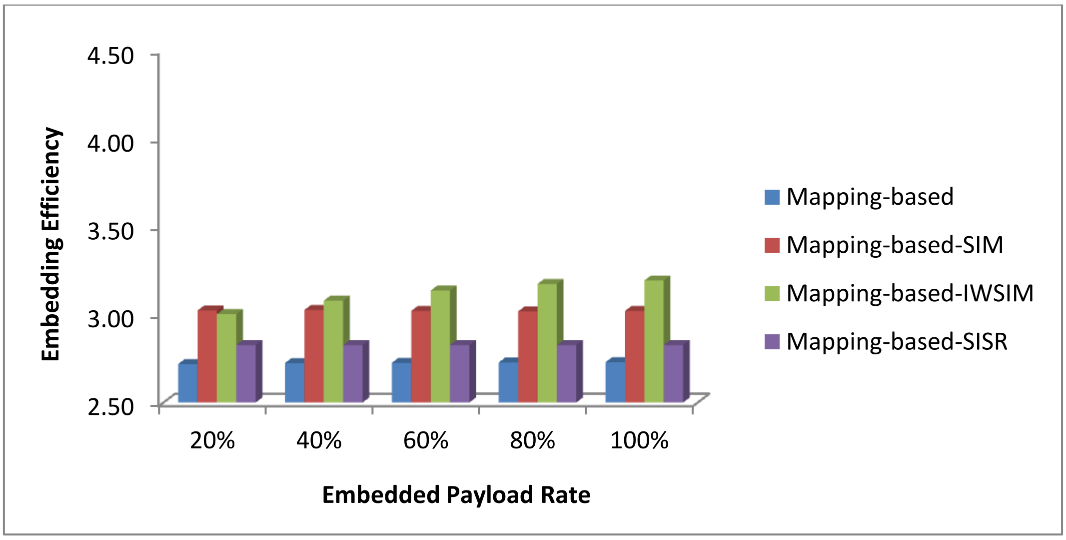


Figure 5-6: Embedding Efficiency for the Fibonacci-Mapping based techniques.

From these two figures, it is noticeable that for all payloads, all the proposed pre-processing schemes yield lower ratio of changed cover pixels and higher embedding efficiency than what is achieved by the original Mapping-based scheme that have no pre-processing. The best performance is achieved by the Mapping-based-IWSIM scheme. This is obviously due to the high 0:1 ratio achieved by these pre-processing schemes. In addition, for all embedding rates, SIM provides the same embedding efficiency, while the embedding efficiency of the IWSIM algorithm increases as the embedding rates increases. This can be explained by the effect of the side information on the efficiency value. In both cases, the side information does not necessarily have the same property on 0:1 ratio as their actual bit-stream. Moreover, due to the fact that the side information of the IWSIM is almost 4 times that of the SIM, the proportion of embedded side information to the embedded actual bit-stream decreases faster for IWSIM than for the SIM, as we increased the payload.

2. Stego-Image Quality Evaluation

To determine the effect of each of the pre-processing schemes on stego-image quality at each payload rate, we computed the PSNR of the stego-images relative to their source cover image for all the embedded bit-streams of the 44 secret images in SIPI. Here we tested all the stego-images obtained from all the 44 SIPI cover images, Figure 5-7 presents the average value of the PSNR of all the tested Fibonacci-Mapping-based embedding techniques at different embedding payloads.

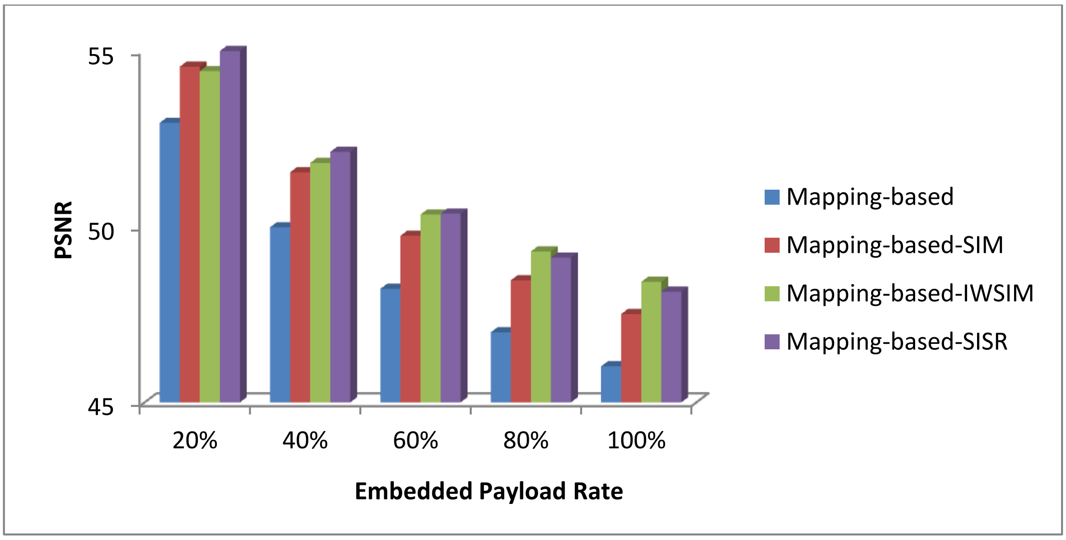


Figure 5-7:The PSNR for the Fibonacci-Mapping based techniques.

For all payloads, embedding the original secret image without pre-processing the secret bit-streams yields the lower average PSNR than that achieved by all the three pre-processing algorithms, IWSIM being the best performing scheme on stego-image quality at the high payloads of 80% and 100%, whereas the SISR perform better at low payloads and at 20% payload SIM has higher PSNR than the IWSIM. This can be explained by the effect of the embedded side information on the PSNR value. The SISR has only 1 bit side information, whereas SIM and SIWSIM side information consist of 0.006 and 0.022 proportion to the actual bit-stream size, respectively. In all cases, the side information does not necessarily have the same property on 0:1 ratio as their actual bit-stream. Moreover, the proportion of embedded side information to the embedded actual bit-stream decreases faster for IWSIM than for the SIM, as we increased the payload.

3. *Detectability Evaluation*

In this section, we evaluate the robustness of all the pre-processed based schemes against the three targeted RS, DIH, and RWS detectors at all payload rates in comparison. We note that the original scheme which does not pre-process the secret bit-stream was shown to be robust against these schemes.

Robustness Against RS Detector

Figure 5-8, Figure 5-9, and Figure 5-10 displays the RS diagram for the Mapping-based-SIM, Mapping-based-IWSIM, and Mapping-based-SISR embedding techniques, from which it is clear that for all embedding techniques and for all payload, there are hardly any differences between RM and RM-, SM and SM-, demonstrating the robustness of the proposed schemes against the RS detector. However, one can notice that the SISR has slightly better robustness. Recall that, the Fibonacci-Mapping based embedding technique in Chapter 4 is also robust against RS detector, see Figure 4-11.

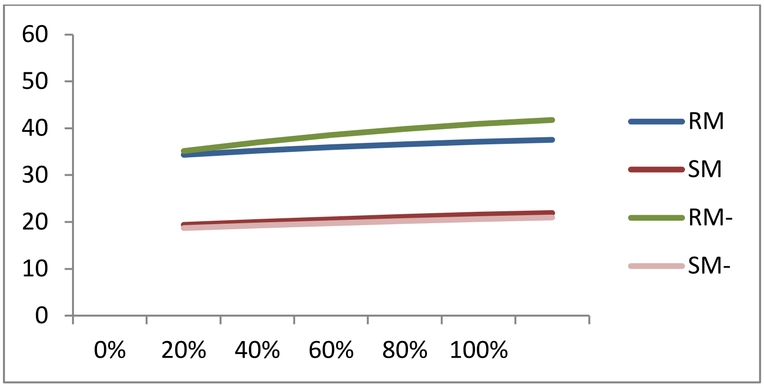


Figure 5-8: RS diagram for Mapping-based-SIM.

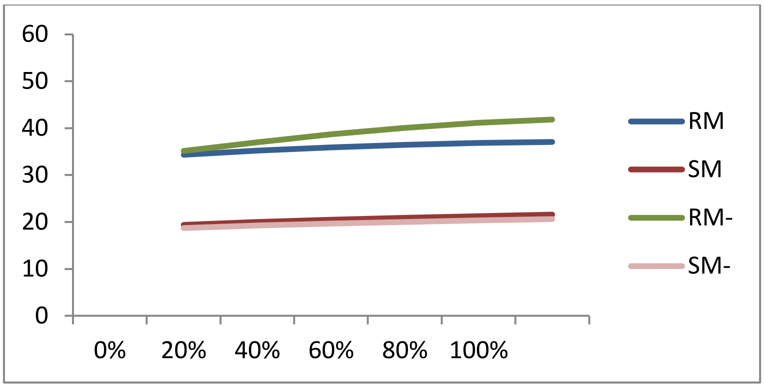


Figure 5-9: RS diagram for Mapping-based-IWSIM.

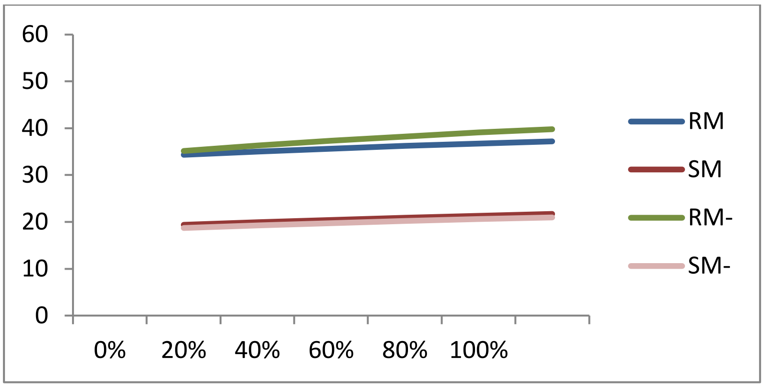


Figure 5-10: RS diagram for Mapping-based-SISR.

Robustness Against DIH Detector

For each embedding ratio, the chart of Figure 5-11 presents the average probability of having a secret hidden with the given embedding payload. From Figure 5-11, we see that embedding the secret images using all Fibonacci-Mapping based embedding techniques are robust against the DIH.

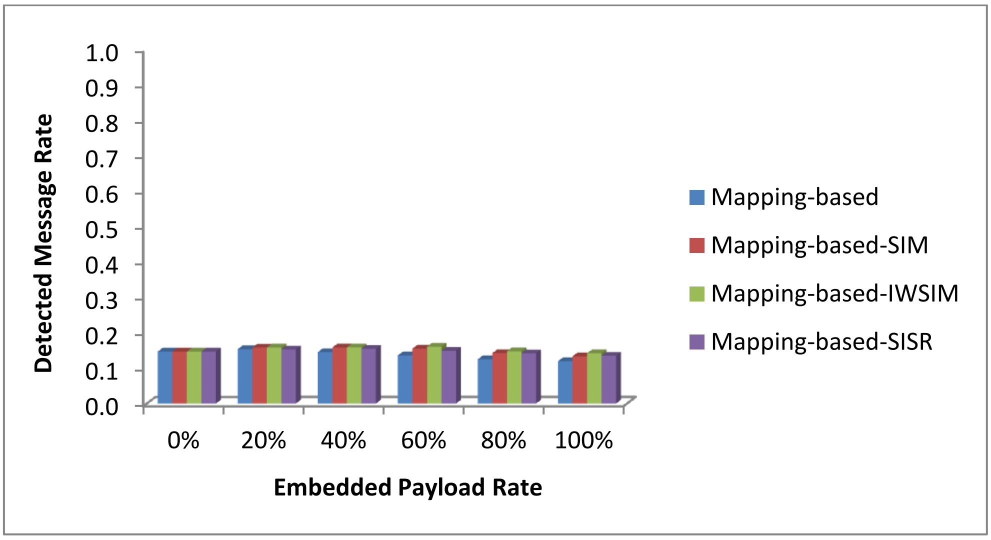


Figure 5-11: DIH steganalysis for the Fibonacci-Mapping based techniques.

Robustness Against RWS Detector

Figure 5-12 presents the average values of the estimation ratios of the flipped cover pixels' LSB of our Fibonacci-Mapping based embedding schemes at different embedding payloads. Figure 5-12 demonstrates the robustness of the all Fibonacci-Mapping based schemes against this steganalysis tool. As before, this is due to the fact that this scheme results in flipping the LSB of fewer cover pixels than the LSBR.

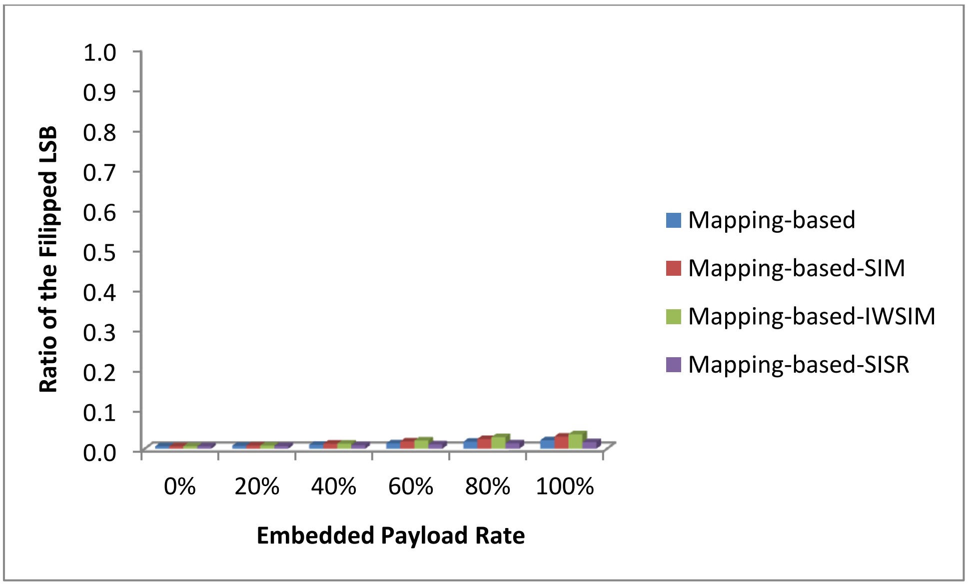


Figure 5-12:RWS steganalysis for the Fibonacci-Mapping based techniques.

5.5 Discussion

This chapter was devoted to pre-process the secret image prior to embedding with the aim of increasing the 0:1 ratio of the secret image bit-stream. This was a follow up on the conclusion made in Chapter 4 on the need to reduce the number of pixel changes post embedding in order to improve stego-images quality. It was realised that a potential solution is to increase similarities between the secret bit-stream and the LSB plane of the cover image. The work in this chapter was focused on the secret image side, and we developed three algorithms that encode the original secret image bit-stream into another bit-stream with significantly increased 0:1 ratio. The first two algorithms, SIM and IWSIM, are based on the similar strategy adopted in statistical coding by exploiting the structure of histograms of the secret image spatial domain and Integer Wavelet sub-bands, respectively. Both algorithms map secret pixel values (sub-band coefficients) according to the descending order of their frequencies so that more frequent values are mapped into bit-strings with the lower number of 1s. The third algorithm, SISR, is directly applied on the spatial domain of the secret image, by first reducing the range of values in blocks as result of subtracting the block’s minimum value and thereby reducing the number of bits needed to represent each block. We have demonstrated that the IWSIM provides highest 0:1 ratio (80% on average), outperforming both SIM and SISR. Our experiments demonstrated that embedding the resulting bit-stream from these three pre-processing algorithms into cover images using the proposed Fibonacci-Mapping based scheme in Chapter 4 result in gaining higher embedding efficiency with maintaining un-detectability, but the improvement in stego-image quality still falls short of our expectation. This is due to the fact that higher bit-planes are changed during secret embedding. Therefore, from now, we consider designing the steganography technique that embeds in only LSB plane (or avoid embedding in other than LSB). To overcome this challenge, in Chapter 7, we design a new Mapping-based embedding scheme that embeds one secret bit per cover pixel.

In addition, in the next chapter, we develop a new cover pixel value decomposition technique, called *Extended-Binary*, that has results in the cover image LSB plane having one of the highest 0:1 ratio among a variety of pixel decomposition schemes. Embedding a secret image bit-stream with higher 0:1 ratio, obtained by one of the above pre-processing schemes, into in the cover LSB plane that also has high 0:1 ratio would be expected to increase the probability of similarity between the secret bits and the cover pixels’ LSB bits. This strategy, discussed at the end of Chapter 4, aims to produce stego-images with minimal distortions by minimising the number of changed cover pixels post secret image embedding.

Chapter 6

Cover Pixel Value Decomposition Schemes

In order to increase the probability of similarity between the bits value in the secret image bit-stream and the cover pixels' LSB value, three proposed algorithms were presented in the last chapter that produce bit-streams with high 0:1 ratio. Those three algorithms are applied on the secret image prior to embedding. In this chapter, we turn our attention to the representation of the cover images' pixel values in order to realise the second part of the declared strategy of increasing similarity between the secret image bit-streams and the cover images LSB plane. In Chapter 4, we found that the Fibonacci pixel value decomposition of any image pixel result in eliminating all but one case of having more 1s than 0s within the lowest 3 bit-planes. In fact, there are only 5 possible 3-bit patterns in the 3 lowest significant planes and only 2 of which have 1 as the LSB value. Hence, in the current investigation we study existing decomposition techniques such as binary, Fibonacci, prime, natural, Lucas, and Catalan-Fibonacci (CF) in terms of the ratio of 0:1 in the cover pixels' LSB plane. All these methods extend the number of bit-planes beyond the 8 bit-planes of binary decomposition. But the inclusion of some odd numbers in their base sequences together with the restrictions that need to be imposed to guarantee unique decomposition (e.g. the Zeckendorf theory) may unintentionally increase the number of 0s in LSB plane. The ultimate objective of this chapter is not to introduce new and more decomposition schemes, but to see if there are other decomposition schemes that provide a higher ratio of 0:1 of LSB than existing schemes.

In Section 6.1, we describe the background of pixel value decomposition. In Section 6.2, we introduce a simple pixel value decomposition scheme that extending the number of bit-planes but has no effect on 0:1 ratio in the LSB, but it may useful for embedding in higher bit-planes. In Section 6.3, we introduce a new pixel value decomposition scheme, called the *Extended-Binary,* and demonstrate that it outperforms all the above schemes, except the natural scheme, in terms of 0:1 ratio in the LSB plane. In Section 6.4, we shall investigate the effect on the performance of the usual LSB embedding scheme when we combine the use of the *Extended-Binary* decomposition scheme to represent cover pixels' value with the 3 pre-processing algorithms to transform the secret images.

6.1 Background

In most spatial domain steganography schemes, grayscale cover images are in most cases decomposed into 8 bit-planes by expressing each pixel value in the range 0..255 as a binary linear sum of the sequence {1,2,4,8,16,32,64,128}. In recent years, few other pixel value decomposition techniques have been used, using different sequences to a different representation of cover images prior to embedding the secrets. A review of several non-binary decomposition techniques was conducted in Chapter 3 including Fibonacci (Picione, et al., 2006), prime (Dey, et al., 2007), natural (Dey, et al., 2007), Lucas (Alharbi, 2013), Catalan-Fibonacci (CF) (Aroukatos, et al., 2012). In general, these decomposition techniques are aimed to provide more bit-planes so that embedding in higher bit-planes do not lead to big changes in pixel values and thus has less impact on visibility in comparison to embedding in higher binary-decomposed cover images bit-planes. One could ask is the decomposition schemes can be exploited for different objectives in steganography.

The intensity values of the typical grayscale images range from 0 to 255 require 8 bits to represent them in binary, whereas Fibonacci, prime, Lucas, Catalan-Fibonacci, and natural representation require 12, 15, 12, 15, and 23 bits respectively. Unlike the binary decomposition technique, the non-binary decomposition techniques do not result in a unique bit-stream representation of pixel values. This problem is resolved by careful selection of a unique bit-stream for each grayscale value. For example, a unique Fibonacci representation is obtained by applying Zeckendorf's theorem, while for the other non-binary decomposition techniques uniqueness is imposed by selecting the bit-stream of lexicographically highest value. Examples of valid and non-valid representation were presented in Chapter 3 for all studied decomposition techniques. Consequently, all these decomposition techniques have a common capacity limitation in that not every cover pixel is suitable for embedding.

All these different decomposition schemes share similar objective and structure, and are based on using sequences of positive integers that are obtained by some interesting mathematical process. Here a question needs to be asked, if the choice of mathematically interesting sequences plays any unforeseen advantages, beyond increasing the number of image bit-plane that could be exploited in steganography? And if so, how strict, these processes need to be? Our interest, in relation to the first question, is related to our aim of using a representation of cover image pixels whose LSB plane has optimally high 0:1 ratio. Since the original steganography-related objective for using these different decomposition schemes was to increase the number of image bit-planes, another question arises as to whether the 0:1 ratio has a clear relationship, or not, to the increased number of bit-planes. In Section 6.2, we shall introduce a simple decomposition scheme, called the SS scheme, which results in 16 bit-planes (i.e. higher number of bit-planes than all but one of the existing schemes) but has the same 0:1 ratio as the binary scheme which is lower than all above decomposition schemes. Table 6-1 illustrates the number sequences for each decomposition techniques, including the simple one, their number of bit-planes and the corresponding weights of these bit-planes. The weight of a bit-plane is linked to the effect of changing the corresponding bit on image quality and hence is dependent on their element in the adopted decomposition sequence. In Section 6.3, we introduce and investigate the performance of a new decomposition scheme, called the *Extended-Binary*, obtained by a simple modification to the binary scheme resulting in 9 bit-planes only. For comparison, the above mentioned decomposition techniques will also be investigated and studied in terms of the ratio of 0:1 in cover image LSB plane in order to determine their suitability for our purpose. We shall demonstrate that there are noticeable variations in their performance in terms of 0:1 ratio, and that our Extended-Binary scheme outperforms all but the natural one. In this respect, we shall demonstrate that the ratio of 0:1 is influenced more by the composition of the adopted sequence and perhaps the frequencies of the odd pixel values that are missing from the adopted sequence. In Section 6.3, we also compare the performance of the various decomposition techniques, including our schemes, in terms of payload capacity. In Section 6.4, we test the performance of our Extended-Binary scheme when it is combined with/without the 3 pre-processing secret images of Chapter 5 in an LSBR-like steganography scheme.

6.2 Simple Sequence based cover pixel value decomposition scheme (SS)

In this section, a new pixel value decomposition scheme (Abdulla, et al., 2014) based on a specific representation is used to decompose pixel intensity values into 16 bit-planes that has less impact on stego-image quality when embedding in bit-planes beyond the first 3 bit-planes. The new pixel value decomposition scheme is based on a set of numbers SS and can be defined as:

$$SS = \{1\} \cup \{2n \mid 1 \leq n \leq 16, \quad \text{where } n \neq 9\} \tag{6. 1}$$

In other words, SS = {1, 2, 4, 6, 8, 10, 12, 14, 16, 20, 22, 24, 26, 28, 30, 32}. The reason for excluding number 18 in the sequence is to make the summation of the set SS equal to 255. All natural numbers between 0-255 can be represented using this proposed scheme. Using the SS sequence, each pixel value P is decomposed into 16 bit-planes, and the weight of the bit-planes can be defined as:

$$P = \sum_{i=1}^{16} b_i \, W_i \tag{6. 2}$$

$$\text{where} \quad b_i \in \{0,1\} \quad \text{and} \quad W_i = \begin{cases} 1 & if\ i = 1 \\ 2(i-1) & if\ 2 \leq i \leq 9 \\ 2i & if\ 10 \leq i \leq 16 \end{cases} \tag{6. 3}$$

If any pixel value has more than one representation in this number system, the lexicographically highest of them is always taken, to assert invertible property (e.g., the number 12 has two different representations, namely 0000000000100010 and 0000000000010100 since there are:

$$(1 * 10) + (0 * 8) + (0 * 6) + (0 * 4) + (1 * 2) + (0 * 1) = 12$$

$$(0 * 10) + (1 * 8) + (0 * 6) + (1 * 4) + (0 * 2) + (0 * 1) = 12$$

As 0000000000100010 lexicographically (from left to right) is higher than 0000000000010100, then the valid SS representation of 12 will be 0000000000100010.

This decomposition scheme differs from the existing non-binary schemes is that every cover pixel can be used for embedding when the 1st LSB is used for secret hiding. The number of bit-planes and their corresponding weights for different pixel value decomposition schemes including the proposed SS scheme are presented in Table 6-1.

It is clear that the competition between the non-binary decomposition techniques is increasing the number of bit-planes as well as reducing their weights in order to embed the secret bit in the higher bit-planes with less effect on the cover pixel value. Differences in weights associated to a given bit-plane between two schemes are related to differences in stego quality between these schemes when secrets are embedded in that bit-plane. For example, embedding a secret bit in 4th bit-plane of the binary scheme may change the pixel value by 8, while embedding the secret bit in the 4th bit-plane by the natural or Lucas may change the pixel value by 4. The only advantage of SS scheme over the binary scheme is that SS scheme could yield a better stego-image quality when the secrets are embedded in higher bit-planes (from 4th bit-plane onward), due to the fact that SS assigns smaller weight to the 4th bit-plane than the binary scheme. However, unlike the binary scheme, the SS scheme has a limitation in that not every cover pixel can be used for message embedding in the 2nd or higher bit-planes. An examination of the weights of the bit-planes beyond the 6th for all the listed sequences that except for natural sequence, the stego-image quality (when embedding secrets in the 6th bit-plane of cover images expressed by the SS scheme) is the best. But in this case, the quality could hardly be acceptable unless the embedded secret bits have large similarity the 6th bit-plane of the cover image. This discussion indicate that designing decomposition schemes for the sake of increasing the number of bit-planes is of limited interest confined to the desire of embedding secrets in higher bit-planes which can only be done at the expense of reduced stego-quality.

It is clear that the SS scheme and the binary scheme have exactly the same 0:1 ratio in the LSB plane of any image. Where all other schemes could have higher 0:1 ratio in the LSB due to the fact that the ratio can be reduced by expressing the odd pixel values without necessarily using 1. Only SS and the binary have no odd numbers >1 present in their sequence.

Table 6-1: Number of bit-planes and their corresponding weights for different pixel value decomposition techniques

Bit-plane #	Binary	Fibonacci	Lucas	prime	CF	SS	Natural
1	1	1	2	1	1	1	1
2	2	2	1	2	2	2	2
3	4	3	3	3	3	4	3
4	8	5	4	5	5	6	4
5	16	8	7	7	8	8	5
6	32	13	11	11	13	10	6
7	64	21	18	13	14	12	7
8	128	34	29	17	21	14	8
9		55	47	19	34	16	9
10		89	76	23	42	20	10
11		144	123	29	55	22	11
12		233	199	31	89	24	12
13				37	132	26	13
14				41	144	28	14
15				43	233	30	15
16						32	16
17							17
18							18
19							19
20							20
21							21
22							22
23							23

In short, the SS is of no interest to the objectives of this thesis beyond using it here to illustrate that increasing 0:1 ratio in the LSB plane of cover images is not dependent on increasing number of bit-planes but rather the ability express as many odd pixel values as possible without using 1 in their partition. In the next section, we use this conclusion to develop a new pixel value decomposition technique so called Extended-Binary is presented that aims to provide a higher 0:1 ratio of cover image LSB plane. This will contribute to guiding us in our effort to increase the probability of similarity between both the cover pixels' LSB value and the secret bits value.

6.3 Extended-Binary cover pixel value decomposition scheme

The defining sequence *K* of any pixel value decomposition scheme includes {1}. The above discussion show that increasing 0:1 ratio for any decomposition scheme can only be possible if pixel values that are expressed in the form $k_1+\ldots+k_r+1>1$ with $\{k_1,\ldots,k_r\}$ can be expressed in an equivalent way without using 1. This cannot be done with the usual binary decomposition scheme, or any scheme whose defining sequence does not include any odd number >1, because 1 is the only odd value in its sequence. The only way to increase the 0:1 ratio for such decomposition schemes is to extend their defining sequences by adding odd integer. The question is which odd number is needed to achieve an optimal increase in 0:1 ratio. It is clear that the answer to such question is image dependent. For example, if an image consist of even pixel values only or very few odd pixel values then very little or no benefits can be gained in terms of 0:1 ratio by adding any odd number. However, in such a case, the LSB plane consists of very few 1s anyway. On the other hand, images whose pixel values are predominantly odd, their LSB plan has proportionately 1s and could greatly benefit from extending the binary decomposition scheme, or any scheme whose defining sequence does not include any odd number >1, to include odd integers. We shall focus first on extending the defining sequence of the binary decomposition scheme to deal with images where the ratio of odd pixel values is not marginal.

Let *I* be an image of size *N* and let hist(*I*) be its histogram. The amount of increase in the 0:1 ratio as a result of adding an odd number *x* to the defining sequence B={1, 2, 4, 8, 16, 32, 64,128} is dependent on hist(*x*) and hist(*y*) for all $y > x$ that can be expressed without using the first element in the defining sequence. To determine which odd number can achieve best 0:1 ratio when added to the defining sequence of the binary scheme, we first observe that adding odd number n > 1 for which $n+1\neq 2^i$ cannot be a good candidates. Due to the use of lexicographically highest decomposition for the sake of uniqueness of representation will mean that beside several odd numbers, some even numbers will also have 1 as their LSB. For example, adding 5 will result in having 1 as the LSB of the even numbers in the set {6, 6+8, 6+(2x8), …, 6+(31x8)} besides some odd numbers such as 3, 9,11,17, and 19. The smallest odd number for which $n+1= 2^i$ is 3 and in this case all odd numbers of the form 3+4k, for some k>0, will have 0 LSB, whereas the LSB of all other odd numbers >3 is 1. Therefore, including odd number 3 increases the 0:1 ratio in the LSB plane by:

$$\frac{\sum_{i=0}^{63} hist\ (3+4*i)}{N} \tag{6. 4}$$

Based on the above discussion, a new cover image pixel value decomposition scheme that expands the binary scheme will be proposed and tested for suitability for embedding in terms of increased 0:1 ratio in the LSB plane. The proposed new is an extended version of the usual binary that adds only one bit-plane with the weight of odd (prime) number 3, and it will be referred to as Extended-Binary. The defining sequence of the Extended-Binary scheme is the set of numbers S defined as:

$$\mathrm{S} = \{3\} \cup \{2^n \mid 0 \le \mathrm{n} \le 7\} \tag{6. 5}$$

In other words, S = {1, 2, 3, 4, 8, 16, 32, 64, 128}. Using the set S, each pixel value P is decomposed into 9 bits and the weight of the bit-plane can be defined as:

$$P = \sum_{i=1}^{9} b_i\, W_i \tag{6. 6}$$

$$\text{where} \quad b_i \in \{0,1\} \quad \text{and} \quad W_i = \begin{cases} i & if\ 1 \le i \le 3 \\ 2^{\,i-2} & if\ 4 \le i \le 9 \end{cases} \tag{6. 7}$$

If any pixel value has more than one representation in this number system, then we select the lexicographically highest such numbers to assert uniqueness of representations and the invertible property. For example, the pixel value 12 has two different representations in the Extended-Binary number system, namely 000011000 and 000010101 such as:

$$(1 \times 8) + (1 \times 4) + (0 \times 3) + (0 \times 2) + (0 \times 1) = 12$$

$$(1 \times 8) + (0 \times 4) + (1 \times 3) + (0 \times 2) + (1 \times 1) = 12$$

Since 000011000 is lexicographically (from left to right) is higher than 000010101, and then it will be chosen to validly representing 12 in the Extended-Binary number system, and 000010101 will be discarded. Table 6- 2 illustrates the valid representation of the Extended-Binary decomposition system for the pixel values from 0 to 255.

In general and from our experiments, the number of cover pixels that their values are even is almost equal to those that their values are odd. This means by decomposing the cover pixel value using usual binary, the number of pixels that their LSB value is zero is almost equal to those that their LSB value is one, see Figure 6-2. Therefore by adding only the one bit-plane with weight 3, results in increasing the 0:1 ratio in the LSB bit-plane by the amounts discussed above in equation (6. 4). For example, the odd pixel value 11 in binary decomposition = (8+2+1) ≡ (0000101$\underline{1}$), while in the proposed Extended-Binary = (8+3) ≡ (00001010$\underline{0}$). Thus, the set of numbers of the proposed decomposition technique is not designed randomly, but the reason of adding only a bit-plane with weight of 3 is to make the LSB value of the some odd pixel value becomes zero. This results in increasing the ratio of cover pixels with LSB value zero.

Table 6- 2: Pixel values and their decomposition using Extended-Binary scheme.

value	Binary rep.	value	Binary rep.	value	Binary rep.	value	Binary rep.
0	000000000	64	010000000	128	100000000	192	110000000
1	000000001	65	010000001	129	100000001	193	110000001
2	000000010	66	010000010	130	100000010	194	110000010
3	000000100	67	010000100	131	100000100	195	110000100
4	000001000	68	010001000	132	100001000	196	110001000
5	000001001	69	010001001	133	100001001	197	110001001
6	000001010	70	010001010	134	100001010	198	110001010
7	000001100	71	010001100	135	100001100	199	110001100
8	000010000	72	010010000	136	100010000	200	110010000
9	000010001	73	010010001	137	100010001	201	110010001
10	000010010	74	010010010	138	100010010	202	110010010
11	000010100	75	010010100	139	100010100	203	110010100
12	000011000	76	010011000	140	100011000	204	110011000
13	000011001	77	010011001	141	100011001	205	110011001
14	000011010	78	010011010	142	100011010	206	110011010
15	000011100	79	010011100	143	100011100	207	110011100
16	000100000	80	010100000	144	100100000	208	110100000
17	000100001	81	010100001	145	100100001	209	110100001
18	000100010	82	010100010	146	100100010	210	110100010
19	000100100	83	010100100	147	100100100	211	110100100
20	000101000	84	010101000	148	100101000	212	110101000
21	000101001	85	010101001	149	100101001	213	110101001
22	000101010	86	010101010	150	100101010	214	110101010
23	000101100	87	010101100	151	100101100	215	110101100
24	000110000	88	010110000	152	100110000	216	110110000
25	000110001	89	010110001	153	100110001	217	110110001
26	000110010	90	010110010	154	100110010	218	110110010
27	000110100	91	010110100	155	100110100	219	110110100
28	000111000	92	010111000	156	100111000	220	110111000
29	000111001	93	010111001	157	100111001	221	110111001
30	000111010	94	010111010	158	100111010	222	110111010
31	000111100	95	010111100	159	100111100	223	110111100
32	001000000	96	011000000	160	101000000	224	111000000
33	001000001	97	011000001	161	101000001	225	111000001
34	001000010	98	011000010	162	101000010	226	111000010
35	001000100	99	011000100	163	101000100	227	111000100
36	001001000	100	011001000	164	101001000	228	111001000
37	001001001	101	011001001	165	101001001	229	111001001
38	001001010	102	011001010	166	101001010	230	111001010
39	001001100	103	011001100	167	101001100	231	111001100
40	001010000	104	011010000	168	101010000	232	111010000
41	001010001	105	011010001	169	101010001	233	111010001
42	001010010	106	011010010	170	101010010	234	111010010
43	001010100	107	011010100	171	101010100	235	111010100
44	001011000	108	011011000	172	101011000	236	111011000
45	001011001	109	011011001	173	101011001	237	111011001
46	001011010	110	011011010	174	101011010	238	111011010
47	001011100	111	011011100	175	101011100	239	111011100
48	001100000	112	011100000	176	101100000	240	111100000
49	001100001	113	011100001	177	101100001	241	111100001
50	001100010	114	011100010	178	101100010	242	111100010
51	001100100	115	011100100	179	101100100	243	111100100
52	001101000	116	011101000	180	101101000	244	111101000
53	001101001	117	011101001	181	101101001	245	111101001
54	001101010	118	011101010	182	101101010	246	111101010
55	001101100	119	011101100	183	101101100	247	111101100
56	001110000	120	011110000	184	101110000	248	111110000
57	001110001	121	011110001	185	101110001	249	111110001
58	001110010	122	011110010	186	101110010	250	111110010
59	001110100	123	011110100	187	101110100	251	111110100
60	001111000	124	011111000	188	101111000	252	111111000
61	001111001	125	011111001	189	101111001	253	111111001
62	001111010	126	011111010	190	101111010	254	111111010
63	001111100	127	011111100	191	101111100	255	111111100

From Table 6- 2, it is noticeable that out of 256 values, 192 values have 0 LSB value. In other words, 75% of the values are their LSB value is zero. While for the same values, using usual binary decomposition technique, 50% of the values have LSB value of zero.

6.3.1 Performance of Extended-Binary

In this section, the performance, over our 2 experimental databases, of the Extended-Binary pixel value decomposition technique will be investigated in terms of the ratio of 0:1 ratio of the cover pixels' LSB plane.

Results

Table 6-3 and Table 6-4 present the 0:1 ratio of the cover pixels' LSB plane when we use the proposed Extended-Binary decomposition technique for the original cover images and their complement versions for the SIPI and BOSSBase databases, respectively. In the tables, Rz and Rz'refer to the average 0:1 ratio, over all images in the databases, of original cover images I and their complement version I_C respectively, while max ($Rz,\ Rz'$) refers to the selecting either Rz or Rz' based on the maximum 0:1 ratio for the images I and I_C .

Table 6-3: Ratio of the cover pixels' LSB zero value of the Extended-Binary decomposition technique for SIPI database.

	$\boldsymbol{Rz}$	$\boldsymbol{Rz'}$	**max**$(\boldsymbol{Rz},\boldsymbol{Rz'})$
μ	0.756	0.770	0.774
σ	0.071	0.055	0.054
$\boldsymbol{M_n}$	0.440	0.701	0.701
$\boldsymbol{M_x}$	1.000	1.000	1.000

Table 6-4: Ratio of the cover pixels' LSB zero value of the Extended-Binary decomposition technique for BOSSBase database.

	$\boldsymbol{Rz}$	$\boldsymbol{Rz'}$	**max**$(\boldsymbol{Rz},\boldsymbol{Rz'})$
μ	0.753	0.753	0.756
σ	0.021	0.014	0.014
$\boldsymbol{M_n}$	0.424	0.667	0.702
$\boldsymbol{M_x}$	0.892	0.906	0.906

From Table 6-3 and Table 6-4, its noticeable that the Rz and Rz' are different from each other, and selecting the one, either Rz or Rz', that has higher ratio of zero bits value has led to the improved result for the proposed Extended-Binary. Interestingly, the achieved 0:1 ratio is almost the same ratio obtained by counting the number of grayscale values whose LSB Extended-Binary bit value was 0 in Table 6- 2. This is particularly true for the larger database BOSSBase but to less extent for the SIPI. These results also demonstrate that the proposed Extended-Binary achieves higher 0:1 ratio that can be estimated reasonably well from the translation tables. Furthermore, the result of max $(Rz,\ Rz')$ is mostly greater than of Rz and Rz', and this proof that applying the proposed Extended-Binary on both the original image and its complement results in increased the 0:1 ratio. In other words, applying the Extended-Binary on both the cover image and its complemented version is better than applying on only the cover image in terms of providing higher 0:1 ratio of LSB plane.

To determine whether adding other odd numbers >3 to the binary defining sequence can yield better performance. We expanded our experiments by testing different extended sequences by adding different prime numbers in the binary sequence to investigate whether provide more LSB = 0 or not. The following are the tested extended binary sequences:

$$S_1 = \{1, 2, 4, 5, 8, 16, 32, 64, 128\} \tag{6. 8}$$

$$S_2 = \{1, 2, 4, 8, 11, 16, 32, 64, 128\} \tag{6. 9}$$

$$S_3 = \{1, 2, 4, 8, 16, 23, 32, 64, 128\} \tag{6. 10}$$

$$S_4 = \{1, 2, 4, 8, 16, 32, 47, 64, 128\} \tag{6. 11}$$

$$S5 = \{1, 2, 4, 8, 16, 32, 64, 97, 128\} \tag{6. 12}$$

Figure 6-1 presents the 0:1 ratio for the above different sequences of numbers plus the sequence S in equation (6. 5) used to decompose cover pixels value. The results are obtained for the same two experimental databases.

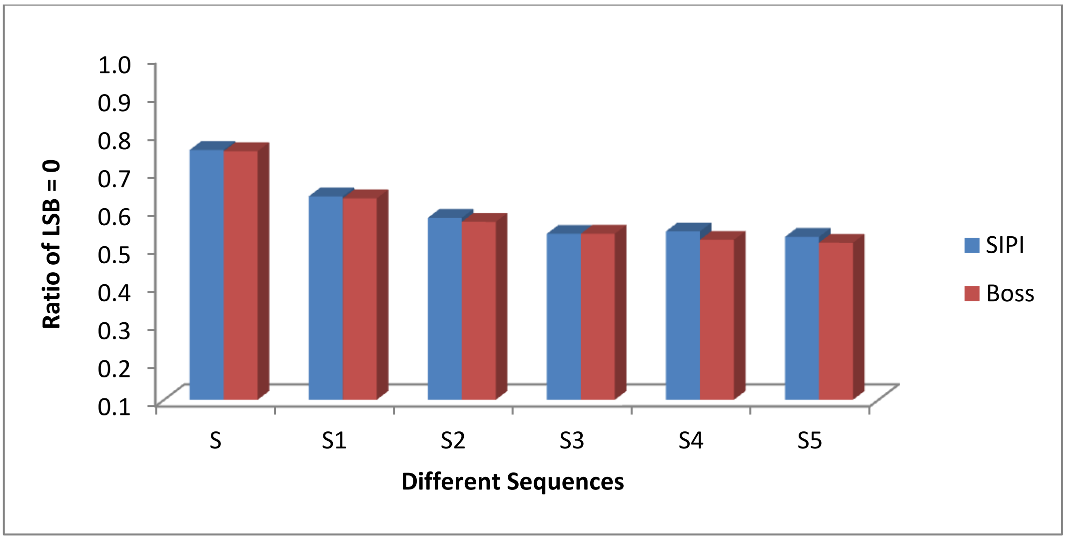


Figure 6-1: Ratio of cover pixels' LSB = 0 for the different sequences of numbers.

Figure 6-1 shows that compared to all the different versions of the Extended-Binary pixel value decomposition technique, the S version has the highest 0:1 ratio of the LSB plane for both databases, SIPI and BOSSBase. Adding prime number in higher bit-planes of the usual binary, results in reducing the 0:1 ratio.

Moreover, Figure 6-2 presents the results of the experiments conducted to test the performance, in terms of 0:1 ratio, of the different decomposition techniques including the S-version of the Extended-Binary, for both databases, SIPI and BOSSBase. Although our objectives in introducing the various decomposition schemes was about increasing the 0:1 ratio in the LSB plane, but we know that in the case of the Fibonacci decomposition scheme the drawback is in reduced payload capacity. This is due to the fact that not every cover pixel is suitable to embed the secret bit, because the embedding scheme may result in violating the Zeckendorf property. Figure 6-3 presents the results of the same experiments in terms of remaining ratio of payload capacity for all decomposition techniques including ours, for all cover images in the two experimental databases, SIPI and BOSSBase.

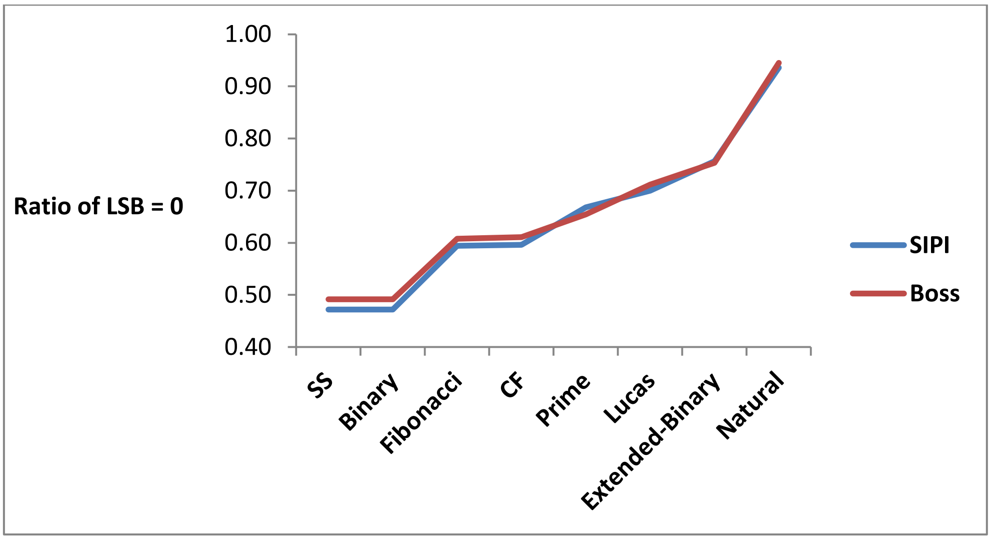


Figure 6-2: Ratio of LSB = 0 for different decomposition techniques.

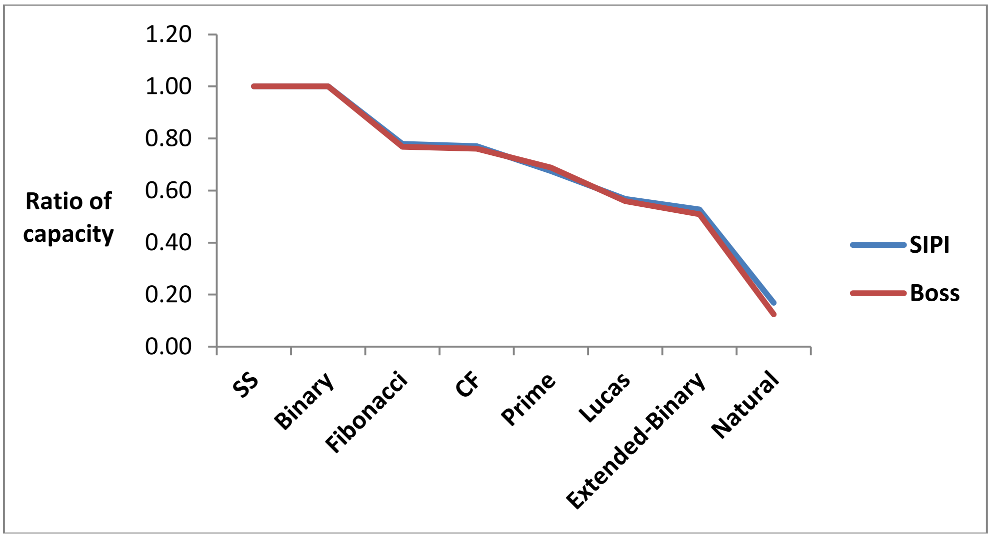


Figure 6-3: Ratio of capacity for different decomposition techniques.

From Figure 6-2, it is noticeable that lowest 0:1 ratio in the LSB plane is obtained by decomposing the cover pixel values using the SS and binary schemes, both have exactly the same ratio. While the highest ratio is obtained when using the natural decomposition technique, and our S version of our Extended-Binary provides second highest 0:1 ratio compared to all other decomposition techniques.

The idea of using image complement, in Chapter 5, to improve the 0:1 ratio of the secret image can be exploited to further improve the 0:1 ratio of the LSB plane cover images. We have modified our Extended-Binary and the Fibonacci decomposition schemes which would be referred to as the Extended-Binary_C, and the Fibonacci_C. In each case, we apply the decomposition scheme on both the image as well as its complement, and then select the image version that has the highest 0:1 ratio as the cover image. The following table shows the effect of using both the image and its complement on the 0:1 ratio in the LSB plane, for the modified schemes. The table also includes the previously established performance of the other unmodified decomposition schemes. It is clear that the performance of the both modified schemes has improved over their unmodified versions. In fact, the improvement in the case of the Fibonacci_C scheme is somewhat significant, and it became better than the CF scheme.

Table 6-5: Ratio of 0:1 LSB for different decomposition techniques.

Decomposition technique	Databases	
	SIPI	**Boss**
SS	0.47	0.49
binary	0.47	0.49
Fibonacci	0.59	0.61
CF	0.60	0.61
Fibonacci_C	0.65	0.63
prime	0.67	0.65
Lucas	0.70	0.71
Extended-Binary	0.76	0.75
Extended-Binary_C	0.77	0.76
natural	0.94	0.94

The effect of using the various pixel value decomposition schemes on the payload capacity, as shown in Figure 6-3, is in the opposite direction of their effect on the 0:1 ratio. In fact, both SS and binary decomposition techniques have full capacity, i.e. every cover pixel is used for message embedding, while the worst capacity ratio results from using natural decomposition technique. The capacity ratio of the S-version of the Extended-Binary is only better than that of the natural decomposition technique. In Chapter 4, we have demonstrated that using a mapping-based has led to increasing the payload capacity for the Fibonacci decomposition scheme. In the next chapter, we shall demonstrate that the capacity drawback of using the Extended-Binary scheme, and some other schemes, can be remedied by adopting mapping-based embedding procedure instead of directly replacing the LSB bits of the cover pixel value.

6.4 Experimental Results

In this section, we test the performance of a simple embedding scheme that simply embeds secret image bit-stream into the LSB of an Extended-Binary decomposed cover image. The performance of this scheme is tested in terms of embedding efficiency, stego-image quality, and robustness against targeted steganalysis tools. In these experiments, we will use the 44 images the SIPI database by creating two size versions of these images: a 512 x 512 to be used as cover images after decomposing their pixel values by the Extended-Binary, and a 128 x 256 for use as secret image. In our experiments, we test the performance of four embedding schemes: (1) *Original_EB* by embedding the secret image bit-stream without pre-processing; (2) *SISR_EB* by embedding the secret images SISR bit-stream; (3) *SIM_EB* by embedding the secret images SIM bit-stream, and (4) *IWSIM_EB* by embedding the secret images IWSIM bit-stream. In total, for each of the 4 cases, we have 1936 stego-images. The experimental results will be presented in the next 3 parts and each case evaluation parameters represent the average value of the 1936 images in each case.

1. Embedding Efficiency Evaluation

Figure 6-4 presents the average value of the ratio of modified pixels to the length of the secret bit-stream, for the 4 embedding schemes, while Figure 6-5 presents the average value of the corresponding embedding efficiency. From Figure 6-4, it is clear that, for all embedding payloads, the IWSIM_EB causes the lower number of modified cover pixels after secret embedding compared to the others, and consequently it has higher embedding efficiency. Together the results in the two figures demonstrate that performance of the 4 schemes in terms of efficiency are in the order IWSIM_EB, SIM_EB, SISR_EB and Originl_EB from best to worst. This was to be expected because the corresponding 0:1 ratio in their secret bit-streams is 80%, 73%, 57%, and 49% (see Chapter 5). Note that a higher 0:1 ratio reflects a higher similarity between the secret bits values and the cover pixels' LSB values, and in all schemes the 0:1 ratio of the Extended-Binary LSB is fixed at 77% (see Table 6-5). However, the achieved efficiency by IWSIM_EB is still lower than what is desired, this may have happened because of the skipping of bad cover pixels candidates and the majority of the skipped pixels may have a 0 LSB value. Note that in the Extended-Binary decomposition scheme, 47% of the cover pixels are skipped for embedding on average, see Figure 6-3. This limitation of skipping cover pixels of Extended-Binary decomposition scheme will be investigated and overcame in the next chapter.

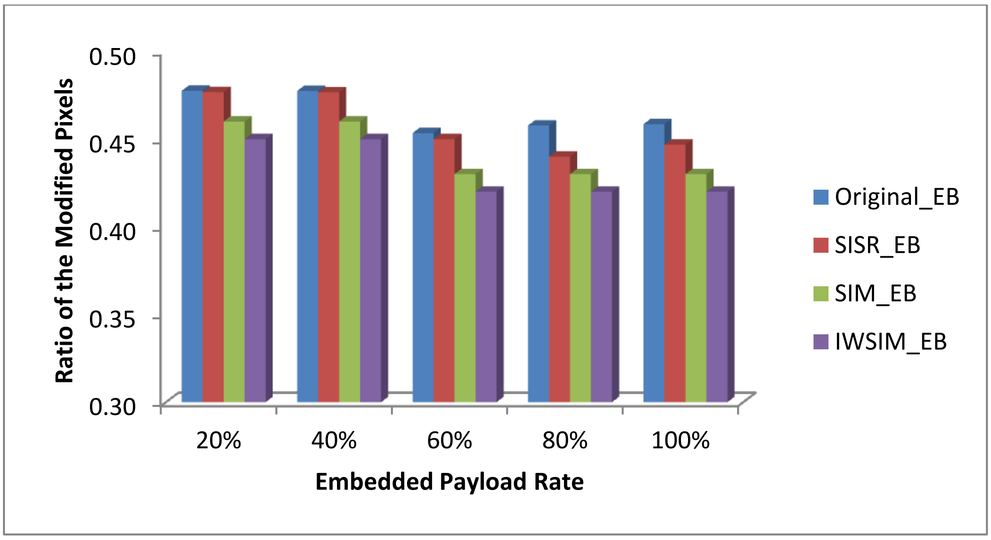


Figure 6-4: The ratio of the modified pixels for the Original_EB, SISR_EB, SIM_EB, and IWSIM_EB schemes.

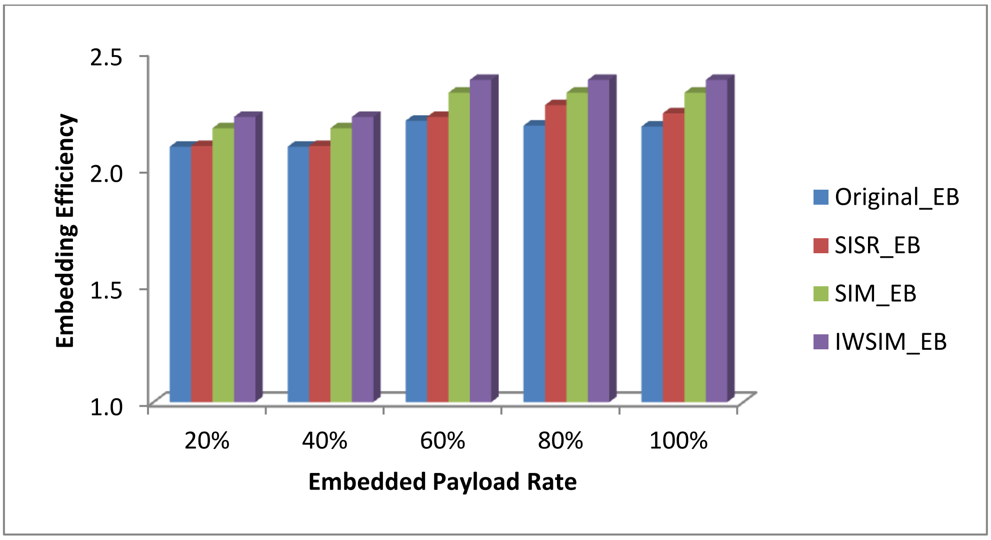


Figure 6-5: The embedding efficiency for the Original_EB, SISR_EB, SIM_EB, and IWSIM_EB schemes.

2. Stego-Image Quality Evaluation

Figure 6-6 presents the average PSNR values of the stego-images relative to the cover images computed for the 4 tested embedding schemes. It is clear that for all embedding payload rate, PSNR of the SISR_EB embedding scheme is higher than that for all 3 other schemes, which is due to the fact that the length of the SISR bit-stream is always less than the length of the other bit-streams. While the PSNR values for other embedding schemes are almost the same.

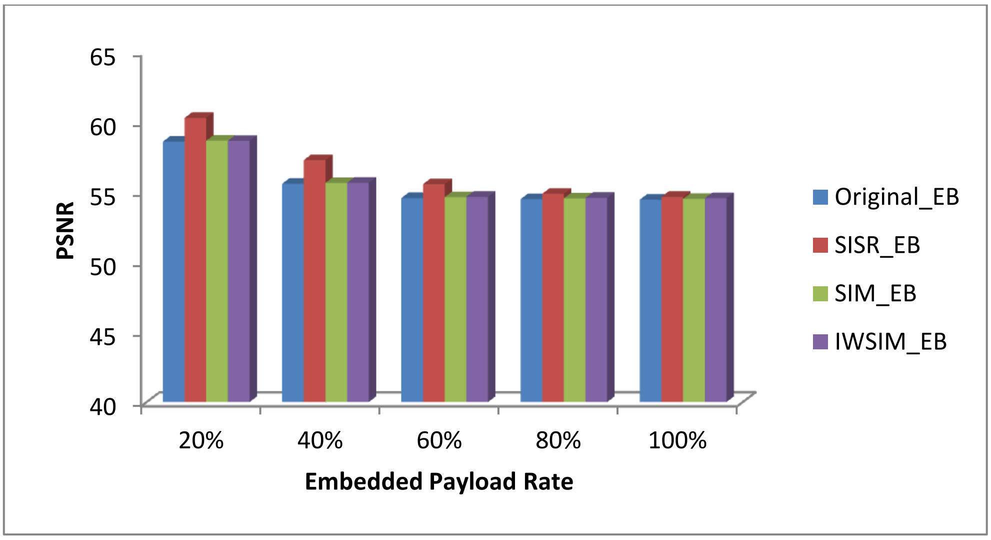


Figure 6-6: The PSNR for the Original_EB, SISR_EB, SIM_EB, and IWSIM_EB schemes.

3. Detectability Evaluation

The detectability evaluation experiments carried out, in this section, for all the four embedding schemes and all payload rates determine their robustness against the three well-known steganalysis detectors (RS, DIH, and RWS). The experimental results again represent average values when the 1936 stego-images were tested by the three tools.

Robustness Against RS Detector

Figure 6-7 displays the RS diagram for tested embedding schemes, from which it is clear that for all tested embedding schemes, there are big differences between RM and RM-, SM and SM-, demonstrating that the tested schemes are not robust against the RS detector. However, the SIM_EB and the IWSIM_EB are slightly more robust against the RS at higher embedding rates. The reason is that in all tested schemes, the cover pixels’ LSB value are flipped when the secret bit not match, and this cause asymmetry problem. Therefore, the embedded message can be detected by RS detector. In the next chapter, the asymmetry problem of Extended-Binary decomposition scheme will be sorted out.

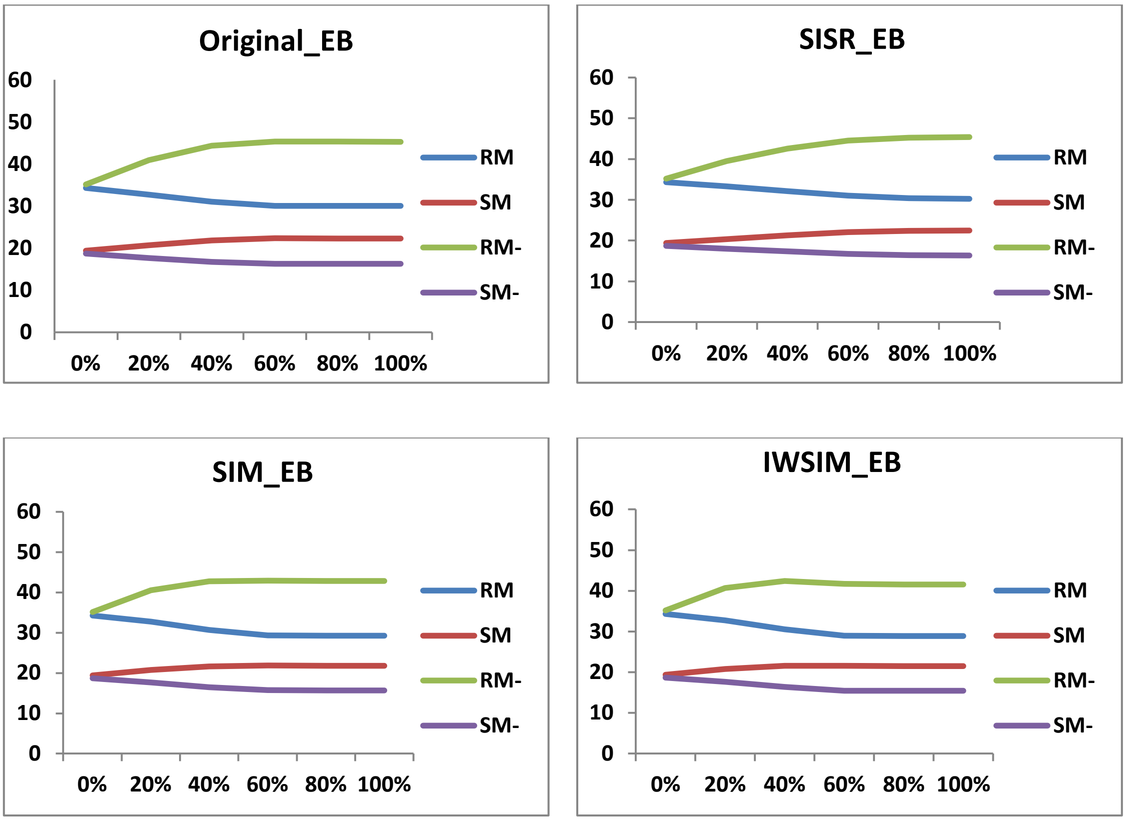


Figure 6-7: RS diagram for the Original_EB, SISR_EB, SIM_EB, and IWSIM_EB schemes.

Robustness Against DIH Detector

For each embedding rate, the chart of Figure 6-8 presents the average values representing the probability of having a secret hidden at the given embedding ratio. We can see that for the higher embedding rates, the IWSIM_EB is more robust against DIH compared to other tested embedding schemes. This is achieved due to a lower ratio of cover pixels are modified after secret embedding, see Figure 6-4. It is also clear that for all embedding rate, schemes that embedded the pre-processed secret image bit-streams are more robust than the one that embeds the original unprocessed secret image bit-stream. This is because in all manipulated schemes in Chapter 5, SISR, SIM, and IWSIM, the ratio of 0:1 in their bit-streams are higher than the original secret image bit-stream, reflecting on increasing the probability of similarity between the secret bit-stream and the LSB of the Extended-Binary representation of the cover pixel value.

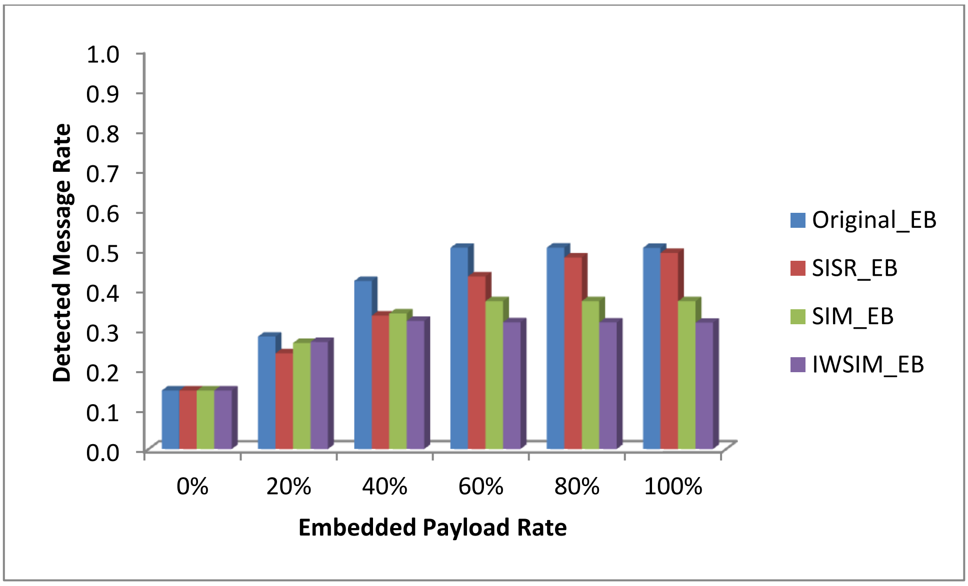


Figure 6-8: DIH steganalysis for the Original_EB, SISR_EB, SIM_EB, and IWSIM_EB schemes.

Robustness Against RWS Detector

Figure 6-9 presents the average values of the estimation ratios of the flipped cover pixels' LSB of the 4 tested embedding schemes at different embedding rates. These results that all schemes are robust against the RWS especially at high payload rates > 40% with a maximum detection rates < 0.19. As for the DIH detector, it is clear that at higher embedding rate, the IWSIM_EB is more robust against RWS compared to other tested embedding schemes. This is achieved due to the same reason discussed in DIH detector. It is also clear that for higher embedding rate, schemes that embedded the manipulated secret image bit-stream are more robust than the one that is embedding the original secret image bit-stream.

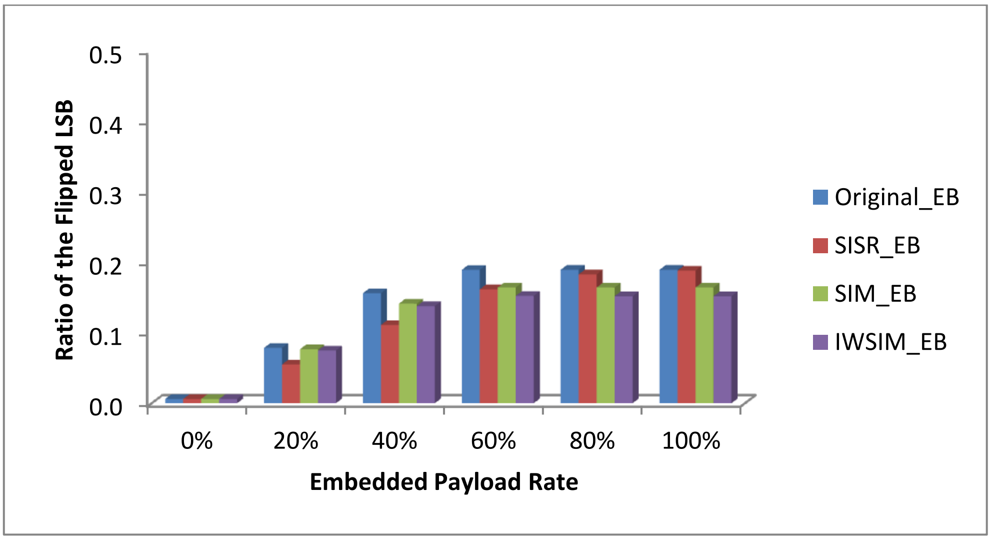


Figure 6-9: RWS steganalysis for the Original_EB, SISR_EB, SIM_EB, and IWSIM_EB schemes.

6.5 Discussion

This chapter was devoted to complement the pre-processing of secret images by attempting to improve the similarity between the LSB plane of cover images with the preprocessed secret image bit-streams. Having found in Chapter 4 that the use of Fibonacci pixel decomposition scheme resulted in increasing the number of 0 LSB's within the three first decomposed bit-planes, was the main motive in investigating various decomposition schemes for their impact on 0:1 ratio in the LSB plane. We investigated existing schemes for cover pixel value decomposition such as binary, Fibonacci, prime, natural, Lucas, Catalan-Fibonacci (CF), and SS and introduced a new technique, Extended-Binary which decomposes cover pixel intensity values into 9 bit-planes suitable for embedding purposes. Experimental results demonstrate that the Extended-Binary cover pixel value decomposition technique offers the 2^{nd} highest 0:1 ratio (approximately 77% on average). The best performing decomposition scheme is based on the 23 bit-plane natural defining sequence which is not practical to use. The experimental work carried out on a sufficiently large number of natural images from two databases, and demonstrates the success of our approach to improve similarity between secret image bit-streams and the cover image LSB plane for improved stego-image quality.

The limitation of the Extended-Binary scheme is payload capacity, since not every pixel is usable for message embedding when the secret bit is embedding in the cover pixel's LSB and this is the case in all other pixel value decomposition schemes except binary based and SS based embedding techniques. Mapping based embedding technique can be used to remedy this drawback on the payload capacity. Moreover, these proposed processes, namely Extended-Binary and image complement, help achieve a steganography system that has high embedding efficiency, when pre-processed secret bit-streams are embedded. Furthermore, embedding the bit-stream, which contains higher ratio of bits that their values are zero in the cover image that higher ratio of its pixels' LSB has a value of zero, results in reducing the number of necessary cover pixels to be changed after message embedding.

We have tested the performance of an embedding scheme that simply embeds secret image bit-stream into the LSB of an Extended-Binary decomposed cover image. The performance of this scheme is tested in terms of embedding efficiency, stego-image quality, and robustness against targeted steganalysis tools. Unfortunately, the embedding efficiency obtained by the IWSIM_EB is still lower than our objective; this may have happened because of the skipping the bad candidate cover pixels. Moreover, the majority of the skipped pixels seem to have 0 LSB value while the corresponding IWSIM secret bit-stream bits were 1's, i.e., dissimilarity. Note that in the Extended-Binary decomposition scheme, 47% of the cover pixels are skipped for embedding on average, see Figure 6-3. This limitation of skipping cover pixels of Extended-Binary decomposition scheme is exacerbated by the fact that in this chapter our schemes simply embed in the LSB by replacement. In the next chapter, we shall demonstrate that using mapping tables, rather than replacement will help overcome this limitation. All the schemes have been shown to be robust, especially high embedding rates, against the DIH and the RWS steganalysis tools, but not so against the RS tool. This is somewhat similar to the robustness of the Fibonacci-Mapping based scheme proposed in Chapter 4.

Chapter 7

Mapping based Steganography for Hiding Secret Images in Cover Images

In Chapter 4, we introduced the Fibonacci-Mapping based scheme to embed two secret bits in the first three bit-planes of the Fibonacci decomposed cover image using a mapping table rather than bit replacement for embedding. It increased capacity, had reasonable embedding efficiency, good robustness against two of the LSB targeted steganalysis tools, but had less than desirable stego-image quality. It helps set out a strategy to increase similarity between secret image bit-stream and the cover image LSB plane. For the secret image, we developed three successful algorithms (SIM, IWSIM, and SISR) in the Chapter 5 to be applied on the secret image prior to embedding which increased 0:1 ratio in the secret image bit-streams. Embedding pre-processed secret image bit-streams into cover images using the Fibonacci-Mapping based scheme resulted in improved embedding efficiency and maintaining un-detectability, but stego-image quality still falls short of our expectation. In Chapter 6, we designed a new pixel value decomposition scheme (Extended-Binary) which resulted in achieving a 77% ratio of 0:1 in the cover images' LSB plane, and thereby the combined effects of this scheme and those in Chapter 5 contributed to increasing the probability of similarity between the secret image bit-stream and the cover image LSB plane. We tested the performance of an embedding scheme that directly replaces the LSB of the decomposed cover image pixel with a single bit of the pre-processed secret image bit-stream. Though, the various schemes performed well on almost all criteria, the embedding efficiency of our schemes were still lower than what is achievable. LSB replacement based embedding seem to force the skipping of many bad candidate cover pixels, while mapping based embedding does not suffer from this problem. Extending the Fibonacci-Mapping table to other decomposition schemes may results in low stego-image quality due to the fact that we had to modify higher bit-planes. To overcome this drawback, in this chapter, we extend the proposed mapping based embedding table that presented in Chapter 4 by embedding one secret bit in each the decomposed cover pixel. In Section 7.1, we design mapping tables for the various decomposition schemes. In Section 7.2, we describe the various mapping based embedding schemes designed by pairing a secret pre-processing algorithm with a cover pixel decomposition model. In Section 7.3, we shall test the performance of various Mapping-based combination schemes in terms of the above stated objectives of this thesis.

7.1 Single bit Mapping Tables for pixel value decomposition schemes

When secret bits are embedded by directly replacing the cover image LSB bits and the cover image pixels are decomposed by a non-binary technique, many cover pixel values will violate the uniqueness representation rule have to be skipped. To avoid this, and maintain capacity, we use mapping table for embedding single secret bits. In this section, we shall introduce a mapping table for each cover image pixel decomposition scheme use to implement embedding of single bits. These mappings are defined in terms of the first 3 bit-planes of the corresponding decomposition scheme. The structure of the investigated decomposition schemes results in reducing the number of possible 3 bit patterns into 4 or 5 out of 8 different random 3-bit patterns. In order to present these tables in a compact and informative manner, we shall divide the rest of the section into two subsections depending on the number of rows in these tables.

7.1.1 The 5-rows Mapping Tables (Fibonacci, prime, natural, and CF)

The first three LSBs of a cover pixel value in Fibonacci, prime, natural and CF representation belong to the set {000, 001, 010, 100, 101}. Based on the mapping in Table 7-1, a secret bit embeds into a cover pixel by mapping it onto the first 3 LSBs. Note that all mentioned decomposition techniques have the same table.

Table 7-1: Mapping for Fibonacci, prime, natural, and CF.

Cover bits	Secret bit	
	0	**1**
000	000	001
001	010	001
010	010	001
100	100	101
101	100	101

From Table 7-1, we observed the following points:

1. The mapping table is applicable for the Fibonacci representation, and the receiver only extracts from the first LSB of the Fibonacci representation of the stego pixel value to get the message.
2. The mapping is not applicable on the prime based embedding techniques because it is not feasible with some cover pixel values. For example, for the cover pixel value 16, its prime representation is (000000001000100) and after the secret bit value 1 is embedded based on the mapping presented in Table 7-1, the stego pixel value becomes 17. Once the receiver decompose the stego pixel value based on prime decomposition technique, this bit-stream (000000010000000) represents the stego pixel value 17, and by extracting from the first LSB, the secret bit value 0 is obtained which is not equal to the embedded bit at the sender.
3. The mapping is not applicable on the natural based embedding techniques because it is not feasible with some cover pixel values. For example, for the cover pixel value 4, its natural representation is (0000000000000000001000) and after the secret bit value 1 is embedded based on the mapping presented in Table 7-1, the stego pixel value becomes 5. Once the receiver decompose the stego pixel value based on natural decomposition technique, this bit-stream (0000000000000000010000) represents the stego pixel value 5, and by extracting from the first LSB, the secret bit value 0 is obtained which is not equal to the embedded bit at the sender.
4. The mapping is not applicable on the CF based embedding techniques because it is not feasible with some cover pixel values. For example, for the cover pixel value 13, its CF representation is (000000000100000) and after the secret bit value 1 is embedded based on the mapping presented in Table 7-1, the stego

pixel value becomes 14. Once the receiver decompose the stego pixel value based on CF decomposition technique, this bit-stream (000000001000000) represents the stego pixel value 14, and by extracting from the LSB, the secret bit value 0 is obtained which is not equal to the embedded bit at the sender.

7.1.2 The 4-rows Mapping Tables (Lucas, and Extended-Binary)

The first three LSBs of a cover pixel value in Lucas and Extended-Binary representation belong to the set {000, 001, 010, 100}. Based on the mapping in Table 7-2, a secret bit embeds into a cover pixel by mapping it onto the first 3 LSBs.

Table 7-2: Mapping for Lucas and Extended-Binary.

Cover 3-LSBs	Mapping for Lucas		Mapping for Extended-Binary	
	Secret bit		Secret bit	
	0	1	0	1
000	000	001	000	001
001	010	001	010	001
010	010	001	010	001
100	100	101	100	001

From Table 7-2, we observed the following points:

1. The mapping is applicable on the Lucas based embedding techniques, and all pixel values are feasible with the mapping presented in Table 7-2. The only drawback of the Lucas is the quality of the stego-image, because the first element in the Lucas sequence starts by 2 and modifying the first LSB leads to change the pixel value by 2. While in other decomposition techniques the pixel value change by 1, when the secret bit is embedded in the LSB.

2. The mapping is applicable on the Extended-Binary based embedding techniques, and all pixel values are feasible with the mapping presented in Table 7-2. The only drawback of the our proposed steganography approach based on the mapping in Table 7-2 is the stego-image quality, since in the Table 7-2 when the first three LSBs of the Extended-Binary representation of the cover pixel value is 100 and the secret bit value is 1, the cover pixels value will be changed by 2 after secret embedding. In other words, 12.5% of the modified pixels' value may change by 2.

The advantage of this mapping for Extended-Binary representation is not only overcome the drawback of the payload capacity but also it has the advantage that do not suffer from the asymmetry problem. For example, in the usual binary based embedding techniques (i.e. LSBR), the even pixels value either increases by one or left unchanged, and odd pixels value are decreased by one or left unchanged. In other words, the odd pixel value either becomes an even value of left unchanged and the even pixel value either becomes an odd value of left unchanged. This creates an imbalance in the embedding distortion in the stego-image and this imbalanced is called asymmetry problem, normally grouping in the pixel values (0, 1); (2, 3); . . . (254, 255), and can be exploited to easily detect the existence of a hidden message in a stego-image using some designed steganalysis techniques, such as PoV, even at a low embedding rate. While in our proposed mapping in Table 7-2, the odd pixel value may increase, decrease or left unchanged. For example, the odd pixel value 1 (000000001 in Extended-Binary representation) becomes 2 (000000010 in Extended-Binary representation) after the secret bit 1 is embedded, while in LSBR becomes 0. Also for the odd pixel value 3 (000000100 in Extended-Binary representation) becomes (000000001 in Extended-Binary representation) after the secret bit 1 is embedded, while in LSBR becomes 2. Beside of the property of our proposed steganography approach of decreasing the ratio of modified pixels after message embedding that makes the stego-image less detectable, another factor that leads to make our proposed steganography approach less detectable against steganalysis is the stego-image has not asymmetry problem.

7.2 Efficient Secure image-based steganography schemes

By combining each of our three secret image pre-processing algorithms (see Chapter 5) with the pixel Extended-Binary decomposition scheme and using the corresponding mapping table, we get three different schemes that referred to by EB_SISR, EB_SIM, and EB_IWSIM. Here, we shall present a general format of the embedding and extracting procedures for each possible paired scheme.

Embedding Procedure

1. Apply the SISR, SIM, or IWSIM on the secret image prior to embedding producing the secret bit-stream of length m.
2. Let I' be the complement image of the cover image I.
3. Decompose pixels value using the S-version of the Extended-Binary decomposition technique for I and I'.
4. Calculate the 0:1 ratio R and R' of the LSB plane of the decomposed image I and I', respectively.

5. If $R >= R'$, then the image I is chosen as a cover, otherwise, image I' is chosen as a cover.

6. PRNG is used to select the cover pixel p_i randomly to be used for message embedding using an agreed seed.

7. Based on the proposed mapping in Table 7-2, the secret bit m_i is embedding in p_i.

Note that one bit is needed to be added to the secret bit-stream to indicate to the receiver whether the secret is embedded in the decomposed version of I or in that of I'. In the first case, the bit is set to 0 otherwise it is set to 1. The flow chart below displays the embedding procedure of our proposed image-based steganography schemes (EB_SISR, EB_SIM, and EB_IWSIM).

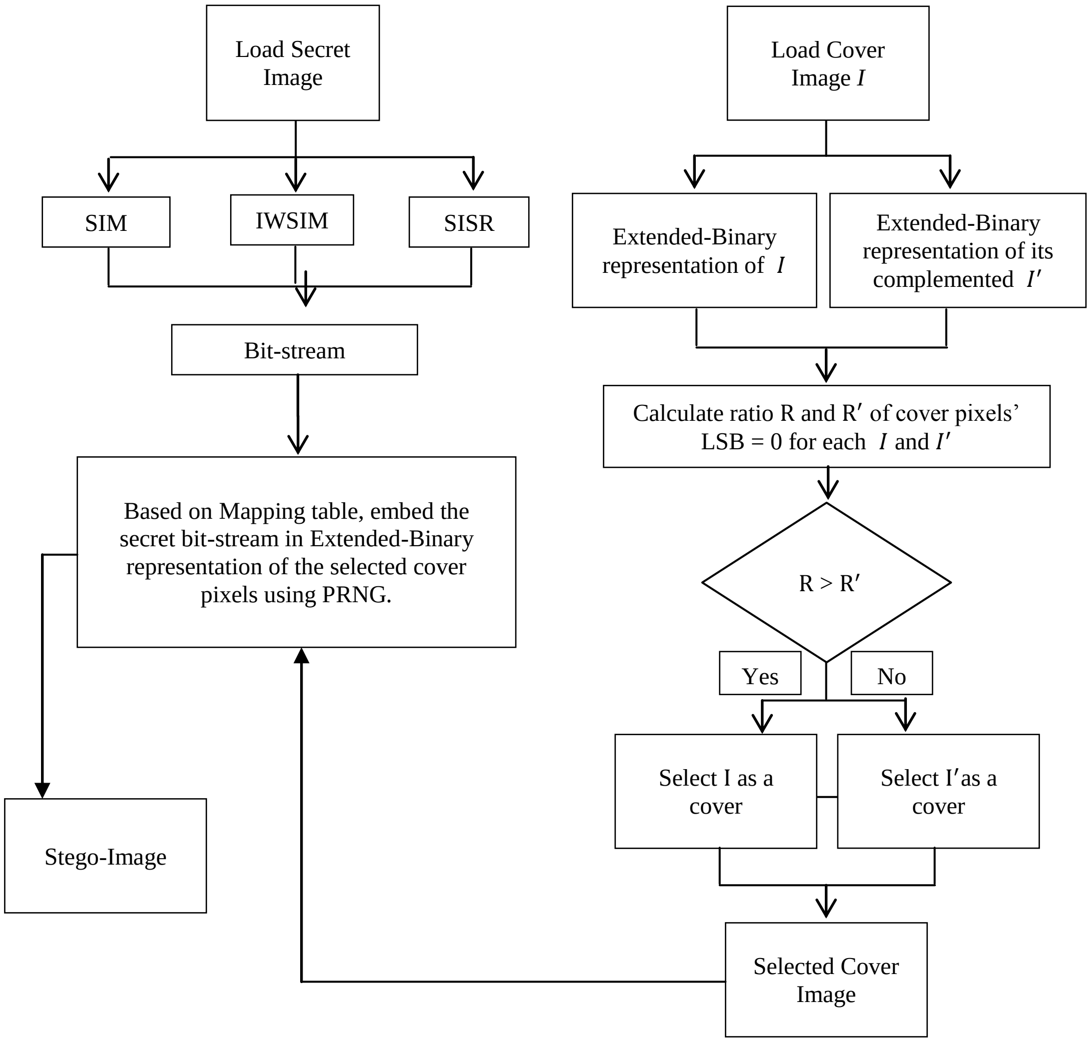


Figure 7- 1: Embedding procedure for our image-based steganography schemes.

Extracting Procedure

On receiving the perceived stego-image S, first the indicator bit should be extracted from the agreed pixel location.

1. If the indicator bit is 0, then extract the secret from S, else extract it from the complement image S'.
2. Use the same PRNG to select the random stego pixel p_i' .
3. Extract the secret bit m_i from the LSB of the Extended-Binary representation of p_i' using the appropriate mapping table.
4. The reverse procedure (decoding) of the SISR, SIM, or IWSIM is applied on the extracted bit-stream to reconstruct the embedded secret image.

For comparison reasons, we also create two other mapping based embedding schemes using the above procedures for the IWSIM pre-processing but instead of the Extended-Binary the cover images will be decomposed by Fibonacci and Lucas. We refer to these schemes as Fib_IWSIM and L_IWSIM. The flow chart below displays the extracting procedure of our proposed image-based steganography schemes.

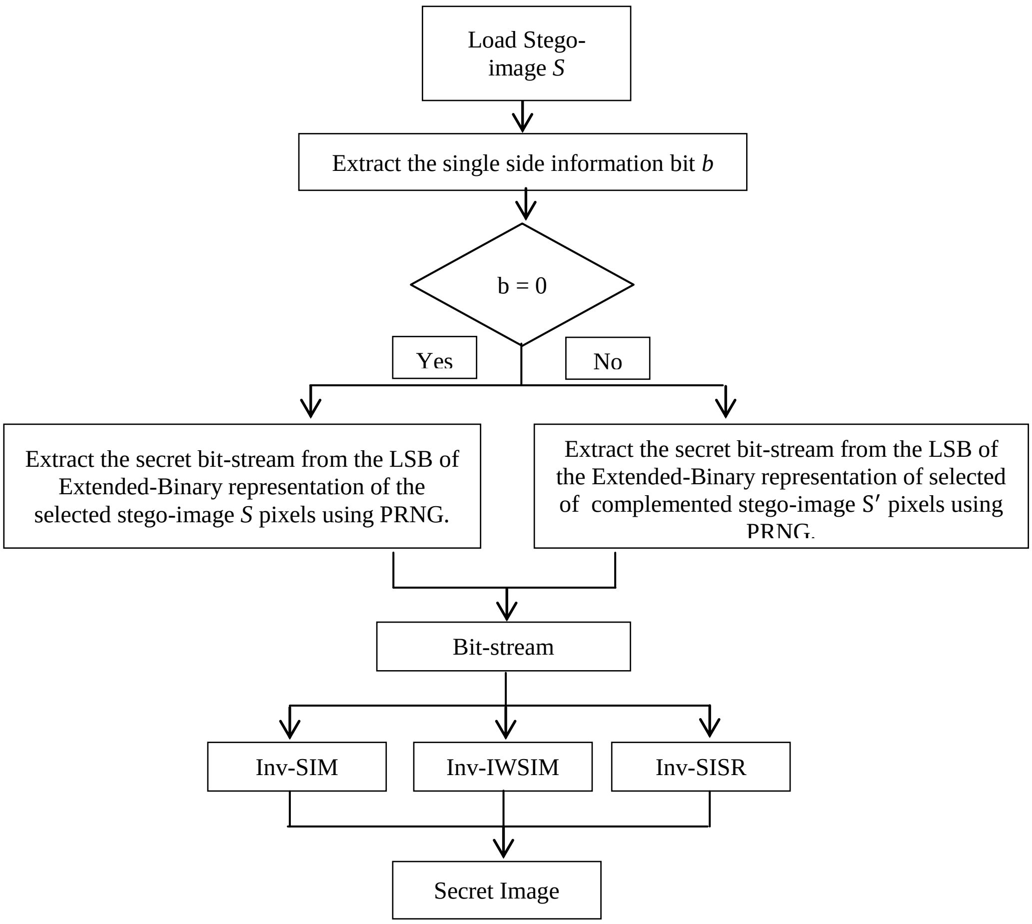


Figure 7-2: Extracting procedure for our image-based steganography schemes.

7.3 Experimental Setup and Results

In this section, the performance evaluation of the proposed image-based steganography schemes (EB_SISR, EB_SIM, EB_IWSIM, Fib_IWSIM, and L_IWSIM) is presented. Four sets of experiments are conducted to evaluate the performance of the proposed steganography schemes: The first is to measure payload capacity, the second is to measure the embedding efficiency, the third is to test the stego-image quality, and the fourth one is to measure the detectability/security of the embedded message. In each of the four experiments, the results are compared with the well-known steganography techniques of LSBR, LSBM, LSBMR, and ILSBMR. The last two techniques have the best embedding efficiency among existing schemes in the literature. Our experimental datasets in these tests are:

1. **SIPI database**: 44 images of size 512 x 512 are used as cover images. For each cover image, we embedded 44 versions of these images but resized to 128 x 256 as secrets resulting in 1936 stego-images for each tested steganography technique including our proposed.
2. **BOSSBase database**: 1000 images of size 512 x 512 are used as cover images. However, embedding 1000 secret images in 1000 cover images is not practical, so we use two standard images, namely Lenna and Jet, are resized to 128 x 256 as secret images, see Figure 7-3. Each of Lenna and Jet images is embedded in each of the 1000 cover images resulting 2000 stego-images for each test.

Note that, the resulted bit-stream contains 78% and 80% of the bits that their value is zero for each image Lenna and Jet respectively after the proposed IWSIM is applied, while resulted bit-stream contains 63% and 72% of the bits that their value is zero for each image Lenna and Jet respectively after the proposed SIM is applied. Moreover, after the proposed SISR is applied on the image Lenna and Jet, the number of resulted bits that represent the image is reduced to 186684 and 185967 bits respectively. Furthermore, the resulted bit-stream from SISR that represent image Lenna contains 55% of bits with zero value, and the resulted bit-stream from SISR that represent image Jet contains 57% of bits with zero value.

Figure 7-3: Secret images: Lenna and Jet.

1. Payload Capacity Evaluation

The capacity of the steganography techniques can be evaluated by measuring the number of the allowed embedded secret bits proportion to the cover image size using equation (2. **3**), in Chapter 2. Table 7-3, displays the capacity of the tested steganography techniques.

Table 7-3: Capacity of the tested steganography techniques.

	LSBR	LSBM	LSBMR	ILSBMR	EB_SISR	EB_SIM	EB_IWSIM	Fib_IWSIM	L_IWSIM
SIPI	1.0	1.0	0.952	0.952	1.0	0.994	0.978	0.978	0.978
BOSSBase	1.0	1.0	0.978	0.978	1.0	0.993	0.979	0.979	0.979

From Table 7-3, it is noticeable that each of the LSBR, LSBM and proposed EB_SISR technique has full capacity, and EB_SIM is only marginally lower. The lowest average capacity (0.952) is achieved by the LSBMR and ILSBMR for the SIPI database. In all other cases, a capacity of around 0.98 is achieved. The loss in capacity by the LSBMR and ILSBMR technique is entirely due to the exclusion of the saturated cover pixel values (i.e. 0 or 255) which account for an average of 4.8% for the SIPI images and 2.2% for the BOSSBase database. Whereas the loss capacity in the cases of EB_SIM and EB_IWSIM is accounted for by the size of the side information needed to send to the receiver, and in the case of EB_IWSIM there is an increase in the number of bits representing coefficients in some Wavelet sub-bands. It is important to realise that in reality, EB_SISR achieves more than full capacity, if we take into account the fact that SISR reduces the secret image bit-stream to 70% of its original size.

2. Embedding Efficiency Evaluation

Theoretically, the probability ratio of pixels that could be modified after message embedding is proportional to the embedded secret image size, and for the EB_IWSIM steganography scheme is 0.338. This is calculated by using equation (7. 1):

$$1 - (R_0 \times R_0') + ((1\text{- } R_0) \times (1\text{- } R_0')) \tag{7. 1}$$

Where R_0 is the ratio of 0:1 in the secret image bit-stream, and R_0' is the ratio of 0:1 of the cover pixels' LSB value.

For instant, on average, the IWSIM achieves 80% ratio of 0:1 in the secret bit-streams while the Extended-Binary LSB plane of the cover images yield a 77% of 0:1 ratio, and therefore, the probability of modifying the cover pixel by the EB_IWSIM scheme is:

$$1 - (0.80 \times 0.77) + ((0.20 \times 0.23)) = 0.338$$

The probability of modifying cover image pixels after embedding secret images using traditional LSBR scheme is:

$$1 - (0.5 \times 0.5) + ((0.5 \times 0.5)) = 0.5$$

This is because on average secret images bit-streams have a 50% ratio of 0:1 (see Table 7-4), and the same is true about the LSB plane of the cover images (decomposed using traditional binary decomposition), see Figure 6-2.

Table 7-4: Ratio of 0:1 in the binary representation of the tested secret images.

Databases	0:1 Ratio in original image
SIPI	0.494
BOSSBase	0.540

On the other hand, we calculated the actual ratios of modified pixels (to the embedded payload) for each of the 8 tested schemes. Figure 7- 4, Figure 7- 5, and Figure 7- 6 present the average of these ratios for the stego-images in the SIPI database, BOSSBase database when the Lenna secret image is embed, and BOSSBase database when the Jet secret image is embed, respectively. The corresponding embedding efficiency charts are presented in Figure 7-7, Figure 7-8, and Figure 7-9, respectively.

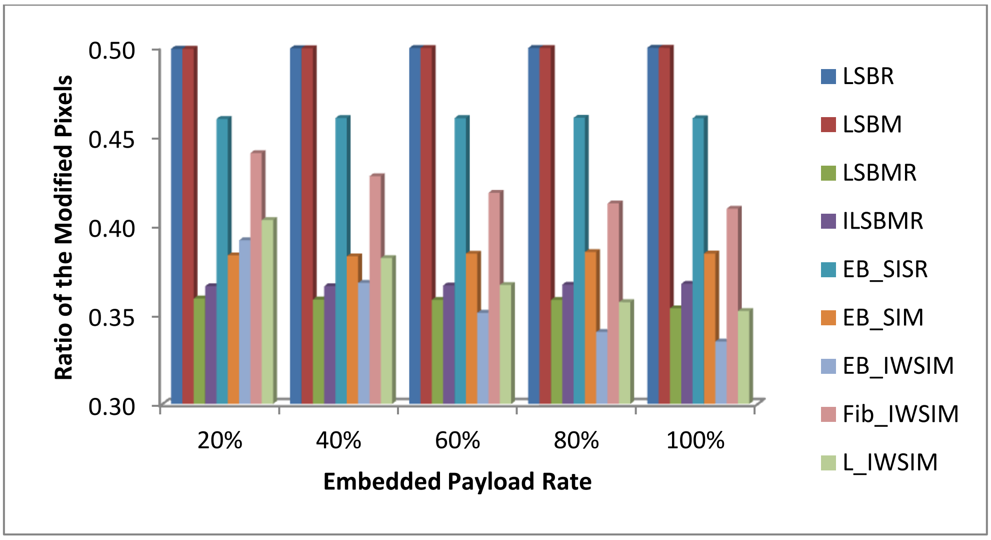


Figure 7- 4: Ratio of modified pixels for the SIPI experimental images.

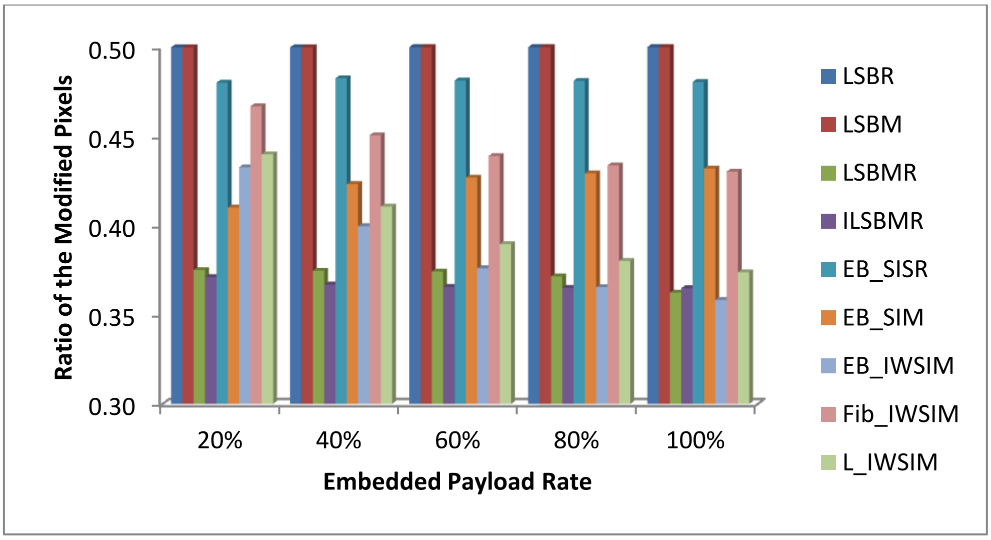


Figure 7- 5: Ratio of modified pixels of the cover BOSSBase image when Lenna is the secret image.

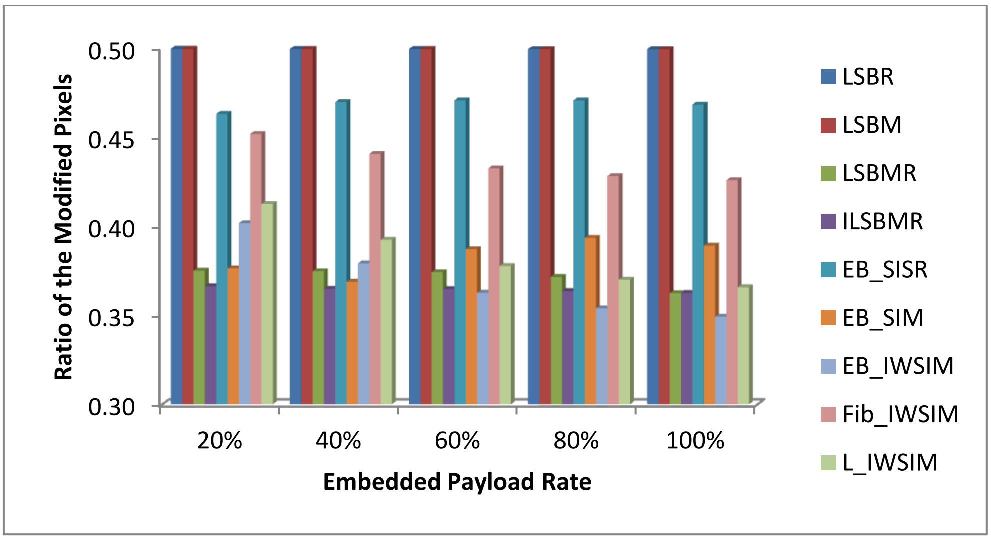


Figure 7- 6: Ratio of modified pixels of the cover BOSSBase image when Jet is the secret image.

From the above charts, one can see that the EB_IWSIM outperforms all other schemes for the payload of 60% or more, but it is outperformed by the LSBMR and ILSBMR at the lower embedding rates. Note that the EB_IWSIM embedding scheme at the lower embedding rate, the effect of including the side information is the main reason for this low performance. These results, also explain a similar pattern of performance of the various schemes with regards to the embedding efficiency as displayed in Figure 7-7 to Figure 7-9. Again EB_IWSIM outperforms all other schemes at embedding rates of 60% and above.

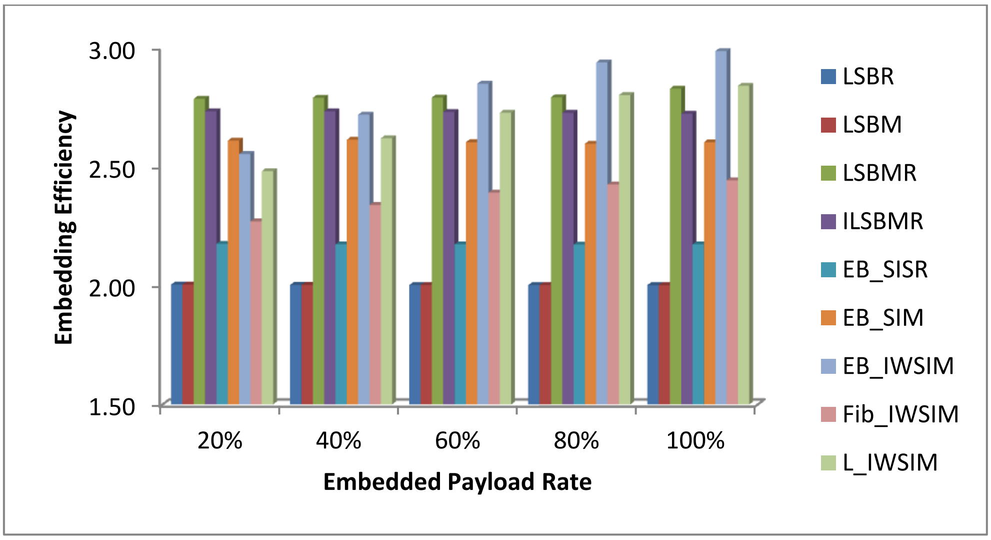


Figure 7-7: Embedding efficiency for the SIPI database.

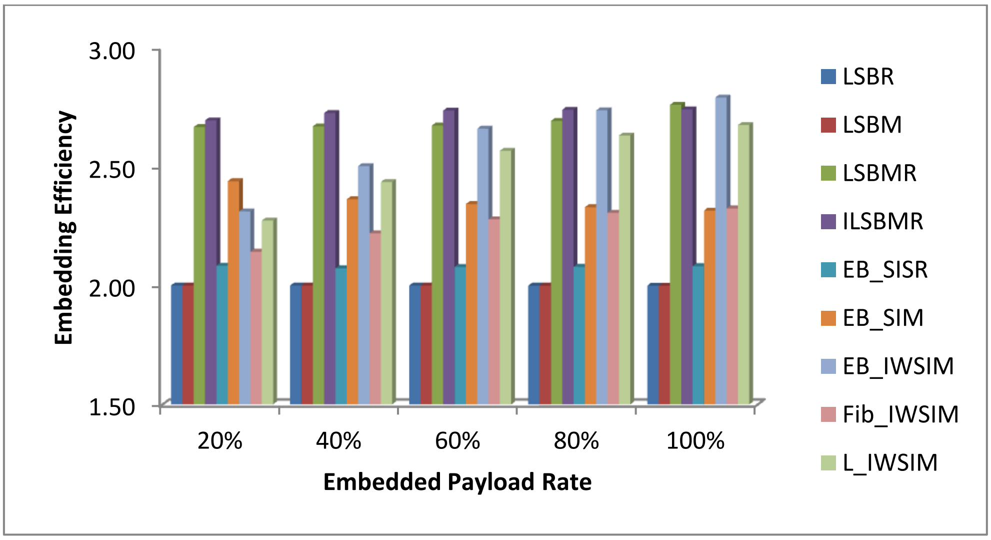


Figure 7-8: Embedding efficiency for the BOSSBase database when the secret image Lenna is embedded.

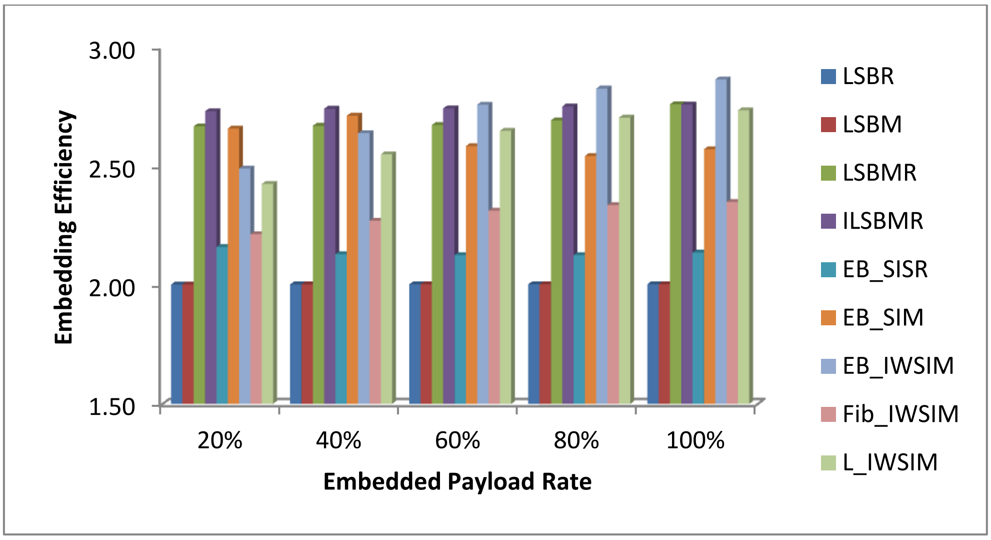


Figure 7-9: Embedding efficiency for the BOSSBase database when the secret image Jet is embedded.

3. Stego-Image Quality Evaluation

We evaluated the stego-image quality for all the above nine embedding schemes in terms of the PSNR values with respect to the original cover images. The results shown in Figure 7-10, Figure 7-11, and Figure 7-12, and present the average PSNR value for the all the experimental data.

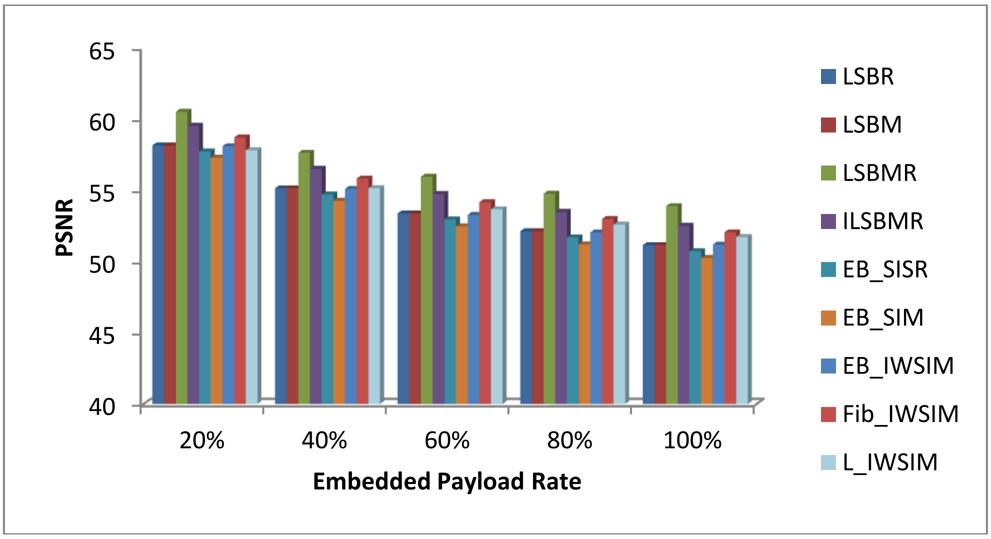


Figure 7-10: Average PSNR for the tested steganography schemes for the SIPI database.

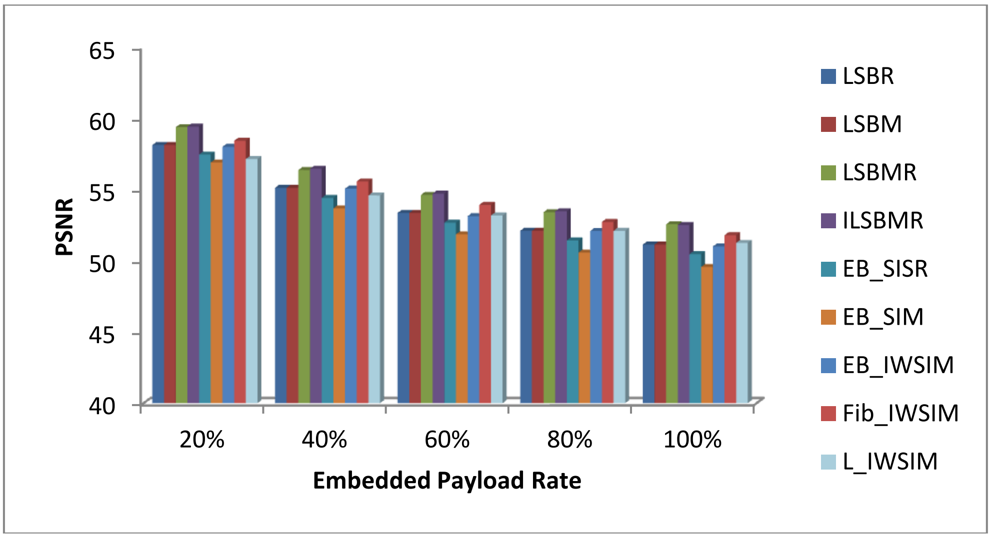


Figure 7-11: PSNR for the BOSSBase stego images when the secret image Lenna is embedded.

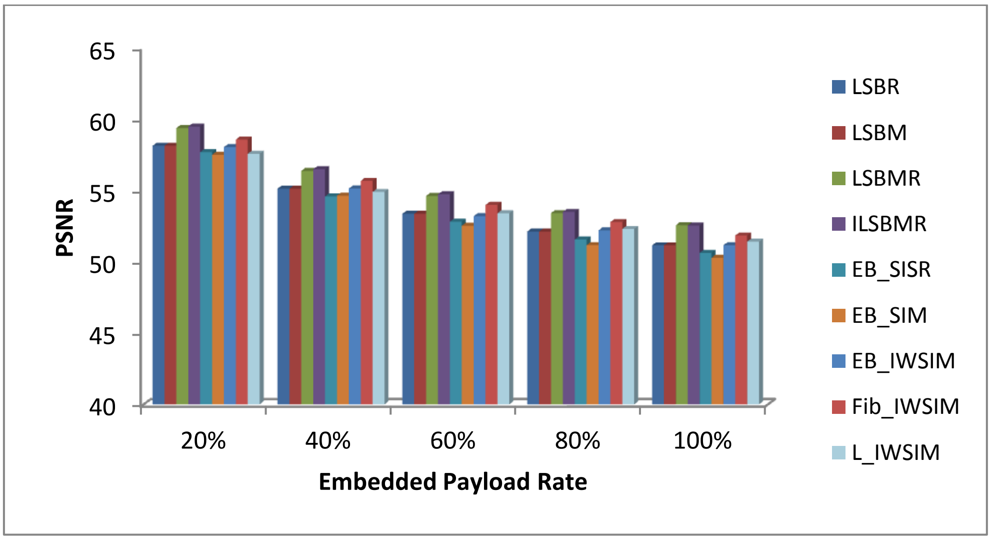


Figure 7-12: PSNR for the BOSSBase stego images when the secret image Jet is embedded.

Clearly, the PSNR of the LSBMR and ILSBMR scored the highest value compared to all other schemes including ours at all embedding rates. Moreover, the PSNR of the EB_IWSIM is slightly higher than EB_SIM and EB_SISR at all embedding rates. The Fib_IWSIM performance is reasonably near that of the LSBMR and ILSBMR schemes. Note that 25% of the lowest 3 bit-planes of the Extended-Binary decomposed cover pixels are 100 and if the secret bit value is 1, the cover pixels value will be changed by 2, i.e. 12.5% of the cover pixels may change by 2. The reason of that the PSNR of the EB_IWSIM is higher than the PSNR of the EB_SIM and EB_SISR is the ratio of 0:1 of the IWSIM is higher than SIM and SISR and this reduces the probability of changing cover pixel value by 2.

4. *Detectability Evaluation*

In this section, we report on experiments conducted to test the robustness against steganalysis tools of the various mapping table schemes that are used for embedding pre-processed secret image bit–streams in cover images whose pixels are decomposed in various ways. Seven well-known steganalysis detectors have been used to evaluate the detectability of the proposed steganography technique. These steganalysis techniques are PoV, RS, DIH, WS, RWS, LSBMS, and SRM that were fully described in Chapter 2 and reviewed in Chapter 3.

Robustness Against RS Detector

Figure 7- 13, Figure 7-14, and Figure 7-15 are presenting the RS diagram for the tested steganography techniques for each SIPI database, BOSSBase database when the Lenna secret image is embedded, and BOSSBase database when the Jet secret image is embedded, respectively. Firstly, these results confirm what is already known that LSBR is not robust against the RS detector. Whereas all other tested embedding schemes including ours, there are no differences between RM and RM-, SM and SM-, such differences, and thereby demonstrating robustness against RS detector.

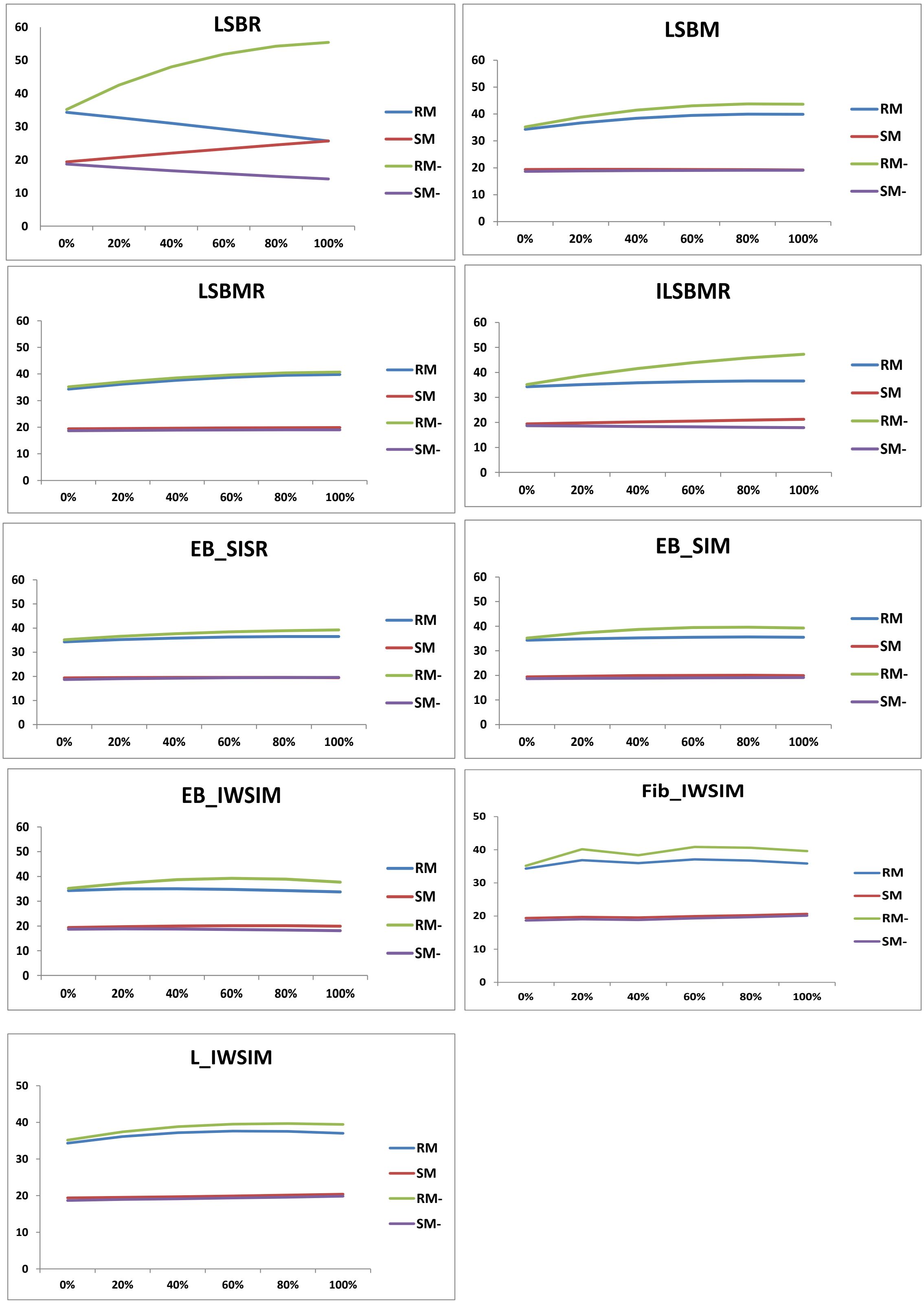


Figure 7- 13: RS diagram for all tested steganography schemes for SIPI database.

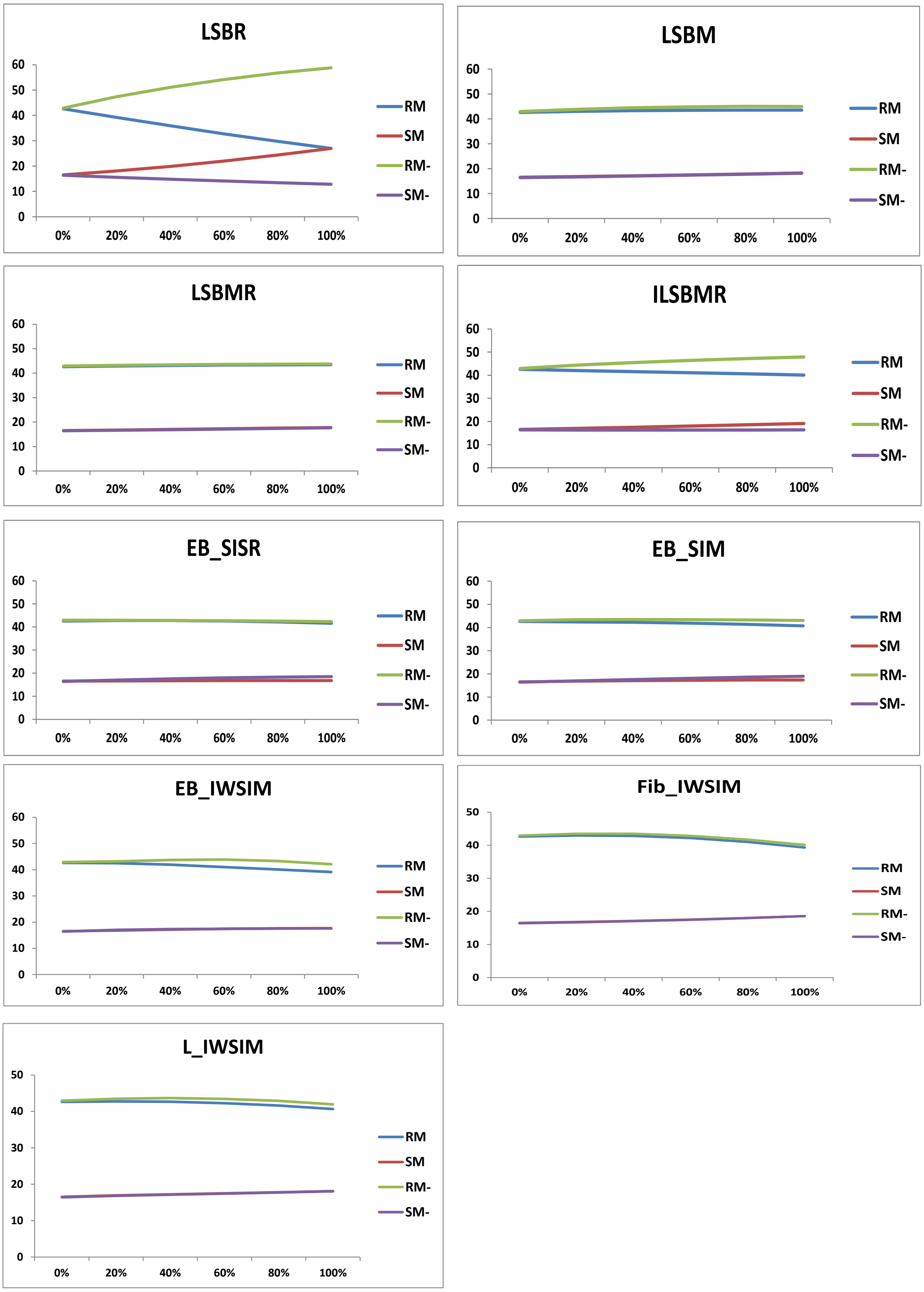


Figure 7-14: RS diagram for all tested schemes for the BOSSBase database when Lenna image is embedded.

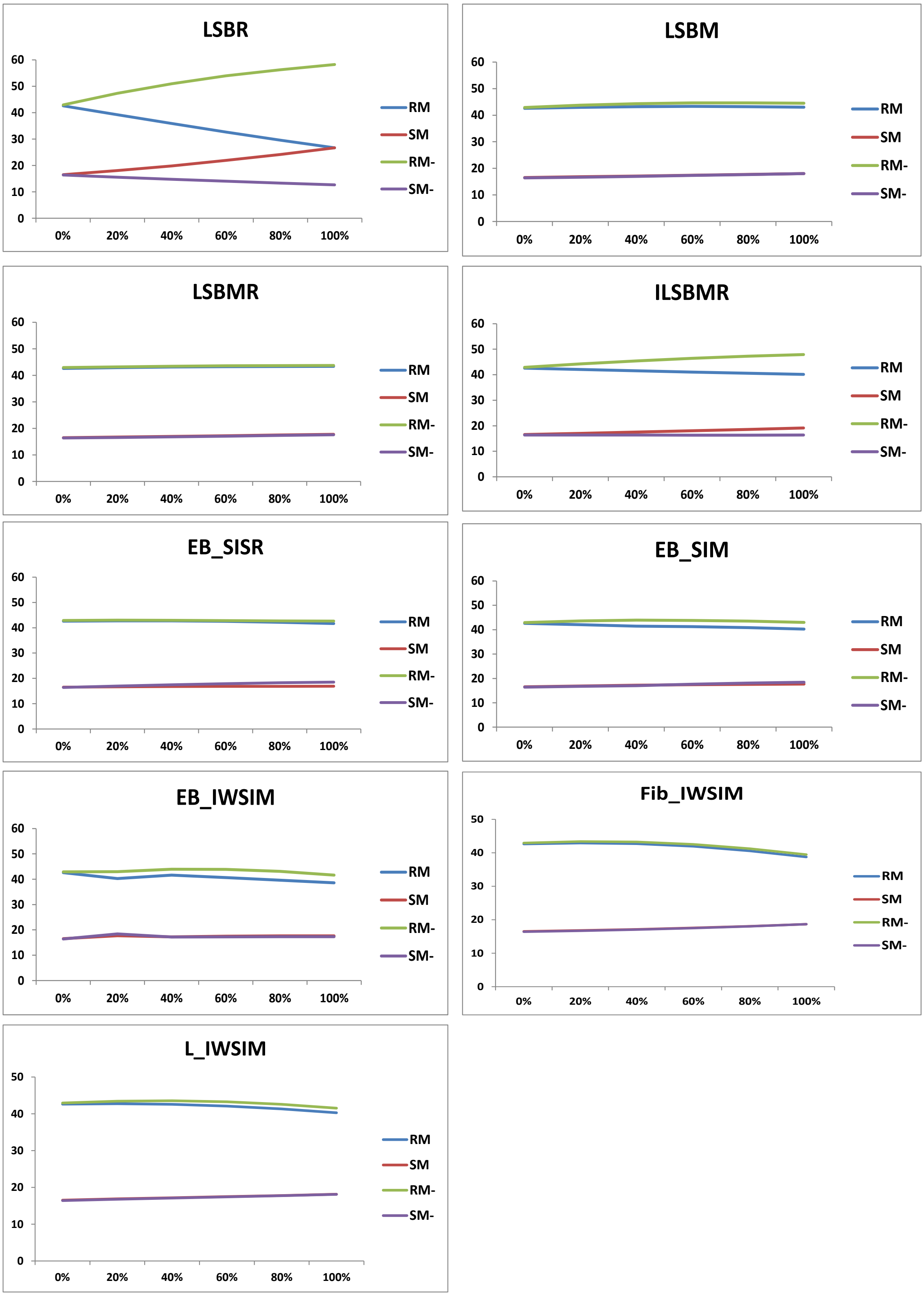
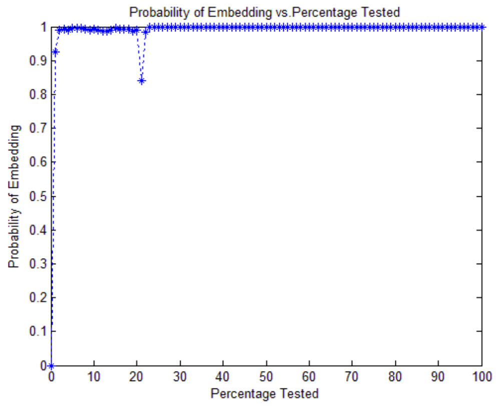


Figure 7-15: RS diagram for all tested schemes for the BOSSBase database when Jet image is embedded.

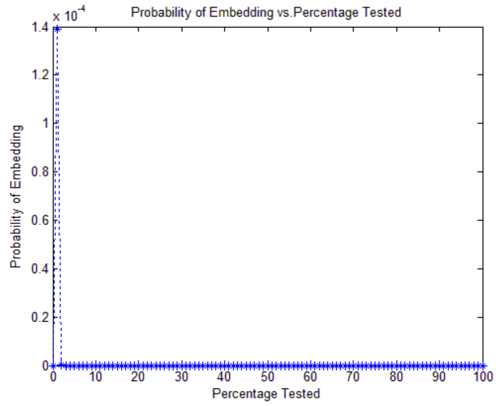
Robustness Against PoV Detector

We repeat the same experiments, done for RS, but to test robustness against the PoV detector, and the corresponding represents the PoV attack for only one stego-image (first stego-image in each database) as a representative sample, but in the appendix we put PoV diagram for 5 randomly selected stego-images for each embedding scheme and both databases. Clearly, all schemes, except LSBR, are undetectable by PoV detector at all payload rates, i.e. are robust against PoV.

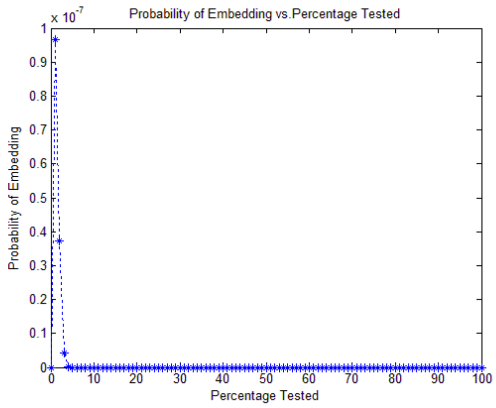
LSBR



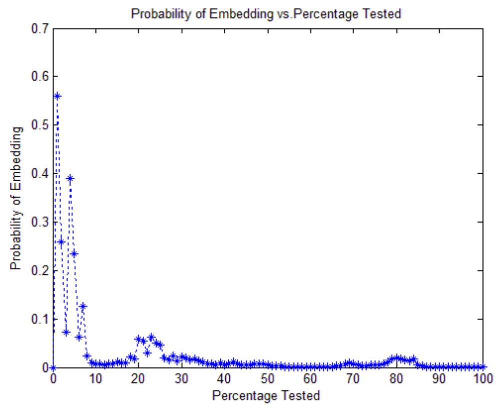
LSBM



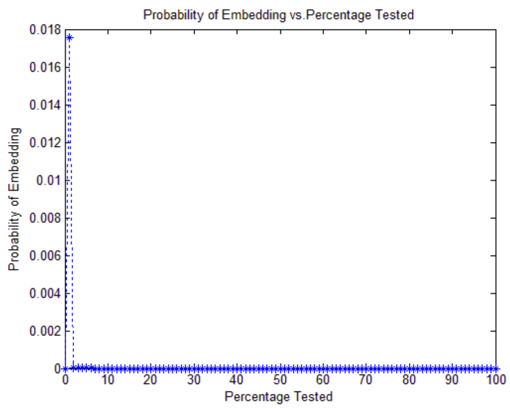
LSBMR



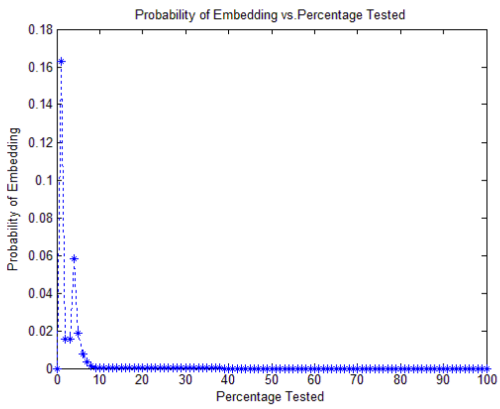
ILSBMR



EB_SISR



EB_SIM



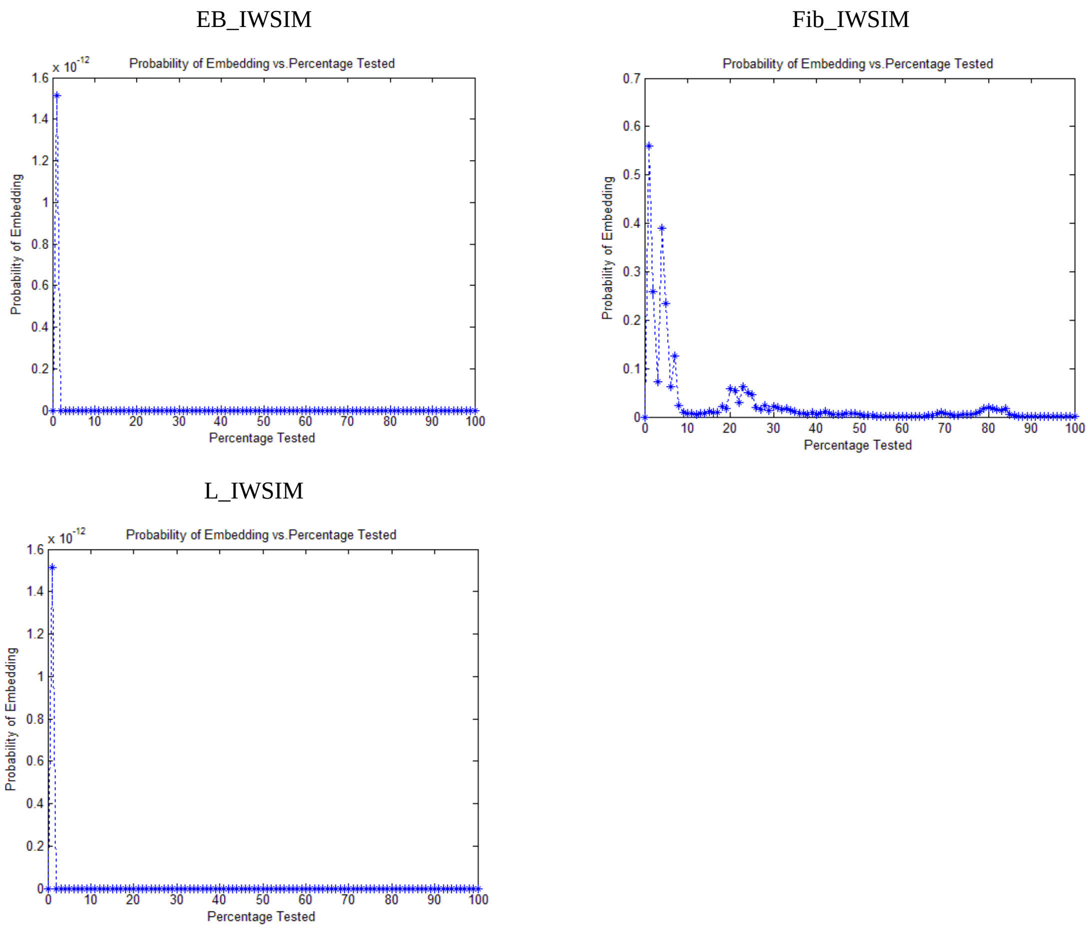
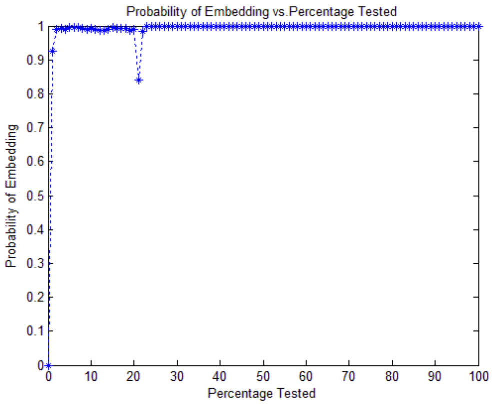
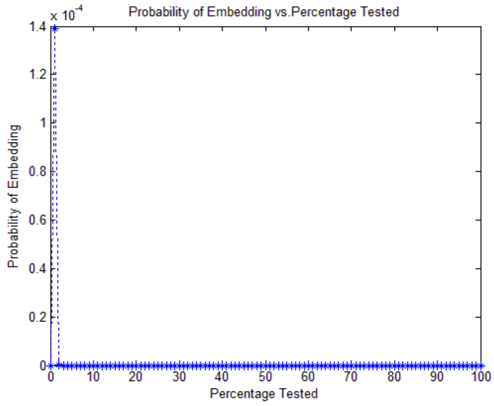


Figure 7-16: PoV diagram for sample cover image from SIPI database.

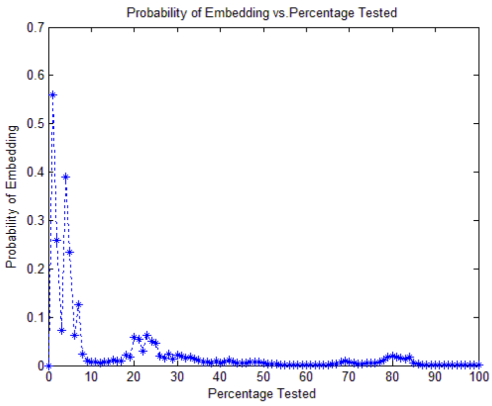
LSBR



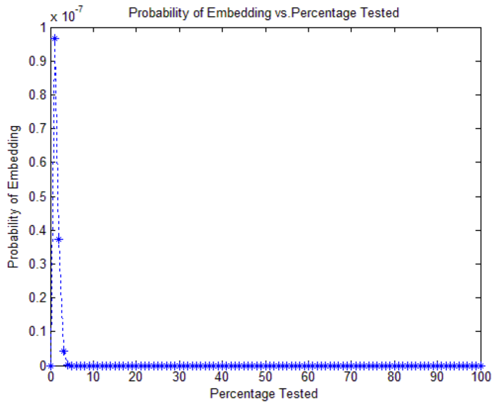
LSBM



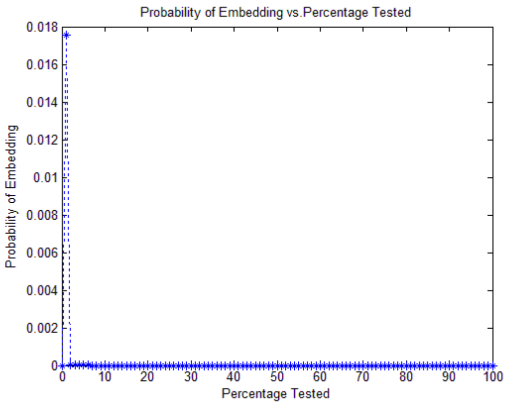
LSBMR



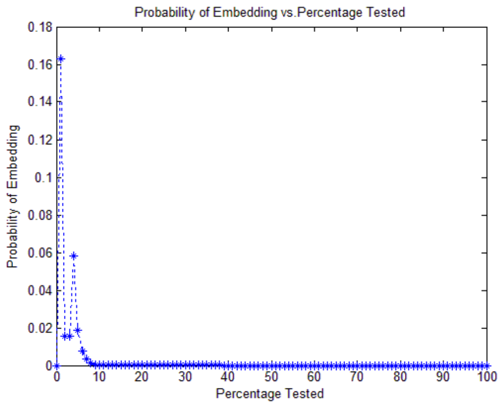
ILSBMR



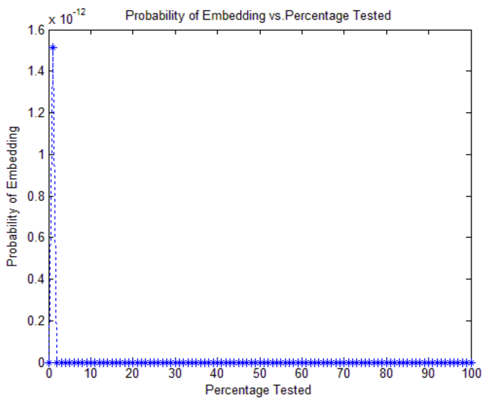
EB_SISR



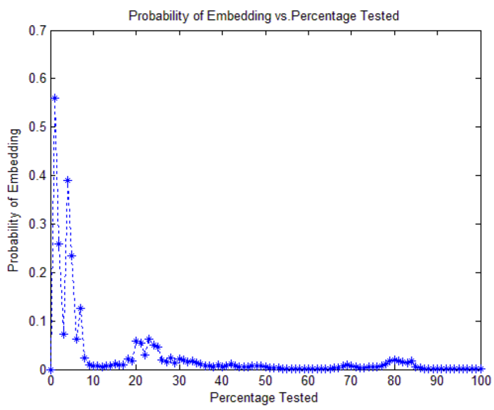
EB_SIM



EB_IWSIM



Fib_IWSIM



L_IWSIM

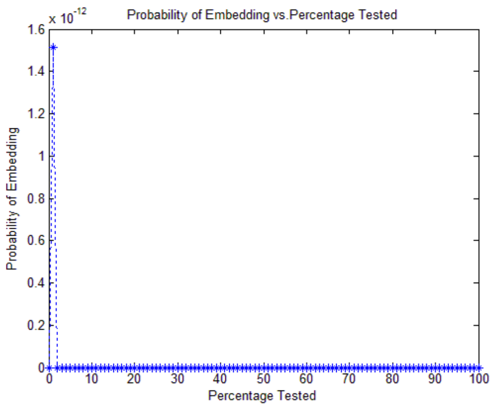
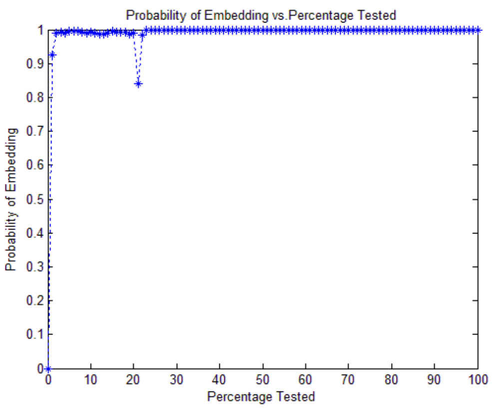
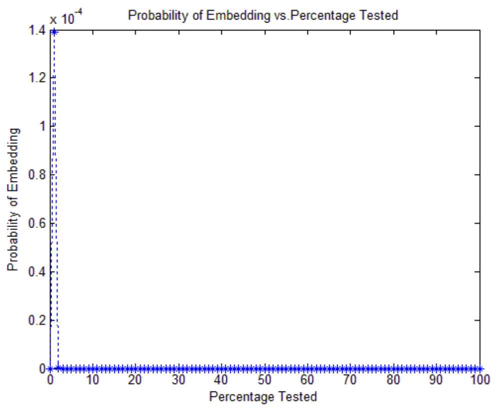


Figure 7-17: PoV diagram for sample cover image from BOSSBase when the Lenna image was embedded.

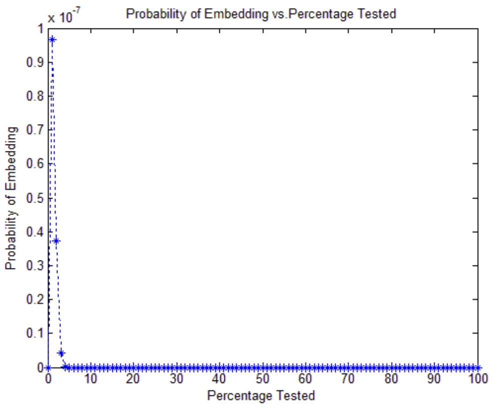
LSBR



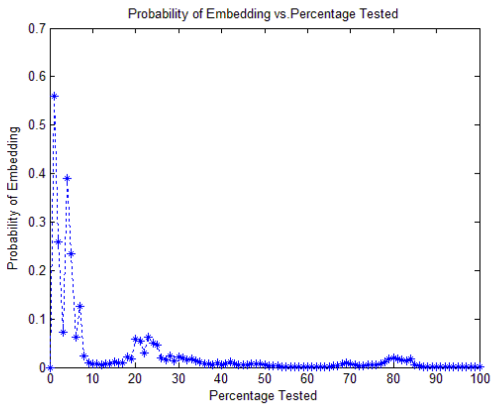
LSBM



LSBMR



ILSBMR



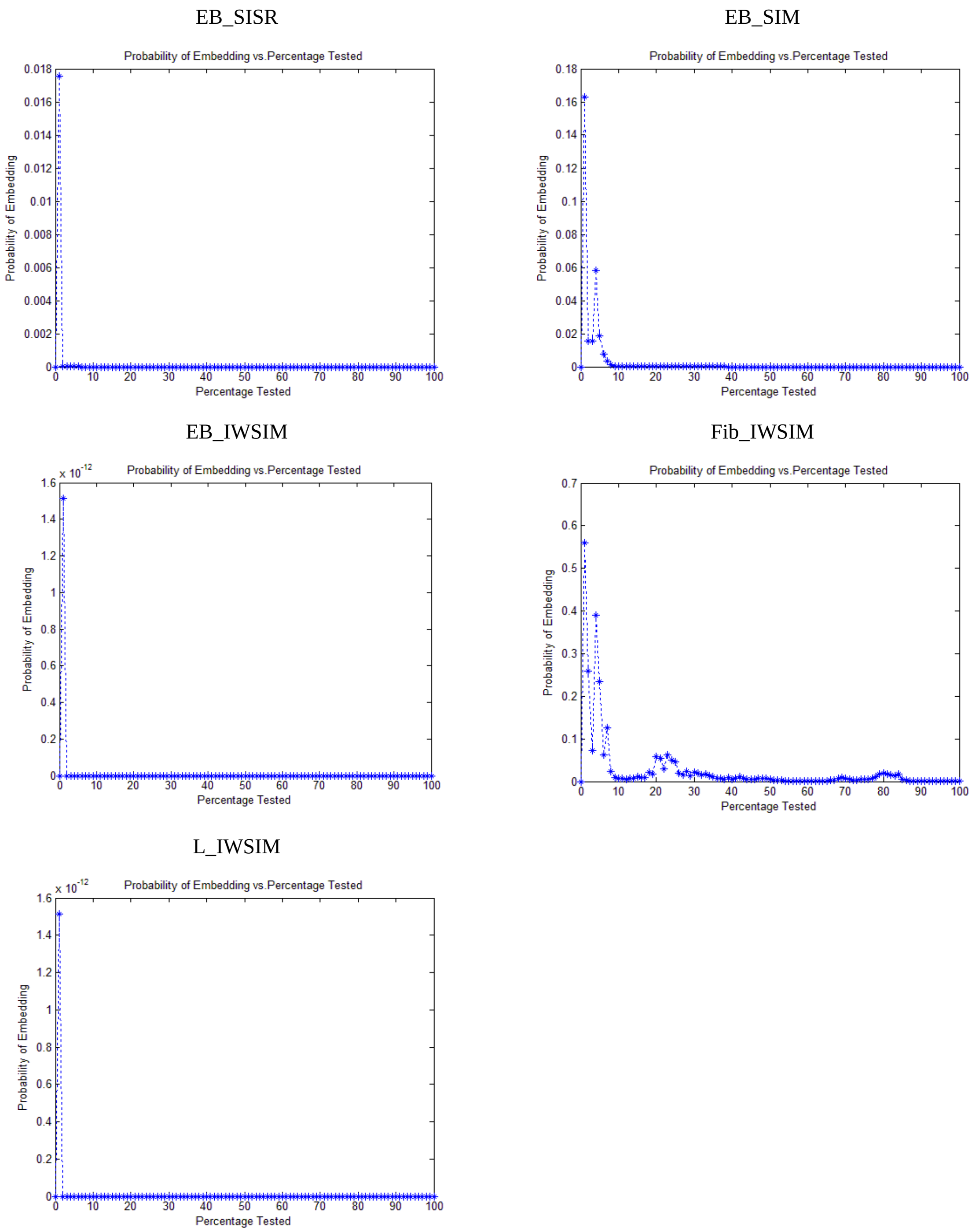


Figure 7-18: PoV diagram for sample cover image from BOSSBase when the Jet image was embedded.

Robustness Against DIH Detector

We now report on the results of testing the same set of embedding schemes, as in the above sections, against the difference image histogram DIH detector using the same set of experimental cover and secret images. Figure 7-19, Figure 7-20, and Figure 7-21 are presenting the average values of the estimated length of the embedded secret image bit-stream at the different payload rates. Again, these results demonstrate that the LSBM and all mapping based embedding including the Fib_IWSIM and L_IWSIM are undetectable by the DIH tool at all embedding rates with LSBM being the best performing scheme but only marginally better than our schemes. Again these experiments re-affirm the known fact that the LSBR is detectable by the DIH, while the DIH is able just to detect the ILSBMR at high embedding rate but not with high confidence.

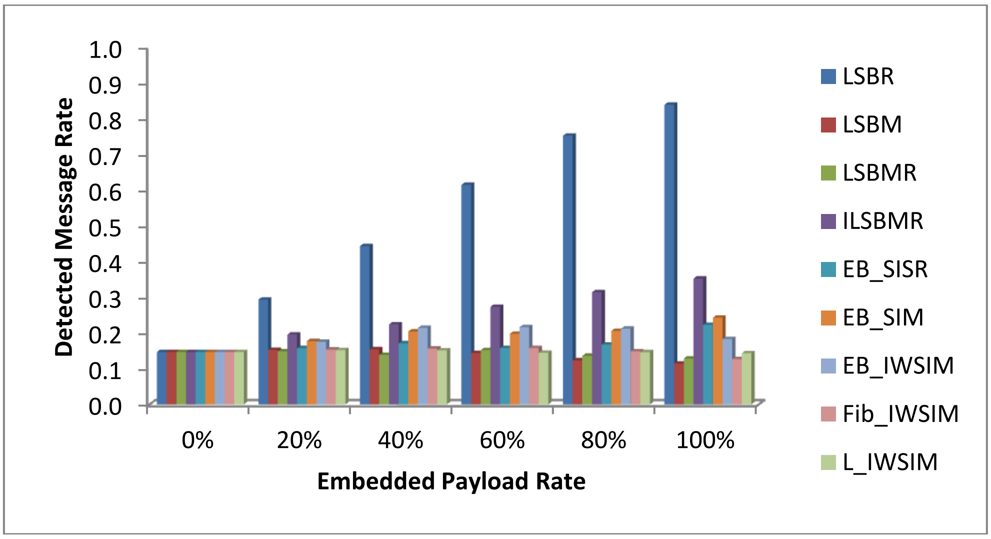


Figure 7-19: DIH steganalysis for all tested steganography schemes for SIPI database.

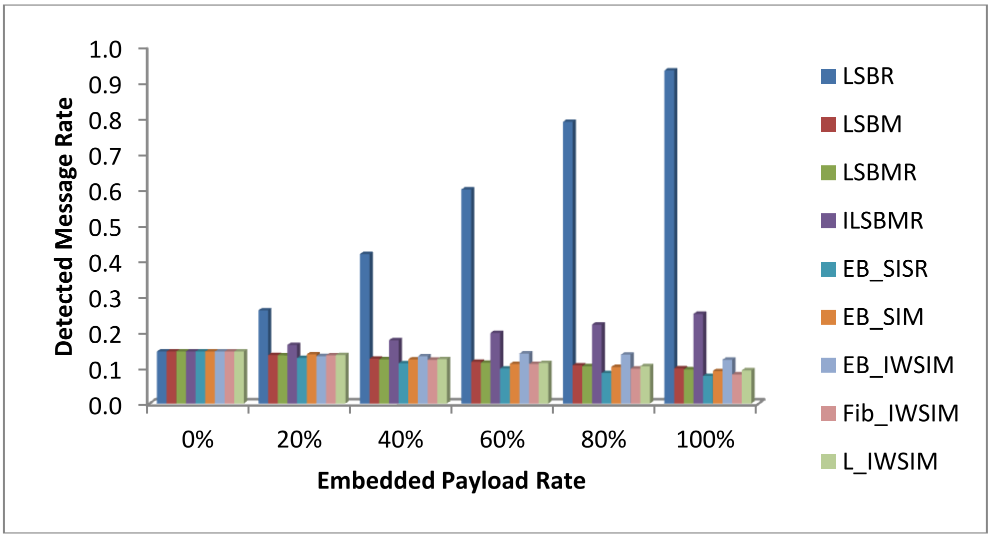


Figure 7-20: DIH steganalysis for BOSSBase database when Lenna image was embedded.

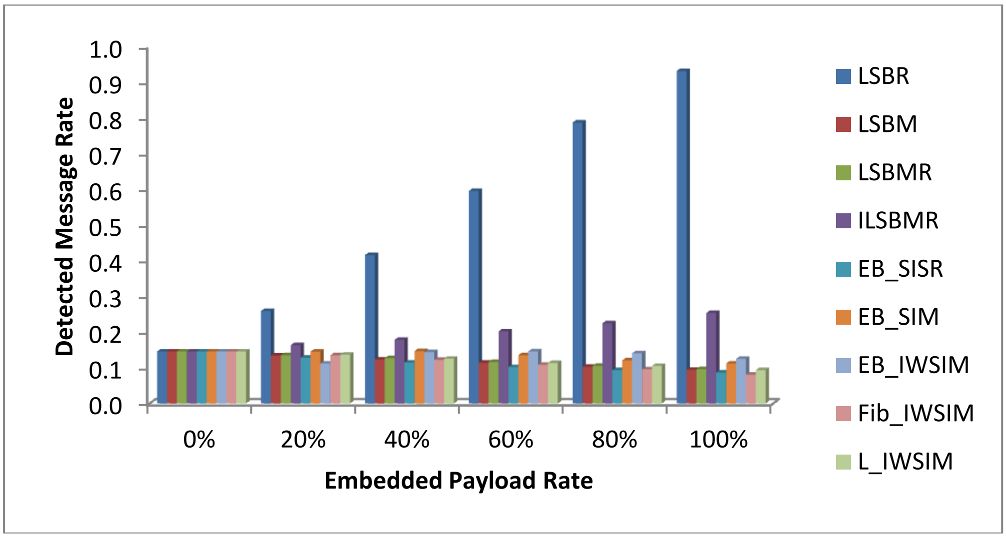


Figure 7-21: DIH steganalysis for BOSSBase database when Jet image was embedded.

Robustness Against WS Detector

We now report on the results of experiments to test the same set of embedding schemes, as in the above sections, against the weighted stego WS detector which estimates the length of embedded secret by solving a least square optimisation problem applied to versions of the input stego-image. For testing, we use the same set of experimental cover and secret images. Figure 7-22, Figure 7-23, and Figure 7- 24 are presenting the average values of the estimation results for the stego-images obtained from the SIPI database, BOSSBase database when the Lenna secret image is embedded, and BOSSBase database when the Jet secret image is embedded, respectively. The pattern of these results almost mimic those obtained when testing for robustness against DIH, i.e. the LSBM and all mapping based embedding including the Fib_IWSIM and L_IWSIM are undetectable by the WS tool at all embedding rates with LSBM being the best performing scheme but only marginally better than our schemes. Moreover, these experiments re-affirm the known fact that the LSBR is detectable by the WS, and WS is able just to detect the ILSBMR at high embedding rate but not with high confidence. WS outputs a slightly higher estimated secret length when EB_IWSIM stego-image is tested, compared to our other schemes, but this is still within the margin of error.

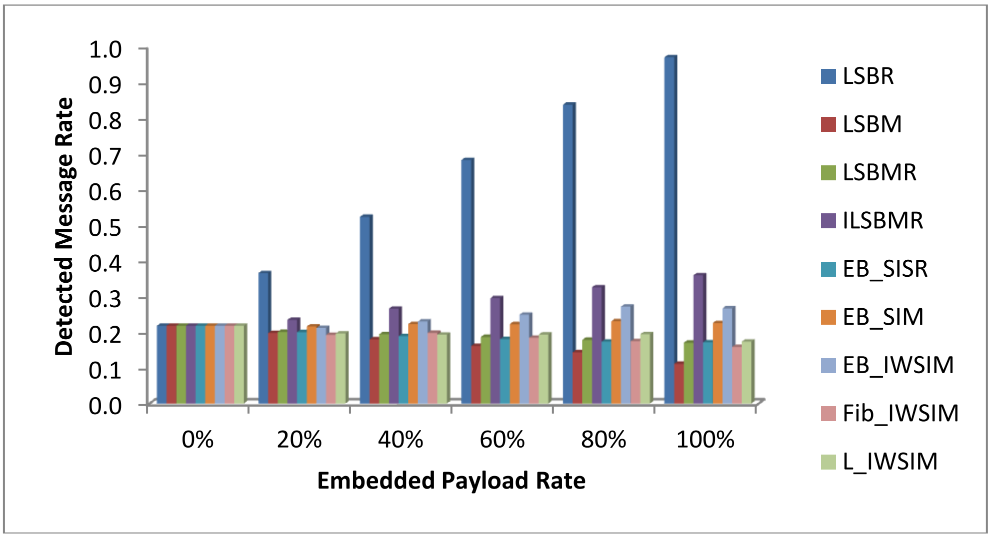


Figure 7-22: WS steganalysis for stego-images in SIPI database.

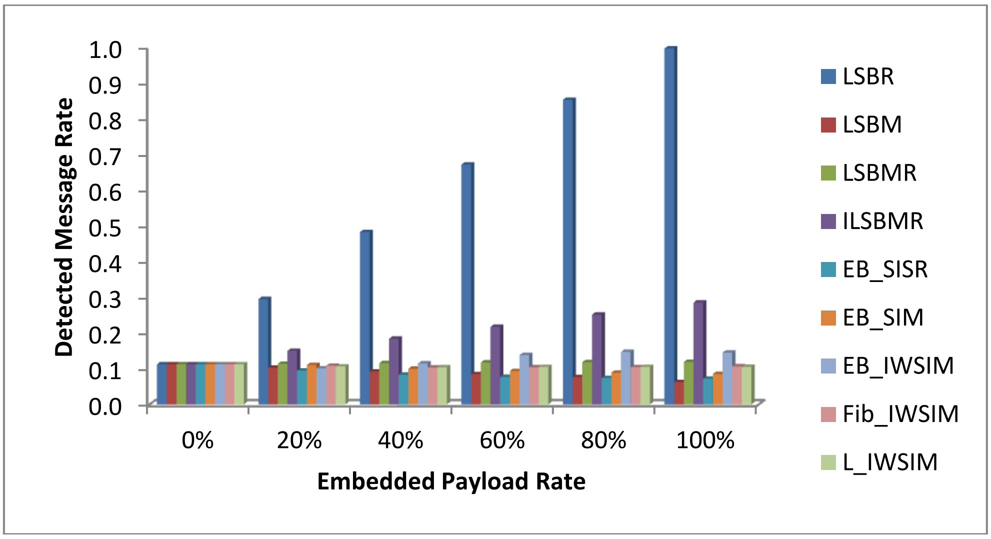


Figure 7-23: WS steganalysis for BOSSBase stego-images when Lenna image was embedded.

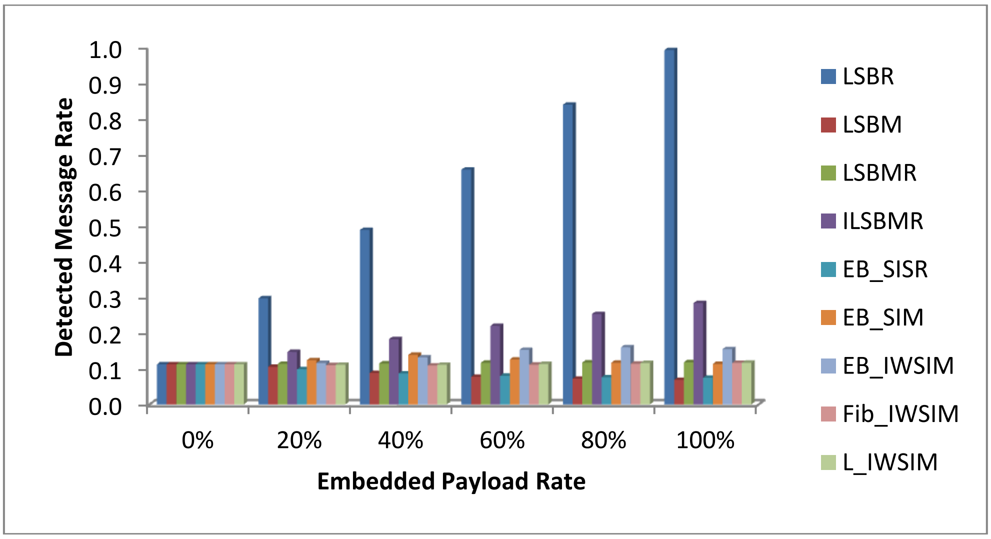


Figure 7- 24: WS steganalysis for BOSSBase stego-images when Jet image was embedded.

Robustness Against RWS Detector

Having shown the robustness of our schemes against the WS tool, we next investigated robustness against the revised version of WS. Here, we present the results of investigation experiments to test the same set of embedding schemes, as in the above sections, against the revised WS detector using the same set of experimental cover and secret images. Figure 7-25, Figure 7-26, and Figure 7-27 are presenting the average values of the estimation results of the flipped cover pixels' LSB for tested steganography schemes for each SIPI database, BOSSBase database when the Lenna secret image is embedded, and BOSSBase database when the Jet secret image is embedded, respectively. The pattern of these results is very similar to those obtained when testing for robustness against WS, except that the LSBM is the best performing scheme. Moreover the LSBM and all mapping based embedding including the Fib_IWSIM and L_IWSIM are undetectable by the RWS tool, at all embedding rates, with marginal differences between these schemes. The RWS predicts a slightly higher estimated secret length for EB_IWSIM than our other schemes. Note that, for all Extended-Binary decomposed schemes, the mapping table embedding may result in changing the pixel values by 2.

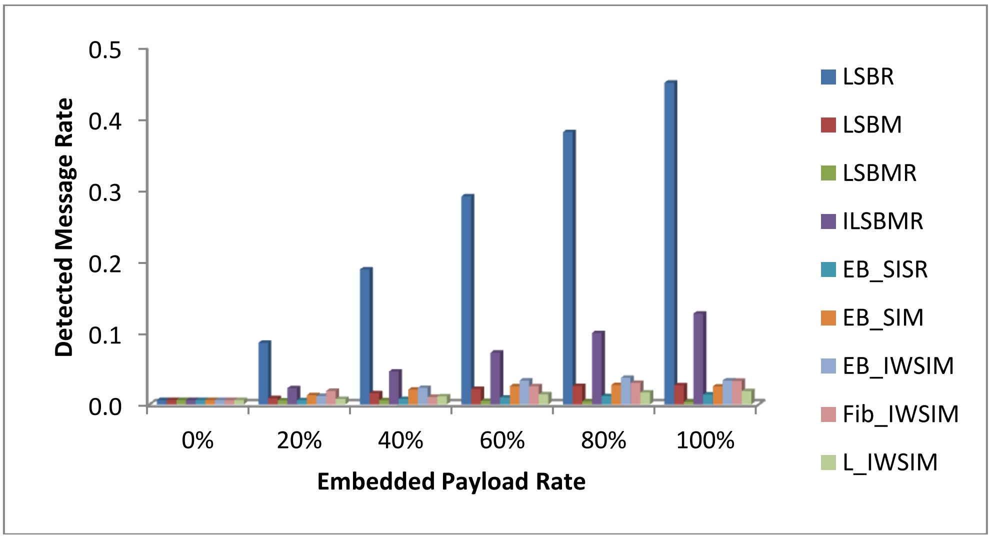


Figure 7-25: RWS steganalysis for all tested steganography schemes for SIPI database.

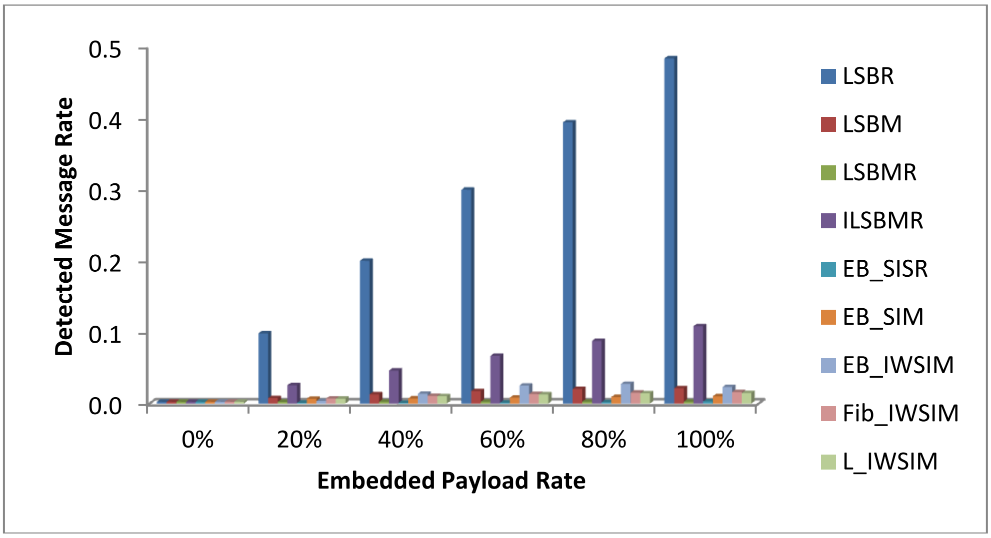


Figure 7-26: RWS steganalysis for BOSSBase database when Lenna image was embedded.

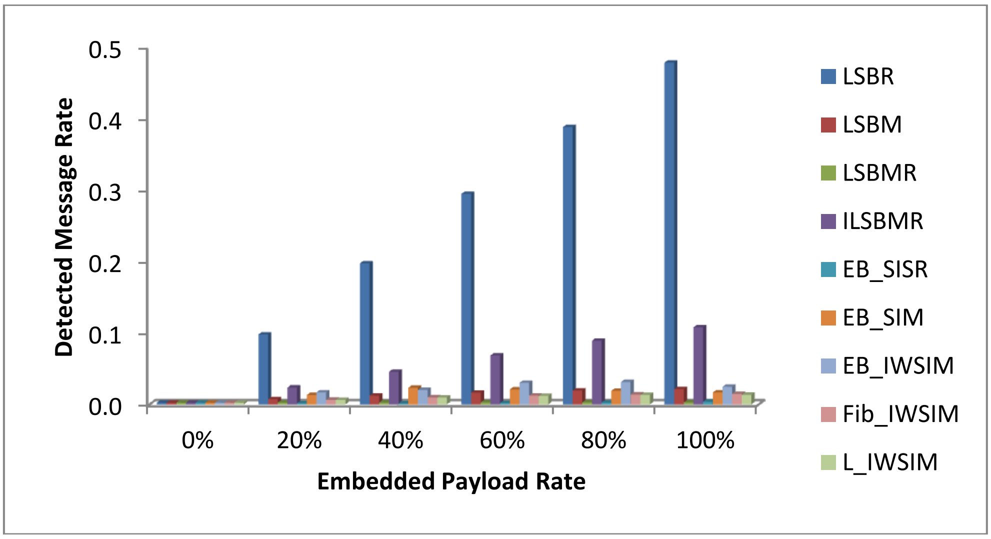


Figure 7-27: RWS steganalysis for BOSSBase database when Jet image was embedded.

Robustness Against LSBMS Detector

The LSBMS is a targeted tool but was designed to detect the LSB matching based embedding techniques. Here we focus on evaluating the robustness of the same set of embedding schemes, tested in this chapter, against the LSBMS. In these experiments, we input the same set of stego-images obtained from the experimental cover and secret images. The tool will return the number of images that are detected as stego-images, i.e. containing a payload. Figure 7-28, Figure 7- 29, and Figure 7- 30 sketches the average curves of the ratio of the detected stego-images proportion to the number of images in the tested database for tested steganography schemes for each SIPI database, BOSSBase database when the Lenna secret image is embedded, and BOSSBase database when the Jet secret image is embedded, respectively. Clearly, all our EB-based schemes

(EB_SISR, EB_SIM, and EB_IWSIM) as well as Fib_IWSIM, and L_IWSIM are robust against the LSBM and are less detectable even than cover images. All other schemes are outperformed by our schemes, but LSBMR is best among them in that few are detected as not cover images at high embedding rate.

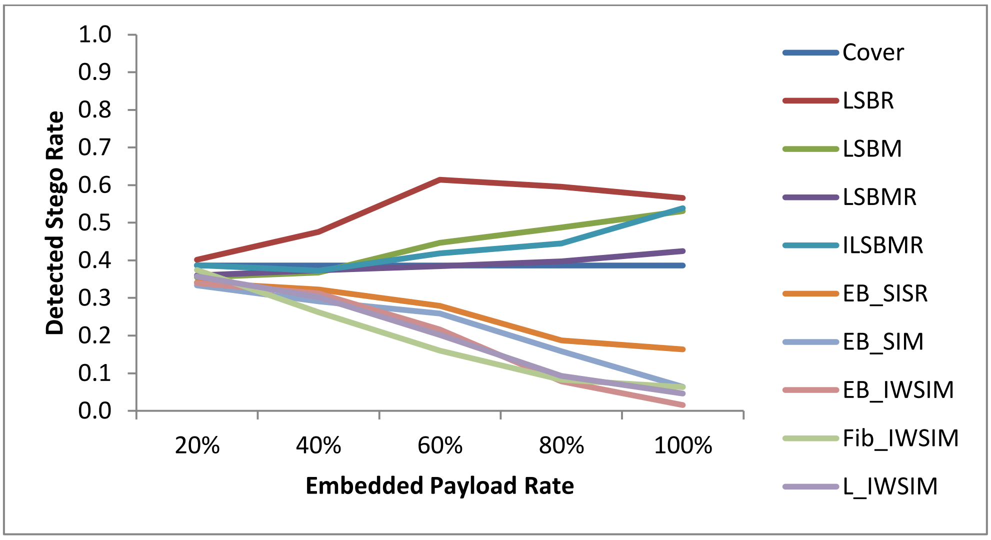


Figure 7-28: LSBMS steganalysis for SIPI database.

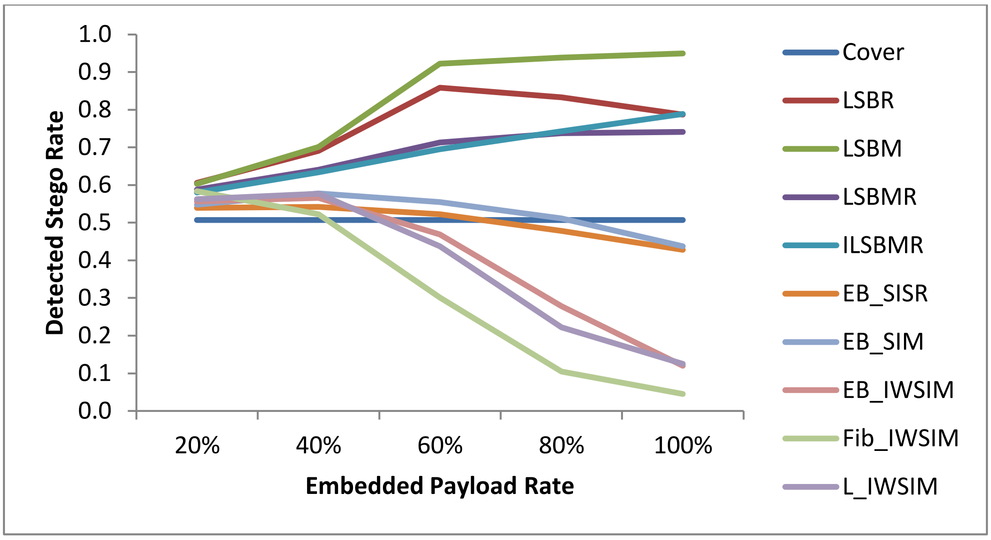


Figure 7- 29: LSBMS steganalysis for BOSSBase database when Lenna image was embedded.

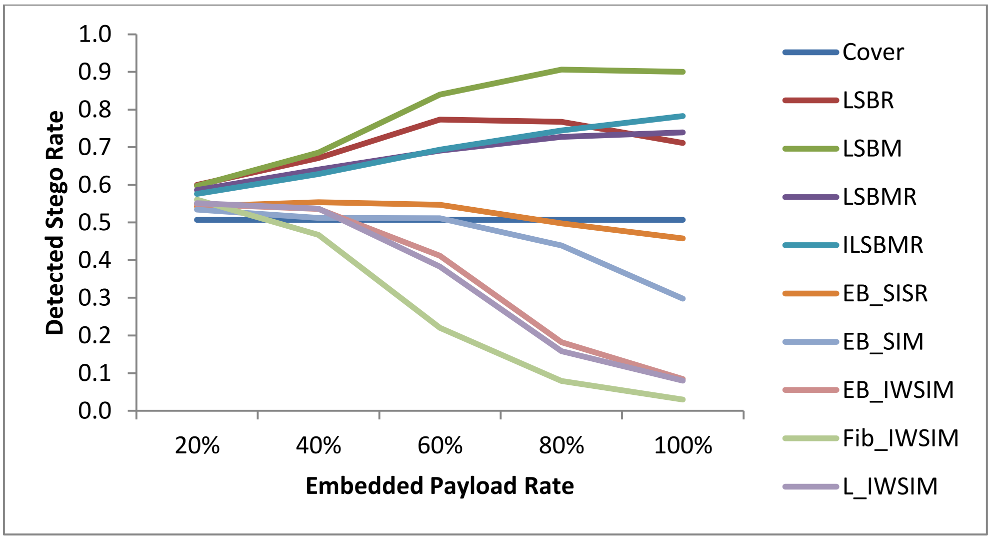


Figure 7- 30: LSBMS steganalysis for BOSSBase database when Jet image was embedded.

Robustness Against SRM Detector

Finally, we shall test the performance of all the above schemes against the only known blind steganalysis tools, namely the SRM tool. We repeated the same set of experiments above but this time to test robustness against the SRM. Note that this tool requires the input of a large set of cover images together with their corresponding stego-images. In the case of BOSSBase, we input the selected 1000 cover images, the corresponding 1000 stego-images after embedding Lenna, as well as the corresponding 1000 stego-images when Jet is embedded. For the 44 images in SIPI, we have 1936 stego-images and we repeated the same original 44 cover images 44 times to make up a total of 1936 cover images. The SRM tool uses half of the cover image set together with the same number of stego-images for training and the rest for testing. The tool is a binary based classification where an input test image is declared as a cover or a stego using a large number of local distortion features. Figure 7-31, Figure 7-32, and Figure 7- 33 show the average ratio of the detected stego-images to the number of tested images. In all figures, it is obvious that all tested schemes are detected by SRM. Obviously, SRM can easily find sufficient distortion features that result from embedding messages. The only way to withstand such attack is to follow the strategy that was used by the UNIWARD (Holub, et al., 2014) whereby embedding avoids smooth and clean edge regions. However, for embedding rate > 40% even this specially designed scheme is detected.

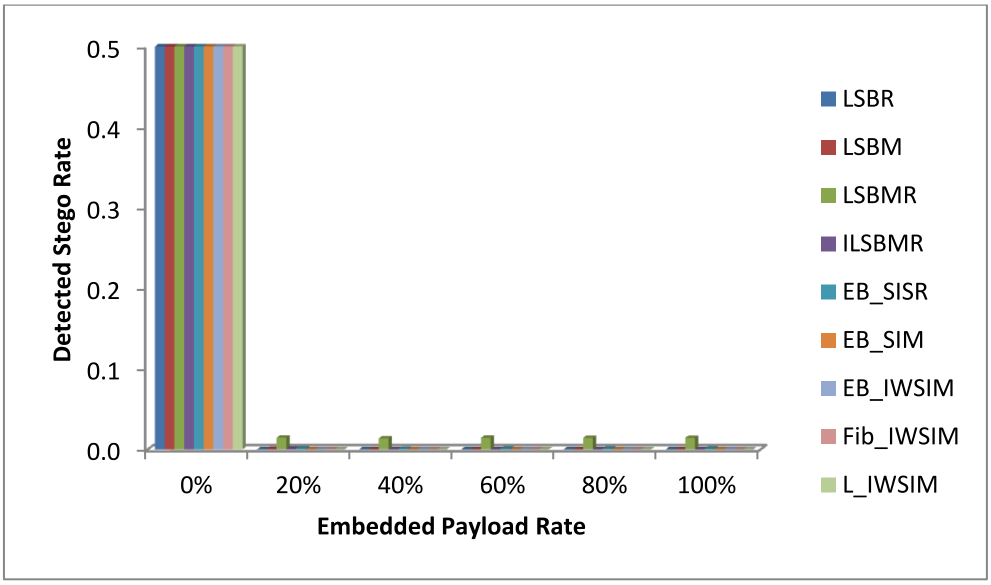


Figure 7-31: SRM steganalysis of SIPI database.

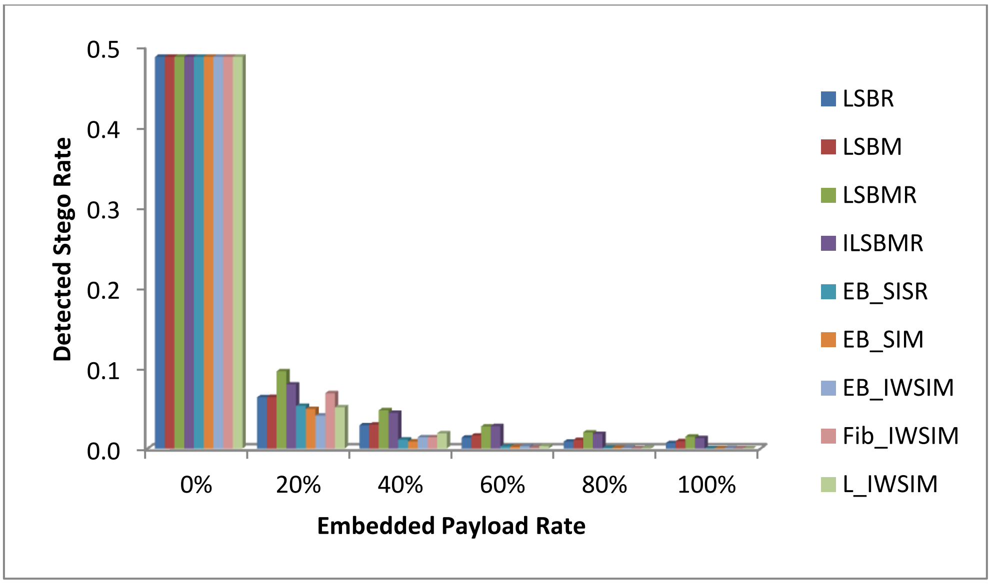


Figure 7-32: SRM steganalysis of BOSSBase database when Lenna image was embedded.

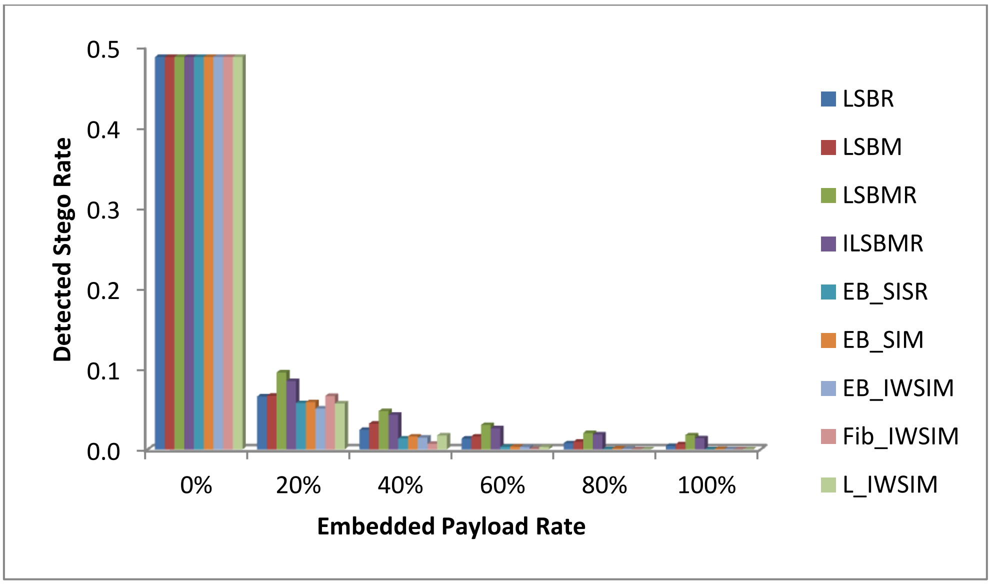


Figure 7- 33: SRM steganalysis of BOSSBase database when Jet image was embedded.

7.4 Summary

In this chapter, we developed the last step of our strategy identified in Chapter 4 for designing robust steganography embedding schemes that meet the main success criteria on stego-image quality, embedding efficiency, payload capacity, and robustness against known steganalysis tool. The strategy was based on increasing similarity between the secret image bit-stream and the cover image LSB plane. Earlier in Chapters 5 and 6, we developed the first two steps in this strategy by pre-processing the secret image bit-streams for increased 0:1 ratio and developed cover pixel decomposition model that also results in increased 0:1 ratio in the cover LSB plane. The variable significant increases in 0:1 ratio in the cover image LSB plane (using different pixel decomposition models) and in the secret image bit-stream (using different algorithms) have increased the probability of similarities between the two input data to any embedding scheme. However, the different non-binary pixel decomposition schemes resulted in decreasing the number possible patterns for the 3 first bit-planes which tilted the balance in favour of using mapping tables rather than any other embedding scheme. Accordingly, various mapping based embedding schemes, each corresponding to a choice of a pixel decomposition scheme and a secret image bit-stream that is pre-processed by one of the 3 such algorithms. .

The high level of success of the adopted strategy and that of the last step is demonstrated by the extensive experimental work done in this chapter to evaluate the performance of the set of proposed and revised steganography techniques (EB_SISR, EB_SIM, EB_IWSIM, Fib_IWSIM, and L_IWSIM) in terms of capacity, embedding efficiency, stego-image quality, and robustness against several well-known steganalysis tools. Experimental results demonstrated these schemes compares well with, and outperform, existing schemes on many of the stated success criteria. The following is a summary of the evaluation tests:

1- Although in terms of payload capacity some of the schemes are outperformed by the LSBR and LSBM, but very marginally and this is because the side information needed for the SIM and IWSIM pre-processing algorithms. Moreover, the EB_SISR has significantly improved capacity compared to LSBR and LSBM as a result of its unintended compression effect on the length of the secret image bit-stream.
2- For embedding rates > 40%, the best of our schemes EB_IWSIM outperforms all tested steganography schemes in terms of the ratio of modified cover pixels and embedding efficiency. In fact, while existing schemes has constant efficiency rates overall embedding payloads, our schemes' efficiency increases with increased payload. This is to the effect of the side information at low payloads.
3- In terms of stego-image quality, the PSNR values achieved by our schemes are almost comparable to all tested existing schemes. In fact, our revised Fib_IWSIM is only marginally outperformed by the LSBM and LSBMR. The shortcoming of our schemes in relation to PSNR is due to the fact that sometimes the cover pixel value is changed by 2 after secret embedding.
4- All our schemes outperform the LSBR in terms of robustness against all targeted steganalysis tools. Unfortunately, like all existing tested schemes, it detected by the universal SRM tool.

The fact that results are averaged over a large number of cover images is an incentive to select cover images carefully to overcome the marginal shortcoming on stego-image quality. Moreover, we can also apply the various secret image bit-streams pre-processing schemes adaptively to achieve optimal results. This is to be done in the future.

Chapter 8

Conclusions and Future Research Directions

8.1 Conclusions

Digital steganography is an information security mechanism that is generally concerned with concealing the presence of secret data by embedding in another innocuous data/object to be transmitted as mundane communication, i.e. making the act of communication itself a secret. It is becoming an alternative, but complementary, to cryptography in protecting sensitive secrets where adversaries are aware of the presences of the secret but cannot decipher.

This thesis is devoted to investigating steganography schemes for hiding secret image files in image files. It was initially motivated by an interest in protecting sensitive communications for use by intelligence and law enforcing agencies in crime fighting/prevention which necessitate the secure and preserving privacy exchange of photos of crime scene and face images of suspects. Moreover, forensic investigators often need to take and transmit left fingerprints, for later comparison without undermining the integrity of the evidence. Armed forces need such as exchanging military maps or surveillance video in hostile environment/situations. Modern health care systems required by law to maintain the privacy of critical information when storing or exchanging patient's medical images such as X-ray. Furthermore, financial and commercial organizations such as banks can benefit from such technology for remote authentication.

The main objectives of the work conducted in this thesis were the design, development and testing the performance of secure and efficient steganography schemes for embedding secret images in cover images. Over the recent history of digital communication, many steganography techniques have been developed for embedding secrets into digital images primarily by manipulating their least significant bit-planes (LSBs). Although, the changes to the content of cover images may not be visible to human eye, but the presences of the secret may become detectable by automatic steganalysis tools that conduct statistical analysis tests and/or search for distortions to local features. Having conducted a literature review of the areas of digital steganography, and steganalysis, we identified the main challenges that steganographers face as well as the criteria for success. The embedding capacity of the cover image while protecting against detectability is a particularly challenge. Embedding longer secret bit-streams, bound to result in some form of local distortion and quality degradation of the stego-image. Robustness of embedding schemes against adversary attacks is closely related to detectability, it is dependent on the maintenance of image quality, and it gets more difficult the higher the payload is. While many existing spatial domain steganography schemes have been developed to perform well with respect to one or more of the above requirements, this thesis aims to achieve, or pave the way to achieve, optimal performance in terms of all these objectives.

Having reviewed the literature on existing spatial domain based digital steganography and steganalysis tools, we designed two embedding schemes: Indexing-based and Fibonacci-Mapping. In both cases, more than the LSB plane used for hiding the secret. The first scheme, embeds only one bit in each cover pixel and uses a combination of pre-processing the cover image pixels to eliminate the possibility of having equal bits in the 2LSB planes, followed by a system that report the index of the bit that matches the secret bit. Compared to the LSBR, this scheme resulted in lower stego-image quality and embedding efficiency, but it was robust against two of the steganalysis tools, DIH and RWS. The second approach extended the Fibonacci-like steganography by bit-plane(s) mapping instead of bit-plane(s) replacement to embed two secret bits in the first three Fibonacci bit-planes. Unlike the original Fibonacci scheme, no cover pixels were excluded from embedding because actions are taken to comply with Zeckendorf theorem. Consequently, this scheme has double the embedding capacity of LSBR. Furthermore, it is secure against steganalysis techniques such as RS, DIH, and RWS. The improved capacity and robustness was at the expense of further reduction of stego-image quality compared to the Indexing-based scheme. Although, these two schemes did not achieve the sought after objectives, but help develop our strategy for the following work that is based on pre-processing the secret image as well as considering various pixel value decomposition models but for different purposes.

The source of difficulty in addressing the different challenges, mentioned above, is related to the fact that the embedding process may result in changing cover image pixel values. Most schemes make changes to the LSB (or higher) bit-plane of cover images that may result in local distortions even when these changes are not easy to detect by the human eye. Embedding efficiency is the most important quantifiable attribute for digital steganography as a measure of the pixel value changes. It has a direct influence on the stego-image quality and message detectability/security without compromising payload capacity. The central focus of our research was therefore on the design of embedding schemes that have high efficiency and message un-detectability while maintaining payload capacity. Our approach is to reduce the effect of the act of hiding a secret image in digital images by minimising the number of changed pixels.

For LSB based embedding schemes, pixel values change whenever there are dissimilarities between the cover LSB plane and the secret image bit-stream, and in these cases the embedding process may lead to change to the statistical parameters of stego-image bit-planes as well as to local image features (computable linear relationship between neighbouring pixels). Steganalysis tools exploit these effects to model targeted as well as blind attacks. Usually, these problems are dealt with by randomising the changes to the LSB, using elaborate schemes to embed one or more secret bits in different/multiple cover bit-planes, or embedding in noise-tolerant regions.

Our innovative approach to minimise the embedding-induced changes was based on developing efficient image procedures and models to manipulate the cover and the secret images, prior to embedding, that increases similarity between the cover image LSB plane and the secret image bit-stream. Note that most existing image-based steganography techniques focus on the embedding strategy and give no consideration to pre-processing the secret/cover image except encrypting or compressing the secret image. One of the premises of this thesis was applying carefully selected pre-processing techniques could help enhance the efficiency and security of the steganography systems. This was achieved in two novel steps that increase the 0:1 ratio in both the secret bit-stream and the cover LSB plane.

Image pixel values, in general, are not uniformly distributed, as is the case of random secrets, and different blocks in the image have a different texture and different statistics.

Therefore, secret images bit-streams are different from general secret bit-stream dealt with in the literature. It is these characteristics that have been exploited in this thesis to develop three secret image pre-processing algorithms that help transform the secret image bit-stream for increased 0:1 ratio. The first two algorithms (SIM, IWSIM) are similar, but one in spatial domain and the other in the Wavelet domain (using integer-valued Wavelet filter), and are based on a modified version of statistical coding used for image compression. The modification is based on mapping the most frequent pixels (Wavelet sub-band coefficients) onto bytes with more 0s. The third pre-processing algorithm (SISR), process blocks by subtracting their means from all pixel values and hence reducing the required number of bits needed to represent the pixel residues in these blocks. In other words, SISR also reduces the length of the secret bit-stream without loss of information. The extensive experimental testing demonstrated that these algorithms yield a significant increase in the secret image bit-stream 0:1 ratio with the Wavelet version, IWSIM algorithm, yielding the best performance with an average ratio of 80:20.

For the second step, i.e. manipulating the cover image, we revisited the various existing models of pixel value decomposition schemes including the Fibonacci and other defining sequences that differ from the usual binary scheme. However, while existing steganography schemes use such models simply to expand the number of bit-planes to enable embedding in higher bit-planes than LSB, we aimed to investigate the capabilities of these models for increasing the 0:1 ratio in the corresponding LSB plane. We investigated a number of pixel value decomposition models (including Fibonacci, prime, natural, Lucas, and Catalan-Fibonacci) and determine the best decomposition that achieves the highest ratio of 0:1 in the cover LSB plane. A number of such existing techniques indeed can lead to increased 0:1 ratio in the corresponding LSB plane. Consequently, we developed a new cover pixel value decomposition technique, referred to as the Extended-Binary, which is an extension of the binary decomposition scheme, that has results in the cover image LSB plane having one of the highest ratios of 0 to 1 among a variety of pixel decomposition schemes, 77% on average.

Having successfully fulfilled the objectives set out in the above 2-steps strategy, we embarked on designing an embedding scheme that benefits from the achieved similarities between the various pre-processed secret image bit-stream and the decomposed cover-image LSB plane. We designed embedding schemes that simply embeds by replacement the various pre-processed secret image bit-streams into the LSB of an Extended-Binary decomposed cover image. The various schemes have shown good but modestly improved performance. Unfortunately, the embedding efficiency obtained by the best-performing scheme IWSIM_EB was still lower than what was desired, due to skipping the bad candidate cover pixel. Skipping cover pixels is a problem that is associated with LSB by the replacement that results in violating the uniqueness of pixel representation by non-binary decomposition schemes.

The Fibonacci-Mapping scheme, designed at the early stages, benefited from the observation that there are only 5 possible patterns for the Fibonacci decomposed cover 3 first bit-planes. On examination of all the non-binary pixel decomposition schemes, we found that in all these cases the number of possible patterns for the 3 first bit-planes is reduced to 4 or 5 instead of 8. We used this fact to design bit-plane mappings suitable for embedding, instead of LSB replacement, in order to make each cover pixel a suitable candidate for secret bit embedding and avoid skipping. We used these mapping-based embedding schemes to create a number of steganography schemes, one for each combination of cover image decomposition model and a secret image bit-stream pre-processing algorithm. The extensive experimental works done to test the performance of these new and revised schemes have shown beyond any doubt the success of our strategy.

In particular, the various stego-images obtained by these schemes are minimally distorted as a result of reduced number of changed cover pixels post embedding and by implication higher embedding efficiency. Overall, the set of proposed and revised steganography techniques (EB_SISR, EB_SIM, EB_IWSIM, Fib_IWSIM, and L_IWSIM) have performed well in terms of all the four stated success criteria (capacity, embedding efficiency, stego-image quality, and robustness against several well-known targeted steganalysis tools). These schemes also compare well with, and outperform, existing schemes on many of the stated success criteria. However, the stego-image quality, in terms of PSNR, output by our schemes was marginally lower than LSBR and LSBM. This is because, in some cases, the cover pixels values were changed by 2 after secret bit embedding. Unfortunately, like all the tested steganography schemes, ours were not robust against the blind SRM tool.

8.2 Future Research Directions

The work reported in this thesis, not only demonstrated the viability of high embedding efficiency and un-detectable image-based steganography schemes but highlights several potential research directions to be explored in the future. Some of the initial plans aim to overcome some the above mentioned limitations.

1. **Cover Image Selection.** The fact that the various performance measures are averaged over a large number of cover images is an incentive to adopt a credible cover image selection strategy to overcome the marginal shortcoming on stego-image quality. For this, we need to investigate the relationship between image texture/entropy information and the 0:1 ratio obtained from the various decomposition schemes.

2. **Adaptive pixel decomposition and secret pre-processing.** Understanding the relationship between texture/entropy information and the 0:1 ratios obtained from both steps can be used to develop adaptive block-based mapping schemes to embed various secret image bit-streams (pre-processed by different block-based schemes) adaptively into the cover image blocks that have been appropriately decomposed using similarity ranking.

3. **Robustness against SRM**. To improve the robustness of some or all of our mapping based schemes against SRM, we need to investigate embedding in non-smooth regions and avoid clean edge regions. This would be similar to the approach adopted in the UNIWARD embedding scheme which was designed to be robust against the SRM and succeeded in doing so for low embedding rates. However, this would require an investigation to identify local distortion feature models that result from our mapping-based embedding.

4. **Alternative Secret Image pre-processing**. Most of image-based steganography researches till date have not considered any pre-processing on the secret image except encryption or compression. Pre-processing algorithms can be designed to be applied on the secret image prior to embedding in order to achieve steganography requirements. Although, in this research, we developed three pre-processing algorithms, and the one that based on the IWT provides better performance than the other algorithms in terms of embedding efficiency and message detectability. Our

future plan is to investigate and test other integer to integer Wavelet-like transform domains for improved ratio of 0:1 in the manipulated secret image bit-streams.

References

Abdelwahab, A. A. and Hassaan, L. A., (2008). A discrete wavelet transform based technique for image data hiding. IEEE, Radio Science Conference, pp. 1-9.

Abdulla, A. A., Jassim, S. A. and Sellahewa, H., (2013). Efficient high-capacity steganography technique. SPIE, International Society for Optics and Photonics, pp. 875508-875508.

Abdulla, A. A., Jassim, S. A. and Sellahewa, H., (2013). Secure Steganography Technique Based on Bitplane Indexes. IEEE International Symposium on Multimedia (ISM), pp. 287-291.

Abdulla, A. A., Sellahewa, H. and Jassim, S. A., (2014). Steganography based on pixel intensity value decomposition. SPIE, International Society for Optics and Photonics, pp. 912005-912005.

Abdulla, A. A., Sellahewa, H. and Jassim, S. A., (2014). Stego Quality Enhancement by Message Size Reduction and Fibonacci Bit-Plane Mapping. Springer, Security Standardisation Research (SSR), pp. 151-166.

Abdullah, K. A., Al-Jawad, N. and Abdulla, A. A., (2014). Effect of using different cover image quality to obtain robust selective embedding in steganography. SPIE, International Society for Optics and Photonics, pp. 913808-913808.

Agaian, S. S., Cherukuri, R. C. and Sifuentes, R., (2006). A new secure adaptive steganographic algorithm using Fibonacci numbers. IEEE, pp. 125-129.

Agaian, S. S., Cherukuri, R. C. and Sifuentes, R. R., (2007). Key dependent covert communication system using fibonacci p-codes. IEEE, International Conference on System of Systems Engineering, pp. 1-5.

Alharbi, F., (2013). Novel Steganography System using Lucas Sequence. International Journal of Advanced Computer Science and Applications (IJACSA), vol. 4, pp. 52-58.

Aroukatos, N., Manes, K., Zimeras, S. and Georgiakodis, F., (2012). Data hiding techniques in steganography using fibonacci and catalan numbers. IEEE, 9th International Conference on Information Technology, pp. 392-396.

Bas, P., Filler, T. and Pevny, T., (2011). Break Our Steganographic System. Springer, Information Hiding, pp. 59-70.

Battisti, F., Carli, M., Neri, A. and Egiaziarian, K., (2006). A Generalized Fibonacci LSB Data Hiding Technique. 3rd International Conference on Computers and Devices for Communication.

Bender, W., Gruhl, D., Morimoto, N. and Lu, A., (1996). Techniques for data hiding. IBM systems journal, vol. 35, pp. 313-336.

Bhattacharyya, D. et al., (2009). Discrete fourier transformation based image authentication technique. IEEE, 8th International Conference on Cognitive Informatics, pp. 196-200.

Bhattacharyya, D. and Kim, T.-h., (2011). Image Data Hiding Technique Using Discrete Fourier Transformation. Springer, Ubiquitous Computing and Multimedia Applications, pp. 315-323.

Cachin, C., (1998). An information-theoretic model for steganography. Springer, Information Hiding, pp. 306-318.

Calderbank, A., Daubechies, I., Sweldens, W. and Yeo, B.-L., (1997). Lossless image compression using integer to integer wavelet transforms. IEEE, ICIP, pp. 596-596.

Chan, C.-S., (2009). On using LSB matching function for data hiding in pixels. IOS Press, Fundamenta Informaticae, vol. 96, pp. 49-59.

Chan, C.-K. and Cheng, L., (2001). Improved hiding data in images by optimal moderately-significant-bit replacement. IET, Electronics Letters, vol. 37, pp. 1017-1018.

Chan, C.-K. and Cheng, L.-M., (2004). Hiding data in images by simple LSB substitution. Elsevier, Pattern recognition, vol. 37, pp. 469-474.

Chang, C.-C. and Tseng, H.-W., (2004). A steganographic method for digital images using side match. Elsevier, Pattern Recognition Letters, vol. 25, pp. 1431-1437.

Cheddad, A., Condell, J., Curran, K. and McKevitt, P., (2008). Enhancing Steganography in digital images. IEEE, Canadian Conference on Computer and Robot Vision, pp. 326-332.

Chen, M., Zhang, R., Niu, X. and Yang, Y., (2006). Analysis of Current Steganography Tools: Classifications and Features. IEEE, International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 384-387.

Chen, P.-Y., Lin, H.-J. and others, (2006). A DWT based approach for image steganography. International Journal of Applied Science and Engineering, vol. 4, pp. 275-290.

Chen, W.-J., Chang, C.-C. and Le, T., (2010). High payload steganography mechanism using hybrid edge detector. Elsevier, Expert Systems with applications, vol. 37, pp. 3292-3301.

Codr, J., (2009). Unseen: An Overview of Steganography and Presentation of Associated Java Application C-Hide. Citeseer, Retrieved January, vol. 8, pp. 2010.

Cogranne, R. et al., (2014). A local adaptive model of natural images for almost optimal detection of hidden data. Elsevier, Signal Processing, vol. 100, pp. 169-185.

Cole, E. and Krutz, R. D., (2003). Hiding in plain sight: Steganography and the art of covert communication. John Wiley and Sons, Inc.

Cox, I. J., Kalker, T., Pakura, G. and Scheel, M., (2005). Information transmission and steganography. Springer, Digital Watermarking, pp. 15-29.

Cox, I. et al., (2007). Digital Watermarking and Steganography. Morgan Kauffman.

Crandall, R., (1998). Some notes on steganography. Posted on steganography mailing list.

Daneshkhah, A., Aghaeinia, H. and Seyedi, S. H., (2011). A more secure steganography method in spatial domain. IEEE, 2nd International Conference on Intelligent Systems, Modelling and Simulation (ISMS), pp. 189-194.

Dey, S., Abraham, A. and Sanyal, S., (2007). An LSB Data Hiding Technique Using Natural Number Decomposition. IEEE, 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP), vol. 2, pp. 473-476.

Dey, S., Abraham, A. and Sanyal, S., (2007). An LSB Data Hiding Technique Using Prime Numbers. 3rd International Symposium on Information Assurance and Security, pp. 101-108.

Filler, T., Judas, J. and Fridrich, J., (2011). Minimizing additive distortion in steganography using syndrome-trellis codes. IEEE Transactions on Information Forensics and Security, vol. 6, pp. 920-935.

Fridrich, J., (1999). Applications of data hiding in digital images. IEEE, 5th International Symposium on Signal Processing and Its Applications, vol. 1, pp. 9.

Fridrich, J. and Goljan, M., (2004). On estimation of secret message length in LSB steganography in spatial domain. International Society for Optics and Photonics, Electronic Imaging, pp. 23-34.

Fridrich, J., Goljan, M. and Du, R., (2001). Reliable detection of LSB steganography in color and grayscale images. ACM, Workshop on Multimedia and Security: New Challenges, pp. 27-30.

Fridrich, J., Goljan, M., Lisonek, P. and Soukal, D., (2005). Writing on wet paper. IEEE Transactions on Signal Processing, vol. 53, pp. 3923-3935.

Fridrich, J. and Kodovsky, J., (2012). Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, vol. 7, pp. 868-882.

Fridrich, J., Lisonek, P. and Soukal, D., (2007). On steganographic embedding efficiency. Springer, Information Hiding, pp. 282-296.

Fridrich, J. and Soukal, D., (2006). Matrix embedding for large payloads. International Society for Optics and Photonics, Electronic Imaging, pp. 60721W--60721W.

Geetha, C. and Giriprakash, H., (2012). Image Steganography by Variable Embedding and Multiple Edge Detection using Canny Operator. Citeseer, International Journal of Computer Applications, vol. 48.

Gonzalez, R. C. and Woods, R. E., (2002). Digital image processing second edition. Beijing: Publishing House of Electronics Industry.

Hempstalk, K., (2006). Hiding behind corners: Using edges in images for better steganography. Computing Women's Congress, Hamilton, New Zealand, pp.11-19

Holub, V. and Fridrich, J., (2013). Digital image steganography using universal distortion. 1st ACM workshop on Information hiding and multimedia security, pp. 59-68.

Holub, V., Fridrich, J. and Denemark, T., (2014). Universal distortion function for steganography in an arbitrary domain. Springer, EURASIP Journal on Information Security, vol. 2014, pp. 1-13.

Holub, V. and Fridrich, J. J., (2012). Designing steganographic distortion using directional filters. IEEE International Workshop on Information Forensics and Security (WIFS), pp. 234-239.

Huang, Q. and Ouyang, W., (2010). Protect fragile regions in steganography LSB embedding. 3rd International Symposium on Knowledge Acquisition and Modeling (KAM), pp. 175-178.

Iranpour, M., (2013). A novel steganographic method based on edge detection and adaptive multiple bits substitution. 18th International Conference on Digital Signal Processing (DSP), pp. 1-6.

Iranpour, M. and Farokhian, F., (2013). Minimal Distortion Steganography Using Well-Defined Functions. 10th International Conference on High Capacity Optical Networks and Enabling Technologies, pp. 21-24.

Islam, M. and Alzahir, S., (2013). A novel qr code guided image stenographic technique. IEEE International Conference on Consumer Electronics (ICCE), pp. 586-587.

Jain, A. K. and Uludag, U., (2002). Hiding fingerprint minutiae in images. 3rd Workshop on Automatic Identification Advanced Technologies, pp. 97-102.

Janakirman, S. et al., (2012). Pixel Bit Manipulation for Encoded Hiding - An Inherent stego. International Conference on Computer Communication and Informatics (ICCCI), pp. 1-6.

Jenifer, K. S., Yogaraj, G. and Rajalakshmi, K., (2014). LSB Approach for Video Steganography to Embed Images. Citeseer, International Journal of Computer Science and Information Technologies (IJCSIT), vol. 5, pp. 319-322.

Johnson, N. F. and Jajodia, S., (1998). Exploring steganography: Seeing the unseen. IEEE, Computer, vol. 31, pp. 26-34.

Juarez-Sandoval, O., Espejel-Trujillo, A., Nakano-Miyatake, M. and Perez-Meana, H., (2013). Robust Steganography Based on QIM Algorithm to Hide Secret Images. International Journal of Computers, vol. 7, pp. 145-152.

Kahn, D., (1996). The history of steganography. Springer, Information Hiding, pp. 1-5.

Katzenbeisser, S. and Petitcolas, F. A., (2002). Defining security in steganographic systems. International Society for Optics and Photonics, Electronic Imaging, pp. 50-56.

Katzenbeisser, S. and Petitolas, F., (2000). Information hiding techniques for steganography and digital watermaking. Artech House, Taylor and Francis.

Kaur, M. and Kaur, G., (2014). Review of Various Steganalysis Techniques. International Journal of Computer Science and Information Technologies (IJCSIT), vol. 5, pp. 1744-1747.

Ker, A. D., (2005). A general framework for structural steganalysis of LSB replacement. Springer, Information Hiding, pp. 296-311.

Ker, A. D., (2005). Improved detection of LSB steganography in grayscale images. Springer, Information Hiding, pp. 97-115.

Ker, A. D., (2005). Steganalysis of LSB matching in grayscale images. IEEE, Signal Processing Letters, vol. 12, pp. 441-444.

Ker, A. D., (2007). Steganalysis of embedding in two least-significant bits. IEEE Transactions on Information Forensics and Security, vol. 2, pp. 46-54.

Ker, A. D. et al., (2013). Moving steganography and steganalysis from the laboratory into the real world. ACM workshop on Information hiding and Multimedia Security, pp. 45-58.

Ker, A. D. and Bohme, R., (2008). Revisiting weighted stego-image steganalysis. SPEE, Electronic Imaging, Forensics, Steganography, and Watermarking of Multimedia Contents, vol. 6819, pp. 681905-681905.

Khamrui, A. and Mandal, J., (2013). A Genetic Algorithm based Steganography Using Discrete Cosine Transformation (GASDCT). Elsevier, Procedia Technology, vol. 10, pp. 105-111.

Khosravi, M. J. and Naghsh-Nilchi, A. R., (2014). A novel joint secret image sharing and robust steganography method using wavelet. Springer, Multimedia Systems, vol. 20, pp. 215-226.

Kipper, G., (2004). Investigator's guide to steganography. crc press.

Kodovsk, J., (2012). Steganalysis of Digital Images Using Rich Image Representations and Ensemble Classifiers. PhD Thesis, State University of New York.

Kodovsky, J. and Fridrich, J., (2013). Quantitative steganalysis using rich models. International Society for Optics and Photonics, IS and SPIE Electronic Imaging, pp. 86650O--86650O.

Kodovsky, J., Fridrich, J. and Holub, V., (2012). Ensemble classifiers for steganalysis of digital media. IEEE Transactions on Information Forensics and Security, ,vol. 7, pp. 432-444.

Kraetzer, C., (2007). Visualisation of benchmarking results in digital watermarking and steganography. Citeseer, IST-2002-507932 ECRYPT, p. 30.

Kumar, V. and Kumar, D., (2010). Performance evaluation of dwt based image steganography. 2nd International Advance Computing Conference (IACC), pp. 223-228.

Lau, S., (2003). An Analysis of Terrorist Groups' Potential Use of Electronic Steganography. Bethesda, Md.: SANS Institute, February, vol. 18.

Liu, J., Tang, G. and Sun, Y., (2013). A secure steganography for privacy protection in healthcare system. Springer, Journal of medical systems, vol. 37, pp. 1-10.

Li, Y., Li, C.-T. and Wei, C.-H., (2007). Protection of mammograms using blind steganography and watermarking. IEEE, 3rd International Symposium on Information Assurance and Security, pp. 496-500.

Luo, W., Huang, F. and Huang, J., (2010). Edge adaptive image steganography based on LSB matching revisited. IEEE Transactions on Information Forensics and Security, vol. 5, pp. 201-214.

Luo, X.-Y., Wang, D.-S., Wang, P. and Liu, F.-L., (2008). A review on blind detection for image steganography. Elsevier, Signal Processing, vol. 88, pp. 2138-2157.

Macaulay, G. C. and others, (1904). The History of Herodotus.Macmillan and Company, vol. 2.

Mammi, E. et al., (2008). A novel spatial data hiding scheme based on generalized Fibonacci sequences. Proc. SPIE6982 C, vol. 69820, pp. 1-7.

Mandal, J. and Khamrui, A., (2011). A genetic algorithm based steganography in frequency domain (GASFD). IEEE, International Conference on Communication and Industrial Application (ICCIA), pp. 1-4.

Mangarae, A., (2006). Steganography faq. Zone-H. Org March 18th.

Marvel, L. M., Boncelet Jr, C. G. and Retter, C. T., (1998). Reliable BLIND information hiding for images. Springer, Information Hiding, pp. 48-61.

Mercuri, R. T., (2004). The many colors of multimedia security. ACM, Communications of the ACM, vol. 47, pp. 25-29.

Mielikainen, J., (2006). LSB matching revisited. IEEE, Signal Processing Letters, vol. 13, pp. 285-287.

Petitcolas, F. A., Anderson, R. J. and Kuhn, M. G., (1998). Attacks on Copyright Marking Systems. Springer, Information Hiding, vol. 1525, pp. 218-238.

Petitcolas, F. A., Anderson, R. J. and Kuhn, M. G., (1999). Information hiding-a SURVEY. Proceedings of the IEEE, vol. 87, pp. 1062-1078.

Pevn, T., Filler, T. and Bas, P., 2010. Using high-dimensional image models to perform highly undetectable steganography. Springer, Information Hiding, pp. 161-177.

Picione, D. D. L. et al., (2006). A Fibonacci LSB data hiding tecnique. IEEE, 14th European Signal Processing Conference, pp. 1-5.

Ponomarenko, N. et al., (2008). Color image DATABASE for evaluation of image quality metrics. IEEE, 10th Workshop on Multimedia Signal Processing, pp. 403-408.

Provos, N. and Honeyman, P., (2003). Hide and seek: An introduction to steganography. IEEE, Security and Privacy, vol. 1, pp. 32-44.

Rabah, K., (2004). Steganography-the art of hiding data. Information Technology Journal, vol .3, pp. 245-269.

Raftari, N. and Moghadam, A. M. E., (2012). Digital Image Steganography Based on Integer Wavelet Transform and Assignment Algorithm. IEEE, 6th Asia Modelling Symposium (AMS), pp. 87-92.

Rashid, R. D., Sellahewa, H. and Jassim, S. A., (2013). Biometric feature embedding using robust steganography technique. SPIE, International Society for Optics and Photonics, pp. 875503-875503.

Raul, R.-C., Claudia, F.-U. and Trinidad-BIas, G. d. J., (2007). Data hiding scheme for medical images. IEEE, 17th International Conference on Electronics, Communications and Computers, pp. 32-32.

Seyyedi, S. A. and Ivanov, N., (2014). High Payload and Secure Steganography Method Based on Block Partitioning and Integer Wavelet Transform. International Journal of Security and Its Applications, vol. 8, pp. 183-194

Sharp, T., (2001). An implementation of key-based digital signal steganography. Springer, Information Hiding, pp. 13-26.

Simmons, G. J., (1984). The prisoners' problem and the subliminal channel. Springer, Advances in Cryptography, pp. 51-67.

Singh, K. M., Singh, L. S., Singh, A. B. and Devi, K. S., (2007). Hiding secret message in edges of the image. International Conference on Information and Communication Technology (ICICT), pp. 238-241.

Stoica, A., Vertan, C. and Fernandez-Maloigne, C., (2003). Objective and subjective color image quality evaluation for JPEG 2000 compressed images. IEEE, International Symposium on Signal, Circuit and Systems,vol. 1, pp. 137-140.

Swanson, M. D., Zhu, B. and Tewfik, A. H., (1996). Robust data hiding for images. IEEE, Digital Signal Processing Workshop Proceedings, pp. 37-40.

Tan, P.-N., Steinbach, M., Kumar, V. and others, (2006). Introduction to data mining. Library of Congress, Second Edition.

Thien, C.-C. and Lin, J.-C., (2003). A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function. Elsevier, Pattern Recognition, vol. 36, pp. 2875-2881.

Viterbi, U., (1981). USC-SIPI image database. Signal and Image Processing Institute, University of Southren California (USC).

Wang, C. et al., (2010). A content-adaptive approach for reducing embedding impact in steganography. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 1762-1765.

Wang, R.-Z., Lin, C.-F. and Lin, J.-C., (2000). Hiding data in images by optimal moderately-significant-bit replacement. IET, Electronics Letters, vol. 36, pp. 2069-2070.

Wang, Z. and Bovik, A. C., (2002). A universal image quality index. IEEE, Signal Processing Letters, , vol. 9, pp. 81-84.

Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P., (2004). Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, vol. 13, pp. 600-612.

Wang, Z., Sheikh, H. R. and Bovik, A. C., (2002). No-reference perceptual quality assessment of JPEG compressed images. IEEE, International Conference on Image Processing, vol. 1, pp. 477-480.

Wang, Z., Sheikh, H. R. and Bovik, A. C., (2003). Objective video quality assessment. CRC press, The handbook of video database: design and applications, pp. 1041-1078.

Wayner, P., (2002). Disappearing Cryptography-Information Hiding: Steganography and Watermarking. Morgan Kaufmann, Second Edition, pp. 291-318.

Westfeld, A., (2001). F5—a steganographic algorithm. Springer, Information Hiding, pp. 289-302.

Westfeld, A. and Pfitzmann, A., (2000). Attacks on steganographic systems. Springer, Information Hiding, pp. 61-76.

Wu, D.-C. and Tsai, W.-H., (2003). A steganographic method for images by pixel-value differencing. Elsevier, Pattern Recognition Letters, vol. 24, pp. 1613-1626.

Wu, N.-I. and Hwang, M.-S., (2007). Data hiding: current status and key issues. IJ Network Security, vol. 4, pp. 1-9.

Yang, B. and Deng, B., (2006). Steganography in gray images using wavelet. 2nd International Symposium on Communication, Control and Signal Processing(ISCCSP).

Zhang, T. and Ping, X., (2003). Reliable detection of LSB steganography based on the difference image histogram. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 3, pp. 545-548.

Zollner, J. et al., (1998). Modeling the security of steganographic systems. Springer, Information Hiding, pp. 344-354.

List of Publications

- **Abdulla, A. A.**, Jassim, S. A. and Sellahewa, H., “Efficient high-capacity steganography technique”, Proceedings of Mobie Multimedia/Image Processing and Applications, SPIE, vol. 8755, **2013**.
- **Abdulla, A. A.**, Jassim, S. A. and Sellahewa, H., “Secure Steganography Technique Based on Bitplane Indexespp”, IEEE International Sumposium on Multimedia (ISM), Proceeding of IEEE Computer Society, **2013**.
- **Abdulla, A. A.**, Sellahewa, H. and Jassim, S. A., “Steganography based on pixel intensity value decomposition”, Proceedings of Mobie Multimedia/Image Processing and Applications, SPIE, vol. 9120, **2014.**
- Abdullah, K. A., Al-Jawad, N. and **Abdulla, A. A.**, “Effect of using different cover image quality to obtain robust selective embedding in steganography”, Proceedings of Optics, Photonics, and Digital Technologies for Multimedia Applications, SPIE, vol. 9138, **2014**.
- **Abdulla, A. A.**, Sellahewa, H. and Jassim, S. A., “Stego Quality Enhancement by Message Size Reduction and Fibonacci Bit-Plane Mapping”, Security Standardisation Research (SSR), LNCS 8893, Springer, **2014**.

Appendix

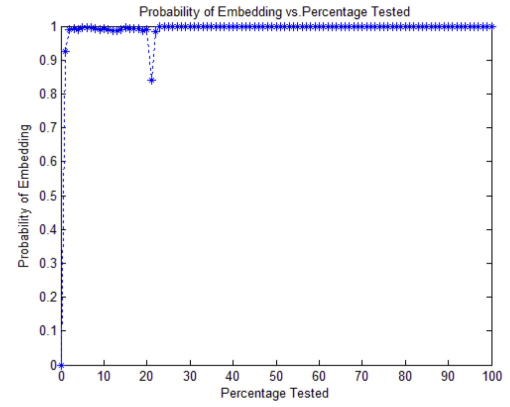
Table A-1: Grayscale values (0-511) in descending order of number of 1s in its binary representation.

value	Binary rep.	value	Binary rep.	value	Binary rep.	value	Binary rep.
0	000000000	416	110100000	31	000011111	111	001101111
1	000000001	448	111000000	47	000101111	119	001110111
2	000000010	15	000001111	55	000110111	123	001111011
4	000000100	23	000010111	59	000111011	125	001111101
8	000001000	27	000011011	61	000111101	126	001111110
16	000010000	29	000011101	62	000111110	159	010011111
32	000100000	30	000011110	79	001001111	175	010101111
64	001000000	39	000100111	87	001010111	183	010110111
128	010000000	43	000101011	91	001011011	187	010111011
256	100000000	45	000101101	93	001011101	189	010111101
3	000000011	46	000101110	94	001011110	190	010111110
5	000000101	51	000110011	103	001100111	207	011001111
6	000000110	53	000110101	107	001101011	215	011010111
9	000001001	54	000110110	109	001101101	219	011011011
10	000001010	57	000111001	110	001101110	221	011011101
12	000001100	58	000111010	115	001110011	222	011011110
17	000010001	60	000111100	117	001110101	231	011100111
18	000010010	71	001000111	118	001110110	235	011101011
20	000010100	75	001001011	121	001111001	237	011101101
24	000011000	77	001001101	122	001111010	238	011101110
33	000100001	78	001001110	124	001111100	243	011110011
34	000100010	83	001010011	143	010001111	245	011110101
36	000100100	85	001010101	151	010010111	246	011110110
40	000101000	86	001010110	155	010011011	249	011111001
48	000110000	89	001011001	157	010011101	250	011111010
65	001000001	90	001011010	158	010011110	252	011111100
66	001000010	92	001011100	167	010100111	287	100011111
68	001000100	99	001100011	171	010101011	303	100101111
72	001001000	101	001100101	173	010101101	311	100110111
80	001010000	102	001100110	174	010101110	315	100111011
96	001100000	105	001101001	179	010110011	317	100111101
129	010000001	106	001101010	181	010110101	318	100111110
130	010000010	108	001101100	182	010110110	335	101001111
132	010000100	113	001110001	185	010111001	343	101010111
136	010001000	114	001110010	186	010111010	347	101011011
144	010010000	116	001110100	188	010111100	349	101011101
160	010100000	120	001111000	199	011000111	350	101011110
192	011000000	135	010000111	203	011001011	359	101100111
257	100000001	139	010001011	205	011001101	363	101101011
258	100000010	141	010001101	206	011001110	365	101101101
260	100000100	142	010001110	211	011010011	366	101101110
264	100001000	147	010010011	213	011010101	371	101110011
272	100010000	149	010010101	214	011010110	373	101110101
288	100100000	150	010010110	217	011011001	374	101110110
320	101000000	153	010011001	218	011011010	377	101111001
384	110000000	154	010011010	220	011011100	378	101111010
7	000000111	156	010011100	227	011100011	380	101111100
11	000001011	163	010100011	229	011100101	399	110001111
13	000001101	165	010100101	230	011100110	407	110010111
14	000001110	166	010100110	233	011101001	411	110011011

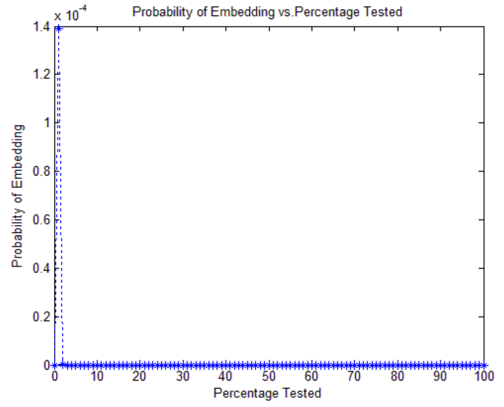
value	Binary rep.	value	Binary rep.	value	Binary rep.	value	Binary rep.
19	000010011	169	010101001	234	011101010	413	110011101
21	000010101	170	010101010	236	011101100	414	110011110
22	000010110	172	010101100	241	011110001	423	110100111
25	000011001	177	010110001	242	011110010	427	110101011
26	000011010	178	010110010	244	011110100	429	110101101
28	000011100	180	010110100	248	011111000	430	110101110
35	000100011	184	010111000	271	100001111	435	110110011
37	000100101	195	011000011	279	100010111	437	110110101
38	000100110	197	011000101	283	100011011	438	110110110
41	000101001	198	011000110	285	100011101	441	110111001
42	000101010	201	011001001	286	100011110	442	110111010
44	000101100	202	011001010	295	100100111	444	110111100
49	000110001	204	011001100	299	100101011	455	111000111
50	000110010	209	011010001	301	100101101	459	111001011
52	000110100	210	011010010	302	100101110	461	111001101
56	000111000	212	011010100	307	100110011	462	111001110
67	001000011	216	011011000	309	100110101	467	111010011
69	001000101	225	011100001	310	100110110	469	111010101
70	001000110	226	011100010	313	100111001	470	111010110
73	001001001	228	011100100	314	100111010	473	111011001
74	001001010	232	011101000	316	100111100	474	111011010
76	001001100	240	011110000	327	101000111	476	111011100
81	001010001	263	100000111	331	101001011	483	111100011
82	001010010	267	100001011	333	101001101	485	111100101
84	001010100	269	100001101	334	101001110	486	111100110
88	001011000	270	100001110	339	101010011	489	111101001
97	001100001	275	100010011	341	101010101	490	111101010
98	001100010	277	100010101	342	101010110	492	111101100
100	001100100	278	100010110	345	101011001	497	111110001
104	001101000	281	100011001	346	101011010	498	111110010
112	001110000	282	100011010	348	101011100	500	111110100
131	010000011	284	100011100	355	101100011	504	111111000
133	010000101	291	100100011	357	101100101	127	001111111
134	010000110	293	100100101	358	101100110	191	010111111
137	010001001	294	100100110	361	101101001	223	011011111
138	010001010	297	100101001	362	101101010	239	011101111
140	010001100	298	100101010	364	101101100	247	011110111
145	010010001	300	100101100	369	101110001	251	011111011
146	010010010	305	100110001	370	101110010	253	011111101
148	010010100	306	100110010	372	101110100	254	011111110
152	010011000	308	100110100	376	101111000	319	100111111
161	010100001	312	100111000	391	110000111	351	101011111
162	010100010	323	101000011	395	110001011	367	101101111
164	010100100	325	101000101	397	110001101	375	101110111
168	010101000	326	101000110	398	110001110	379	101111011
176	010110000	329	101001001	403	110010011	381	101111101
193	011000001	330	101001010	405	110010101	382	101111110
194	011000010	332	101001100	406	110010110	415	110011111
196	011000100	337	101010001	409	110011001	431	110101111
200	011001000	338	101010010	410	110011010	439	110110111
208	011010000	340	101010100	412	110011100	443	110111011
224	011100000	344	101011000	419	110100011	445	110111101
259	100000011	353	101100001	421	110100101	446	110111110
261	100000101	354	101100010	422	110100110	463	111001111
262	100000110	356	101100100	425	110101001	471	111010111
265	100001001	360	101101000	426	110101010	475	111011011
266	100001010	368	101110000	428	110101100	477	111011101
268	100001100	387	110000011	433	110110001	478	111011110
273	100010001	389	110000101	434	110110010	487	111100111

value	Binary rep.	value	Binary rep.	value	Binary rep.	value	Binary rep.
274	100010010	390	110000110	436	110110100	491	111101011
276	100010100	393	110001001	440	110111000	493	111101101
280	100011000	394	110001010	451	111000011	494	111101110
289	100100001	396	110001100	453	111000101	499	111110011
290	100100010	401	110010001	454	111000110	501	111110101
292	100100100	402	110010010	457	111001001	502	111110110
296	100101000	404	110010100	458	111001010	505	111111001
304	100110000	408	110011000	460	111001100	506	111111010
321	101000001	417	110100001	465	111010001	508	111111100
322	101000010	418	110100010	466	111010010	255	011111111
324	101000100	420	110100100	468	111010100	383	101111111
328	101001000	424	110101000	472	111011000	447	110111111
336	101010000	432	110110000	481	111100001	479	111011111
352	101100000	449	111000001	482	111100010	495	111101111
385	110000001	450	111000010	484	111100100	503	111110111
386	110000010	452	111000100	488	111101000	507	111111011
388	110000100	456	111001000	496	111110000	509	111111101
392	110001000	464	111010000	63	000111111	510	111111110
400	110010000	480	111100000	95	001011111	511	111111111

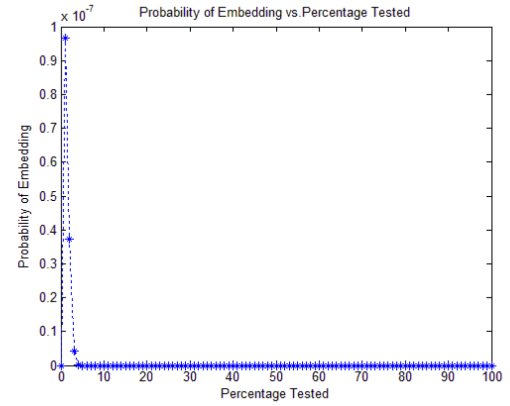
LSBR



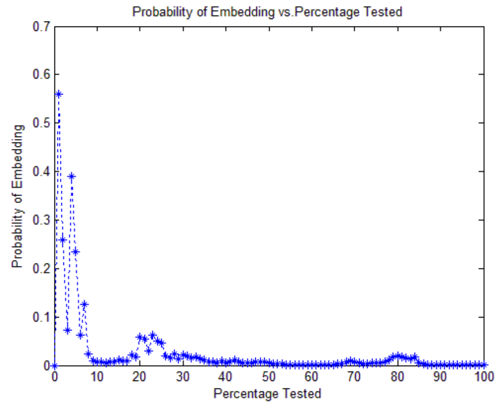
LSBM



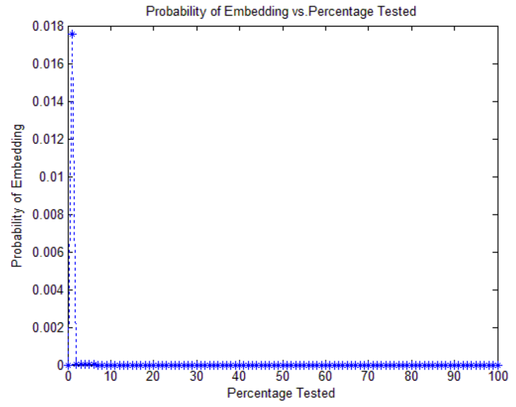
LSBMR



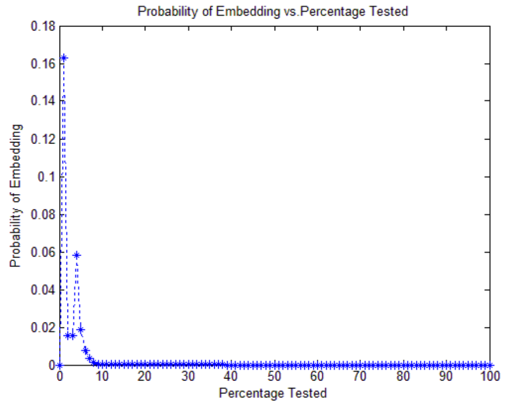
ILSBMR



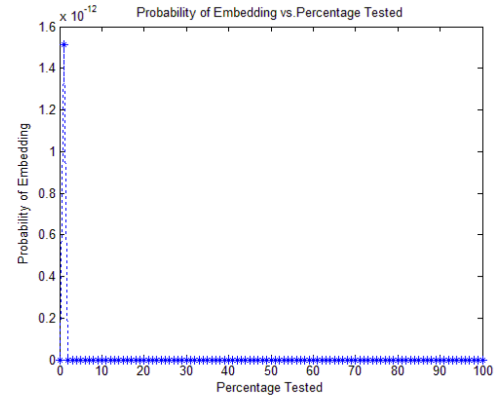
EB_SISR



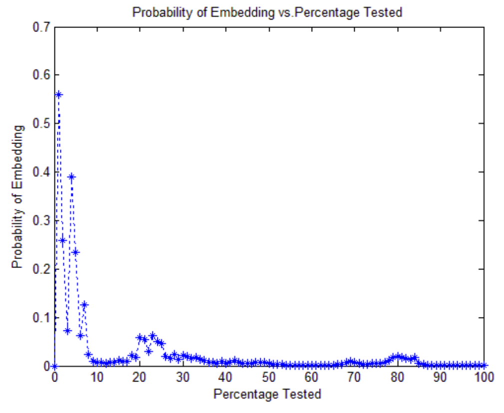
EB_SIM



EB_IWSIM



Fib_IWSIM



L_IWSIM

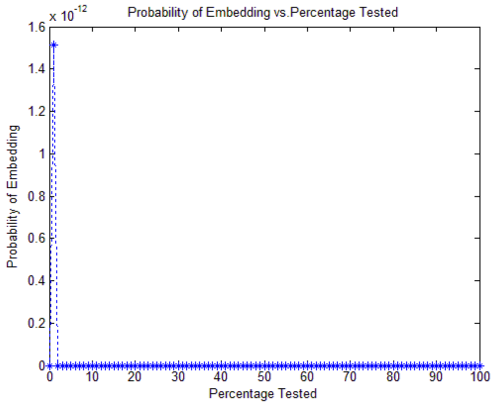
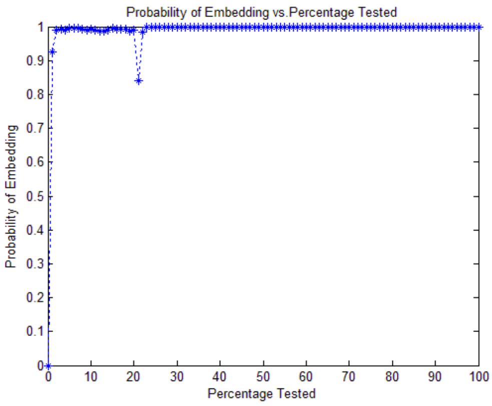
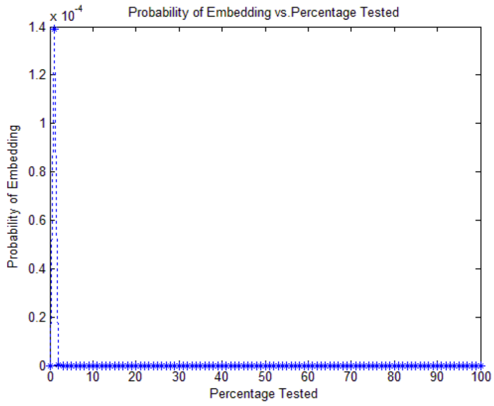


Figure A-1: PoV diagram for stego-image number 330 from SIPI database.

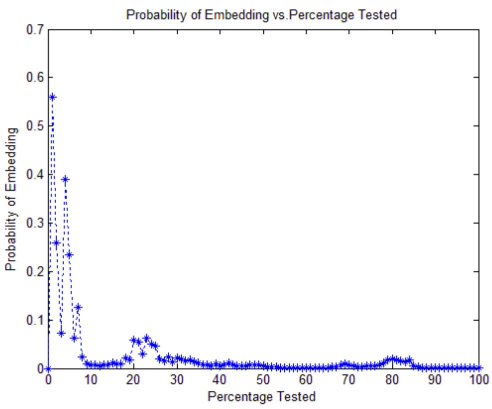
LSBR



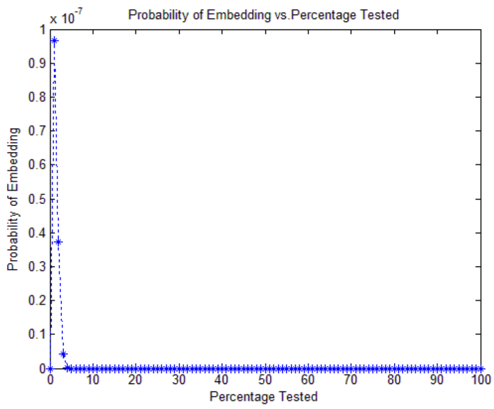
LSBM



LSBMR



ILSBMR



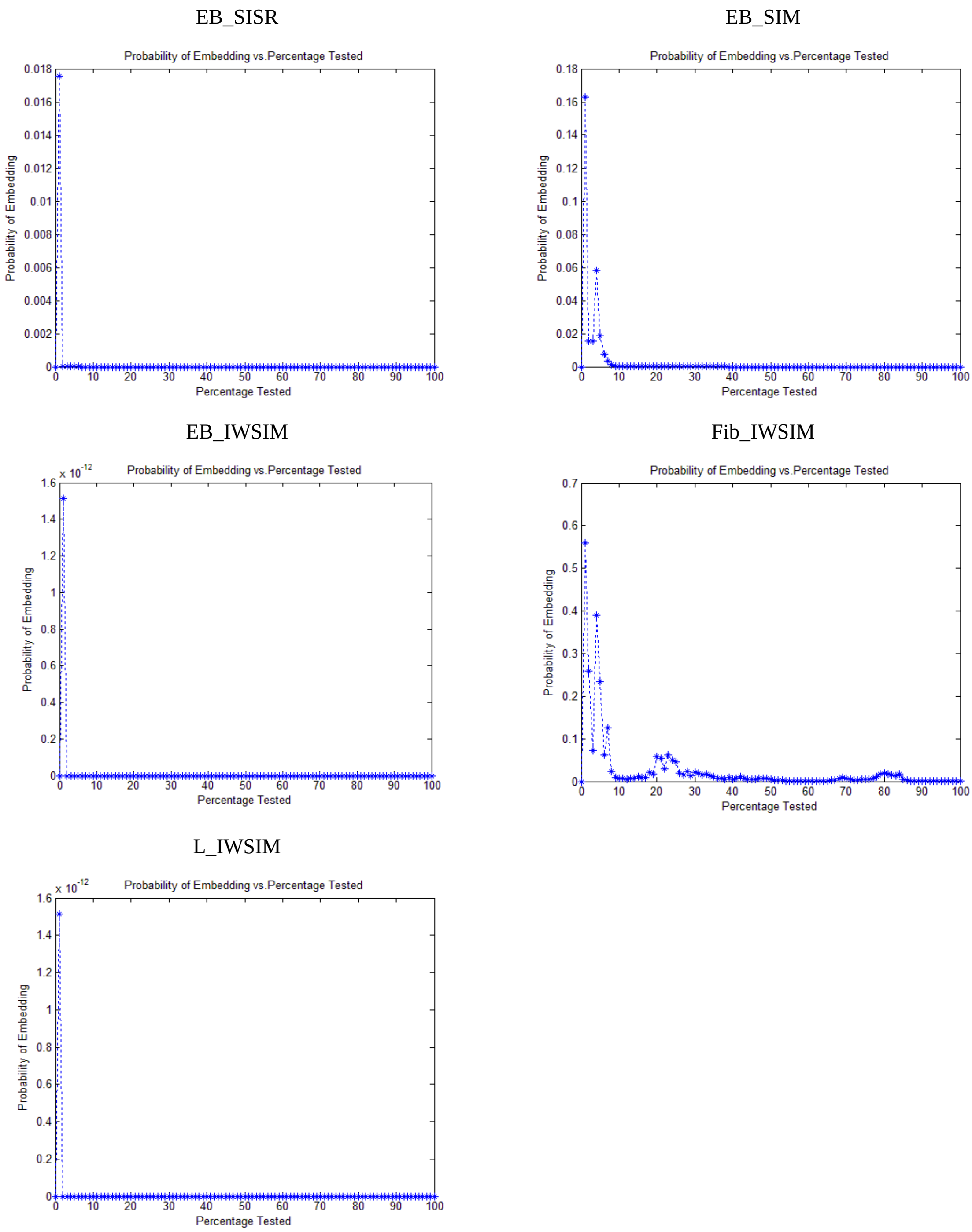
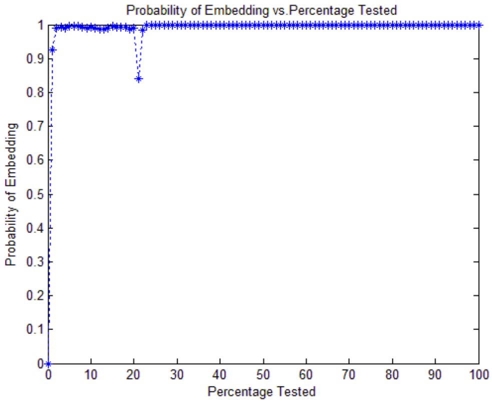
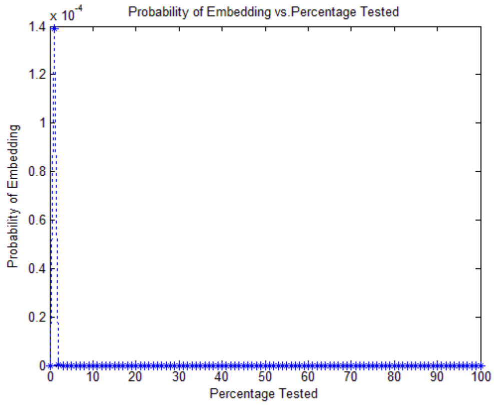


Figure A-2: PoV diagram for stego-image number 965 from SIPI database.

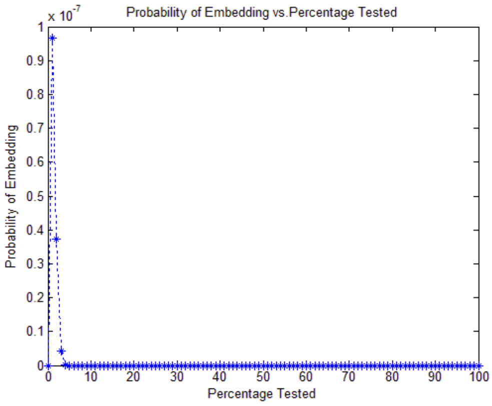
LSBR



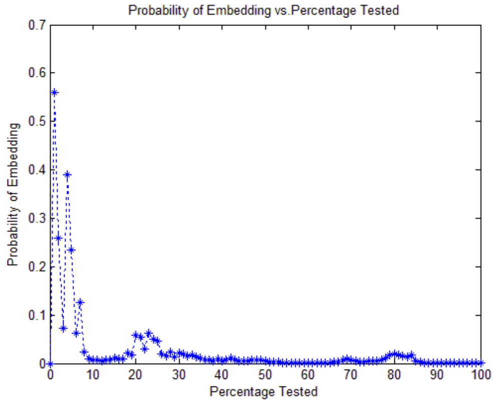
LSBM



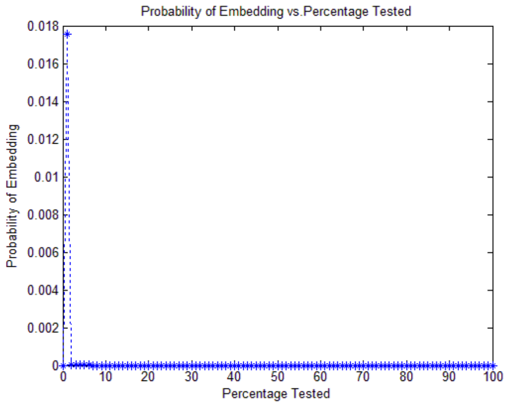
LSBMR



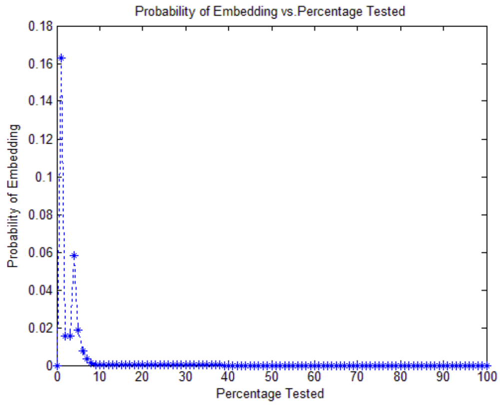
ILSBMR



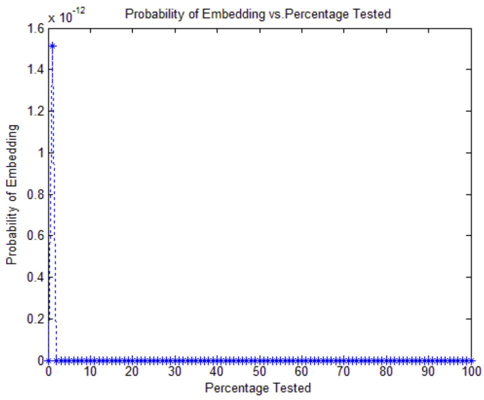
EB_SISR



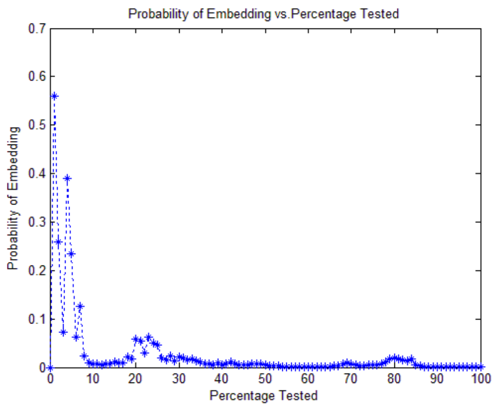
EB_SIM



EB_IWSIM



Fib_IWSIM



L_IWSIM

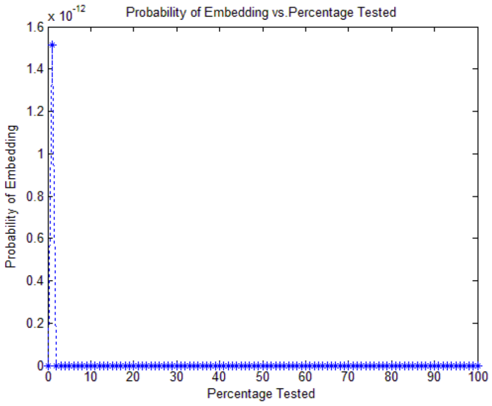
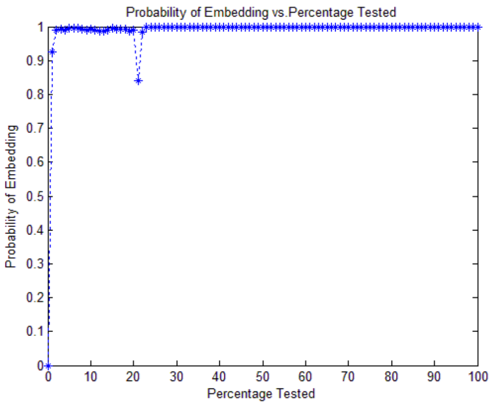
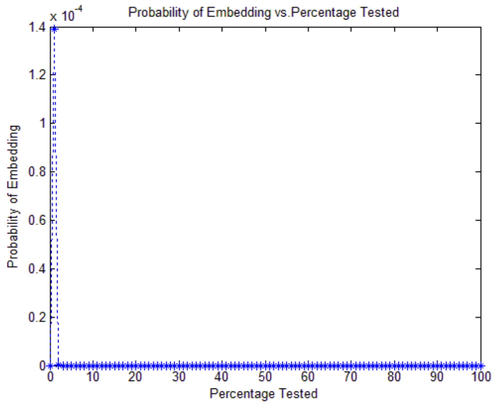


Figure A-3: PoV diagram for stego-image number 1023 from SIPI database.

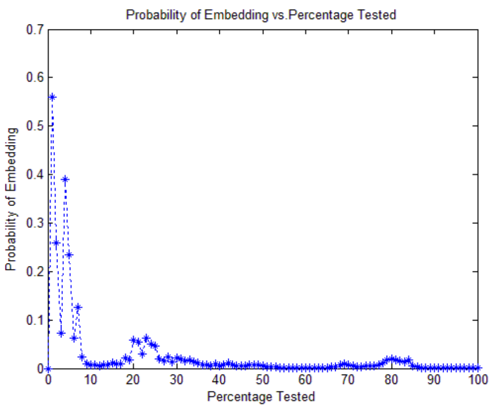
LSBR



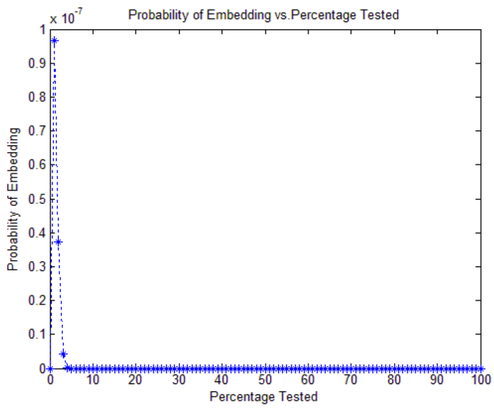
LSBM



LSBMR



ILSBMR



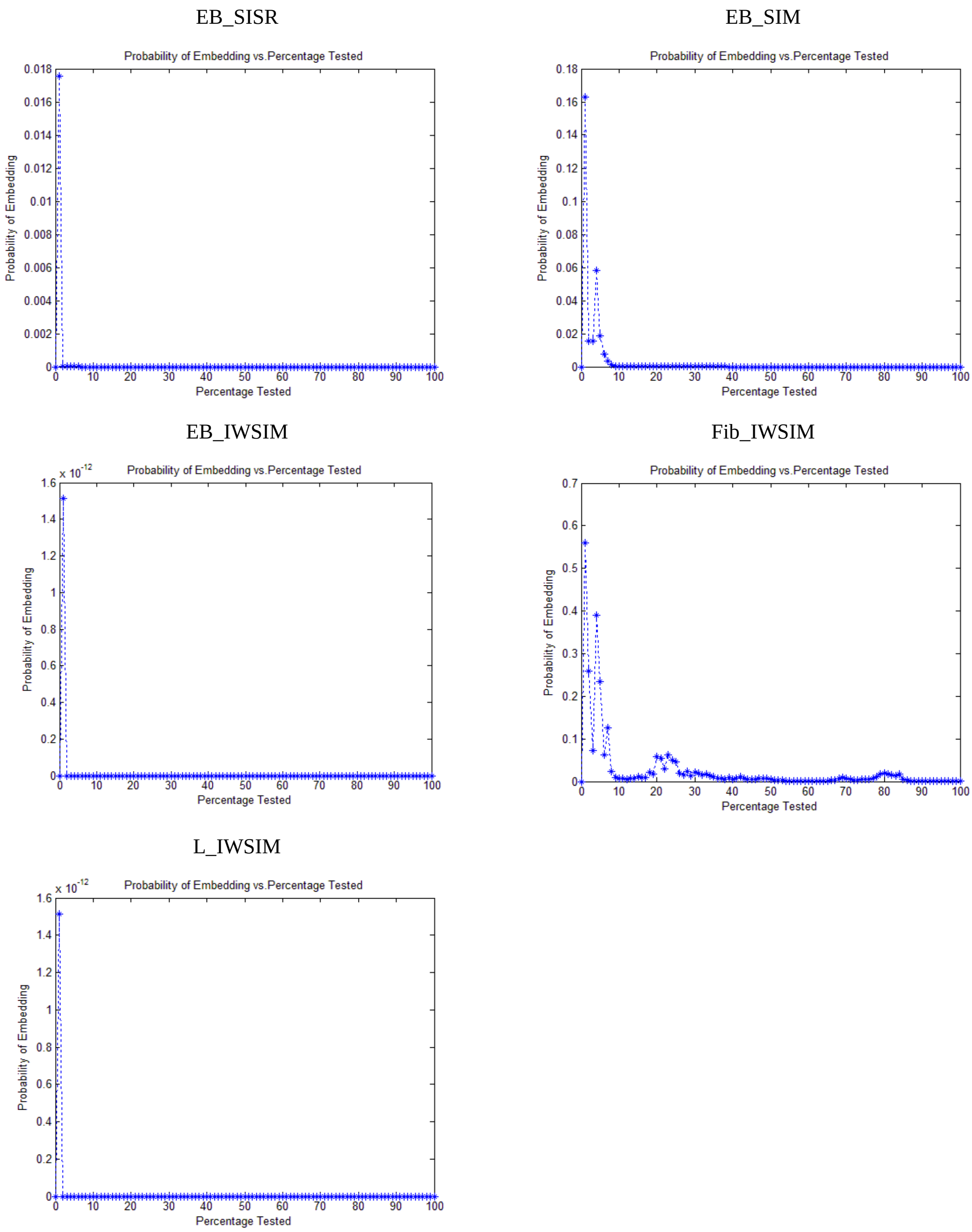
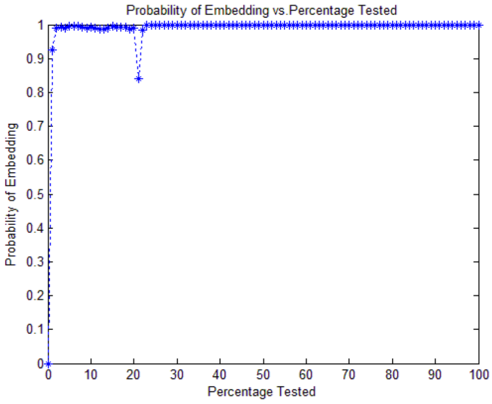
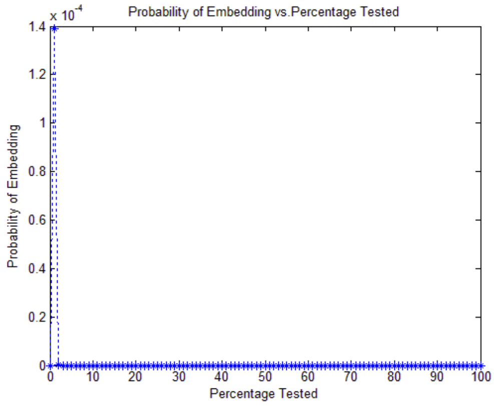


Figure A-4: PoV diagram for stego-image number 1417 from SIPI database.

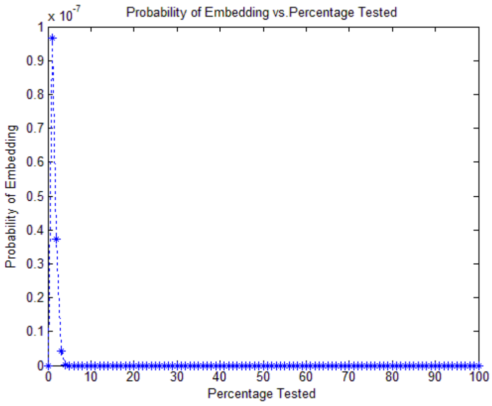
LSBR



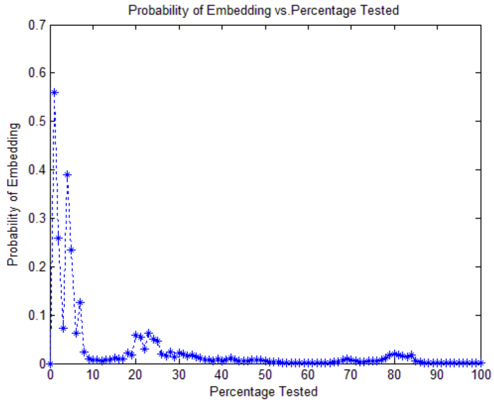
LSBM



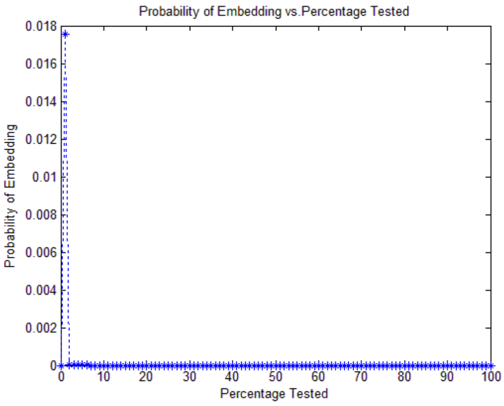
LSBMR



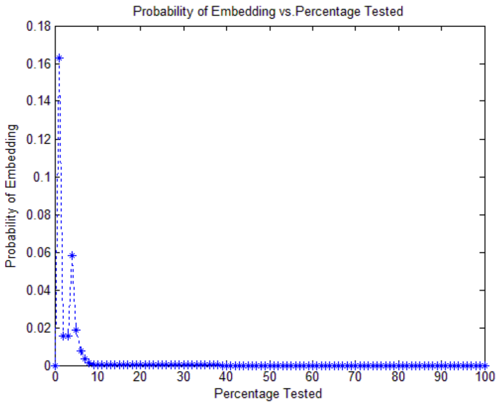
ILSBMR



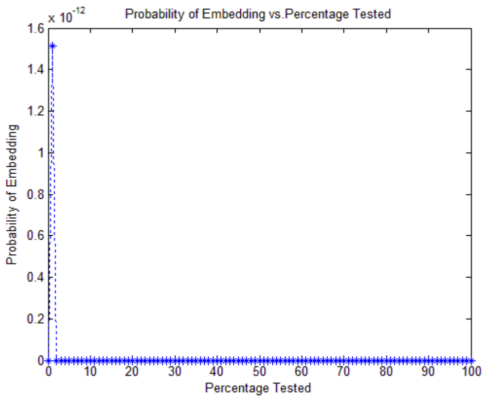
EB_SISR



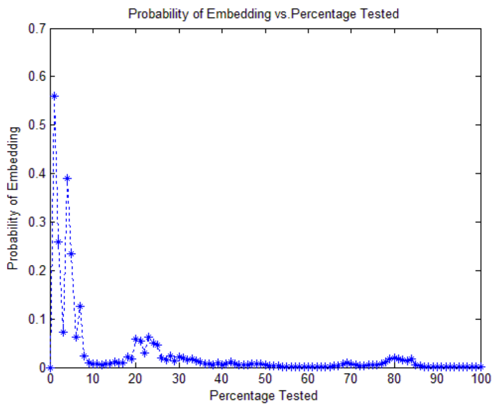
EB_SIM



EB_IWSIM



Fib_IWSIM



L_IWSIM

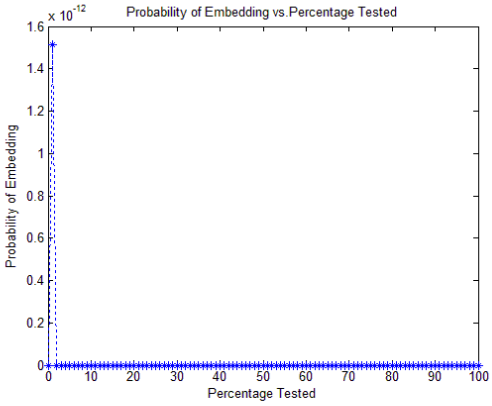
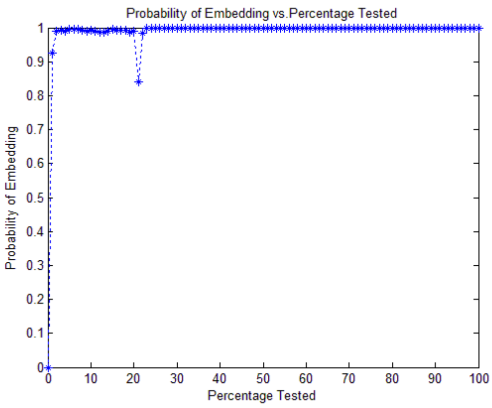
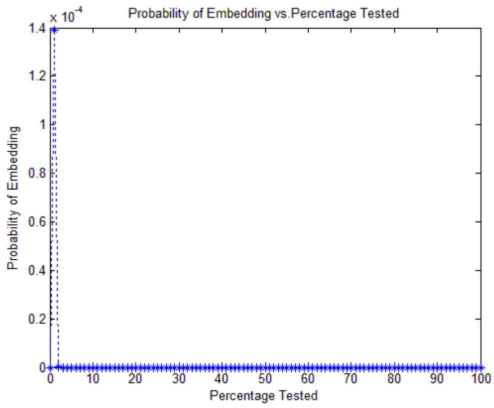


Figure A-5: PoV diagram for stego-image number 1832 from SIPI database.

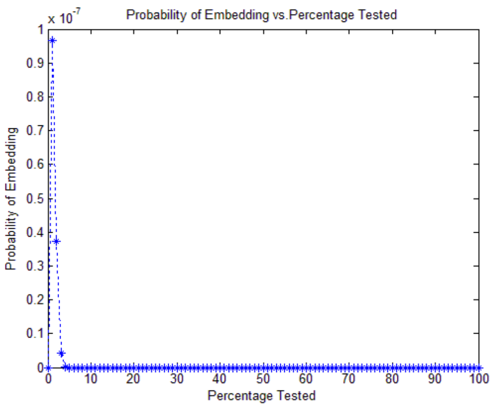
LSBR



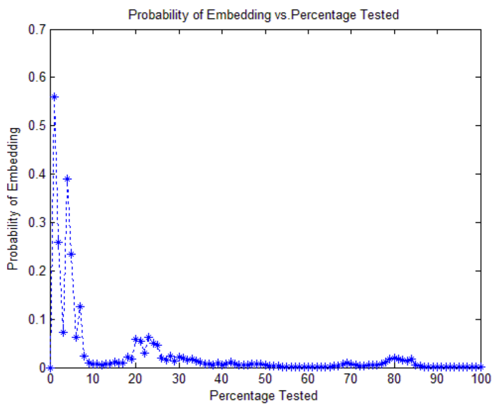
LSBM



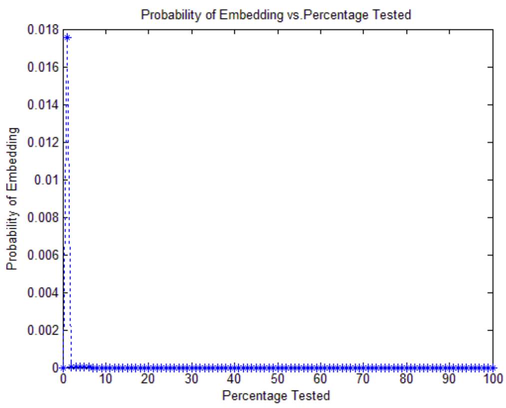
LSBMR



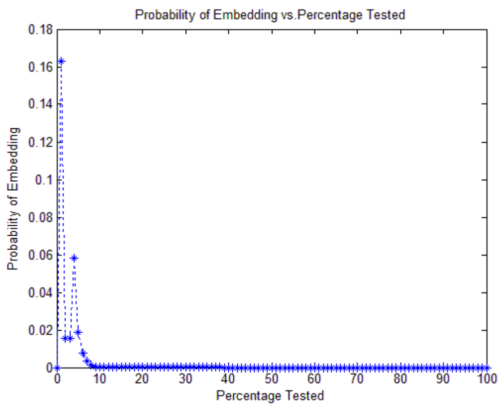
ILSBMR



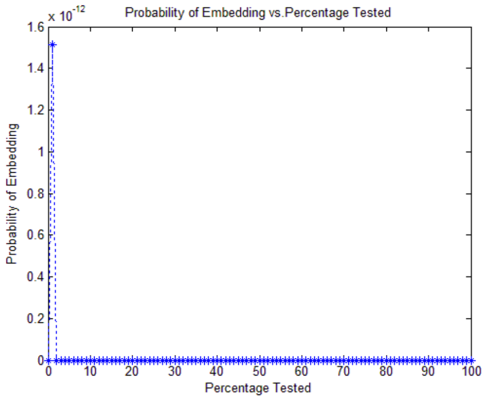
EB_SISR



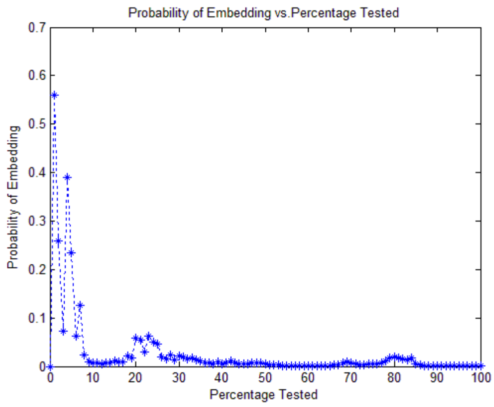
EB_SIM



EB_IWSIM



Fib_IWSIM



L_IWSIM

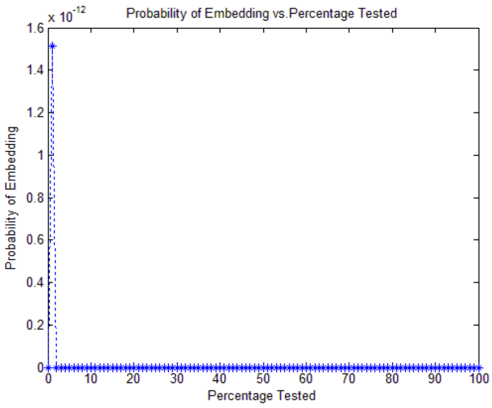
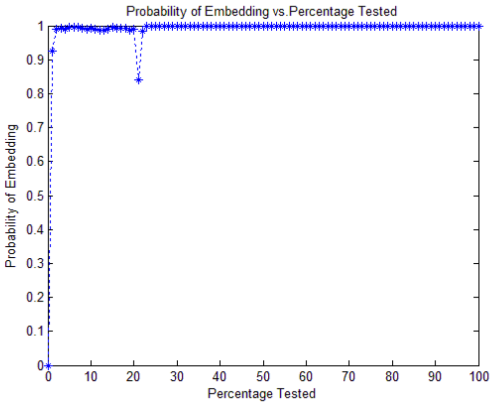
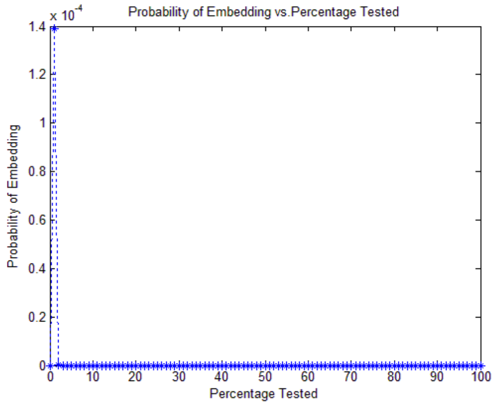


Figure A-6: PoV diagram for Stego-image number 122 from BOSSBase when the Lenna image was embedded.

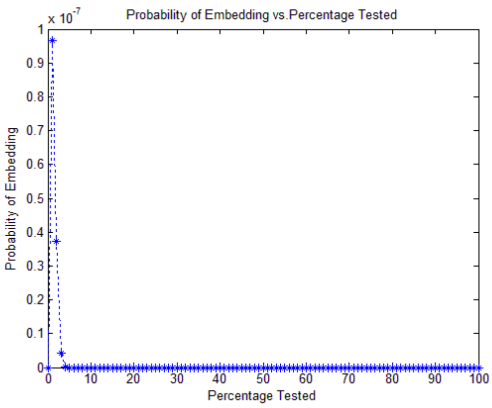
LSBR



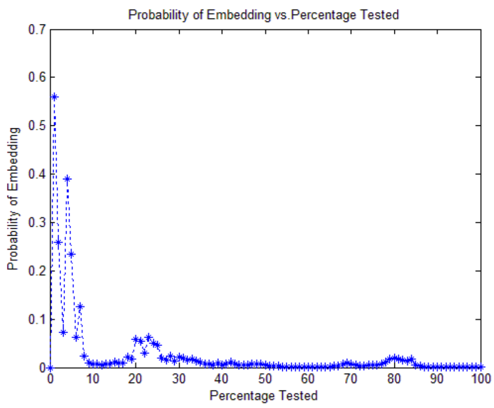
LSBM



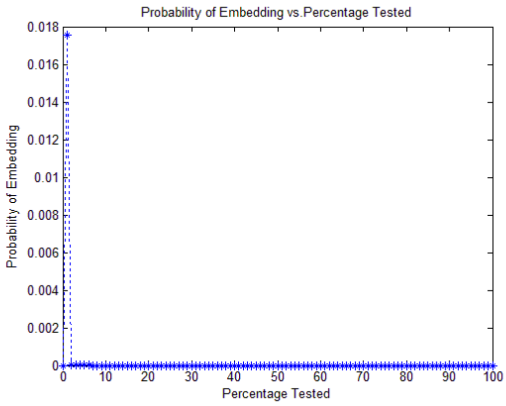
LSBMR



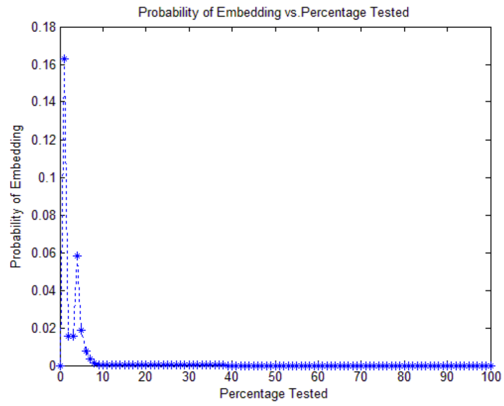
ILSBMR



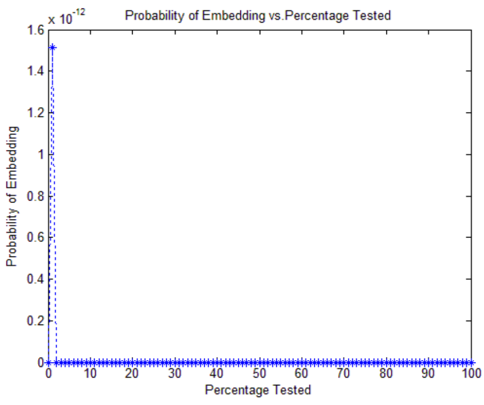
EB_SISR



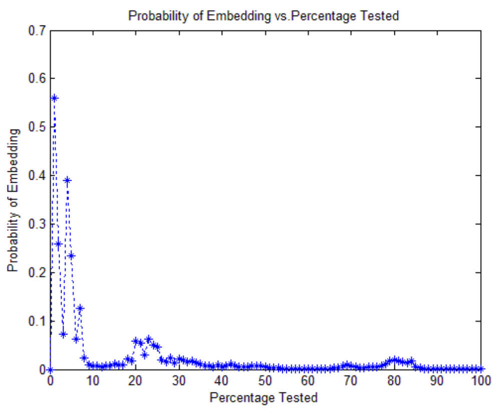
EB_SIM



EB_IWSIM



Fib_IWSIM



L_IWSIM

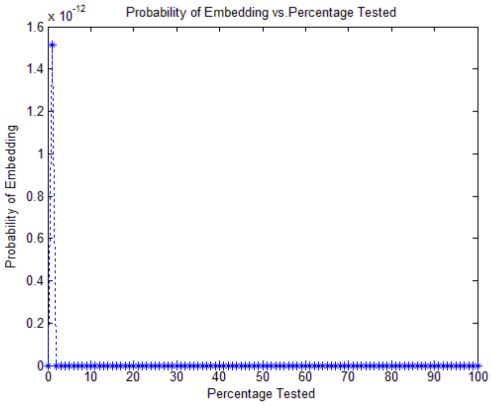
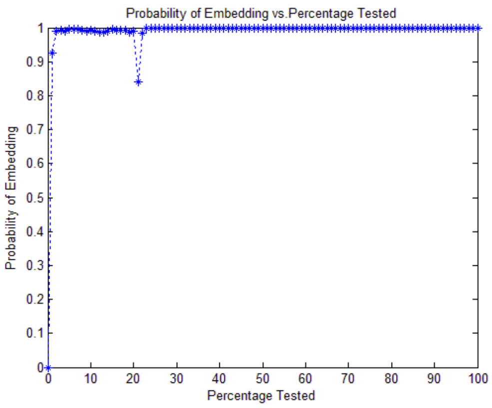
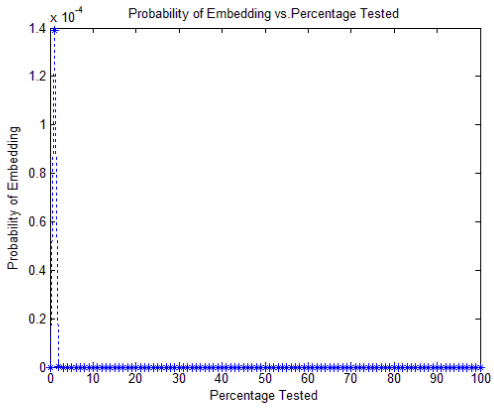


Figure A-7: PoV diagram for Stego-image number 489 from BOSSBase when the Lenna image was embedded.

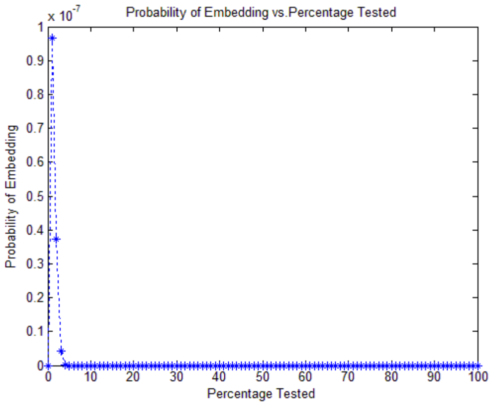
LSBR



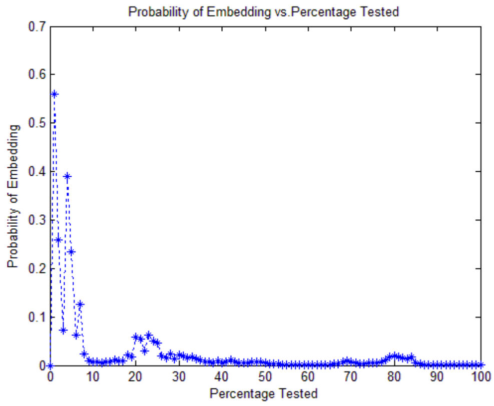
LSBM



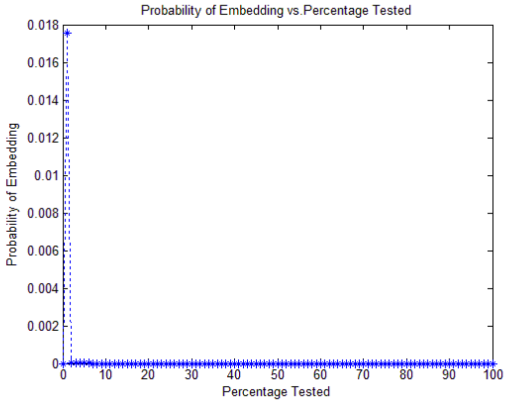
LSBMR



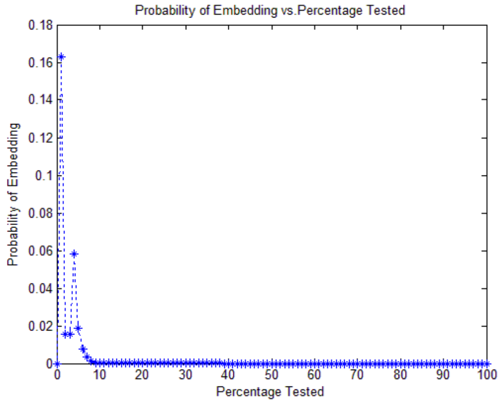
ILSBMR



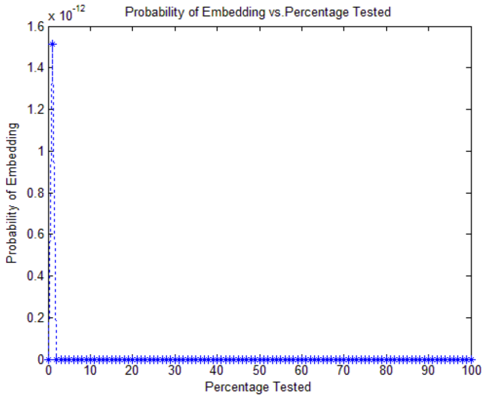
EB_SISR



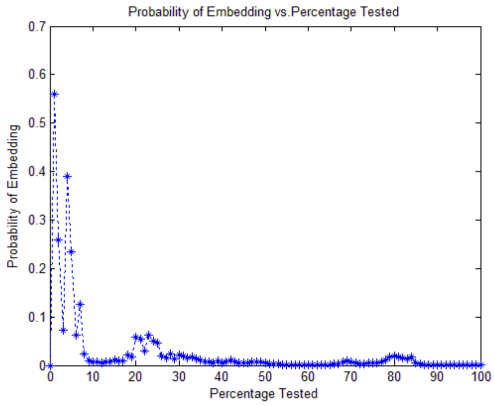
EB_SIM



EB_IWSIM



Fib_IWSIM



L_IWSIM

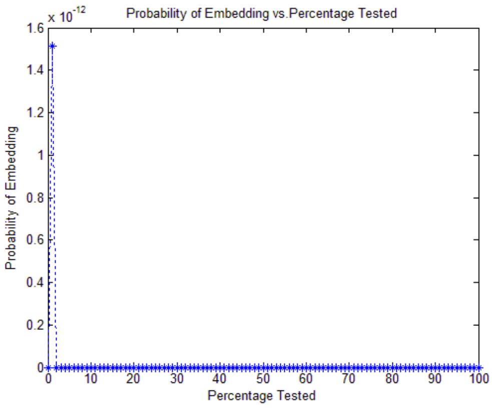
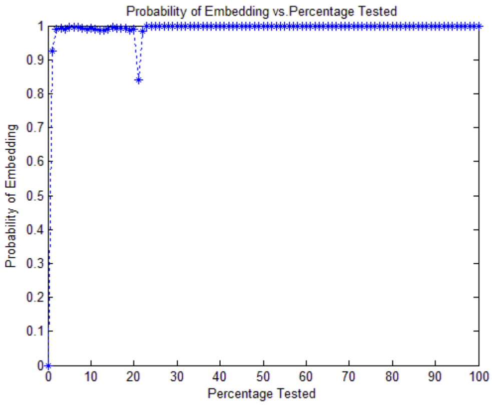
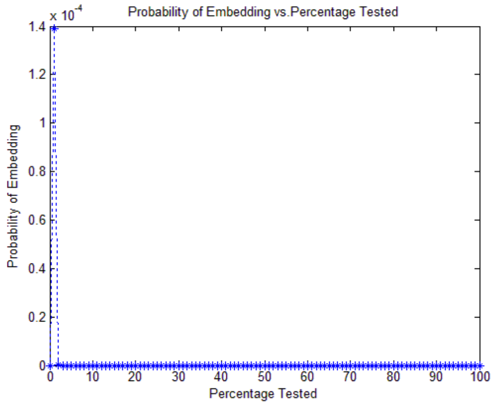


Figure A-8: PoV diagram for Stego-image number 664 from BOSSBase when the Lenna image was embedded.

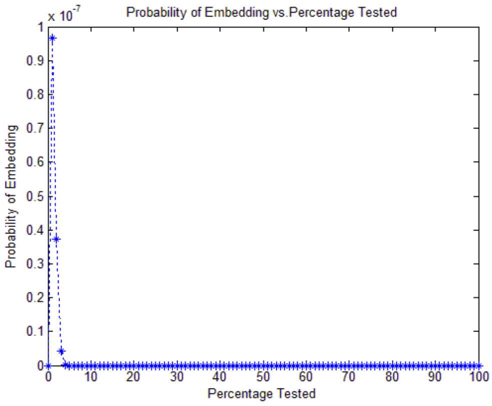
LSBR



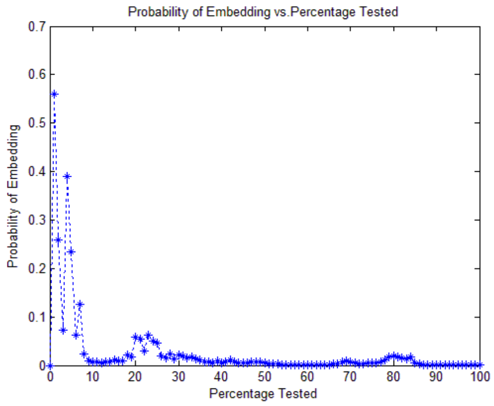
LSBM



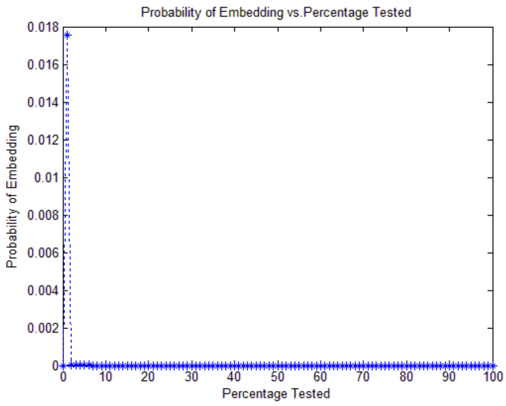
LSBMR



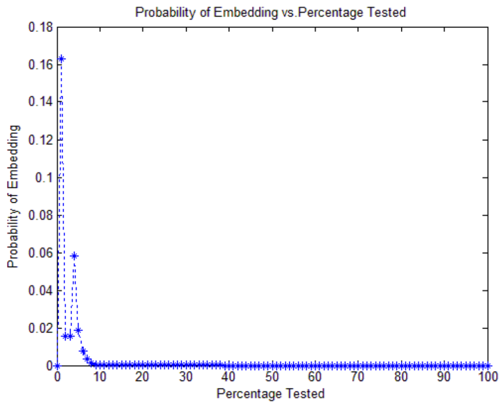
ILSBMR



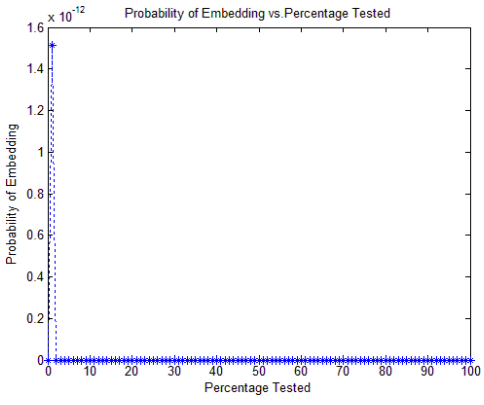
EB_SISR



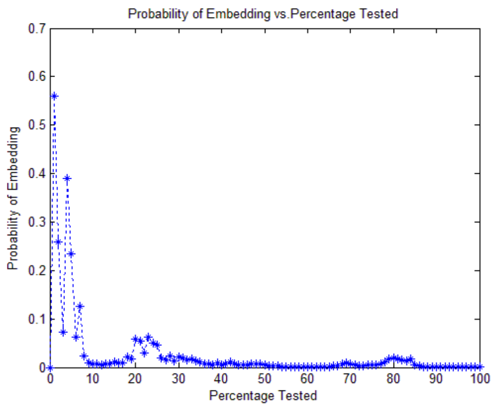
EB_SIM



EB_IWSIM



Fib_IWSIM



L_IWSIM

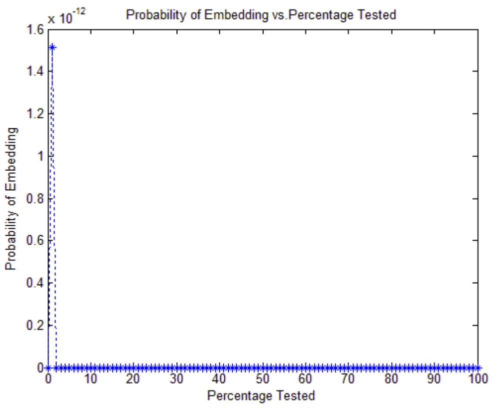
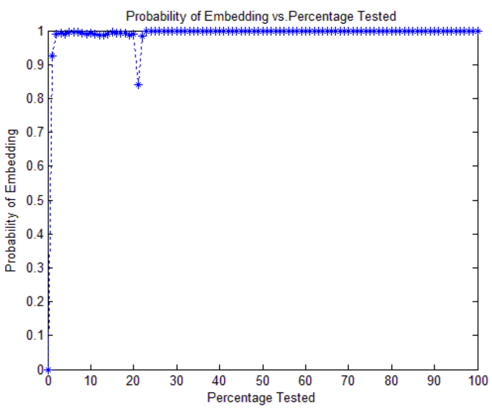
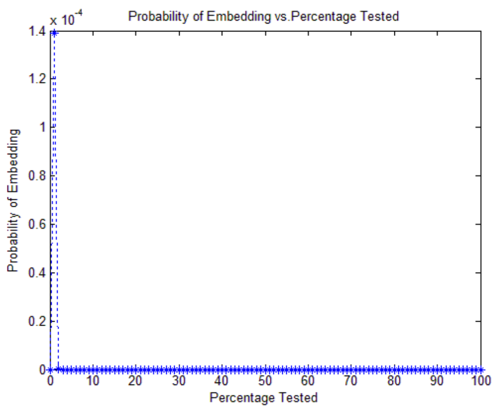


Figure A-9: PoV diagram for Stego-image number 855 from BOSSBase when the Lenna image was embedded.

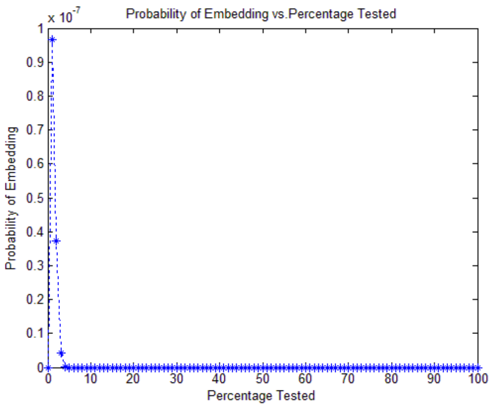
LSBR



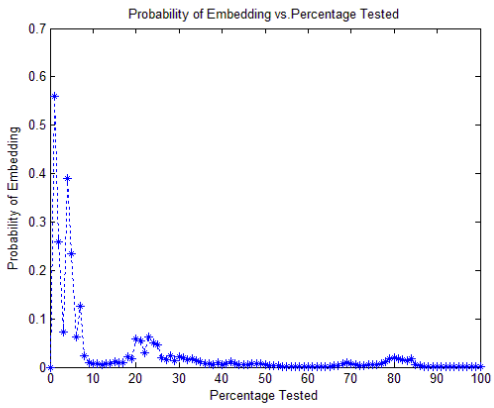
LSBM



LSBMR



ILSBMR



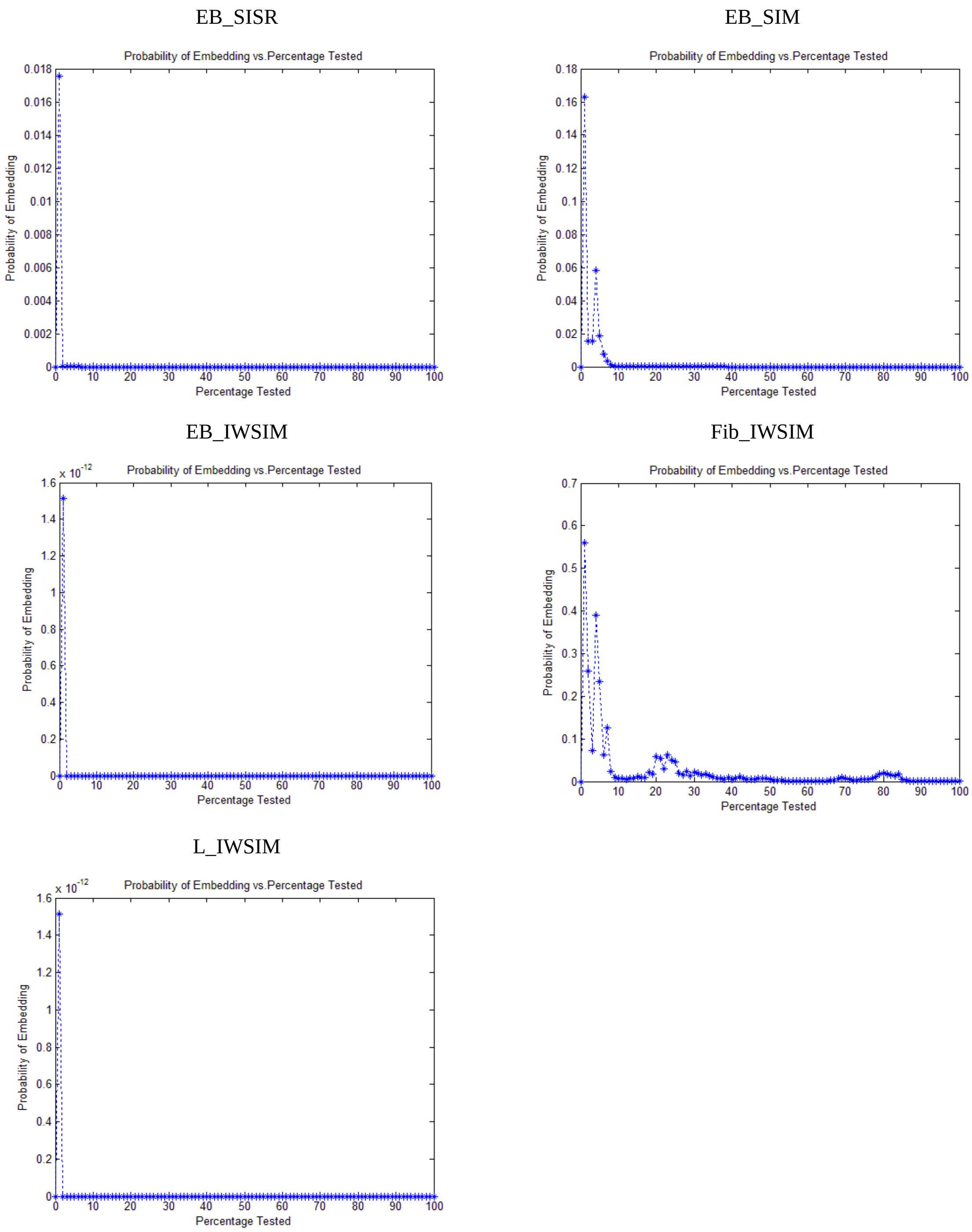
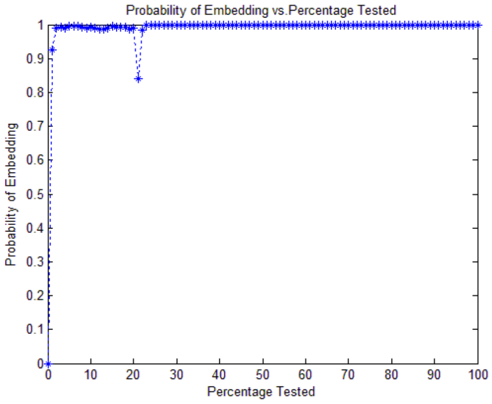
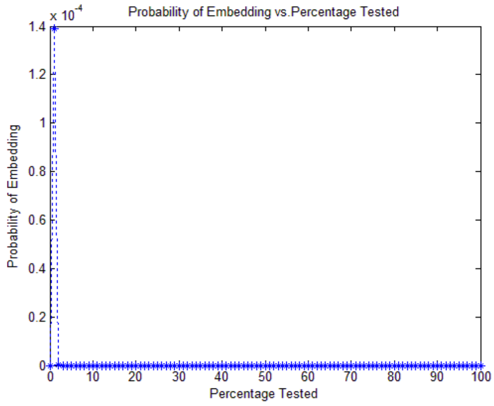


Figure A-10: PoV diagram for Stego-image number 970 from BOSSBase when the Lenna image was embedded.

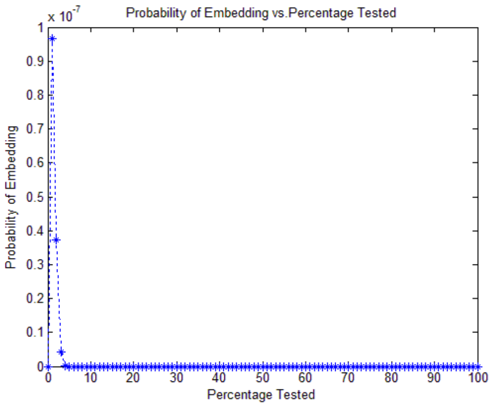
LSBR



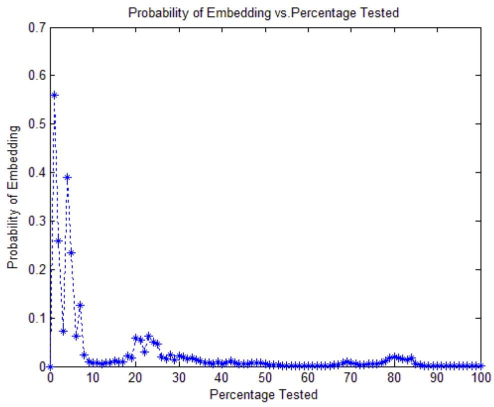
LSBM



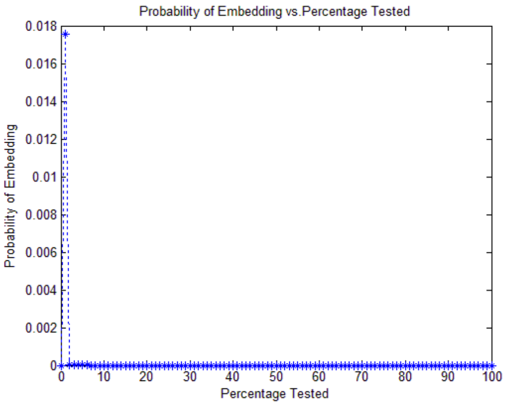
LSBMR



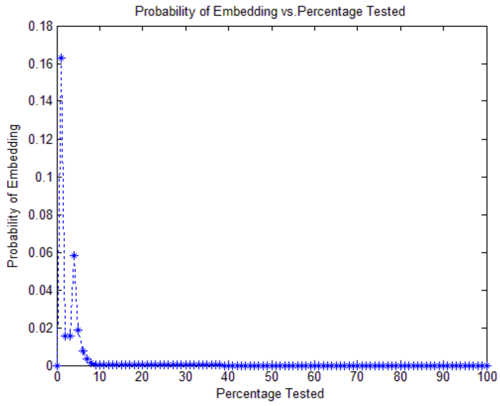
ILSBMR



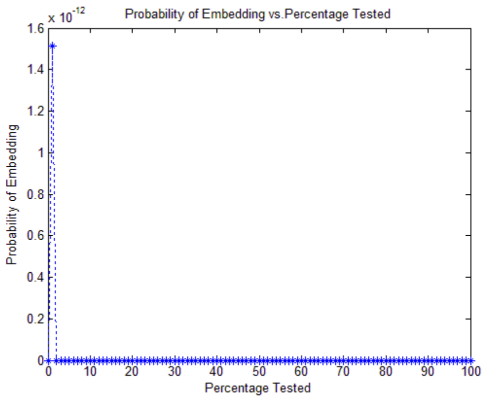
EB_SISR



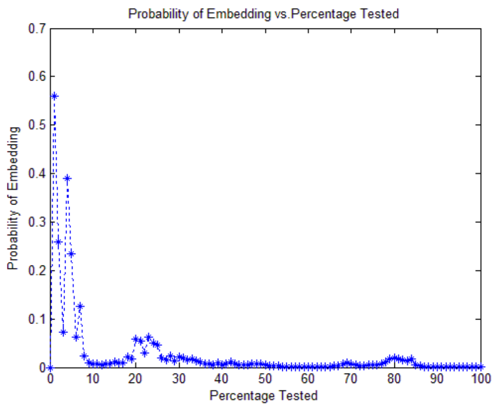
EB_SIM



EB_IWSIM



Fib_IWSIM



L_IWSIM

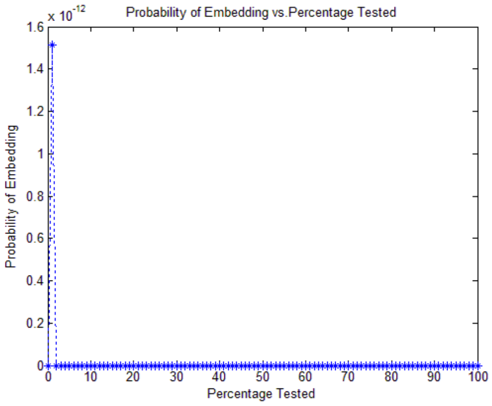
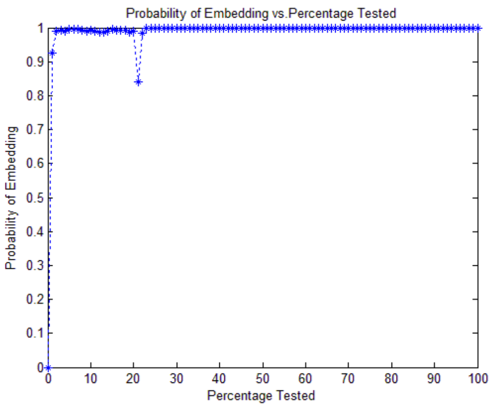
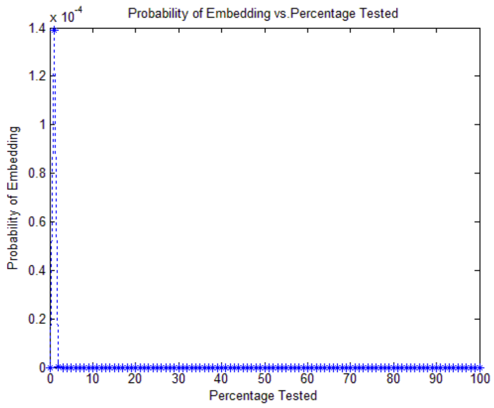


Figure A-11: PoV diagram for Stego-image number 122 from BOSSBase when the Jet image was embedded.

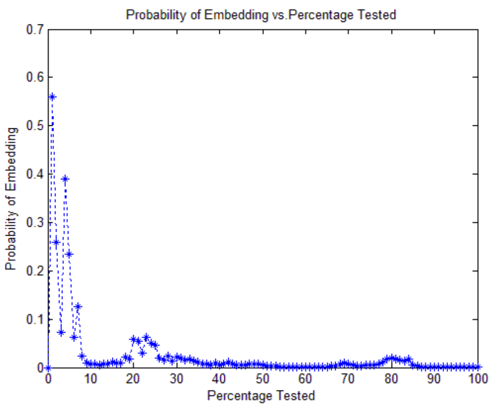
LSBR



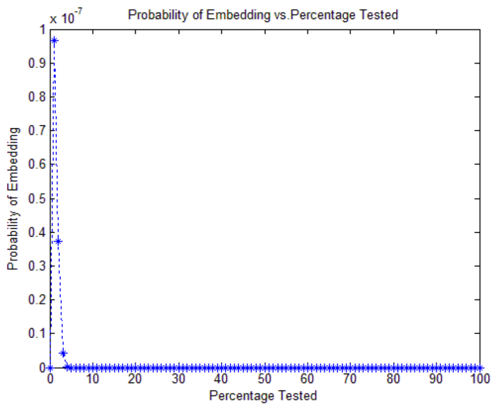
LSBM



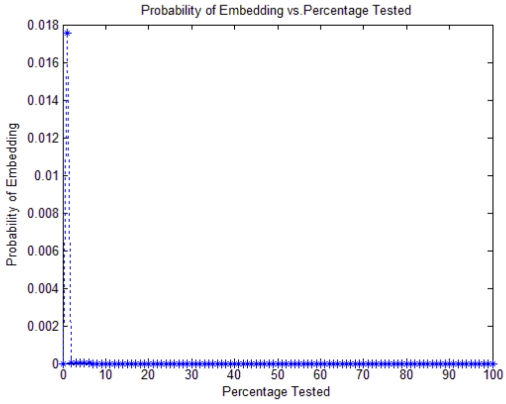
LSBMR



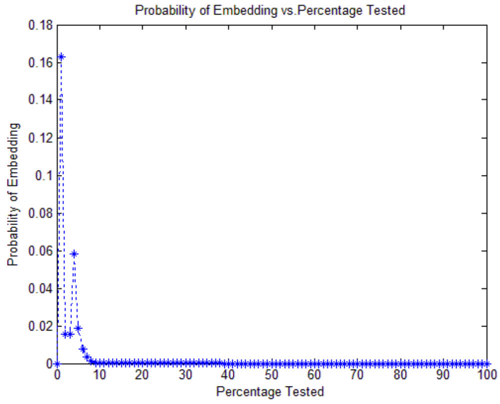
ILSBMR



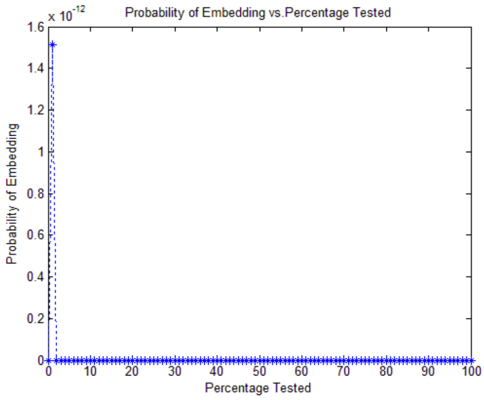
EB_SISR



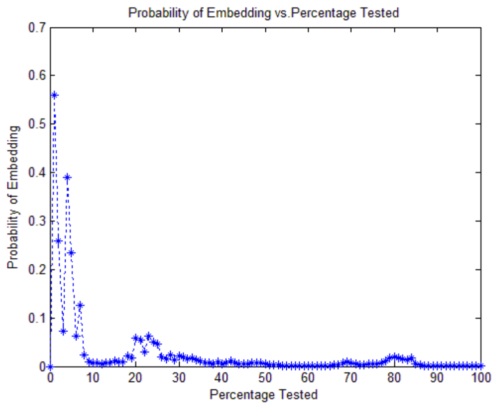
EB_SIM



EB_IWSIM



Fib_IWSIM



L_IWSIM

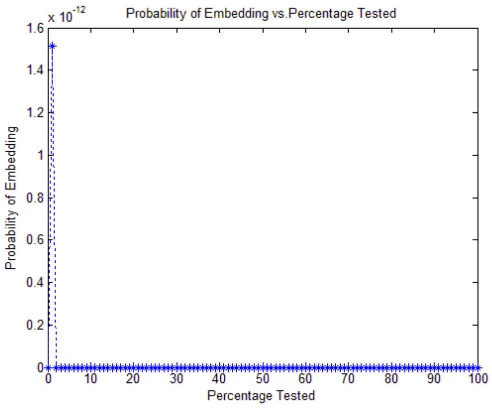
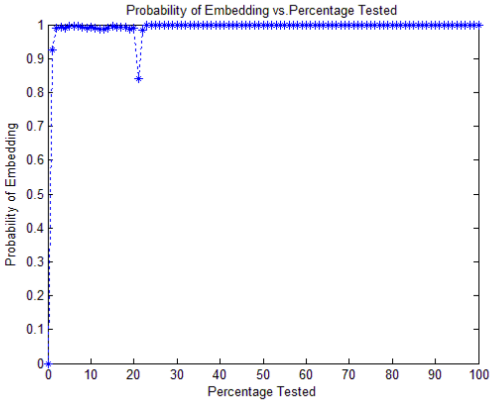
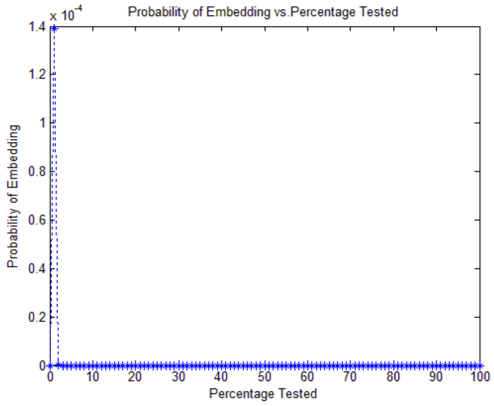


Figure A-12: PoV diagram for Stego-image number 489 from BOSSBase when the Jet image was embedded.

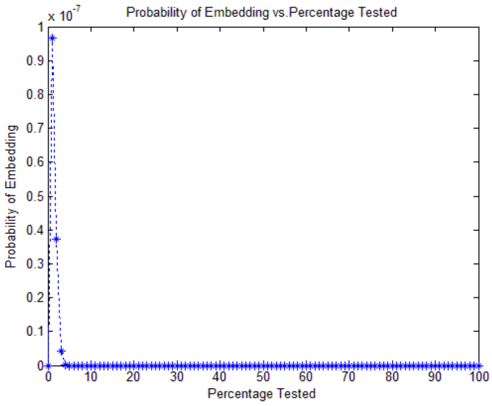
LSBR



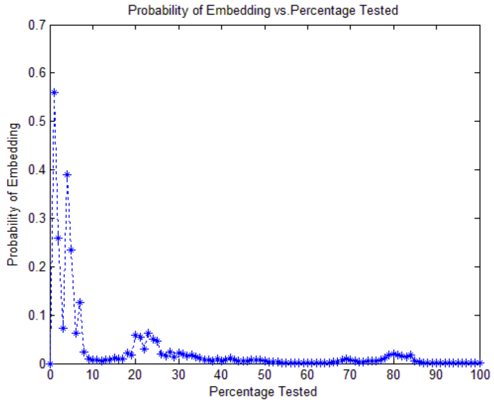
LSBM



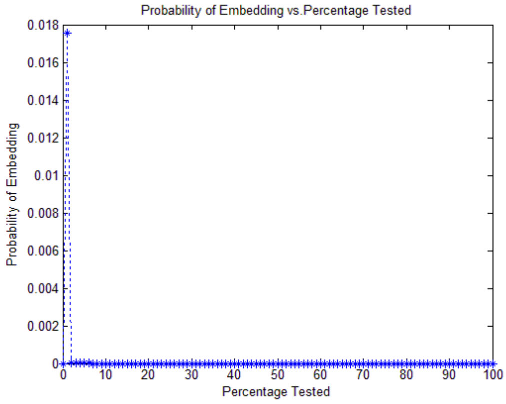
LSBMR



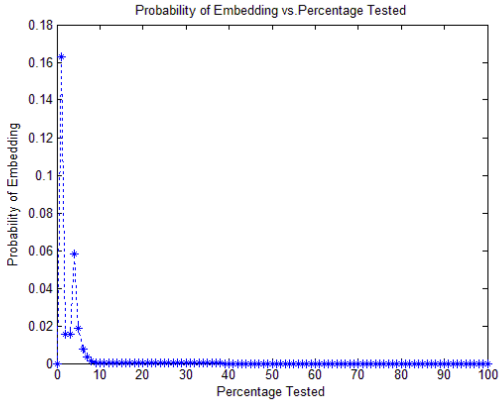
ILSBMR



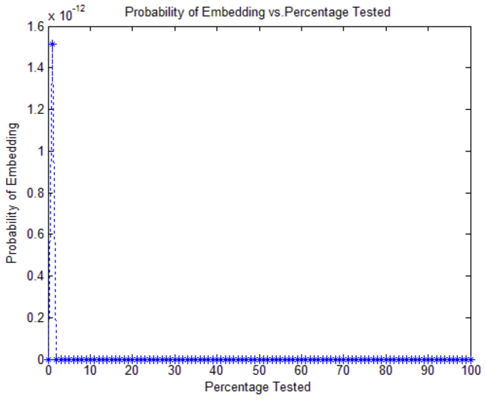
EB_SISR



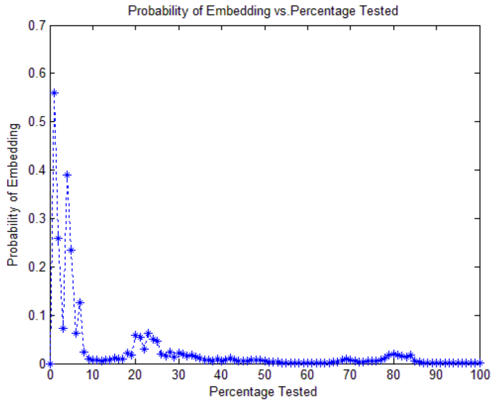
EB_SIM



EB_IWSIM



Fib_IWSIM



L_IWSIM

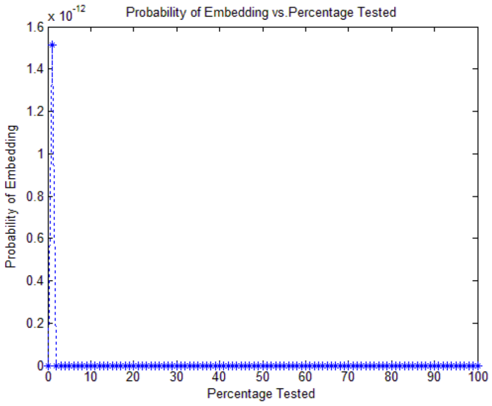
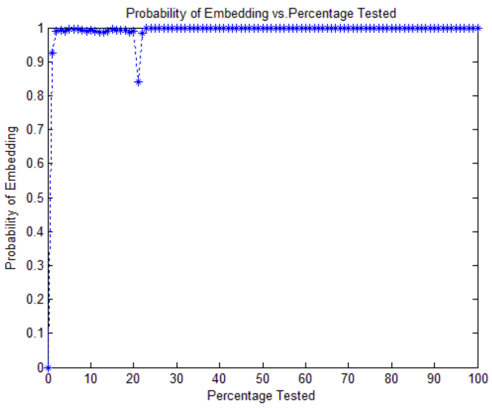
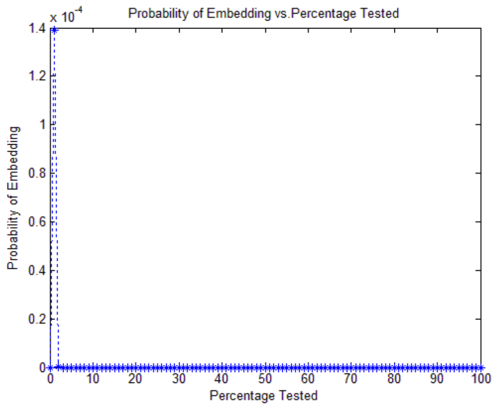


Figure A-13: PoV diagram for Stego-image number 664 from BOSSBase when the Jet image was embedded.

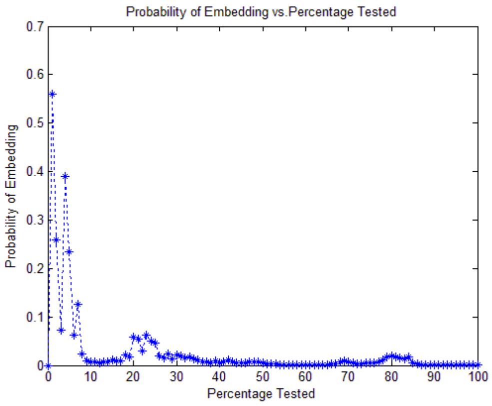
LSBR



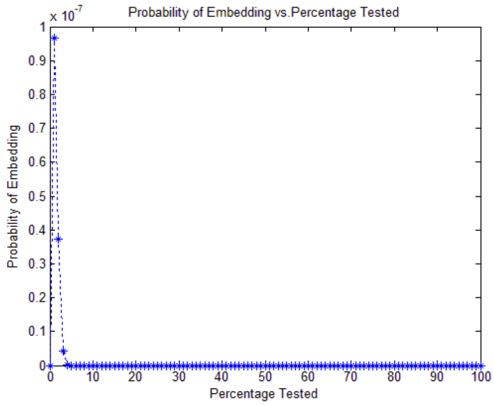
LSBM



LSBMR



ILSBMR



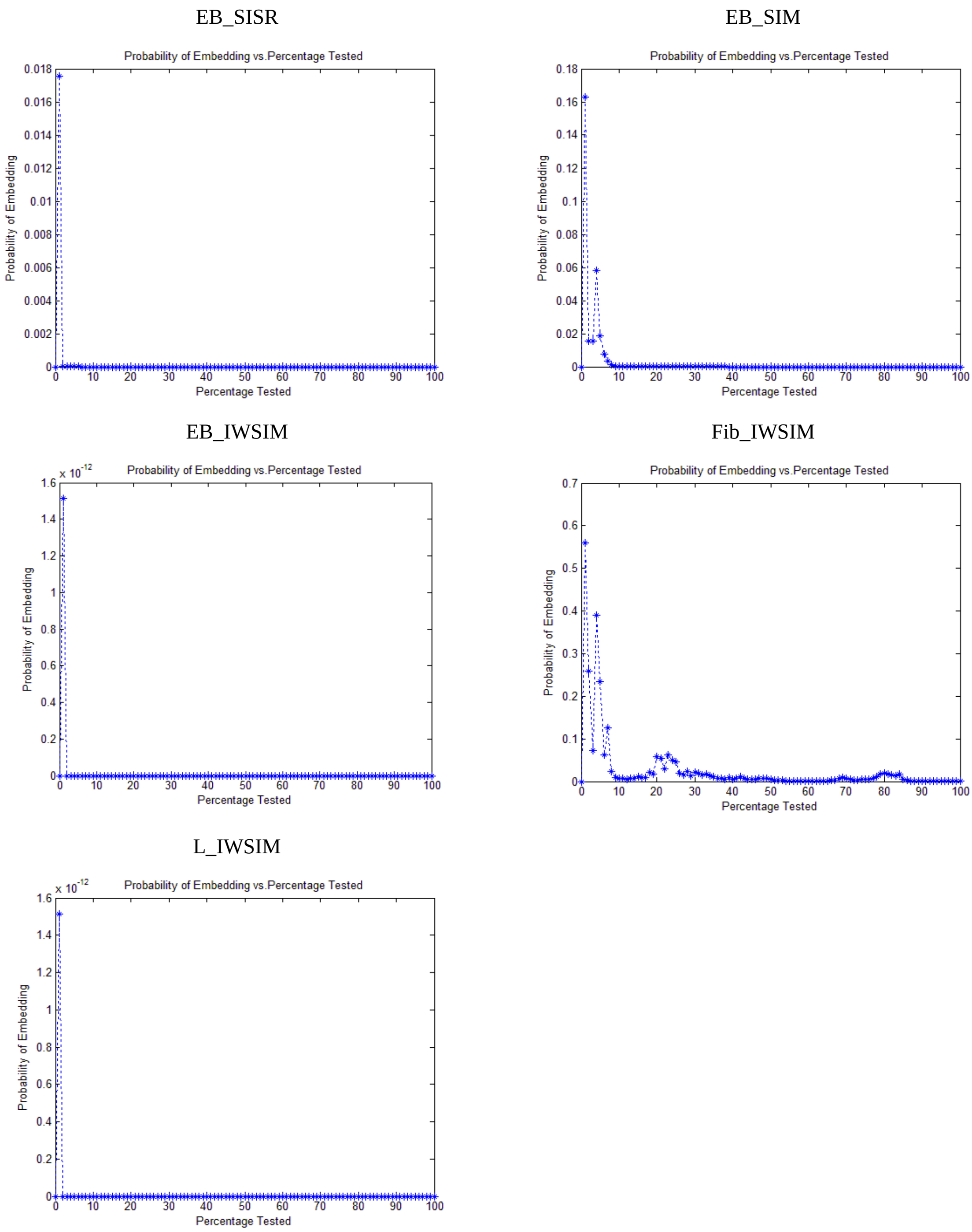
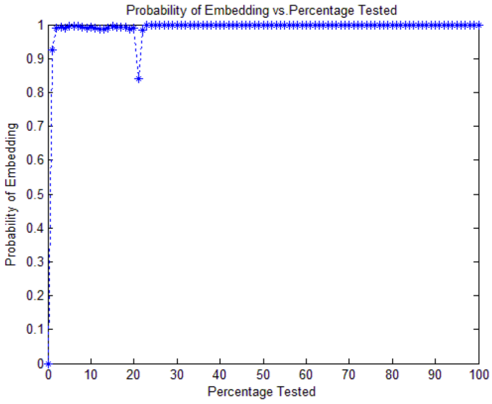
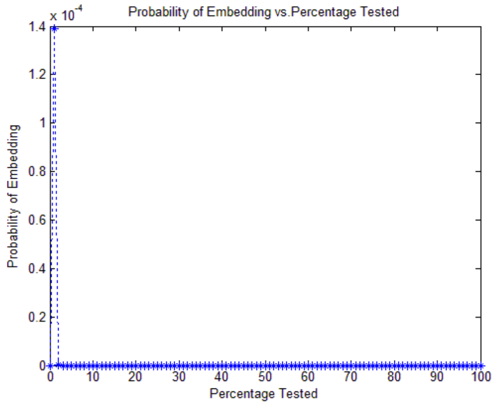


Figure A-14: PoV diagram for Stego-image number 855 from BOSSBase when the Jet image was embedded.

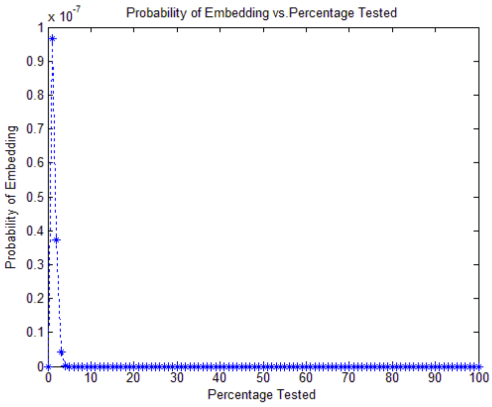
LSBR



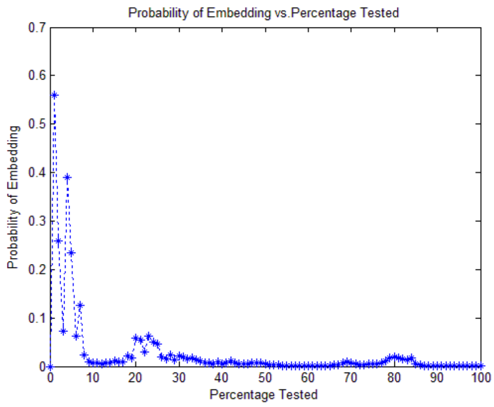
LSBM



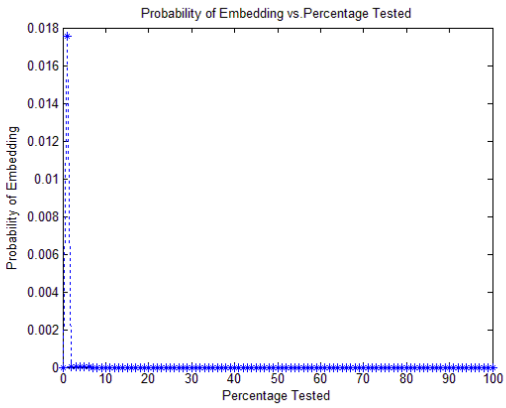
LSBMR



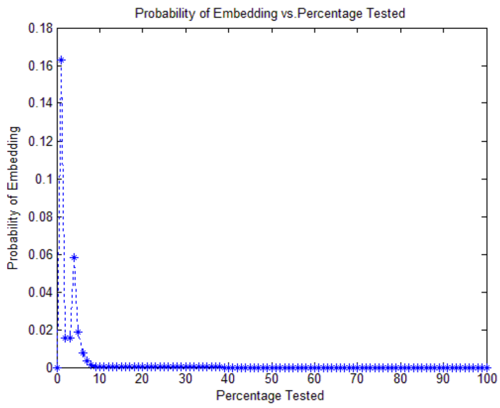
ILSBMR



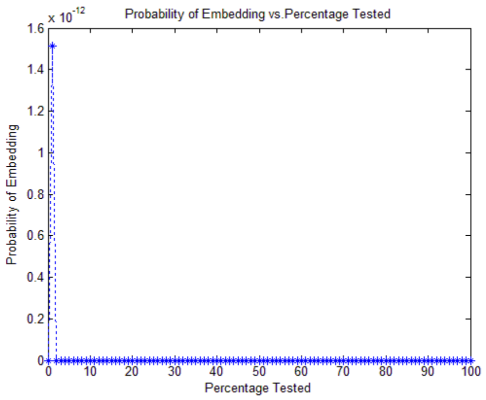
EB_SISR



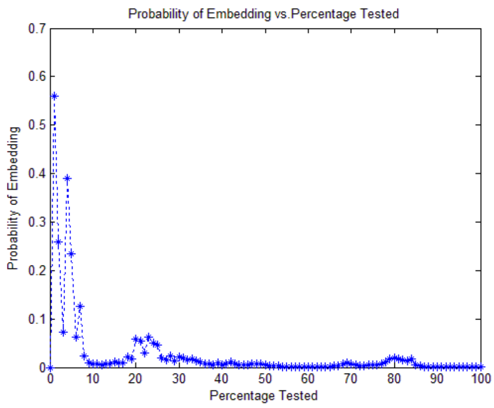
EB_SIM



EB_IWSIM



Fib_IWSIM



L_IWSIM

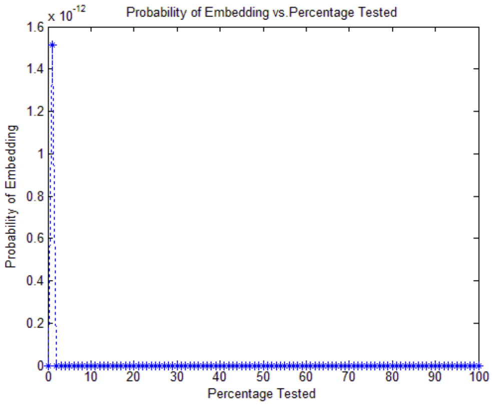


Figure A-15: PoV diagram for Stego-image number 970 from BOSSBase when the Jet image was embedded.