University of Geneva

Faculty of Science
Department of Computer science

Master's Thesis

XOR Message Encryption Based on JPEG Perturbed Quantization Steganography

Author:
Sarah Sabbagh

Supervisor:
Dr. Eduardo Solana

September 2017

UNIVERSITY OF GENEVA

Abstract

Faculty of Science

Master of Computer Science

XOR Message Encryption Based on JPEG Perturbed Quantization Steganography

by Sarah SABBAGH

Covert communication provided by steganography has various applications, mainly in military agencies, healthcare industries, and government intelligence agencies. In some countries where the use of cryptography is prohibited and punishable by law, steganography can be a safer alternative. Every steganographic system aims to provide users with a communication channel that cannot be detected by third parties. A system that fails at this requirement is unreliable and cannot be used. As steganographic techniques are numerous, a variable-rate random linear code, named Wet Paper Codes method, stands out by its high channel capacity and arbitrary selection channel that is not shared between the users. This system is used jointly with Perturbed Quantization wherein an information-reducing operation is applied on the cover image in order to use the elements which values are the most unpredictable to embed message bits using groups of these changeable elements. This thesis proposes a modification of the system by an additional XOR processing of non-random messages before embedding and the testing of a more advanced selection of changeable elements in cover images in order to improve the security of the system. This is followed by a series of experiments on 1000 images that showed that the system performs as good as its precursor by being less detectable than other JPEG steganographic techniques while running faster embedding. The accessible steganographic system developed in this work could be used to exchange covert messages between users or groups of people using current social networking services.

"If you don't respect your enemy's capabilities, you're in for one nasty surprise after another."

Garrus Vakarian

Contents

Abstract **2**

Acknowledgements **6**

List of Figures **8**

List of Tables **9**

List of Algorithms **10**

Abbreviations **11**

1 Introduction **12**
1.1 Structure of the thesis 14

2 State of the art **15**
2.1 Introduction 15
2.2 Background information 15
2.2.1 Image sensors 16
2.2.2 Color filter arrays 17
2.2.3 Demosaicing 18
2.2.4 Noise 19
2.3 JPEG compression 20
2.4 Steganographic methods and respective attacks 23
2.4.1 Least Significant Bit embedding 24
2.4.2 ± 1 embedding 27
2.4.3 F5 algorithm and matrix embedding 30
2.4.4 Steganography by cover-source swtiching 33
2.5 Conclusion 35

3 Theoretical foundations of image steganography **36**
3.1 Channel and security 36
3.1.1 Principle 37
3.1.2 Notations 37
3.1.3 Steganographic channel 38
3.1.4 Cachin's definition of steganographic security 39
3.1.5 Distortion limited embedding and capacity 41
3.2 Steganalysis 44
3.2.1 Attacks 44
3.2.2 Detection problem 45

3.2.3 ROC curve . . . 47
3.2.4 Ensemble classifiers . . . 49
3.3 Conclusion . . . 49

4 Perturbed quantization steganography with wet paper codes 50
4.1 Introduction . . . 50
4.2 Wet paper codes . . . 50
4.3 Double compression perturbed quantization . . . 56
4.3.1 Perturbed quantization . . . 56
4.3.2 Double compression . . . 57
4.3.3 Selection rule . . . 58
4.3.4 JPEG quality factors . . . 60

5 Proposed method 61
5.1 Selection rule for more robust embedding . . . 61
5.2 Processing of messages . . . 64
5.3 Requirements and practical limitations . . . 65

6 Steganalysis 68
6.1 Data acquisition . . . 68
6.1.1 Image dataset . . . 68
6.1.2 Plaintext and ciphertext datasets . . . 69
6.2 Key considerations . . . 69
6.3 CC-PEV features . . . 71
6.4 Embedding rate . . . 72
6.5 Detection results . . . 73
6.6 Results analysis . . . 80
6.7 Conclusions . . . 81

7 Conclusion 82

Bibliography 83

Acknowledgements

I would like to express my gratitude to my project supervisor Doctor Eduardo Solana who has been very supportive towards me and who worked actively to provide me the chance to pursue my goal. I am also grateful to all of those with whom discussions have contributed to the project.

List of Figures

2.1 Charge-coupled-device architecture. 16
2.2 Bayer Color Filter Array 17
2.3 Image acquisition pipeline. 19
2.4 JPEG compression pipeline. 20
2.5 Two-dimensional DCT basis functions. 21
2.6 Zigzag ordering of coefficients. 23

3.1 Steganographic channel with cover modification. 38
3.2 Distortion limited embedder with noisy channel. 41
3.3 Detection problem with a passive warden. 46
3.4 Comparison of two detectors with their ROC curves. 48

4.1 Probability for a matrix $k - r \times k$ of having a rank $k - r$. 52
4.2 Histogram of values of DCT coefficients from [20]. 58
4.3 Stego images and an overview of their modified elements. 59
4.4 Embedding capacity according to Q_1 and Q_2. 60

5.1 Selection rule with $f(x_i)$. 62
5.2 Pair of cover and stego objects after the embedding of a message of 1000 characters. 63
5.3 Using the XOR with m_c and m_p. 65
5.4 Global view of the embedding system. 66

6.1 Image and message preprocessing before embedding. 69
6.2 Calibration process. 71
6.3 Embedding different message lengths. 72
6.4 ROC curves of images embedded with XORed messages and random bitstreams with embedding rate of 0.05 bpc. 73

6.5 ROC curves of images embedded with XORed messages and random bitstreams with embedding rate of 0.1 bpc. 74
6.6 ROC curves of images embedded with the proposed selection rule and with the original selection rule. 75
6.7 ROC curves of images embedded with XORed messages using $Q_1 = 85$ and $Q_2 = 70$ with the proposed selection rule. 76
6.8 ROC curves of images embedded using different then identical keys. . 77
6.9 ROC curves of images embedded using different then identical keys. . 78
6.10 ROC curves for 0.05 bpc. 79

List of Tables

5.1 Comparison of embedding time when using the classical selection rule and the proposed one. 63

6.1 Detection reliability ρ for XORed messages and random bitstreams. . 74
6.2 Detection reliability ρ for original and proposed selection rules. . . . 75
6.3 Detection reliability ρ for PQ with the proposed selection and using XORed messages. 76
6.4 Detection reliability ρ when changing the key parameter with embedding rates of 0.05 and 0.1 bpc. 77
6.5 Detection reliability ρ for 0.05 bpc. 79

List of Algorithms

1 WPC Encoding algorithm [16] 54
2 WPC Decoding algorithm [16] 55
3 WPC Encoding with additional $f(x_i)$ 67

Abbreviations

AES	**A**dvanced **E**ncryption **S**tandard
ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
AUC	**A**rea **U**nder **C**urve
CCD	**C**harge **C**oupled **D**evice
CFA	**C**olor **F**ilter **A**rray
CMOS	**C**omplementary **M**etal-**O**xide-**S**emiconductor
DCT	**D**iscrete **C**osine **T**ransform
IDCT	**I**nverse **D**iscrete **C**osine **T**ransform
LSB	**L**east **S**ignificant **B**it
PQ	**P**erturbed **Q**uantization
QT	**Q**uantization **T**able
ROC	**R**eceiver **O**perating **C**haracteristic
SVM	**S**upport **V**ector **M**achine
WPC	**W**et **P**aper **C**odes

Chapter 1

Introduction

Information hiding can be traced back to 440 B.C., when according to Herodotus in his *Histories*, the tyrant Histiæus tattooed the head of one of his slaves after shaving it and sent him to his vassal after the hair had regrown [35]. Steganography, composed by the Greek words *steganos*, meaning "covered" and *graphein*, meaning "writing", is the art of concealing messages into innocuous-looking media.

In the present day, communication using digital media is omnipresent in the daily life of billions of users that have seen their access to computers and smartphones increase in the past decades. Along with digital communication come the issues of privacy and security of the exchanged data. When communicating using a public channel, two users expose the content of their messages. Even with private messaging applications or emails providers, the communications can be monitored and/or collected by third parties at the expenses of the sender. In the presence of adversaries, the practice of cryptography has allowed secure communications by the construction of protocols that prevent third parties from reading the encrypted information [40]. However, the presence of encrypted data is dubious since the messages contain nonsensical text that cannot be read. The art of steganography differs from cryptography by concealing the very existence of secret communication. Messages are hidden into innocent looking supports which, unlike encrypted data, do not look suspect.

The field of steganography gathers multiple classes of techniques and their respective algorithms. Inevitably, these algorithms are broken, either by attackers or by researchers who want to improve the security of their steganographic systems. A steganographic technique has the fundamental objective of being undetectable while conveying a non-negligible amount of information that should be correctly extracted even after attacks.

A steganographic scheme is composed of two elements: the embedding and extraction algorithms. When two entities want to communicate by using steganography, they need to have compatible embedding and extraction schemes. They can also decide to share a secret key on which they agreed before starting to communicate. The embedding algorithm takes the message to be sent, the eventual key, and a *cover object* in order to create what is referred as a *stego object*. The cover object

is a media that acts as a carrier for the message. The secret information is concealed in the cover which results in the creation of the stego object. Then, the extraction algorithm uses the stego object jointly with the key to obtain the hidden message.

The media file used as a cover object is modified according to the encoding technique and several types of data files can be used as cover objects such as text files, video files, or audio files [11, 15, 25]. However, it is the field of **image** steganography that has been the most studied [15] and a great majority of current steganographic applications work with digital images. It is crucial to know that in opposition to cryptography where a communication is compromised if data is decrypted or guessed by an adversary, in steganography, if the mere existence of hidden information is revealed, the steganographic scheme that is used is a failure and cannot be considered secure.

The users of steganography do not only want to protect the content of their messages, as cryptography provides, but have also a need to conceal the existence of communications. Military communications can use these techniques to hide sensitive information but also indications about the sender and the recipient [35]. In the same fashion, criminals tend to use all available technologies that allow them to operate anonymously, and steganographic systems that cannot be detected are perfect for their illegal exchanges. On the other hand, in some countries, some oppressed populations are being constantly monitored and restricted in their communications and access to social networking services. Cryptography regulation around the world is constantly changing, however, these countries often prohibit the use of encryption algorithms [1], justifying the use of steganography. There is a crucial duality between the good usage of steganographic technologies and the immoral practice by criminal organizations. Nonetheless, most of technologies can be deviated from their original purpose for malicious intentions and this idea must not restrain the progress in the development of these technologies.

Similarly to what is practiced in cryptography, a steganographic scheme is not secret. From its working principle to its algorithms, all parties are able to implement and use the technique. More importantly, they are also able to search for weaknesses in order to develop targeted attacks. It is this constant threat that motivates the improvement of existing techniques; the perpetual game between attackers and new systems has driven the evolution of current methods.

The Wet Paper Codes steganographic method is distinct from previous techniques by its use of an arbitrary selection channel [16]. This channel is dependent on the cover image and is not determined by the sender. Besides, it is never communicated to the recipient, which is a major security improvement considering that if the recipient ignores the location of the message bits, a potential attacker will also be clueless. The location of the elements that will be modified is determined with the help of an information-reducing operation, where the elements with the most uncertain values are chosen. In combination with Perturbed Quantization, the Wet Paper Codes technique was shown to be considerably less detectable than other techniques [19].

Selecting the elements to change during the embedding can be adapted to suit different conditions. As stated in the original work of Fridrich et al. in [19], only the most robust elements could be kept for embedding. The work of this thesis proposes to adjust the selection rule in order to see the impact on the security by picking robust elements. With this further selection, an increase in the embedding speed will be observed. The second contribution concerns the messages. When unprocessed messages are usually embedded, this work proposes to use an additional non-secret message that is publicly available to convey a message, while the real one is a XORed sequence hidden in the image. This is motivated by the massive use in current society of image sharing on social networking services. The system lets the users preprocess their messages to share an image with a description while the real description is in the image. The real message can then be read by all recipients in possession of the secret key.

The goal of this study is to develop the proposed steganographic system and to determine its performance.

1.1 Structure of the thesis

A total of seven chapters constitute this thesis, this introduction being the first one.

The second chapter provides the reader with some background information about image sensors and JPEG compression in order to understand the importance of noise and distortion in steganographic schemes. These explanations are followed by an overview of the most known steganographic techniques, along with some of the targeted attacks that exist currently.

Fundamental theory for steganographic systems is explained in Chapter 3 where the principles of steganographic channel and steganographic security are analyzed. Then, theory concerning steganalysis and the detection problem encountered by steganalysts is explained. The ROC curves used to interpret the final results are also described along with an introduction to Ensemble Classifiers used to run the experiments.

Chapter 4 is constituted of the core theory of the steganographic method used in this work. It covers the Wet Paper Codes theory with its embedding and decoding algorithms and the concept of Perturbed Quantization with the special case of JPEG double compression.

Next, Chapter 5 describes the proposed method that uses the theory of previous chapter with additional message preprocessing and selection rule. The system is then tested on a dataset of 1000 images with a series of experiments in Chapter 6 where all results and ROC curves are reported along with some observations. An analysis of these observations is done in the last section, explaining the performance of the proposed system.

The final chapter 7 concludes the thesis with a discussion on the proposed method and some further works.

Chapter 2

State of the art

2.1 Introduction

The field of image steganography is vast and the methods are constantly evolving. From the simplest algorithm to the most sophisticated one, a lot of progress has been made in the field. It is possible to separate all methods into two domains: spacial and frequency domain techniques [15]. Methods that work in the spatial domain will change directly the values of the cover image pixels while frequency domain methods are used after projecting data into the frequency domain. Modifications are done in the given domain before transforming the data back to the spatial domain.

New steganographic systems are developed while existing algorithms are improved to overcome attacks that target some of their flaws. Teams of steganalysts use all the knowledge acquired with previous schemes in order to create systems that cannot be detected with current steganalysis techniques, while being able to contain a good amount of message bits.

First, this chapter will provide some background information on the acquisition of digital images. This process involves multiple components and introduces a crucial element for steganography which is the noise. Then, the JPEG compression is detailed in order to understand the Perturbed Quantization technique used in this work. Finally, an overview of most known steganographic techniques is proposed, along with their flaws and some of their targeted attacks.

2.2 Background information

There are two approaches to produce a digital image: there are computer-generated images and images acquired with scanning devices or cameras. Hiding information in an image that was synthesized with a computer is a hard and limited task. These images hold in each pixel a determined amount of color, making them noiseless if no external perturbation has been applied. Whether a simple computer graphics application or a more complex 3D computer graphics software is used, the

modification of a pixel will disrupt a plain color or a surface that has been generated using deterministic algorithms. This is the reason why steganography is mostly used with images acquired with dedicated devices: noisy and imperfect images are more interesting because of their ability to tolerate modifications [15, 26].

2.2.1 Image sensors

The acquisition of digital images relies on devices that need to have a core element which is the sensor. Two sensor technologies are widely used nowadays: the charge-coupled-device (CCD) and the complementary metal-oxide-semiconductor (CMOS) sensor. Both technologies rely on the use of a semiconductor to absorb photons in order to produce electrons. In present-day sensors, almost all semiconductors encountered are in silicon [44]. The reason is that the bandgap of silicon is ideal for capturing light visible by the human eye [44]. After the production of electrons, the charge is collected, transferred, and eventually converted into a voltage. Transfer and conversion steps are different in CCDs and CMOS sensors.

A sensor (CCD or CMOS) is a 2D rectangular array of photosites [8, 15, 32]. Each photosite contains a photodetector (a capacitor) that captures light during exposure time and converts accumulated photons into electrons. A photon that physically reaches the silicon will be absorbed more or less deeply in the silicon wafer based on its wavelength [32, 44].

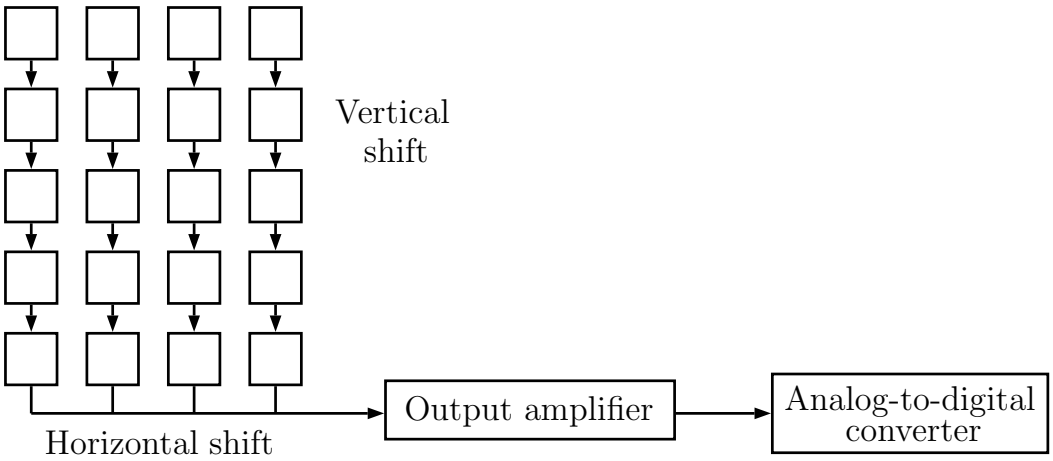


Figure 2.1: Charge-coupled-device architecture.

Invented in 1969, the charge-coupled device is particular in its charge transfer step: each capacitor transfers the charge that it contains to the neighboring capacitor. In a 2D array, this process operates with row transfers [15, 32] where charges are shifted from one row to another, from top to bottom (Figure 2.1). Charges of the last row are transferred into an amplifier that converts them into voltage and amplify them by a gain K. Invented at the same period, CMOS sensors are particular by the existence of an active amplifier at each photosite of the array [15, 32]. Charges

are read directly at each photosite and there is no need for a sequential row shift. CMOS sensors consume less power than CCDs, are less expensive, and allow a direct access to a given location, which are the reasons why they equip most of cameras on the market [9, 15].

Each step of a digital image acquisition adds noise to the final output and both conversion and amplification steps introduce additional noise to the signal. Given that CMOS sensors have an amplifier at each photosite, the noise generated by these electronic components is more important than the one in a CCD, which is the reason why CCDs are preferred in scientific applications such as in astronomy [15].

2.2.2 Color filter arrays

When exposed to light, a photodetector will capture all photons in the spectrum range of the semiconductor material that is used. This means that all colors will be collected, resulting in a grayscale digital image [15]. To avoid this result, and in order to acquire color images as they are perceived by the human eye, the use of color filters is necessary. A filter is added to each photosite detector in order to absorb undesired wavelengths and let a given color pass through to the semiconductor.

To capture all colors at all positions (all photosites), the use of a specific camera that splits the light beam and sends it to three separate sensors is necessary [2, 15]. Each sensor will acquire one color channel which will give three colors for each position of the array. However, calibration in order to align perfectly the three light beams and sensors is a difficult task, and having three times more components makes the device less affordable [31]. This is the reason why in order to reduce costs, size, and errors, nearly all digital cameras use color filter arrays.

A color filter array (CFA) is a particular arrangement in 2D of RGB filters on all the photosites of the sensor [15, 31]. This principle can be found in the retina of the human eye where cone cells, which are responsible for color vision, exist in three types [2]. The stimulation of a cone of a given type allows us to perceive a specific wavelength. CFAs and human eyes share the property of capturing one color per spatial location. When acquiring an image of a real scenery, light is captured at discrete positions: this is a sampling of the real word and it will produce aliasing if the frequency is too low [2]. Multiple color filter arrays exist, but the most popular one is the Bayer CFA [2, 15, 31], called after the name of its inventor (Figure 2.2).

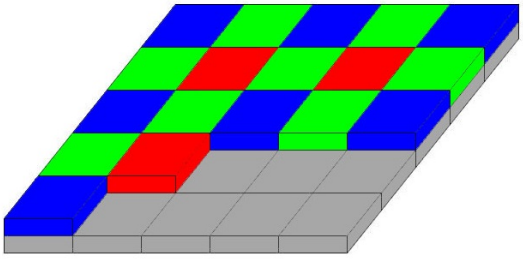


Figure 2.2: Bayer CFA on a sensor represented in gray.

The Bayer filter is particular by its disposition of RGB filters: there are twice as many green pixels as red or blue pixels. This higher sampling rate of the green channel is related to properties of the human visual system, which is the most sensitive to wavelengths corresponding to the green color [2, 15].

Only one color channel is acquired per location during exposure time. In order to obtain each color at each discrete location of the image, several interpolating methods exist. This process is called *demosaicing* (or demosaicking) because the CFA pattern is like a mosaic that will be broken in order to obtain three different layers, one for each color. In fact, the CFA can be seen as a down-sampling from a full resolution image $X = (R, G, B)$ [31] containing all pixels for each color channel to a mosaiced image Y where

$$Y = \sum_{X=R,G,B} M \cdot X \tag{2.1}$$

with M a mask that allows to extract the desired CFA pattern.

2.2.3 Demosaicing

The demosaicing is either a linear or non linear process [5] that aims to fill missing colors of each channel by using the values measured on photosites. These values are not modified by the demosaicing process.

Most of demosaicing algorithms are developed for the Bayer CFA but can be extended to other patterns [31]. There are two interpolation problems in the demosaicing of the Bayer pattern. First one is to perform the reconstruction of the missing half of green pixels. The second one is to do the same with the missing three-quarters of red and blue pixels. It is possible to perform interpolation independently for each channel, however, demosaicing with respect to inter and intra dependencies of channels is a more complicated and rewarding task by the fact that it will minimize aliasing and artifacts resulting from reconstruction [2, 31].

Representing and analyzing the demosaicing problem in the frequency domain is an alternative that will benefit from multiple techniques that already exist [2]. However, regardless of the representation and the demosaicing method, the process of interpolation can be seen as a filtering that will inevitably introduce dependencies between pixels. These dependencies will be disrupted by modifications introduced during the hiding of data in an image, which will compromise the hidden information [2, 15].

The demosaicing step is followed by several processing stages that are accomplished by the camera before obtaining the final image. After interpolation, a color transformation is done in order to map colors into a given color space (RGB, sRGB, etc.). Then, colors are rectified using the gamma correction, and other procedures such as denoising, contrast adjustment, cropping, lens-distortion compensation, etc, can be applied depending on the manufacturer of the camera. Finally, the image is saved in either a lossless format or a lossy compressed format (Figure 2.3).

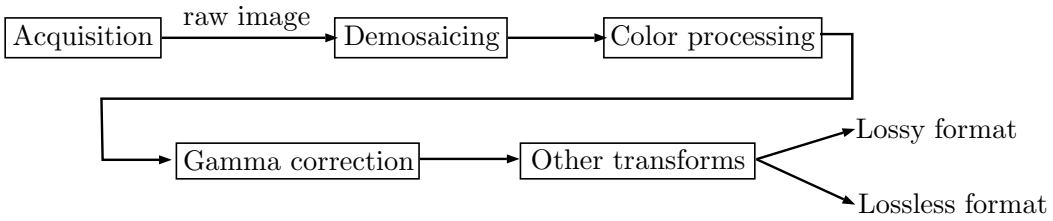


Figure 2.3: Image acquisition pipeline.

All these transformations imply that data can be hidden before or after the processing of the image. Some of the processing steps described above are not reversible [15]. Therefore, it is a requirement to understand in detail how every step acts in the final image in order to know where information cannot be hidden in case of irreversible processing.

2.2.4 Noise

Ideally, when using image sensors, it is expected that each photon that arrives at the sensor will produce one electron [38, 15]. Because of imperfections in the silicon wafer, the efficiency in today's sensors, although being really high, can never reach 100%. There are multiple other elements that cause the signal not to be the perfect image that was expected during the acquisition phase. This additional undesired signal that is merged with the expected signal is described as *noise*. Various factors contribute differently to the build up of noise on the original signal. We distinguish several types of noise [38, 8, 15, 44].

Dark current noise is caused by irregularities in the silicon lattice and by thermally generated electrons that accumulate in the photosites. These electrons are present whether the sensor is exposed to light or not. Dark current can be reduced by cooling the sensor and clearing the pixels before exposure. It is also possible to subtract the dark current by using a dark frame: a picture of a dark frame is taken (or with the cap on the sensor) and the image is subtracted to remove noise from following images.

Photon or shot noise is due to the quantum nature of light. All photons do not arrive at the same time at the sensor and each arrival is an independent event. The consequence is that, during a fixed exposure time, some photosites will receive more photons than others. This random arrival during exposition to light is described by a Poisson distribution.

Pixel non-uniformity is present because not all photosites of a given sensor have the same sensitivity to light. This is due to imperfections in the manufacturing of the silicon wafer that cause a systematic artifact on a specific sensor. This artifact is not random, thus it is possible to use it as a fingerprint for a camera [14]. The pixel

non-uniformity does not allow to embed more information because of its non-random nature.

Cameras integrate several electronic components in a small space. All this circuitry it not always compatible and some units can cause additional noise for others [38]. After the reading of electrons on photosites, the conversion to voltage and the amplification phase will produce the **readout noise** and the **amplifier noise**, respectively. Several phases of processing take place during the acquisition of an image. Therefore, there are multiple moments during which additional noise is introduced on the signal.

2.3 JPEG compression

JPEG is a lossy compression format designed to represent images in a compact and efficient manner by exploiting the characteristics of the human visual system. The acronym stands for Joint Photographic Experts Group, which is a comity of experts who developed during several years a standard for still images [41]. The final version of the standard was specified in 1991 and approved in 1992. The compression is based on the Discrete Cosine Transform (DCT) and includes several steps.

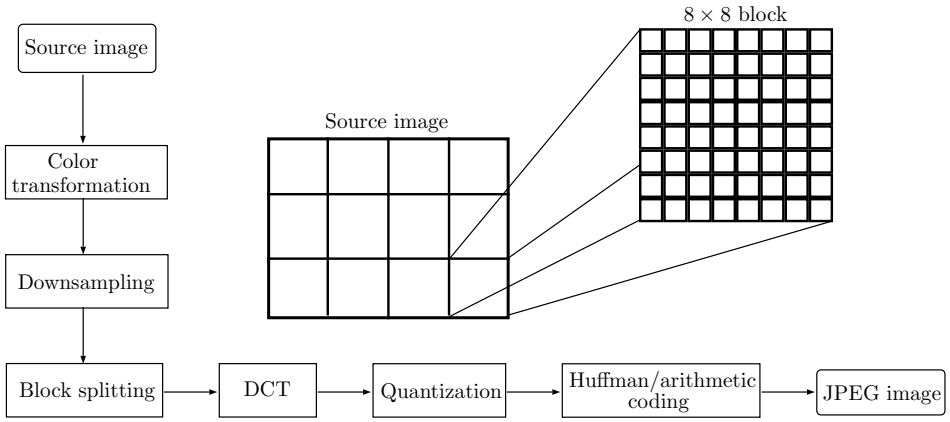


Figure 2.4: JPEG compression pipeline.

Color transformation: The JPEG algorithm is designed to be able to perform on images described by any color model. However, the process is widely used with the $Y'C_bC_r$ color space for the main purpose of embedding efficiency. The $Y'C_bC_r$ model is composed of the luma component Y' that represents the brightness of a pixel and the C_b and C_r chrominance components, respectively depicting the blue and red differences in color.
Compression using the $Y'C_bC_r$ color space shows better results. This is a consequence of the human eye that holds the characteristic of being more sensitive to luminance than to chrominance. In fact, differences in color tones are less noticeable than changes in the brightness of a color. Thus, images are transformed in order to focus on the highly compressible elements C_b and C_r.

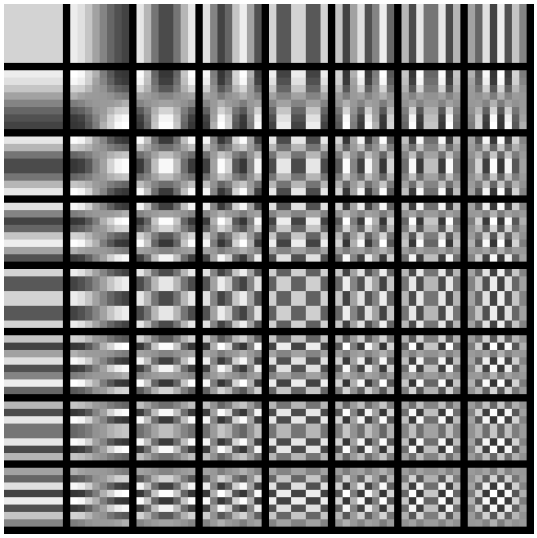


Figure 2.5: Two-dimensional DCT basis functions.

Downsampling: The second step takes also advantage of this lower sensitivity to hue and color saturation by downsampling the chrominance components. Decimation is only applied to C_b and C_r, and Y' remains the same.

Block splitting: The JPEG algorithm operates on an image by transforming it gradually by parts. Thus, the image is divided into blocks of usually 8×8 elements. Each color component is split up independently.

DCT transform: The compression is based on the discrete cosine transform (DCT) which changes a block from the spacial domain to the frequency domain. This transformation uses only real numbers to represent a signal with a sum of sinusoids, each one having its own frequency and amplitude. Each 8×8 block is transformed into a linear combination of cosine functions by using the two-dimensional DCT basis functions shown in Figure 2.5. Before this step, the block values are shifted from the positive range to one centered in zero in order to reduce the dynamic range for the DCT process. Thereby, for an RGB image with pixel values in $[0, 255]$, the midpoint is 128 and all values are mapped to the set $[-127, 128]$ by subtracting 127.

Let $\boldsymbol{B}$ be a 8×8 block ($N = 8$) of Y',C_r, or C_b elements and $B_{u,v}$ the value at the u-th row and v-th column of the block with $\{0 \leq u < 8,\ 0 \leq v < 8\}$. The DCT coefficients are given by the two-dimensional DCT [41] :

$$b_{i,j} = \frac{\alpha(i)\alpha(j)}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} B_{u,v} \cdot \cos\left[\frac{(2u+1)i\pi}{16}\right] \cos\left[\frac{(2v+1)j\pi}{16}\right] \tag{2.2}$$

Where

$$\alpha(i) = \begin{cases} 1/\sqrt{2}, & \text{if } i = 0 \\ 1, & \text{otherwise} \end{cases}$$

Each 8×8 block is a 64-point discrete signal that is decomposed into a 64 orthogonal basis signal. The coefficient with zero frequency $b_{0,0}$ is called *DC coefficient* while the remaining coefficients are called *AC coefficients*.
In theory, the DCT is a lossless transformation that can be reversed by applying its inverse (IDCT), which should recover the original values. However, because of the numerical precision and rounding errors, it may not be the case in practice.

Quantization: This step is crucial in the process of JPEG compression due to its non-reversible nature. Given that high frequencies brightness changes are not strongly perceptible by the human eye, the amount of information carried by these elements can be greatly reduced. This is achieved by the the use of a Quantization Table (QT) that removes non-significant visual information. The most common quantization matrix is given by:

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{2.3}$$

In a 8×8 block, each DCT coefficient is divided by the corresponding quantizer step size from the QT and then rounded to the nearest integer.
Let $\boldsymbol{Q} = \{q_{0,0}, q_{0,1}, \ldots, q_{7,7}\}$ be the quantization table of size 8×8, the quantized values of a block $\boldsymbol{b}$ are given by [15, 41]:

$$b^Q_{i,j} = round\left(\frac{b_{i,j}}{q_{i,j}}\right), \quad 0 \leq i, j < 8 \tag{2.4}$$

Redundant information is discarded therefore fewer bits are allocated to represent DCT coefficients. Reversing this step is impossible because of its lossy nature.

Coding and zig-zag sequence: The elements of each DCT block are reordered such that the *DC* component is on the upper left corner and the remaining coefficients are arranged in a zig-zag sequence by placing low-frequency coefficients before high-frequency coefficients. This ordering simplifies the entropy coding. JPEG proposes two lossless entropy coding methods: Huffman coding and arithmetic coding [41].

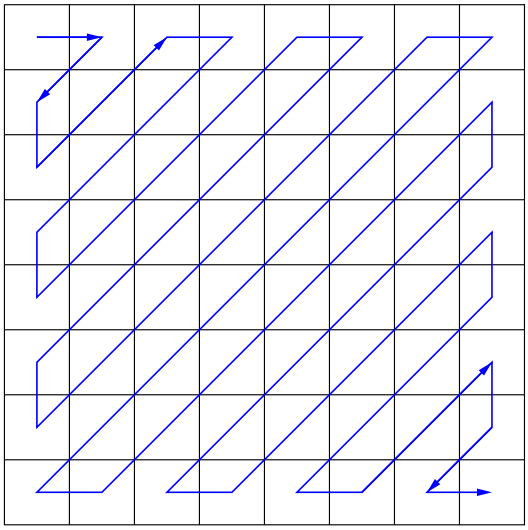


Figure 2.6: Zigzag ordering of coefficients.

2.4 Steganographic methods and respective attacks

A great number of steganographic systems have been developed using knowledge from previous schemes and the solid theoretical foundations developed in the past decades. In order to respect as much as possible the principal conditions of non-detectability, robustness, and capacity, many practical methods are created and modified through trial and error with the use of steganalysis.

The steganalysts expect for each steganographic method to undergo a steganalysis and multiple attacks. Some attack techniques are targeted against one class of embedding methods where information about the algorithm is known by the steganalyst, while other techniques have no knowledge of the embedding method and try to cover as many steganographic techniques as possible during detection.

The following section attempts to give an overview of most known steganographic techniques. For each major technique, targeted attacks are presented in order to understand the evolution and the improvement of steganographic systems.

2.4.1 Least Significant Bit embedding

The Least Significant Bit (LSB) embedding is an elementary steganographic method that can be easily detected using steganalysis. It is considered as the simplest algorithm [15] and is not secure at all. The method works in the spatial domain and can be used with digital images as well as with videos, audio files, or other digital media. Its main advantage is that, when used with images, the human eye cannot detect the small perturbations that are introduced (perceptual invisibility).

Each pixel value of a digital image is represented using a certain number n_b of bits that can vary according to the image format and/or the precision. For instance, most grayscale images use one byte ($n_b = 8$ bits) to express each pixel. As suggested by its name, the LSB embedding hides information by replacing least significant bits from the cover image with message bits in order to generate the stego image.

Let $\boldsymbol{x}$ be the cover image of size $n \times m$, $\boldsymbol{m} = m_1, m_2 \dots m_l \in \{0,1\}^l$ the message bit stream of size l, and x_{ij} a pixel of the cover image with $\{0 \leq i < n, 0 \leq j < m\}$. Each x_{ij} is constituted of n_b bits $x_{ij}[1] \dots x_{ij}[n_b]$ and its value is given by:

$$x_{ij} = \sum_{k=0}^{n_b-1} x_{ij}[k] \cdot 2^{n_b-k-1} \tag{2.5}$$

The LSB embedding algorithm aims to replace the bit $x_{ij}[n_b]$ with a message bit. The hiding of the u-th bit of message in the x_{ij} pixel in order to get the stego pixel y_{ij} is given by [10, 15]

$$\begin{aligned} LSBflip(x_{ij}) &= x_{ij} - x_{ij} \mod 2 + m_u = y_{ij} \\ &= \begin{cases} x_{ij} + 1 & \text{if } x_{ij} \mod 2 = 0 \text{ (even)} \\ x_{ij} - 1 & \text{if } x_{ij} \mod 2 = 1 \text{ (odd)} \end{cases} \end{aligned} \tag{2.6}$$

It is possible to alter all pixels x_{ij} of the cover image or only some of them by the use of a secret key shared between the sender and the recipient that indicates the positions of pixels to flip. With a message length of l and image dimensions $n \times m$, the relative payload is given by $\alpha = \frac{l}{n \times m}$ [15].
The extraction step consists in reading least significant bits ($y_{ij} \mod 2$) at the positions given by the secret key.

Histogram attack: the chi-square test

Flipping a least significant bit is an idempotent operation [15], which means that $LSBflip(LSBflip(x)) = x$. When an even value has its LSB flipped it will result in an increment and for an odd value it will result in a decrement of the value. Westfeld and Pfitzmann [43] proposed an approach where these values are grouped into disjoint *pairs of values* (PoVs) $\{0,1\}, \{2,3\}, \dots, \{2^{n_b} - 2, 2^{n_b} - 1\}$. Flipping a value will not change the pair to which it is associated, which is a considerable

information for steganalysis. Let $\boldsymbol{h}[x],\ x = 0, \ldots, 2^{n_b} - 1$ be the histogram of the value x which counts how many times this value is encountered in the image.

$$\boldsymbol{h}[x] = \sum_{i=1}^{n} \sum_{j=1}^{m} \delta(x_{ij} - x) \quad ; \quad \delta(x) = \begin{cases} 1 & \text{if} \quad x = 0 \\ 0 & \text{if} \quad x \neq 0 \end{cases} \tag{2.7}$$

where δ is the Kronecker function. It is observed that within one PoV $\{2k, 2k+1\},\ k = 0, \ldots, 2^{n_b-1}$, the sum $\boldsymbol{h}[2k] + \boldsymbol{h}[2k+1]$ does not change after embedding. However, after LSB embedding, a balancing of the histogram bins of each pair will be perceived. Even if the histogram sum of a pair stays unchanged, the values $\boldsymbol{h}[2k]$ and $\boldsymbol{h}[2k+1]$ will have a tendency to become close to each other which leads to a very characteristic artifact in the histogram that takes the shape of a staircase [15, 43].

This observation, generally untrue for cover images, will allow the construction of a histogram attack with a composite hypothesis testing problem using the chi-square test (χ^2-test). The chi-square test is a popular test used for composite hypothesis testing that determines if the observed sample distribution follows the theoretically expected distribution. For a given PoV $\{2k, 2k+1\}$, the observed distribution $\{o_k\}$ [30] is given by:

$$o_k = h[2k] \tag{2.8}$$

And under assumption that the sender fully embeds the cover image, the expected distribution [15, 30] is given by:

$$e_k = E[\boldsymbol{h}[2k]] = \frac{\boldsymbol{h}[2k] + \boldsymbol{h}[2k+1]}{2} \tag{2.9}$$

Using the observed and expected distributions, the chi-square computes the statistics S.

$$S = \sum_{e_k \neq 0} \frac{(o_k - e_k)^2}{e_k} = \frac{1}{2} \sum_{k=1}^{2^{(n_b-1)}-1} \frac{(\boldsymbol{h}[2k] - \boldsymbol{h}[2k+1])^2}{\boldsymbol{h}[2k] + \boldsymbol{h}[2k+1]} \tag{2.10}$$

If the observed values follow the expected distribution, the statistics will approximately follow the chi-square distribution $S \sim \chi^2$ and S will be small. Consequently, it is possible to build a detector that uses the value of S and a threshold γ can be set on this value in order to decide whether a stego image is encountered or not. However, the attack is only efficient with sequential embedding (all pixels are used) and is not able to detect a stego image if less than 99% pixels are used for embedding [15, 30].

RS steganalysis

Fridrich et al. [18] propose a reliable method to detect LSB embedding in both 24-bit color images and 8-bit grayscale or color images. The algorithm is able to work with randomly scattered message bits (non-sequential embedding) by the use of a discrimination function. The image is first divided into disjoint groups $G = (x_1, \ldots, x_n)$

of pixels to which the discrimination function, e.g. the variation computation, is applied.

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \tag{2.11}$$

Additionally, an invertible function F called *flipping* is defined and $F(F(x)) = x$, $\forall x \in P$, where P is all possible pixel values, e.g. for a 8-bit grayscale image $P = \{0, \ldots, 255\}$.

$$F_0 = identity, \forall x \in P \tag{2.12}$$

$$F_1 : 0 \leftrightarrow 1, 2 \leftrightarrow 3, \ldots, 254 \leftrightarrow 255, \forall x \in P \tag{2.13}$$

$$F_{-1} : -1 \leftrightarrow 0, 1 \leftrightarrow 2, \ldots, 255 \leftrightarrow 256, \forall x \tag{2.14}$$

These operations are applied on groups of pixels selected according to a mask M and three distinct groups are defined:

$$\begin{aligned} &\text{Regular groups:} \quad G \in R \Leftrightarrow f(F(G)) > f(G) \\ &\text{Singular groups:} \quad G \in S \Leftrightarrow f(F(G)) < f(G) \\ &\text{Unusable groups:} \quad G \in U \Leftrightarrow f(F(G)) = f(G) \end{aligned}$$

A stego-detector is built by analyzing the number of regular and singular groups and their evolution when the message length is increased. The number of regular groups for a mask M is denoted by R_M and its equivalent for singular groups is denoted by S_M. R_{-M} and S_{-M} are the number of groups for the negative mask. In a typical image, it is observed that $R_M \cong R_{-M}$ and $S_M \cong S_{-M}$. After randomizing the LSB plane, typically what the LSB embedding does, the difference between R_M and S_M decreases. With an embedding using all pixels, 50% of pixels are flipped and $R_M \cong S_M$.

This steganalysis method surpasses the chi-square test, which definitely makes the LSB embedding unusable because of its detectability.

2.4.2 ±1 embedding

The ±1 embedding is a technique that has been developed after observing the poor resistance of the LSB embedding to steganalysis. The operation of bit flipping, as seen before, introduces characteristic artifacts in the histogram because of its non-symmetrical process. The ±1 embedding, also called LSB matching [15], is a modification of the LSB embedding. Instead of flipping an LSB that does not match the message bit, a random addition or subtraction by one is applied to the pixel value. Thus, it is not possible to define pairs of values anymore, as well as disjoint triplets $\{x-1, x, x+1\}$. The special cases of the minimum and maximum pixel values (0 and 255) will only be increased or decreased, respectively. Without the symmetrical characteristic of its predecessor, LSB matching is much harder to detect and the performance of different detectors is not constant over different cover sources [37].

HCF COM steganalysis

The modification of pixels without consideration toward the cover content and its statistical properties is considered as a simple addition of noise, referred as *stegonoise*. Harmsen et al. [23] propose a blind detection scheme that exploits the fact that altered images are subjected to a decrease of the histogram characteristic function center of mass. When a steganographic technique modifies a cover image by the addition of independent noise, the resulting stego image can be represented as a sum of two independent random variables. More precisely, the histogram of the stego image is a convolution between the histogram of the cover image and the noise probability mass function (PMF) because the sum of two independent random variables is a convolution of their PMFs.

Let f be the stego noise PMF. The histogram $\boldsymbol{h}_s$ of the stego image is given by

$$\boldsymbol{h}_s[n] = \boldsymbol{h}_c[n] * f[n] \tag{2.15}$$

which corresponds to a low-pass version of the cover image histogram. Working in the frequency domain is more convenient given the fact that a convolution in spatial domain is translated by a simple multiplication in the frequency domain.

$$\boldsymbol{H}_s[k] = F[k] \cdot \boldsymbol{H}_c[k] \tag{2.16}$$

where the transformation to frequency domain is done with the discrete Fourier transform (DFT):

$$X[k] = DFT(x[n]) = \sum_{n=0}^{N-1} x[n] e^{-\frac{2\pi i n k}{N}} \tag{2.17}$$

The DFT of a histogram is referred as the *histogram characteristic function* (HCF). Next, the HCF center of mass (COM) is computed with only half of the values because of the symmetry of the DFT's absolute value. This metric provides information

about the energy distribution in the HCF.

$$COM(\boldsymbol{H}[k]) = \frac{\sum_{k=0}^{\frac{N}{2}-1} k|\boldsymbol{H}[k]|}{\sum_{k=0}^{\frac{N}{2}-1} |\boldsymbol{H}[k]|} \tag{2.18}$$

It is then shown with the use of the Chebyshev inequality that this center of mass will either decrease or stay unchanged after embedding.

$$COM(\boldsymbol{H}_s[k]) \leq COM(\boldsymbol{H}_c[k]) \tag{2.19}$$

This metric is used as a feature to build a simple steganalyzer that will separate cover from stego images. According to (2.19), the COM measure is expected to be lower in stego images. With the knowledge of the embedding algorithm, a classifier is trained with both cover and stego images which is an application of non-blind steganalysis. The training will result in a threshold that separates the data into two sets. This threshold will then be used to identify new stego images created with the same embedding technique. However, from a cover source to another, the performance is not the same and a threshold that allows to separate clearly the two sets cannot be always defined.

SPAM steganalysis

During image acquisition, as seen in section 2.2.4, noises that are added to the signal have a source usually modeled as a signal independent from the content. However, all following in-camera processings (demosaicing, color correction, etc.) introduce dependencies between pixels, and these characteristic dependencies are used by most steganalysts. The work of Pevny et al. [37] takes advantage of the fact that dependencies are disrupted by the embedding in order to describe a new steganalysis method. A Subtractive Pixel Adjacency Model (SPAM) is developed in order to compute features for steganalysis. Eight directions are considered $\{\leftarrow, \rightarrow, \uparrow, \downarrow, \nwarrow, \searrow, \swarrow, \nearrow\}$.

Given an image of size $n \times m$, an array denoted by $\boldsymbol{D}$ is created by computing the differences between pixels for each direction. This will result in a model constituted of values in a small range $[-T, T]$. In the case of left-to-right direction, the array is given by:

$$\boldsymbol{D}^{\rightarrow}_{i,j} = I_{i,j} - I_{i,j+1} \tag{2.20}$$

with I_{ij} the pixel value at position (i, j) with $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$. This array of differences is equivalent to a high-pass filtered image, which implies that the image content is removed, revealing the stego noise. The model assumes that the differences $I_{i,j} - I_{i,j+1}$ are independent of $I_{i,j}$, involving that for the value $r = k - l$,

$$P(I_{i,j+1} = k, I_{i,j} = l) = P(I_{i,j+1} - I_{i,j} = r) \cdot P(I_{i,j} = l) \tag{2.21}$$

In fact, using a difference model instead of a full neighborhood model does not lead to a loss of information. Mutual information between $I_{i,j+1} - I_{i,j}$ and $I_{i,j}$ is experimentally estimated from 10'800 grayscale images to be $7.615 \cdot 10^{-2}$. It suggests that

differences and pixel values are almost independent.

SPAM features of the first order $\boldsymbol{F}^{1st}$ model the array $\boldsymbol{D}$ with the use of a first order Markov process.

$$\boldsymbol{M}^{\rightarrow}_{u,v} = P(\boldsymbol{D}^{\rightarrow}_{i,j+1} = u | \boldsymbol{D}^{\rightarrow}_{i,j} = v) \tag{2.22}$$

Because of a lack of complexity in the first order model, second order SPAM features $\boldsymbol{F}^{2nd}$ are also considered.

$$\boldsymbol{M}^{\rightarrow}_{u,v,w} = P(\boldsymbol{D}^{\rightarrow}_{i,j+2} = u | \boldsymbol{D}^{\rightarrow}_{i,j+1} = v, \boldsymbol{D}^{\rightarrow}_{i,j} = w) \tag{2.23}$$

where $u, v, w \in [-T, T]$. With eight different directions, the feature dimensionality is considerable. It can be reduced by making the assumption that in natural images, the statistics are symmetrical with respect to flipping and mirroring. Therefore, the matrices of horizontal directions are averaged, as well as the matrices of diagonal directions.

$$\boldsymbol{F}^{\cdot}_{1,\ldots,k} = \frac{\boldsymbol{M}^{\rightarrow}_{\cdot} + \boldsymbol{M}^{\leftarrow}_{\cdot} + \boldsymbol{M}^{\uparrow}_{\cdot} + \boldsymbol{M}^{\downarrow}_{\cdot}}{4} \tag{2.24}$$

$$\boldsymbol{F}^{\cdot}_{k+1,\ldots,2k} = \frac{\boldsymbol{M}^{\nwarrow}_{\cdot} + \boldsymbol{M}^{\searrow}_{\cdot} + \boldsymbol{M}^{\swarrow}_{\cdot} + \boldsymbol{M}^{\nearrow}_{\cdot}}{4} \tag{2.25}$$

where $k = (2T+1)^2$ and $k = (2T+1)^3$ for the first and second order features, respectively. These averaged Markov transition probabilities constitute the SPAM features and are used to train classifiers which can be either linear or non-linear with a preference to soft-margin SVMs. They are evaluated using the minimal average decision error: with P_f the probability of false alarm, P_m the probability of miss, and under equal probabilities of cover and stego images, the performance is given by:

$$P_{Err} = \min \frac{1}{2}(P_f + P_m) \tag{2.26}$$

Once more, it is observed that the accuracy of the steganalyzer changes depending on the cover source and that very noisy images make the detection much more difficult.

2.4.3 F5 algorithm and matrix embedding

The F5 algorithm was developed by Pfitzmann and Westfield [42] for JPEG images with the main intention of overcoming histogram attacks encountered with LSB embedding (section 2.4.1). The method is one of the first to use matrix encoding which allows to decrease the number of changes made to the cover image. It works in the frequency domain by considering LSBs of DCT coefficients and not LSBs of pixel values. Well designed models exist for the histogram of DCT coefficients and embedding by a simple LSB flipping is easily detectable. Instead of flipping LSBs, the F5 method operates by reducing the absolute value of DCT coefficients by one while skipping AC terms equal to zero and DC coefficients. Unlike its predecessor, this technique does not modify the shape of the DCT coefficients histogram.

Let $\boldsymbol{x} = x_1, x_2, \ldots$ be the cover image and $\boldsymbol{m}$ the message to embed. With matrix embedding, it is possible to embed a message of two bits m_1, m_2 in three modifiable bit locations $a_1 = LSB(x_1), a_2 = LSB(x_2), a_3 = LSB(x_3)$ with at most one modification of the cover. Four cases are possible:

$$
\begin{aligned}
&m_1 = a_1 \oplus a_3, m_2 = a_2 \oplus a_3 \Rightarrow \text{no modifications} \\
&m_1 \neq a_1 \oplus a_3, m_2 = a_2 \oplus a_3 \Rightarrow \text{change } a_1 \\
&m_1 = a_1 \oplus a_3, m_2 \neq a_2 \oplus a_3 \Rightarrow \text{change } a_2 \\
&m_1 \neq a_1 \oplus a_3, m_2 \neq a_2 \oplus a_3 \Rightarrow \text{change } a_3
\end{aligned}
$$

More precisely, the sender and the recipient need to share a binary matrix denoted by $\boldsymbol{H}$. If the message size is k, the matrix $\boldsymbol{H}$ will be composed of all non-zero binary vectors up to $2^k - 1$ and will be of size $k \times (2^k - 1)$. For instance, with a message $\boldsymbol{m}$ of size $k = 3$, the binary matrix will be [15]:

$$
\boldsymbol{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \tag{2.27}
$$

Here, $2^k - 1$ pixels are needed from the cover image and the LSBs of these pixels are denoted by the vector $\boldsymbol{a}$. The sender will modify the elements of $\boldsymbol{a}$ by decreasing the absolute value of the DCT coefficients in order to satisfy:

$$
\boldsymbol{m} = \boldsymbol{H}\boldsymbol{y} \tag{2.28}
$$

where $\boldsymbol{y}$ is the modified vector $\boldsymbol{a}$ and $\boldsymbol{H}\boldsymbol{y}$ is called the "syndrome" of $\boldsymbol{y}$. If no modification is needed, then $\boldsymbol{y} = \boldsymbol{a}$. The F5 algorithm only allows one change in the vector $\boldsymbol{a}$, such that:

$$
d_{Hamming}(\boldsymbol{a}, \boldsymbol{y}) = 1 \tag{2.29}
$$

The embedding process does not modify coefficients equal to zero and the extraction will also ignore all null coefficients. However, decreasing the absolute value of a coefficient initially equal to 1 or -1 will give zero. These message bits will be

ignored during extraction, resulting in the loss of information for the recipient. This phenomenon is called *shrinkage*.

The F5 method uses this matrix encoding scheme to embed longer messages by dividing them into segments. First, the RGB image is compressed to JPEG but the procedure is stopped after the quantization (section 2.3). Then, the embedding capacity is estimated. The F5 algorithm does not embed in DC terms and AC coefficients equal to zero. Moreover, when shrinkage occurs with coefficients that have an absolute value of 1, the message segment is re-embedded in the next group of DCT coefficients. With $\boldsymbol{h}_{DCT}$ the total number of DCT coefficients, $\boldsymbol{h}[0]$ the number of DCT coefficients equal to zero, and $\boldsymbol{h}[1]$ the number of DCT coefficients with absolute value 1, the estimated capacity is given by [21]:

$$C = \boldsymbol{h}_{DCT} - \frac{\boldsymbol{h}_{DCT}}{64} - \boldsymbol{h}[0] - \boldsymbol{h}[1] + 0.49\boldsymbol{h}[1] \tag{2.30}$$

where $\boldsymbol{h}_{DCT}/64$ is the number of DC elements and $\boldsymbol{h}[1] + 0.49\boldsymbol{h}[1]$ is an estimation of the loss due to shrinkage.

By using the total length of the message and the expected capacity C, the optimal segment length k is computed. A good choice of k will allow to change a minimum number of LSBs. Next, a pseudo-random number generator (PRNG) is seeded by a key derived from a user password. It will be used to select DCT coefficients from the cover image along a pseudo-random path. The same PRNG is used to generate a bit-stream that will be XORed with the message to make it look more randomized.

For each message segment of k bits, a group of $2^k - 1$ coefficients is used by the matrix embedding. When shrinkage does occur for one subset of the message, it will be embedded again with the next group of coefficients. If the estimated capacity is finally not enough for the message, the embedding fails. This is mainly caused by an optimistic estimation of the shrinkage.

Cover image histogram steganalysis

The F5 method does not overwrite pixel bits of the cover image which makes attacks using the chi-square test useless. While characteristic artifacts such as staircase shapes do not occur, the histogram of DCT coefficients is still modified. In general, the number of DCT coefficients equal to zero will increase after embedding while the number of non-zero coefficients will decrease. Fridrich et al. [21] propose a steganalysis method to break the F5 algorithm by estimating the cover image histogram from the stego image. This attack has good detection performance and is also able to estimate the message size.

The first step is to make an accurate estimation $\hat{\boldsymbol{h}}$ of the cover image histogram. This estimation will then be used to compute the expected values of the stego image histogram. The stego image is decompressed to the spatial domain, cropped by 4 columns and recompressed using the same quality factor. A uniform blurring operation with a 3×3 kernel is applied to remove potential JPEG blocking artifacts.

This histogram estimation is possible because the stego image obtained with F5 is visually very close to the cover image.

Modifications introduced by the F5 algorithm are publicly known. Therefore, it is possible to estimate the evolution of the number of DCT coefficients of a given value. The number of AC coefficients in the cover image with absolute value equal to d is denoted by $\boldsymbol{h}[d]$ and $\boldsymbol{h}_{kl}[d]$ represents the number of AC coefficients corresponding to the frequency (k,l), $1 \leq k,l \leq 8$ that have an absolute value equal to d. If n AC coefficients from a total of P non-zero AC coefficients are modified during the embedding, the probability that a non-zero coefficient will be modified is given by $\beta = n/P$. The expected values of the stego image histograms are denoted by the capital letter $\boldsymbol{H}$ and are given by:

$$\begin{aligned} \boldsymbol{H}_{kl}[d] &= (1-\beta)\boldsymbol{h}_{kl}[d] + \beta \boldsymbol{h}_{kl}[d+1], \quad \text{for} \quad d > 0 \\ \boldsymbol{H}_{kl}[0] &= \boldsymbol{h}_{kl}[0] + \beta \boldsymbol{h}_{kl}[1], \quad \text{for} \quad d = 0 \end{aligned} \tag{2.31}$$

Using the previous estimation of $\hat{\boldsymbol{h}}$, it is possible to calculate the expected values $\boldsymbol{H}_{kl}[d]$. The parameter β is best estimated using the least square approximation in order to minimize the square error between the real stego image histogram $\boldsymbol{H}_{kl}$ and the expected values $\boldsymbol{H}_{kl}[d]$ obtained with equation (2.31).

$$\beta_{kl} = \arg\min_{\beta} \left(\boldsymbol{H}_{kl}[0] - \hat{\boldsymbol{h}}_{kl}[0] - \beta \hat{\boldsymbol{h}}_{kl}[1] \right)^2 + \left(\boldsymbol{H}_{kl}[1] - (1-\beta)\hat{\boldsymbol{h}}_{kl}[1] - \beta \hat{\boldsymbol{h}}_{kl}[2] \right)^2 \tag{2.32}$$

Which leads to the following formula:

$$\beta_{kl} = \frac{\hat{\boldsymbol{h}}_{kl}[1] \left(\boldsymbol{H}_{kl}[0] - \hat{\boldsymbol{h}}_{kl}[0] \right) + \left(\boldsymbol{H}_{kl}[1] - \hat{\boldsymbol{h}}_{kl}[1] \right) \left(\hat{\boldsymbol{h}}_{kl}[2] - \hat{\boldsymbol{h}}_{kl}[1] \right)}{\hat{\boldsymbol{h}}_{kl}^2[1] + \left(\hat{\boldsymbol{h}}_{kl}[2] - \hat{\boldsymbol{h}}_{kl}[1] \right)^2} \tag{2.33}$$

Only low frequency coefficients $(k,l) \in \{(1,2),(2,1),(2,2)\}$ are considered for the estimation in order to avoid using higher frequency coefficients that have insufficient statistics. The detection algorithm uses the estimation β of the number of relative changes introduced by F5 to decide if a secret message is embedded in the image.

2.4.4 Steganography by cover-source swtiching

A steganographic technique based on an embedding process that switches between two sources is proposed by P. Bas and named *natural steganography* [5]. The method mimics a change of ISO sensitivity by adding a particular non-random noise. This additional noise will simulate a new source to which the embedding will switch from the primary one.

Cover-source switching uses the statistical noise that is present in sensors in order for the image to appear being generated from a source $\mathcal{S}_2$ when it was generated by $\mathcal{S}_1$. The author works with the model for sensor noise cited previously [4] which provides a simple noise model that will imitate the change of ISO, thus simulating another source. Thereby, the statistical properties of the stego image will be the same as if it was generated by this second source.

The signal $x(i,j)$ at one location of the image can be described by a combination of two signals: the "dark" signal and the "electronic" signal. The dark signal is denoted by $x_d(i,j)$ and the expectation $E[X_d(i,j)] = \mu_d$ is the mean number of electrons present without light. The electronic signal is denoted by $x_e(i,j)$ and the expectation $E[X_e(i,j)] = K\mu_e$ is the mean number of electrons coming from the scene, where K is the gain (section 2.2.1). The dark signal depends on exposure time and ambient temperature.
The expectation of the signal at one photosite is expressed as:

$$\mu_{i,j} = E[X_d(i,j)] + E[X_e(i,j)] = K\mu_e + \mu_d \tag{2.34}$$

Other mutually independent noises are also considered. The shot noise that follows a Poisson distribution can be approximated by a normal distribution $\mathcal{N}(\mu_e, \sigma_e^2)$. With $\sigma_e^2 = \mu_e$, the electronic signal is therefore given by $\mathcal{N}(0, \mu_e)$. The readout noise is normally distributed as $\mathcal{N}(0, \sigma_d^2)$ and the quantization noise with variance σ_q is uniformly distributed.
The sensor noise derived from the model of [4] is given by:

$$\sigma_s^2 = K^2\sigma_d^2 + \sigma_q^2 + K(\mu - \mu_d) \tag{2.35}$$

With the assumptions that the gain K is constant, that the dark signal is also constant ($\sigma_d^2 = 0$), and that the quantization noise is negligible ($\sigma_q^2 = 0$), there will be a linear relation between μ and the sensor noise variance.

$$N_{i,j}^{(1)} \sim \mathcal{N}(0, a_1\mu_{i,j} + b_1) \tag{2.36}$$

The signal at location (i,j) from the first source, denoted by $^{(1)}$, is then expressed as:

$$x_{i,j}^{(1)} = \mu_{i,j} + n_{i,j}^{(1)} \tag{2.37}$$

with $X \sim \mathcal{N}(\mu, a_1\mu_{i,j} + b_1)$.

Then, the parameters a_1 and b_1 are estimated. This is done by using a given camera with a fixed ISO parameter. A set of raw images is captured and no processing

of the dark signal is applied. Outputs of the photosites are assigned to subsets according to their normalized outputs, then, empirical means and unbiased variances are computed from these sets. Finally, a_1 and b_1 are estimated by linear regression.

The embedding principle aims to mimic an image captured with a different ISO parameter $ISO_2 > ISO_1$. Since the development of a color image includes several processing steps (demosaicing, gamma correction, etc.) the steganographic technique is at first demonstrated with a simple monochrome image that will only undergo quantization.

For a second source characterized by the parameter ISO_2, the noise will be given by $N_{i,j}^{(2)} \sim \mathcal{N}(0, a_2\mu_{i,j} + b_2)$ and the signal by $x_{i,j}^{(2)} = \mu_{i,j} + n_{i,j}^{(2)}$ which can be written as:

$$x_{i,j}^{(2)} = \mu_{i,j} + n_{i,j}^{(1)} + s'_{i,j} \tag{2.38}$$

$$= x_{i,j}^{(1)} + s'_{i,j} \tag{2.39}$$

where $S'_{i,j} \sim \mathcal{N}(0, (a_2 - a_1)\mu_{i,j} + b_2 - b_1)$ is the additional noise to get from $x_{i,j}^{(1)}$ to $x_{i,j}^{(2)}$. It is then assumed that the observed pixel $x_{i,j}^{(1)}$ is very close to its practical expected value $\mu_{i,j}$. This implies the principle of cover-source switching:

$$x_{i,j}^{(2)} \simeq x_{i,j}^{(1)} + s_{i,j} \triangleq y_{i,j} \tag{2.40}$$

with $S_{i,j} \sim \mathcal{N}(0, (a_2 - a_1)x_{i,j}^{(1)} + b_2 - b_1)$ and $Y_{i,j} \sim \mathcal{N}(x_{i,j}^{(1)}, \sigma_S^2)$.

The embedding process described above is a simple addition of an independent noise that only works with monochrome sensors because no further transformation is involved. When color sensors are used, additional advanced computations need to be applied in order to include transformations such as gamma correction, demosaicing, color transform, or down and up-sampling (see [5].III for more details). It involves the knowledge of development techniques presented in Chapter 2.2 . All this processing is achieved to provide a correct model in the processed domain of the perturbations caused by embedding in the raw domain.

2.5 Conclusion

The steganographic techniques that were reviewed in this chapter are part of the fundamental methods through which steganography evolved to its current state. But the field is vast and existing techniques are numerous, not to mention all systems that derive from the main ones and implement subtle differences. These techniques are currently some of the most known and are often used in works as benchmark. In addition, this review also demonstrated the evolution of existing systems and how researchers develop new methods based on previous ones while taking into consideration the flaws that made them easily detectable. Hence, the construction of new targeted attacks contributes significantly in the improvement of the security of steganographic methods.

Many techniques can be found for free on the Internet today, however it mainly concerns simplistic steganographic systems with poor security [17]. As seen with the basic LSB embedding, changes introduced to LSBs in the spatial or frequency domain have been shown to be easily detectable. However, with JPEG being the most present image file format on the Internet [45], JPEG steganography is a field where progress is still in evolution.

Steganographic systems presented in this chapter embed message bits without consideration to the image content. Bits are hidden in a sequential manner or using a secret key. In contrary, the method selected for this work and described in Chapter 4 is different in its way of selecting message bits locations.

Chapter 3

Theoretical foundations of image steganography

Steganographic techniques do not operate all in the same manner. As seen in the previous chapter, some use spatial domain representations of images while others operate in the frequency domain. When all methods aim to deliver perfect steganographic systems that are safe against attacks, it is necessary to understand the concept of a steganographic system along with its channel.

Therefore, this chapter's first objective is to explain the main theory established in the field of steganography and the improvements that have been made to the concept in order to adapt to situations where it cannot always be used. Then, an introduction to steganalysis and attacks is proposed to understand the issues to which all systems are confronted. These elements intend to provide a fundamental basis for every steganographic technique.

3.1 Channel and security

A well known formulation of the steganographic problem is given by Simmons [39] with the *prisoners' problem*. Two prisoners, Alice and Bob, are kept in separate cells and want to communicate to conceive an escape plan. Communication is allowed between prisoners but a warden named Eve keeps an eye on all messages that are exchanged. If she suspects anything, communications will be forbidden and Alice and Bob will never have another opportunity to escape. Eve may be an active or a passive warden. In case of an active warden, Eve will modify stego objects in order to make communication using steganography impossible for the prisoners, while a passive warden will only observe all messages transmitted between the prisoners and try to distinguish stego objects from innocent cover messages.

3.1.1 Principle

Prior to all communications, Alice and Bob need to agree on several elements. They must decide which type of data will be used for hiding and then build an embedding function $Emb()$ and its corresponding extraction function $Ext()$ such that

$$Ext(Emb(message)) = message$$

This pair of functions establishes a *stegosystem* [7, 15]. The two algorithms may depend on a key, in which case Alice and Bob need to agree on beforehand [25]. Usually, the key is randomly drawn from a set of all possible keys with uniform distribution.

The two accomplices communicate by sending messages through a *channel* monitored by the warden Eve. The steganographic channel is composed of: the embedding function, the extraction function, and sources of messages, covers and keys [15].

There are three fields of steganography that vary in the way that Alice has to embed her messages [15]. She can convey information by selecting from a database a specific cover that corresponds to the message (cover selection); she can create the desired object that will reflect the message (cover synthesis); or she can alter an object in order to hide the message in it (cover modification). With cover selection, Alice could simply select an image from a set of images shared with Bob and send it without any modification. A hash function on which they both agreed is used by Bob in order to obtain the message bit(s) from the image. Even if the computation complexity needed by Alice to find an image that conveys the desired message is not considered, there is still one major problem: she has not necessarily the choice of the cover image. This is the reason why steganography by cover modification is the most used and will be the method discussed in this paper.

3.1.2 Notations

For the sake of consistency, this paragraph defines the notations that will be used through this work.

X	Random variable
$\boldsymbol{X}$	Random vector/matrix
x	Realization of random variable X
$\boldsymbol{x}$	Realization of random vector/matrix $\boldsymbol{X}$
$\mathcal{X}$	Set of possible values
i, j	Vector/matrix indices

3.1.3 Steganographic channel

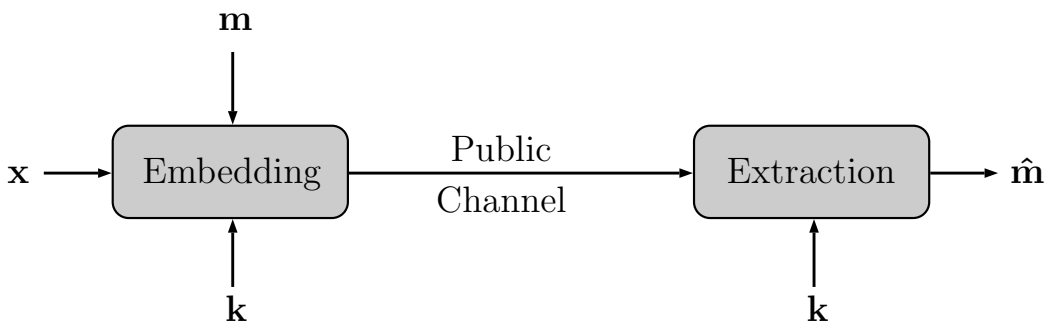


Figure 3.1: Steganographic channel with cover modification.

Given a cover $\boldsymbol{x}$ selected from the set $\mathcal{C}$ of all cover objects, the sender has to build an embedding function that takes $\boldsymbol{x}$, a key $\boldsymbol{k}$ from the set $\mathcal{K}(\boldsymbol{x})$ of all keys for $\boldsymbol{x}$, and a message $\boldsymbol{m}$ from the set $\mathcal{M}(\boldsymbol{x})$ of all messages that can be embedded in $\boldsymbol{x}$. This process will result in a stego object $\boldsymbol{y} \in \mathcal{S}$ [11, 15, 25, 26].

$$Emb : \mathcal{C} \times \mathcal{K} \times \mathcal{M} \to \mathcal{S}$$
$$Emb(\boldsymbol{x}, \boldsymbol{k}, \boldsymbol{m}) = \boldsymbol{y}$$

where $\boldsymbol{x} \in \mathcal{C}$, $\boldsymbol{k} \in \mathcal{K}(\boldsymbol{x})$, $\boldsymbol{m} \in \mathcal{M}(\boldsymbol{x})$. According to the embedding method, changes will be made in all elements of the cover image or in some of them. These pixels are selected beforehand along a pseudo-random path generated from the key or according to the cover properties. Modifications are restricted to this set of pixels which are designated as the *selection channel*.
The extraction function at the recipient side needs the stego object $\boldsymbol{y}$ and the key $\boldsymbol{k}$ used during the embedding process in order to uncover the original message $\boldsymbol{m}$.

$$Ext : \mathcal{S} \times \mathcal{K} \to \mathcal{M}$$
$$Ext(\boldsymbol{y}, \boldsymbol{k}) = \boldsymbol{m}$$

It is assumed that any attacker knows everything about the algorithms. Therefore, the security of a steganographic system only depends on the secrecy of its key and not on its design. This assumption is known as the *Kerckhoffs' principle* [3, 7, 15, 25, 26].

A steganographic system has to be undetectable, which means that the attacker should not be able to detect a single stego object among all communications. A distinction must be made when speaking about undetectability of a system. A data file containing hidden information can be undetected by the human eye because no visible artifact is present, this is known as the *perceptual invisibility*. At the same time, the hidden data can be detected by the use of stochastic analysis. If the warden Eve has her eyes as the only detection tool, perceptual invisibility will be sufficient. However, it is always assumed that Eve knows how the system works and that all means and technologies are at her disposal. Thus, having a system that is perceptually and statistically undetectable is essential.

This property implies that it is not possible to distinguish a cover object $\boldsymbol{x} \in \mathcal{C}$ from a stego object $\boldsymbol{y} \in \mathcal{S}$ without possession of the key [25]. These elements can be captured by probabilistic models [7, 15]; cover objects are generated according to a distribution denoted by p_C. This distribution is known by all parties, even by the attacker, given the fact that the amount of information that is available to her is unknown. When Alice uses her algorithm to embed a message, the resulting stego object $Emb(\boldsymbol{x}, \boldsymbol{k}, \boldsymbol{m})$ follows the distribution p_S. Naturally, if she wants her messages not to be detected, the distribution p_S must be as close as possible to the distribution p_C. As seen in section 2.2.4, images acquired with dedicated devices contain noise from many sources. The only possible approach for Alice is to design an embedding algorithm that mimics these noises. The message will be camouflaged as noise and the warden Eve will consider it as naturally present in the image.

Fundamental attributes of a steganographic system are *non-detectability*, *robustness*, and *capacity* [15].

3.1.4 Cachin's definition of steganographic security

Cachin [7] contributed considerably in the field of steganography by giving formal information-theoretic definitions of the security of stegosystems.

The security of a stegosystem is given in terms of the relative entropy: the *Kullback-Leibler divergence* (KLD). This information measures the difference between two probability distributions. Thus, the system's security is measured by the similarity between cover and stego distributions.

$$D(p_C||p_S) = \sum_{\boldsymbol{x}\in\mathcal{C}} p_C(\boldsymbol{x}) \log \frac{p_C(\boldsymbol{x})}{p_S(\boldsymbol{x})} \tag{3.1}$$

with a logarithm that can be either in base 2 or the natural logarithm. The result of this measure cannot be negative $D(p_C||p_S) \geq 0$.

If the sender can build a stegosystem with a relative entropy of zero, he will have achieved a *perfectly secure system* that an attacker will not be able to detect even after observing all messages during a long period of time [15]. This is accomplished if and only if the relative entropy is null, which means that the distributions of cover and stego objects are *identical*.

Definition 3.1 (**Stegosystem validity**). [7] Given the spaces $\mathcal{M}$, $\mathcal{K}$, and $\mathcal{C}$, a *stegosystem* $\mathfrak{S}$ composed of the functions $Emb()$ and $Ext()$ is only *valid* if, for all $\boldsymbol{m} \in \mathcal{M}$ such as $H(\boldsymbol{m}) > 0$, we have

$$I(\boldsymbol{m}; \hat{\boldsymbol{m}}) > 0 \tag{3.2}$$

where $H(X)$ is the *entropy* [12] of a discrete random variable X with probability mass function p_X over the alphabet $\mathcal{X}$ given by:

$$H(X) = -\sum_{x\in\mathcal{X}} p_X(x) \log p_X(x) \tag{3.3}$$

and $I(X;Y)$ is the *mutual information* [12] between two random variables X and Y given by:

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{3.4}$$

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y;X) \tag{3.5}$$

Definition 3.2 (**Perfectly secure system**). [7] Let $\mathfrak{S}$ be a stegosystem with defined functions $Emb()$ and $Ext()$, and let p_C be the probability distribution of covers and p_S the probability distribution of stego objects.

- $\mathfrak{S}$ is called *perfectly secure* against passive attackers if:

$$D(p_C||p_S) = 0 \tag{3.6}$$

- $\mathfrak{S}$ is called $\epsilon-$*secure* against passive attackers if:

$$D(p_C||p_S) \leq \epsilon \tag{3.7}$$

Hiding a message by using cover modification will inevitably introduce distortions to the stego object. These modifications can be measured by using a *distortion function* $d : \mathcal{C} \times \mathcal{S} \to \mathbb{R}^+ \cup \{0\}$ [11, 15, 33] such as

$$d_k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} |\boldsymbol{x}[i] - \boldsymbol{y}[i]|^k \tag{3.8}$$

The well-known Peak Signal-to-Noise Ratio (PSNR) can also be used.

$$PSNR = 10 \cdot \log_{10} \frac{\max(\boldsymbol{x})^2}{MSE} \quad \text{with} \quad MSE = \frac{1}{n} \sum_{i=1}^{n} |\boldsymbol{x}[i] - \boldsymbol{y}[i]|^2 \tag{3.9}$$

It is possible to decide to modify only some areas of the cover image in order to avoid changes that can be visually detected. In that case, flat areas and edges are avoided while embedding into textures is preferred. This is known as *adaptive steganography*.

Definition 3.3 (**Robustness**). [25] Let $\mathfrak{S}$ be a stegosystem and $\mathfrak{p}$ a class of mappings $\mathfrak{p} : \mathcal{C} \to \mathcal{C}$. $\mathfrak{S}$ is said to be *robust* if for all mappings $p \in \mathfrak{p}$,

$$Ext(p(Emb(\boldsymbol{x}, \boldsymbol{k}, \boldsymbol{m})), \boldsymbol{k}) = Ext(Emb(\boldsymbol{x}, \boldsymbol{k}, \boldsymbol{m}), \boldsymbol{k}) = \boldsymbol{m} \tag{3.10}$$

A stego object that is sent through the public channel must be able to endure compression and image processing operations such as rotation, cropping, etc. These transformations are described by the mapping $\mathfrak{p}$.

Let $\boldsymbol{x} \in \mathcal{C}$ be any cover image of size $n \times m$ and M_c the set of messages that can be embedded in $\boldsymbol{x}$ [15]. If the encoder is able to embed one message bit per pixel, by using an LSB embedding algorithm for instance (section 2.4.1), then, the space of all

possible messages is of size $|M_c| = 2^{n \times m}$ in order to have all possible combinations. The *embedding capacity* of the cover $\boldsymbol{x}$ can thus be obtained with:

$$\log_2 |M_c| \tag{3.11}$$

which corresponds in our example to $n \times m$. It is important to emphasize on the fact that this embedding capacity does not take into account distortions, that the measure is different for each cover, and that it heavily depends on the embedding algorithm. If adaptive steganography is used instead, the capacity will be greatly reduced.

3.1.5 Distortion limited embedding and capacity

When communicating through a steganographic system, Alice and Bob need to know which quantity of information can be hidden without Eve noticing any difference between the stego image and an innocent image. Security of stegosystems as defined by Cachin (3.6, 3.7) implies that the probability distributions p_C and p_S are known to the users of the system. This assumption is conservative because accurate models for natural images are missing and current models are unreliable [15, 34]. It is also assumed that as long as p_S is identical to p_C, distortions generated by the embedding are not important and can even be very large if the assumption $D(p_C||p_S) = 0$ is still respected [15].

When using steganography by cover modification, it can be more legitimate to focus the security condition on the amount of distortion introduced by the modifications [15, 33, 34]. During the embedding stage, the sender must ensure that the stego image will not differ from the cover image, therefore, the embedding distortion needs to be bounded. The communicating parties will use a distortion-limited embedder and they will not be able to hide as many bits as desired in order to prevent detection.

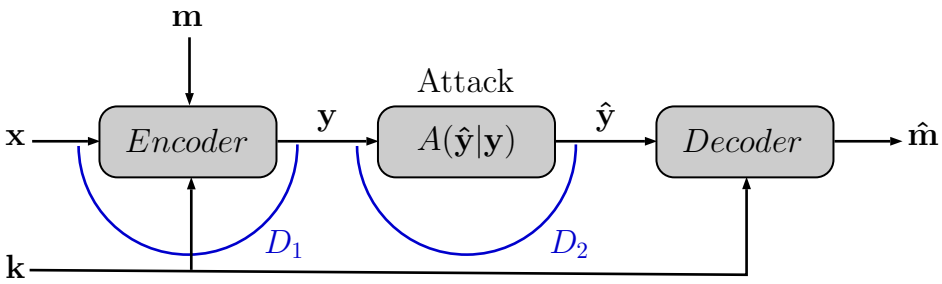


Figure 3.2: Distortion limited embedder (D_1) and noisy channel (D_2).

A stegosystem that has two new constraints is considered [34]. During embedding, the encoder hides a message $\boldsymbol{m}$ in a cover $\boldsymbol{x}$. This will produce the stego object $\boldsymbol{y}$ that will be transmitted to the decoder through the monitored channel. A first constraint is considered on the expected distortion that is introduced during embedding. This bounds the maximal distortion that can be introduced by the encoder

in order for the stego object to stay as close as possible to the cover object. The second constraint corresponds to a bound on the expected distortion of the channel when the stego object is sent. If Eve is an active warden, she needs to keep the stego object that she modifies close to the original stego that was sent by one of the prisoners. This system can be seen in Figure 3.2.

Let $\mathfrak{S}$ be a stegosystem with the functions $Emb()$ and $Ext()$, and the probability distributions p_C, p_K, p_M, and p_S that describe covers, keys, messages and stego objects, respectively. The expected embedding distortion is bounded as follows [15, 33, 34]:

$$E[d(x,y)] = \sum_{\boldsymbol{x}\in\mathcal{C}} p_C(\boldsymbol{x}) \sum_{\boldsymbol{k}\in\mathcal{K}} p_K(\boldsymbol{k}) \sum_{\boldsymbol{m}\in\mathcal{M}} p_M(\boldsymbol{m}) d(\boldsymbol{x}, Emb(\boldsymbol{x},\boldsymbol{k},\boldsymbol{m})) \leq D_1 \tag{3.12}$$

When Alice sends the stego object $\boldsymbol{y}$ to Bob through the public channel, the warden monitoring the channel, that is active in that case, will attack the message. This active warden, that can be seen as a discrete memoryless noisy channel $A(\hat{\boldsymbol{y}}_i|\boldsymbol{y}_i)$ [15], is described by the conditional probability that the noisy stego object $\hat{\boldsymbol{y}}$ is received by Bob when $\boldsymbol{y}$ was sent by Alice. This conditional probability is given by:

$$p_{\hat{\boldsymbol{S}}|\boldsymbol{S}}(\hat{\boldsymbol{y}}|\boldsymbol{y}) = \prod_{i=1}^{n} A(\hat{\boldsymbol{y}}_i|\boldsymbol{y}_i) \tag{3.13}$$

where n is the length of the stego object. The discrete memoryless attack channel is achievable if the expected distortion between S and $\hat{S}$ is bounded as follows [15, 33, 34]:

$$\sum_{\boldsymbol{y},\hat{\boldsymbol{y}}} p_S(\boldsymbol{y}) A(\hat{\boldsymbol{y}}|\boldsymbol{y}) d(\boldsymbol{y},\hat{\boldsymbol{y}}) \leq D_2 \tag{3.14}$$

Under constraint of perfect security ($p_C = p_S$), it can be written as

$$\sum_{\boldsymbol{y},\hat{\boldsymbol{y}}} p_C(\boldsymbol{y}) A(\hat{\boldsymbol{y}}|\boldsymbol{y}) d(\boldsymbol{y},\hat{\boldsymbol{y}}) \leq D_2 \tag{3.15}$$

In the case of a passive warden, the channel is noiseless and $D_2 = 0$ because the warden does not modify the stego object ($\boldsymbol{y} = \hat{\boldsymbol{y}}$).

Definition 3.4 (Rate). [33, 34] Errors may appear when using the extraction algorithm such that the possibility that $Ext(Emb(\boldsymbol{x},\boldsymbol{k},\boldsymbol{m}),\boldsymbol{k}) \neq \boldsymbol{m}$ exists. The *rate* R of a stegosystem $\mathfrak{S}$ is achievable if there exist mappings of embedding and extraction functions such that $|\mathcal{M}| \geq 2^{nR}$ and

$$\sup_{P_{\hat{S}|S}} \Pr\{\hat{\boldsymbol{m}} \neq \boldsymbol{m}\} \to 0 \quad \text{as} \quad n \to \infty \tag{3.16}$$

This definition is constrained by (3.12) and (3.15) due to the embedding function and the supremum over all conditional probabilities.

Definition 3.5 (**Steganographic capacity**). [15, 33, 34] The *steganographic capacity* is defined as the supremum of all achievable rates and depends also on the bounds defined above, which is why it is denoted as $C_{steg}(D_1, D_2)$ (see Theorem 3.1 below).

Definition 3.6 (**Steganographic channel**). In order to operate, the embedder/encoder and the extractor/decoder need to share a random codebook.
Let $\mathcal{U}$ be a finite alphabet of arbitrary large size and a random variable $U \in \mathcal{U}$. This random variable decides which codebook will be used between the sender and the recipient from the set of all shared codebooks.

A *steganographic channel* [15] $\mathcal{Q}_{Steg} = p_{SU|C}(\boldsymbol{y}, u|\boldsymbol{x})$ is a conditional probability distribution such that

$$p_{SU|C}(\boldsymbol{y}, u|\boldsymbol{x}) = \Pr\{Y = \boldsymbol{y}, U = u|X = \boldsymbol{x}\} \tag{3.17}$$

This channel is subject to the embedding distortion constraint and respects the perfect steganographic constraint ($p_C = p_S$). If the second condition is not valid, then the channel is a simple *covert channel* denoted by $\mathcal{Q}$ with $\mathcal{Q}_{Steg} \subseteq \mathcal{Q}$.
A covert channel [33, 34] is feasible if:

$$\sum_{\boldsymbol{y},u,\boldsymbol{x}} p_{SU|C}(\boldsymbol{y}, u|\boldsymbol{x}) p_C(\boldsymbol{x}) d(\boldsymbol{x}, \boldsymbol{y}) \leq D_1 \tag{3.18}$$

$$\sum_{u,\boldsymbol{x}} p_{SU|C}(\boldsymbol{y}, u|\boldsymbol{x}) p_c(\boldsymbol{x}) = p_C(\boldsymbol{y}) \tag{3.19}$$

Theorem 3.1 (**Steganographic capacity**). The *steganographic capacity* is given by the supremum over all feasible covert channels $\mathcal{Q}$ of the infimum over all attack channels $A(\hat{\boldsymbol{y}}_i|\boldsymbol{y}_i)$ defined previously.

$$C_{Steg}(D_1, D_2) = \sup_{\mathcal{Q}} \inf_{A \in \mathcal{A}} \{I(U; \hat{S}) - I(U; C)\} \tag{3.20}$$

where $(U, C) \to S \to \hat{S}$ forms a Markov chain.
It is possible to derive from this theorem the case for a passive warden where $D_2 = 0$. In that case, the definition (3.20) is taken with $U = S$:

$$C_{Steg}(D_1, 0) = \sup_{\mathcal{Q}} H(S|C) \tag{3.21}$$

Detailed proofs can be found in [34].

3.2 Steganalysis

The goal of steganography is to communicate hidden messages through innocuous objects without drawing any suspicion. In the prisoners' problem, the channel used by Alice and Bob is monitored by the warden Eve that is designated as the *steganalyst*. The steganalyst's role is to try to distinguish stego objects from cover objects, and if she succeeds, the steganographic channel is considered broken. When establishing the nature of a given observed object, the decision has to be better than random guessing [15, 26]. In most scenarios, the simple confirmation of the existence of a covert message is sufficient and its extraction may be unnecessary.

It is always essential to perform analyses on a stegosystem. Steganalysts must take advantage of existing attacks in order to build systems that are more resistant [6]. The fields of steganography and steganalysis are complementary and one cannot evolve without the other, it is a perpetual game.

3.2.1 Attacks

There are three types of attacks that Eve can perform [3, 15, 25].

Passive warden The steganalyst can be a simple eavesdropper and observes all objects that are communicated in order to determine which ones are stego objects. Given the fact that the channel is under her control, she can accordingly decide not to let a message pass if it is suspicious.

Active warden The steganalyst is not interested by the hidden content of the objects but wants to hinder all communications between parties. In this case, she will only modify the stego objects in order for the recipient to not be able to extract the messages.

Malicious warden The steganalyst knows which stegosystem is used and is able to extract the hidden messages. She then modifies the stego objects in order to convey false information to other parties, without them knowing that their stegosystem has been compromised.

The Kerckhoffs' principle cannot be always put into practice and the steganalyst may have multiple possible setups of attack. The embedding and extraction algorithms may be unknown to her, as well as the cover source. As a consequence, when some elements are missing, the steganalysis task becomes more complicated. Every side information is crucial and can help mount an attack.

In cryptography, attacks used by cryptanalysts are classified according to the side information available. Similarly, steganographic attack techniques can be classified into categories, e.g., *stego-only attacks* where only the stego object is available to

the steganalyst, *known-cover attacks* where both stego and cover objects are available, or *chosen-stego attacks* where the steganographic algorithm and stego objects are known [25]. The existence of a setup where the steganalyst has access to the algorithm is also possible if the communicating parties have access to an embedding oracle that can also be queried by other actors [26].

In general, all techniques are grouped into two main categories [15]: attacks where the embedding technique is unknown to the steganalyst (blind) and attacks where this information is available (non-blind). In the first case, the attack system is built to be able to detect as many stegosystems as possible. In the second case, the attack system is conceived in order to only target specific features that are known to appear with the use of a specific embedding method. Both setups require a lot of research in order to design specific features that highlight potential weaknesses in stego objects.

3.2.2 Detection problem

From here, only the case of a passive warden is considered. Regardless of the attack setup and the side information available, the steganalyst has to deal with what is called a *detection problem* [7, 15]. When monitoring the steganographic channel, it is possible to establish an estimation of p_C by collecting enough samples $x \in \mathcal{C}$. The detection problem leads to a *binary hypothesis testing* [15]. The steganalyst needs to distinguish between two hypotheses: the case where an object is distributed according to p_C (hypothesis H_0) and the case where it is distributed according to p_S (hypothesis H_1). The purpose is to separate all observations into two mutually exclusive classes.

$$H_0 : \boldsymbol{x} \sim p_C \quad \text{(there is no hidden message)}$$
$$H_1 : \boldsymbol{x} \sim p_S \quad \text{(there is a hidden message)}$$

When there is no information about the method that was used to embed a message (blind steganalysis), the detection problem becomes a composite hypothesis testing problem [15], which is a more complex test.

$$H_0 : \boldsymbol{x} \sim p_C$$
$$H_1 : \boldsymbol{x} \nsim p_C$$

The steganalyst has to deal with a great number of images and working with full representations of images is not practical because of the large complexity involved. Hence, a set of features needs to be extracted from each image and these features must be well chosen according to the side information available. For each image $\boldsymbol{x} \in \mathcal{C}$, a mapping is applied in order to obtain a feature vector of dimension d denoted by $\boldsymbol{f_x} \in \mathbb{R}^d$. In the case of non-blind steganalysis, knowledge of the embedding algorithm allows the attacker to extract specific features that are known to change after the use of this particular stegosystem. The selected features must provide a clear separation between p_C and p_S into two regions with as little overlap as possible.

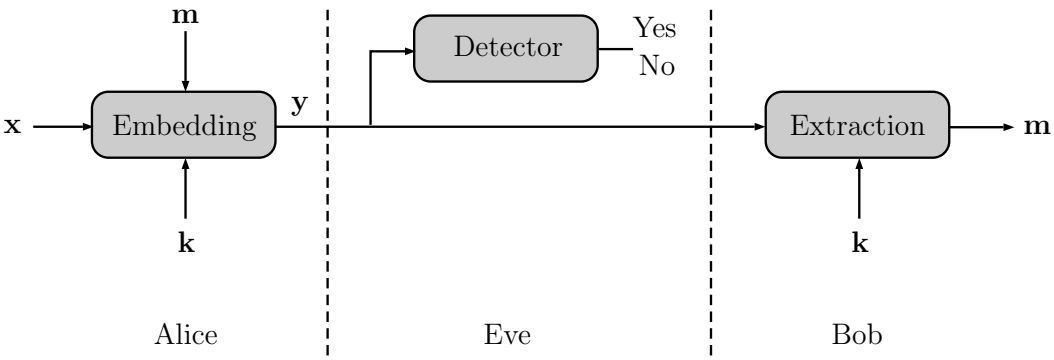


Figure 3.3: Detection problem with a passive warden.

With the problem of binary hypothesis testing, the steganalyst builds what is called a detector that can be expressed by a map $\phi : \mathbb{R}^d \to \{0, 1\}$.

$$\phi(\boldsymbol{x}) = \begin{cases} 0, & \text{if } \boldsymbol{x} \in \mathcal{R}_0 \\ 1, & \text{if } \boldsymbol{x} \in \mathcal{R}_1 \end{cases} \tag{3.22}$$

The observed object $\boldsymbol{x}$ is detected as a cover if $\phi(\boldsymbol{x}) = 0$, and as a stego if $\phi(\boldsymbol{x}) = 1$. Two disjoint regions $\mathcal{R}_0$ and $\mathcal{R}_1$ are defined for the two hypotheses where the critical region $\mathcal{R}_1 = \{\boldsymbol{x} \in \mathbb{R}^d | \phi(\boldsymbol{x}) = 1\}$ defines the objects that will be detected as stego if $\boldsymbol{x} \in \mathcal{R}_1$.

When the warden Eve is monitoring the channel, she needs to be almost sure that a transmitted image contains hidden information before denouncing the prisoners. However, it is nearly impossible to clearly separate $\mathcal{R}_0$ from $\mathcal{R}_1$, thus, the detection is not 100% reliable. The detector may incur in two types of errors when analyzing an object. First, there is a probability p_m of *missed detection*, also called Type I error, that occurs when the detector decides that an object is a cover image when in reality it is a stego image. Then, there is a probability p_f of *false acceptance*, also called Type II error, when an innocent cover object is labeled as a stego object. Given an observed object $\boldsymbol{x}$, the two error types are defined as follows: [15]

$$p_f = \Pr\{\phi(\boldsymbol{x}) = 1 | \boldsymbol{x} \sim p_C\} = \int_{R_1} p_C(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{3.23}$$

$$p_d = \Pr\{\phi(\boldsymbol{x}) = 1 | \boldsymbol{x} \sim p_S\} = \int_{R_1} p_S(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{3.24}$$

$$p_m = \Pr\{\phi(\boldsymbol{x}) = 0 | \boldsymbol{x} \sim p_S\} = 1 - p_d \tag{3.25}$$

where p_d is the probability of correct detection.

In general, a good detector must be designed to have a low probability of false acceptance [15]. Eve wants to be sure that Alice and Bob are secretly communicating before taking further actions and will, in consequence, lower p_f even if the probability of miss p_m increases. Both probabilities cannot be minimized at the same time, which involves a trade-off between false acceptance and miss.

Two methods are commonly used to find an optimal criterion of decision, the Bayesian hypothesis testing and the Neyman-Pearson threshold.

Bayesian A cost function is defined for each error type that may occur when a decision is taken by the detector:

- $C_{11} \leq 0$, the cost when H_1 is correctly detected as H_1.
- $C_{00} \leq 0$, the cost when H_0 is correctly detected as H_0.
- $C_{10} > 0$, the cost of false detection.
- $C_{01} > 0$, the cost of missed detection.

The principle is to minimize the risk of erroneous decision by minimizing the total cost C. Let $P(H_i^D, H_j)$ be the joint probability of detecting i when the real decision must be j.

$$P(H_i^D, H_j) = P(H_i^D|H_j)P(H_j) = \Pr(\boldsymbol{x} \in \mathcal{R}_i|\boldsymbol{x} \sim p_j)P(H_j) \tag{3.26}$$

$$C = \sum_{i,j\in\{0,1\}} C_{ij}P(H_i^D, H_j) \tag{3.27}$$

Neyman-Pearson Priority is given to the probability of false alarm that should not exceed a given value. A bound is imposed on this probability such that $p_f \leq \epsilon$. For a given p_f, the objective is to minimize p_m, or maximize $p_d = 1 - p_m$ in order to find an optimal division for the disjoint subsets $\mathcal{R}_1$ and $\mathcal{R}_2$.

$$p_f = \int_{R_1} p_C(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \leq \epsilon \tag{3.28}$$

In both situations, the optimal detector is the *Likelihood ratio test*. Given the probabilities $p_S(\boldsymbol{x})$ and $p_C(\boldsymbol{x})$, the detector decides H_1 if:

$$L(\boldsymbol{x}) = \frac{p_S(\boldsymbol{x})}{p_C(\boldsymbol{x})} > \lambda \tag{3.29}$$

In the case of the Bayesian detector, the threshold λ is given by:

$$\lambda = \frac{p_0}{p_1}\frac{C_{10} - C_{00}}{C_{01} - C_{11}} \tag{3.30}$$

And for the Neyman-pearson alternative, the threshold λ is the solution of the following equation:

$$\int_{L(\boldsymbol{x})>\lambda} p_C(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \epsilon \tag{3.31}$$

3.2.3 ROC curve

The performance of a detector is usually illustrated by the receiver operating characteristic curve (ROC curve). This function $p_d(p_f)$ is created by plotting the true positive rate against the false positive rate with various values for the parameter

λ. Measuring the ability of a detector to perform an accurate separation is done by comparing its ROC curve to other detectors' curves.
A ROC curve that is close to the diagonal line reveals a poor quality classification. It does not always mean that the classifier is terrible, but generally it is the sign of a bad choice of features.

A scalar measure of performance can be obtained by computing the area under the ROC curve: AUC or AUROC to be more precise.

$$\text{AUROC} = \int_0^1 p_d(x)dx \tag{3.32}$$

The area between the ROC curve and the diagonal line is also a good indicator of performance. The area is denoted by ρ and is normalized in order to have $\rho = 0$ for a very bad classification and $\rho = 1$ for a nearly perfect classification.

$$\rho = 2 \cdot \text{AUROC} - 1 \tag{3.33}$$

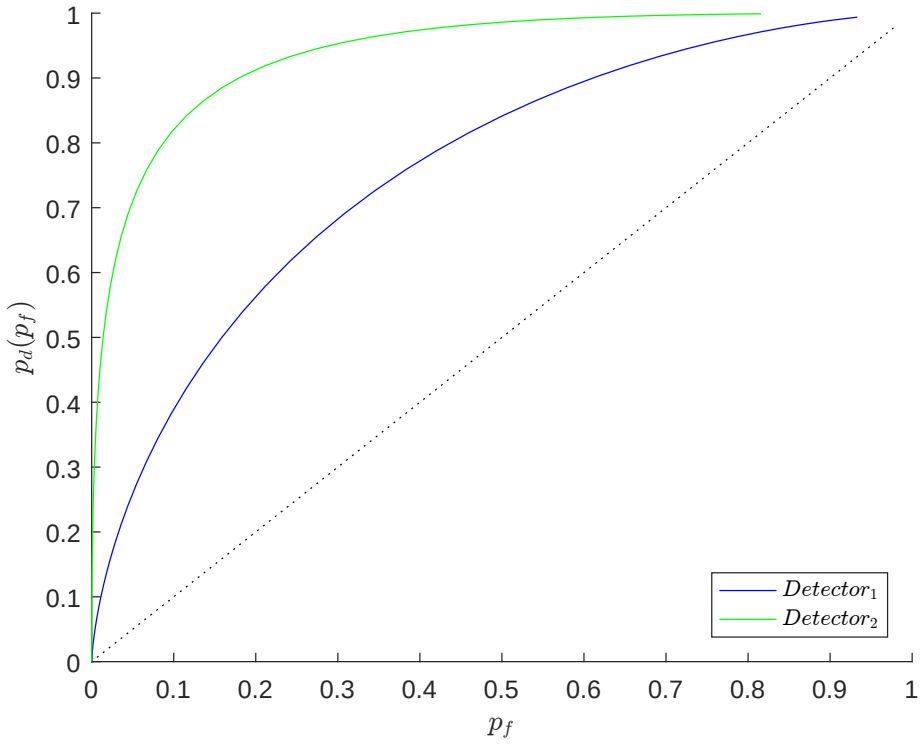


Figure 3.4: Comparison of two detectors with their ROC curves.

As stated in section 3.1.5, the major problem in steganalysis is the accurate estimation of the distributions p_C and p_S. Instead of using the likelihood-ratio test to solve the the detection problem, most of current steganalyst use supervised learning [29] where a classifier is trained with a large collection of cover and stego images.

3.2.4 Ensemble classifiers

The field of machine learning is today mostly occupied by Support Vector Machines (SVMs) that are favored when it comes to classification. This popularity is due to their resistance to over-fitting and the solid mathematical foundation that has been reinforced as time goes by. However, the complexity of SVMs is not suited for steganalysis where the feature dimensionality can be very large and the training dataset is not as big. A lot of constraints must be respected in order to obtain an accurate classification with a SVM, which makes their application in the field of steganalysis rather complicated.

In order to solve this issue, ensemble classifiers where adapted to steganalysis [29]. They are able to perform much more faster than SVMs with big feature dimensionalities due to their lower complexity. Because of the classification speed of ensemble classifiers, steganalysts are able to work with models that are more complex. Early works of steganalysis only used a few dozen of features, while current researches can use several thousand features to describe an image. The detection is less constrained and it is possible to use more complex descriptors that react to more steganographic techniques.

An ensemble classifier is composed of multiple independent base learners. Each base learner is trained independently on a subset of the training data. The independent subsets are chosen randomly and must contain the covers and their corresponding stego feature vectors. This is a specification developed only for steganalysis. Training different base learners on different training sets introduces diversity and contributes to lower the computational complexity. The classification result is finally obtained by fusing the individual decisions [28].

3.3 Conclusion

This chapter covered the essential theory that has been developed so far in the field of steganography. The definition of a steganographic system was formally described and the requirements for a steganographic channel were given in order to understand how to assess the security of such systems.

As stated, not all stegosystems can be classified under the Cachin theory for security, thus, further theories were developed to cope with the lack of formal distributions for p_C and p_S. The main concern of stegosystems is to introduce as little distortion as possible between the cover and the stego objects.

In the last section, different attack scenarios were given and the setup of a passive warden was explained along with the detection theory. The steganalysis field mainly consists in building detectors using supervised learning and the security of steganographic systems are compared by using their ROC curves.

Chapter 4

Perturbed quantization steganography with wet paper codes

4.1 Introduction

To improve the security of a steganographic scheme, it is possible to embed information in the regions of the cover image that can bear more modifications than others, which is the principle of adaptive steganography. However, the selection rule is usually publicly known and the attacker can apply it in order to know where information is more likely to be hidden. The same problem is encountered with selection channels weakly dependent on the secret key shared between Alice and Bob.

Some steganographic techniques try to embed information by minimizing the distortions on the cover image. Each modification of an element of the cover image has an impact $\boldsymbol{\rho}[i]$ that contributes to the total impact given by [15]:

$$d_\rho(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} \boldsymbol{\rho}[i](1 - \delta(\boldsymbol{x}[i] - \boldsymbol{y}[i])) \tag{4.1}$$

with $\boldsymbol{x}$ the cover image, $\boldsymbol{y}$ the stego image and δ the Kronecker function (2.7). This total embedding impact is minimized by a good choice of the selection channel. The main problem is that the impact needs to have knowledge of the cover object, which is only available at the sender side. As a result, the recipient has no access to information that enables him to do the same computation.

4.2 Wet paper codes

The principle of writing on wet paper is introduced by Fridrich et al. [15, 16, 20] and describes a variable-rate random linear code that uses a selection channel only available to the sender. They imagined a metaphor where the cover image is exposed

to rain and the sender can only modify dry pixels, which correspond to the selection channel. During transmission, the image will dry and the receiver will have no information about the pixels that were used by the sender. Given the fact that the selection channel is neither public nor communicated to the recipient(s), the attacker is not able to mount an attack using the selection channel as starting point.
The main novelty is the use of knowledge only available to the encoder and usually discarded after processing the image. In terms of security, the fact that the selection channel can only be determined at the sender side is a major improvement considering that if the side-information is not available to the receiver, it will also not be available to the attacker.

Given the fact that it is not possible for the recipient to know which are the pixels that contain information, the message bits cannot be communicated directly with one element per pixel but will be instead hidden within a group of individual elements. This can be achieved using syndrome coding which is very similar to the principle of matrix embedding explained previously.

In a setting where a message $\boldsymbol{m} = \{m_1, m_2, \ldots, m_q\}$ of q bits is communicated in a cover image, the sender and all recipients need to share a secret key and a parity function. Let $\boldsymbol{x}$ be the cover image containing n cover elements and $\boldsymbol{b} = \{b_1, b_2, \ldots, b_n\}$ the sequence of parities of these n elements, such that $b_i = Parity(x_i)$, $0 < i \leq n$. The parity function can be for instance defined by the least significant bits of pixels. With the use of the *selection rule*, the sender defines k *changeable elements* $b_i, i \in \mathcal{C} \subset \{1, \ldots, n\}, |\mathcal{C}| = k$, such that only these k elements can be modified (dry pixels) while the remaining $n-k$ elements are left untouched (wet pixels). Using the shared secret key, the sender and the recipient generate a binary matrix $\boldsymbol{D}$ of size $q \times n$. Similarly to syndrome coding, some of the changeable elements b_i, $i \in \mathcal{C}$, will be modified in order to satisfy the following equation:

$$\boldsymbol{Db'} = \boldsymbol{m} \tag{4.2}$$

where the stego vector $\boldsymbol{b'}$ is the modified version of $\boldsymbol{b}$.

In order to decode the message, the recipient does not need to know $\mathcal{C}$ and will simply have to apply the following multiplication:

$$\boldsymbol{m} = \boldsymbol{Db'} \tag{4.3}$$

At the recipient side, this multiplication is the only computation, however, the sender needs to solve a system of linear equations in GF(2) which is a more complex and time consuming task than the message extraction. The probability of solving equation (4.2) decreases when the size of the message m increases.

Given the fact that not all elements of $\boldsymbol{b}$ are modified, it is possible to rewrite equation (4.2) . Let $\boldsymbol{v} = \boldsymbol{b'} - \boldsymbol{b}$ be a vector where non-zero elements correspond to the bits that have to be changed by the encoder. Equation (4.2) is then rewritten as:

$$\boldsymbol{Dv} = \boldsymbol{m} - \boldsymbol{Db} \tag{4.4}$$

Not all n elements can be changed during embedding, thus, it is possible to remove from $\boldsymbol{D}$ all columns where $v_i = 0$ for $i \notin \mathcal{C}$ and rewrite (4.4) as:

$$\boldsymbol{H}\boldsymbol{v} = \boldsymbol{m} - \boldsymbol{D}\boldsymbol{b} \tag{4.5}$$

where $\boldsymbol{H}$ is a $q \times k$ submatrix of $\boldsymbol{D}$. A solution exists for any right-hand side if $rank(\boldsymbol{H}) = q$. The probability for a random binary matrix of size $q \times k$ of having the rank $s, s \leq \min(q, k)$ is given by [16, 19]:

$$P_{q,k}(s) = 2^{s(q+k-s)-qk} \prod_{i=0}^{s-1} \frac{(1-2^{i-q})(1-2^{i-k})}{1-2^{i-s}} \tag{4.6}$$

It has been proved in [16] that for a large fixed k, this probability approaches 1. This allows the sender to communicate a number of bits close to k, which is the theoretical upper bound.

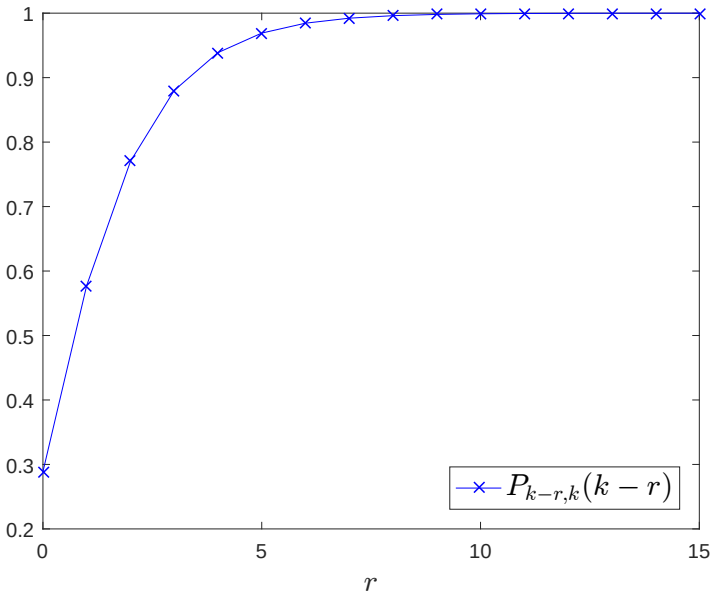


Figure 4.1: Probability for a matrix $k - r \times k$ of having a rank $k - r$.

In order to have an efficient way of solving the system, the only element that can be changed is the matrix $\boldsymbol{H}$. Since the structure of $\boldsymbol{H}$ cannot be directly chosen, it is however possible to choose the matrix $\boldsymbol{D}$ from a special class of random sparse matrices that will allow to solve the system with lower complexity [15]. If $\boldsymbol{D}$ is sparse, $\boldsymbol{H}$ will be sparse too and it will be possible, with high probability, to transform it into upper-triangular form by simply permuting the rows and columns. Once $\boldsymbol{H}$ in an upper-diagonal form, $\boldsymbol{H}_{ii} = 1$ for $i = 1, \ldots, m$, $\boldsymbol{H}_{ij} = 0$ for $i > j$, solving the system will be done by simple back-substitution as in the Gaussian elimination.

Solving equation (4.5) with a Gaussian elimination has a complexity of $O(k^3)$ if a message of maximal length $q = k$ is embedded [19]. This can take up to several minutes of computation on a computer, which makes the system unusable. In order to improve the performances, the vector $\boldsymbol{b}$ is divided into disjoint subsets. If the unique

system of equations is replaced by β smaller systems that are solved independently, the complexity will be reduced from $O(k^3)$ to $O(k^3/\beta^2)$ [16].

With a single block for $\boldsymbol{b}$, the sender needs to communicate the size of the message to the recipient in order for him to be able to build correctly the matrix $\boldsymbol{D}$. This can be achieved for instance by allocating a small portion of the cover to hide the value q with a small uniform matrix $\boldsymbol{D}_0$ (not random). The stego key can be used to determine the fragment of cover in which q will be hidden. However, when using the faster alternative, each subset has a different number of changeable elements k_i and each message size q_i of each subset is communicated by appending its value to the message bitstream. These sizes will occupy a fixed number of bits that can be precomputed, but they will inevitably decrease the channel capacity by using a portion of the bitstream that could have been used for message bits. Only the size q_β of the last block needs to be hidden in a certain way to be communicated to the receiver, other q_i are extracted afterwards during the decoding phase. Choosing to split the image into several blocks to increase the speed of embedding at the expense of capacity is a crucial trade-off to provide a practical system.

In order for the encoder and the decoder to work jointly, both sender and receiver need to agree on several parameters r_1, r_2 and k_{avg} where $r_1 \leq r \leq r_2$ with $r = k/n$ being the rate, and k_{avg} the average number of changeable elements in each subset. The number of subsets β and their sizes n_i are computed in the same manner from both parties as follows:

$$\beta = \lceil n r_2 / k_{avg} \rceil \tag{4.7}$$

$$n_i \in \{\lfloor n/\beta \rfloor, \lceil n/\beta \rceil\} \quad , \quad n = n_1 + n_2 + \ldots + n_\beta \tag{4.8}$$

The value of k_{avg} is set to 250 as in [16]. As the decoder will have to gradually add rows in the matrix $\boldsymbol{D}$ depending on the size q_i, it is required from the encoder to build this matrix element per element or in a row by row manner. The content of $\boldsymbol{D}$ is dependent of the secret key.
If r changes substantially from an image to the other, r_2 may be too big for a given image, which causes the encoder to divide the image into too many subsets. The more subsets there are, the more q_i need to be added to the bitstream. This can be solved by communicating directly the rate r in the image in the same way as q_β.

The original encoding (Algorithm 1) and decoding (Algorithm 2) methods are shown in [16].

Algorithm 1 WPC Encoding algorithm [16]

E0. Calculate $\beta = \lceil nr_2/k_{avg} \rceil$. Using a PRNG, generate a random binary matrix $\boldsymbol{D}$ with $\lceil n/\beta \rceil$ columns and sufficiently many rows.

E1. Determine the header size $h = \lceil log_2(nr_2/\beta) \rceil + 1$ and $q = |\boldsymbol{m}| + \beta h$

E2. $\boldsymbol{b'} \leftarrow \boldsymbol{b}$, $i \leftarrow 1$

E3. $q_i = \lceil k_i(q+10)/k \rceil$, $q_i = \min\{q_i, 2^h - 1, |\boldsymbol{m}|\}$m $\boldsymbol{m}^{(i)} \leftarrow$ the next q_i bits in $\boldsymbol{m}$

E4. Select the first n_i columns and q_i rows from $\boldsymbol{D}$ and denote this submatrix $\boldsymbol{D}^{(i)}$. Solve q_i equations $\boldsymbol{H}^{(i)}\boldsymbol{v} = \boldsymbol{m}^{(i)} - \boldsymbol{D}^{(i)}\boldsymbol{b}^{(i)}$ for k_i unknowns $\boldsymbol{v}$, where $\boldsymbol{H}^{(i)}$ is a $q_i \times k_i$ submatrix of $\boldsymbol{D}^{(i)}$ consisting of those columns of $\boldsymbol{D}^{(i)}$ that correspond to changeable bits in $\boldsymbol{b}^{(i)}$. If this system does not have a solution, the encoder decreases q_i till a solution is found

E5. According to the solution $\boldsymbol{v}$, obtain the i-th segment $\boldsymbol{b'}^{(i)}$ of the vector $\boldsymbol{b'}$ by modifying or leaving $\boldsymbol{b}^{(i)}$ unchanged

E6. Binary encode q_i using h bits and append them to $\boldsymbol{m}$

E7. Remove the first q_i bits from $\boldsymbol{m}$

E8. $q \leftarrow q - q_i$, $k \leftarrow k - k_i$, $i \leftarrow i + 1$

E9. If $i < \beta$ go to step E3

E10. If $i = \beta$, $q_\beta \leftarrow q$

E11. Binary encode q_β using h bits and prepend to $\boldsymbol{m}$, $\boldsymbol{m}^\beta \leftarrow \boldsymbol{m}$

E12. Select the first n_β columns and q_β rows from $\boldsymbol{D}$ and denote this submatrix $\boldsymbol{m}^{(\beta)}$. Solve q_β equations $\boldsymbol{H}^{(\beta)}\boldsymbol{v} = \boldsymbol{m}^{(\beta)} - \boldsymbol{D}^{(\beta)}\boldsymbol{b}^{(\beta)}$ for k_β unknowns $\boldsymbol{v}$. If this system does not have a solution, exit and report failure to embed the message.

E13. According to the solution $\boldsymbol{v}$, obtain the β-th segment $\boldsymbol{b'}^{(\beta)}$ of the vector $\boldsymbol{b'}$ by modifying or leaving $\boldsymbol{b'}^{(\beta)}$ unchanged

Algorithm 2 WPC Decoding algorithm [16]

D0. Calculate $\beta = \lceil nr_2/k_{avg} \rceil$. Using a PRNG, generate a random binary matrix $\boldsymbol{D}$ with $\lceil n/\beta \rceil$ columns and sufficiently many rows

D1. Determine the header length $h = \lceil log_2(nr_2/\beta) \rceil + 1$

D2. $i \leftarrow \beta$

D3. $\boldsymbol{D} \leftarrow$ the first n_β columns of $\boldsymbol{D}$.

D4. $\boldsymbol{D}^{(\beta)} \leftarrow$ the first h rows of $\boldsymbol{D}$, read q_β as $\boldsymbol{D}^{(\beta)}\boldsymbol{b'}^{(\beta)}$

D5. $i \leftarrow i - 1$

D6. Decode q_i from the last h bits of $\boldsymbol{m}$ and remove the last h bits from $\boldsymbol{m}$

D7. Select the first n_i columns and q_i rows from $\boldsymbol{D}$ and denote this submatrix $\boldsymbol{D}^{(i)}$. $\boldsymbol{D} \leftarrow$ the first q_i rows of $\boldsymbol{D}^{(i)}$, prepend $\boldsymbol{D}\boldsymbol{b'}^{(i)}$ to $\boldsymbol{m}$, $\boldsymbol{m} \leftarrow \boldsymbol{D}\boldsymbol{b'}^{(i)} \& \boldsymbol{m}$

D8. If $i > 1$ go to step D5, else $\boldsymbol{m}$ is the extracted message

WPC in public-key steganography

Public-key steganography is a special domain of steganography. It can be used with most of existing steganographic techniques and consists in encrypting the message bitstream in order to make it look randomized [15]. Similarly to public-key cryptography, Alice and Bob will both have a pair of keys: the public key, as its name suggests, is available to any sender and is used to encrypt data while the private key is used by the recipient for decryption. When Alice uses a public selection channel, Eve can extract the hidden sequence but cannot make the difference with a random sequence if the cryptosystem is strong. However, having a public selection channel can be a starting point to mount an attack by observing and collecting information. This problem can be solved by using WPC in order to have random selection channels.

The WPC technique is an efficient steganographic tool that can be used with various schemes. Steganographic methods using this tool differ from previous techniques by the fact that the selection channel is arbitrary. Neither the sender nor the recipient can have the control of it and it is not shared between them. Because the key is not directly implied in the hiding, schemes that use WPC are not subject to brute force. More studies have been conducted in order to improve the embedding efficiency, particularly in [20].

4.3 Double compression perturbed quantization

The WPC technique requires from the user to have some side information that is only available to him. This can be achieved by using an information reducing operation at the sender side before the embedding, such as downsizing, lossy compression, decreasing of the color depth, color to grayscale conversion, dithering, etc.

This work focuses on wet paper codes using lossy compression, and more precisely JPEG compression. In section 2.3, the JPEG encoding process was briefly explained to understand how the loss of information occurs.
The following sections will cover the global principle of perturbed quantization introduced by Fridrich et al. [19] and its application with double JPEG compression.

4.3.1 Perturbed quantization

When the sender uses an image that will undergo an information-reducing process that ends with a quantization step, it is possible to slightly *perturb the rounding* of values in order to embed message bits. The perturbed quantization method uses an unprocessed raw image and the side-information available to the sender comes from the extra information that the raw image carries, compared to its compressed version. This additional information will be lost after the embedding, making it unavailable to other parties.

Let $\boldsymbol{X} \in \mathcal{Z}^N$ be the unprocessed cover image, usually called *precover*, where $\mathcal{Z}$ is the range of its pixels, coefficients, or colors, depending on the format. The information-reducing process F that is applied to the image is expressed as follows:

$$F = Q \circ T : \mathcal{Z}^N \to \mathcal{X}^n \tag{4.9}$$

where $\mathcal{X}$ is the integer dynamic range of the image after the transformation $\boldsymbol{Y} = F(\boldsymbol{X})$. The process is composed of a real-valued transformation $T : \mathcal{Z}^N \to \mathbb{R}$ and an integer quantizer $Q : \mathbb{R} \to \mathcal{X}^n$.

When JPEG compression is chosen as the information-reducing transformation, the sender must have the control on the last steps of the compression. With a raw image $\boldsymbol{X}$ of size n, he will start a JPEG compression and stop the process before the quantization step in order to obtain the DCT coefficients. During the next phase of compression, these n coefficients will be rounded to the closest integer. However, some of these DCT coefficients have a fractional part that is very close to 0.5 and can be rounded differently according to the values needed for syndrome coding.
JPEG compression is not reversible and information removed during quantization is permanently lost. Even if an attacker tries to make an estimation of the uncompressed version of the image, it will not be sufficient to confirm that the original image has been perturbed during compression [19].

Let $d_i,\ i = 1, \ldots, n$, be the DCT coefficients before rounding and D_i the coefficients after being rounded to integers. The sender identifies the k DCT coefficients $d_i,\ i \in \mathcal{C}$, that have a fractional part close to 0.5 within an interval given by a tolerance parameter ϵ such that $d_i - \lfloor d_i \rfloor \in [0.5 - \epsilon, 0.5 + \epsilon]$. These coefficients are the k changeable elements (dry pixels) that will be rounded up or down during embedding in order to satisfy the wet paper codes equation (4.2). The tolerance parameter should be set to a small value.
With k changeable elements identified, the sender may try to fully embed the image with a message of size $q = k$. As seen in section 4.2, this is achieved by solving equation (4.5) with q linearly independent linear combinations.

4.3.2 Double compression

Encountering a JPEG image that underwent more than one compression is not uncommon. The simple action of sending a picture by e-mail can reduce its quality for practical reasons and the user is rarely aware of these modifications. Several other operations can lead to modifications, thus the existence of a double compressed image should not be suspect.

Double compression consists in decompressing a JPEG image that has been compressed with a quality factor Q_1 and resaving it as JPEG once again but with another quality factor Q_2. This double compression perturbed quantization relies on a simple observation established by Fridrich et al. in [19]. The decompression of a JPEG image involves the reverse procedure of JPEG compression where, after entropy decoding, each block $\boldsymbol{b}^Q$ is dequantized (4.10), submitted to the IDCT (4.11) and finally rounded (4.12).

$$d_{i,j} = b^Q_{i,j} q_{i,j} \tag{4.10}$$

$$\boldsymbol{B}^{raw} = IDCT(\boldsymbol{d}) \tag{4.11}$$

$$\boldsymbol{B} = \begin{cases} 0, & \text{if } B^{raw}_{i,j} < 0 \\ round(B^{raw}_{i,j}), & \text{if } 0 \leq B^{raw}_{i,j} \leq 255 \\ 255, & \text{if } B^{raw}_{i,j} > 255 \end{cases} \tag{4.12}$$

Let $q^{(1)}_{i,j}$ and $q^{(2)}_{i,j}$ be respectively the elements from the first quantization matrix with quality factor Q_1 and the elements from the second quantization matrix with Q_2. After dequantization, it can be observed that DCT coefficients $d_{i,j}$ from all blocks are multiples of $q^{(1)}_{i,j}$. This is illustrated in the above paper with quality factors $Q_1 = 88$, $Q_2 = 76$ and the position $(i = 1, j = 2)$. Elements $d_{1,2}$ from each block will be multiples of $q^{(1)}_{1,2} = 3$. However, after the second DCT transform, the values will spread in peaks around multiples of 3.

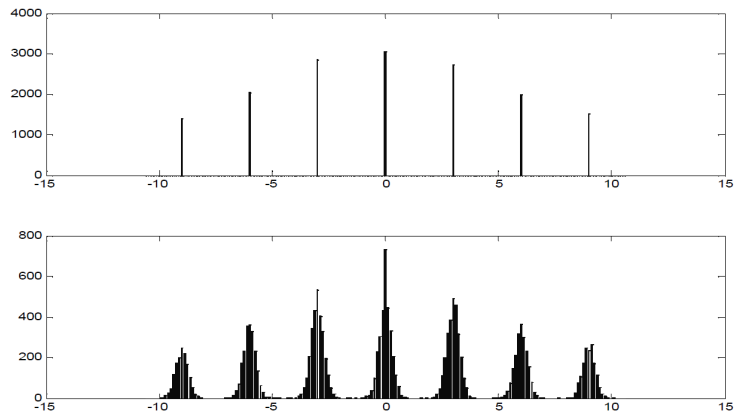


Figure 4.2: From [19]. Top: histogram of values of the DCT coefficient $d_{1,2}$ in the original 88% quality JPEG file. Bottom: histogram of the same DCT coefficient $b_{1,2}$ after decompressing the JPEG file to the spatial domain and DCT transforming.

Given that the second quantization step is $q^{(2)}_{1,2} = 6$, these peaks will be quantized differently according to their position. While peaks around even multiples $2k \times 3$, $k = 0, 1, \ldots$ are simply quantized to $6k$, the peaks that are around odd multiples $(2k+1) \times 3$ are first split in half, then the first half will be quantized to $6k+2$ and the second half to $6k+4$. This is where the perturbed quantization can operate: odd multiples can be set to one value or the other, depending on the result required by the parity function.

4.3.3 Selection rule

Considering the observations made in the previous section, it is now possible to define a selection rule [19]. This rule will choose the appropriate elements that can be safely used for embedding. Let $(q^{(1)}_{i,j}, q^{(2)}_{i,j})$ be a pair of quantization elements, the pair is called *contributing* if there exists integers k and l such that:

$$kq^{(1)}_{i,j} = lq^{(2)}_{i,k} + \frac{q^{(2)}_{i,k}}{2} \tag{4.13}$$

The integer k is a *contributing multiple* of $q^{(1)}_{i,j}$ and is situated exactly in the middle of the second quantization interval of length $q^{(2)}_{i,j}$. The integers $l, l+1$ are contributing multiples of $q^{(2)}_{i,j}$.

Theorem 4.1 (**Contributing pair**). [19] The pair $(q^{(1)}_{i,j}, q^{(2)}_{i,j})$ is contributing if and only if $q^{(2)}_{i,j}/g$ is even, where $g = GCD(q^{(1)}_{i,j}, q^{(2)}_{i,j})$ is the greatest common divisor of $q^{(1)}_{i,j}$ and $q^{(2)}_{i,j}$. Furthermore, all contributing multiples k of $q^{(1)}_{i,j}$ are expressed by the formula

$$k = (2m+1)\frac{q^{(2)}_{i,j}}{2g}, m = \ldots, -2, -1, 0, 1, 2, \ldots \tag{4.14}$$

Detailed proof can be found in [19].

The contributing pairs are computed for a pair of quantization tables with different quality factors, and since the same QT is applied to each block of the image during compression, the contributing elements of the image will always be at the same positions in all blocks, which produces visual patterns. This can be easily illustrated by showing the DCT coefficients that were modified in each block of a stego image obtained with the scheme implemented for this work.

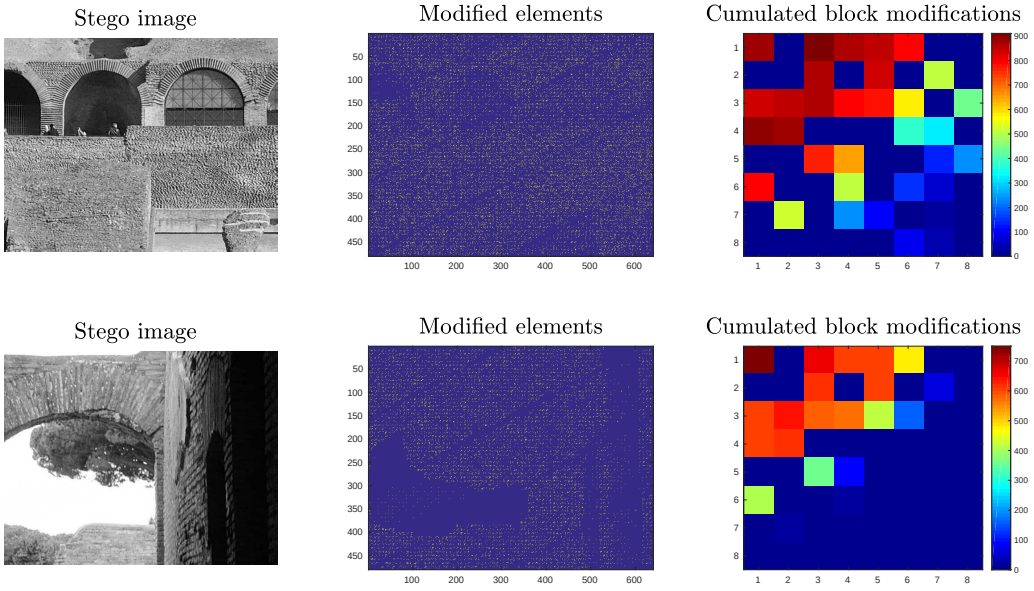


Figure 4.3: Left: stego images. Center: modified elements are shown in light color while untouched elements are in dark color. Right: addition of all block modifications, where dark red is the location with the most changed elements.

Not all blocks have a contributing element for each contributing pair and some regions are too flat to hold discrete modifications. This can be observed in Figure 4.3 with the second image: both all white and all black areas are not modified.

4.3.4 JPEG quality factors

The quality factor is used to control the level of JPEG compression: the smaller it is, the more the image is compressed and shows residual blockiness. It is used in combination with the quantization table (2.3) to generate the quantization matrix that will be used for the current compression. The table generation is computed by the Independent JPEG Group [24] in the following manner:

$$S = \begin{cases} \frac{5000}{Q}, & \text{if } Q < 50 \\ 200 - 2Q, & \text{otherwise} \end{cases} \tag{4.15}$$

$$q_{i,j} = \frac{q_{i,j}^{basic} \cdot S + 50}{100} \tag{4.16}$$

Where $\boldsymbol{q}^{basic}$ is the quantization table given at (2.3). When the table contains large values, the division during the quantization step (2.4) reduces greatly the image quality.

In double compression embedding, it is imperative for the two quality factors to respect the rule $Q_1 < Q_2$. The principle of WPC, as stated before, relies fundamentally on the loss of information when generating the stego image from the cover image. The purpose of the second compression is to lose the material that allowed the creation of the stego, therefore, this second compression must use an inferior quality factor. Some works not only detect the presence of double compression but are also able to estimate the first quantization table [22], therefore, the choice of a Q_2 larger than Q_1 will indubitably put the system in jeopardy.

In this work, the quality factors were set to $Q_1 = 85$ and $Q_2 = 70$ in order to compare the results with the experiments from [19, 20]. These values are not chosen arbitrarily but come from the observation that best embedding capacity is obtained when $Q_2 = 2(Q_1 - 50)$.

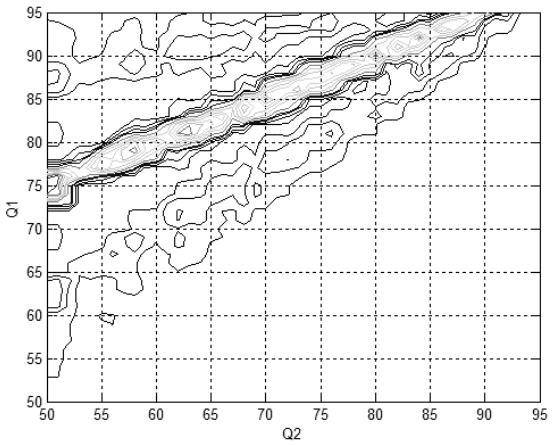


Figure 4.4: From [19]. Embedding capacity according to Q_1 and Q_2.

Chapter 5

Proposed method

The WPC methodology using double compression PQ offers great channel capacity along with low detection rates [19]. The attacker's task is made difficult because of the non-fixed selection channel, and even if the presence of steganography is detected, it is hard to identify the DCT coefficients that have been incorrectly quantized. This is guaranteed by the fact that the loss of side information cannot be reversed neither at the attacker side nor at the recipient side.

The work of this thesis relies on the double compression PQ by adding a processing step to the messages involving an XOR and improving the selection of contributing elements.

5.1 Selection rule for more robust embedding

The selection rule determines which elements from the image are suitable for secure embedding. In general, the communicating parties want to exchange messages that are as small as possible to avoid the risk of revealing the system, but sufficiently large to convey clear information. The WPC encoding algorithm makes use of all the contributing elements that are found when solving the linear equations systems (4.2).

In order to minimize the impact of embedding, the number of changeable elements that are selected can be reduced as mentioned in [19]. When perturbing the quantization of a real value, the error resulting from the process will be smaller if the fractional part of the original value was very close to 0.5. For instance, if the value 15.49 is rounded to 16 instead of 15, the error is very minor. However, rounding 15.9 to 15 is risky because the possible outcome of rounding 15.49 is more unpredictable than the outcome of rounding 15.9. Hence, for all changeable elements x_i, $i \in \mathcal{C}$, a function $f(x_i)$ that computes a numerical value is defined. This function will assign to each element a quantity that informs about the likeliness of a value to be chosen in priority in its block. More precisely, it will compute the closeness of the fractional

part to 0.5 in the following manner:

$$f(x_i) = |0.5 - frac(x_i)| \tag{5.1}$$

$$frac(x_i) = |x_i - \lfloor x_i \rfloor| \tag{5.2}$$

Then during embedding, for each one of the β blocks, only a subset of contributing x_i is chosen according to the size of the sub-message q_i. As stated previously, a matrix of size $(k - r) \times k$ has a solution as long as its rank is bigger or equal than $k - r$. The probability for this matrix of having such a rank is given by (4.6) and it can be observed on the corresponding Figure 4.1 that for $r > 10$, this probability is very close to 1. Under these circumstances, if $k_i - q_i > \delta$, the encoder reduces its value to $k_i = q_i + 10$ and only the k_i elements with the smallest $f(x_i)$ values are used for embedding. The parameter δ is set to 15 in this work.

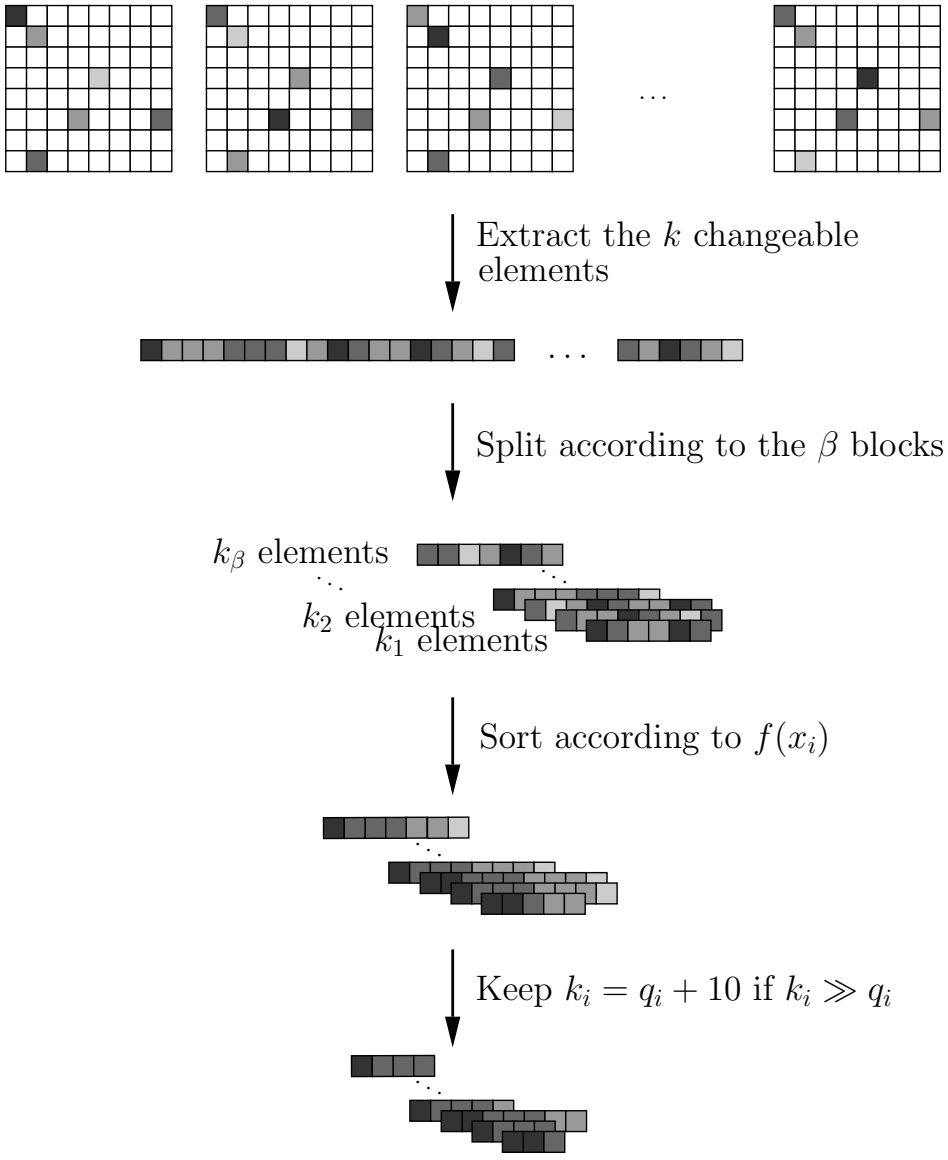


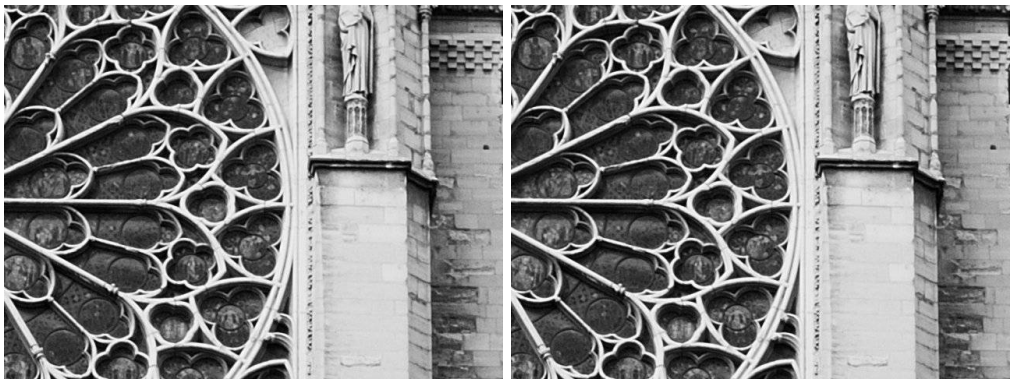
Figure 5.1: Selection rule with $f(x_i)$.

This further selection of elements reduces the size of the matrix $\boldsymbol{D}$ and the main impact of this shrinkage is the increase of speed in the solving of linear systems, especially if the user sends a message which length is very small compared to the total number of changeable elements $k \gg q$. To illustrate the differences in embedding speed, messages of 100, 500, and 1000 characters are embedded in an image holding a maximum of $k = 36'631$ changeable elements. The results are the average values on 10 runs.

	Time in seconds	
	Classical selection	Proposed selection
100 char	2.0049	0.3396
500 char	7.9273	1.4671
1000 char	20.3045	5.3138

Table 5.1: Comparison of embedding time when using the classical selection rule and the proposed one.

The speed-up of the embedding process offers a more practical setup for the sender. Large computation time hinders all usages and should be considered when building a stegosystem.



(a) Cover image. (b) Stego image.

Figure 5.2: Pair of cover and stego objects after the embedding of a message of 1000 characters.

The additional rule (5.1) is not the only function that can be computed, other parameters can be considered such as the location of a changeable element in the image or its neighboring elements. The encoder can combine the perturbed quantization with adaptive selection channels by selecting first the changeable coefficients, then, for instance, keeping only the elements that are located in textured regions. It is also possible to choose to not embed in DC coefficients.

5.2 Processing of messages

The English language, like any other language, is not random. Many known patterns can be found in words and letter frequency analysis has been well studied in the field of cryptanalysis, leading to effective attacks [40]. Therefore, for security reasons, the message should not be embedded without any preprocessing.

The most simple solution is to randomly permute the elements of the message $\boldsymbol{m}$ with the use of a secret key. It is also possible to encrypt the content using a symmetric-key cryptographic algorithm such as AES. This setup requires the use of two keys, one for the stegosystem to generate the matrix $\boldsymbol{D}$ and one for the cryptographic system. Using a single key is tempting but if it is compromised, all messages can be decrypted, whereas with two different keys, if one of them is compromised, the attacker has still some work to do. Finally, if the communication takes place between only two parties, one can consider the use of a public-key cryptographic scheme in addition to the WPC as stated at the end of section 4.2. However, the existence of a public key can be suspect in the eyes of an attacker. Moreover, if the sender needs to broadcast the image to multiple recipients, a different stego image has to be generated for each person. With a symmetric-key algorithm only one key and one embedding is required. If the sender transmits the same message in the same image but using different public keys, the stegosystem is immediately exposed, as discussed in section 6.2.

In this work, a simple scheme is proposed in order to give a randomized aspect to the message. With current technologies and the growth of social networking services, sharing images has never been so easy and most of platforms allow the user to add a description along with the image. Using a simple operation as the XOR, the secret message can be in a certain way randomized before embedding. For each image, the sender needs a ciphertext m_c along with the secret message m_p that is designated as the plaintext. First, both messages m_c and m_p are converted to binary sequences. Then, a XOR is applied and the resulting sequence m is hidden using the WPC scheme while the ciphertext is communicated with the image via the description field.

Let $m_p = \{m_{p_1}, m_{p_2}, \ldots, m_{p_q}\}$ be the secret message of size q and m_c a public message of size r free of secret information defined as $m_c = \{m_{c_1}, m_{c_2}, \ldots, m_{c_r}\}$ such that $q \leq r$. The message m that will be embedded in the image is given by:

$$m_i = m_{p_i} \oplus m_{c_i} \quad \text{with} \quad 1 \leq i \leq q \tag{5.3}$$

This additional processing of the message does not require a key but only a ciphertext. The sequence m_c can be easily transmitted with the image without raising any suspicion, especially with current social medias. The users only need the secret key that generates the matrix $\boldsymbol{D}$. The sender can target different persons by sharing different keys with some groups of users.

Without further complexity, it will be possible to perform tests on the PQ scheme with the increased selection of elements and non-random messages.

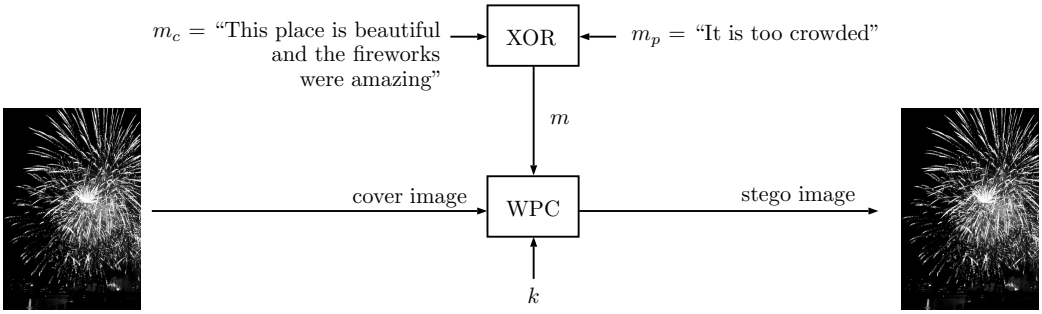


Figure 5.3: Using the XOR with m_c and m_p.

Of course, this additional processing does not provide perfect secrecy or additional security, given the fact that m_c is publicly available. If an attacker can reveal the sequence m, he can read the message m_p. Even if the ciphertext m_c is not communicated with the image but using another communication medium, the security of the system does not improve because it is always considered that the attacker knows everything except the key (Kerckhoffs' principle). The security depends on the ability of the stegosystem to stay undetected, hence the additional selection rule.

In the event that attacks are performed by a malicious entity that is able to change a secret message without the parties being aware of the modifications, it is possible to add an additional step to the processing of messages. With a compromise on the embedding capacity, the sender can decide to append to the message the result of a cryptographic keyed hash function. This deterministic function acts like the fingerprint of the data and produces for each unique message a unique hash value called *digest* [40]. This function requires a supplementary key for the sender and the recipient(s) but guarantees the integrity of messages as long as the attacker is not in possession of the key. If a message is altered, the recipient will discover that the digest computed from the message that he received is not identical to the one in the image. Nevertheless, the use of this additional step can be arguable. Back to the prisoners setup, if the warden Eve is malicious, she is in possession of the secret key that generates the matrix $\boldsymbol{D}$. She obtained it in one way or another from one of the prisoners, using for instance rubber-hose cryptanalysis. Thus, it cannot be excluded that she also obtained the other keys.

5.3 Requirements and practical limitations

Jpeg library The C++ program uses the *libjpeg* library, which is a widely used C library developed by the Independent JPEG Group [24]. The library supports the manipulation of JPEG images and more precisely it allows the extraction of DCT coefficients, which is mandatory in the PQ scheme.

Image requirements For the sake of simplicity, the method that has been implemented runs only with grayscale images. However, all the theory described in this work can also be applied in the case of color images and the application

can be extended to the color scenario.
Furthermore, the images must be compressed to JPEG with a quality factor of 85 and must have a size multiple of 8. The fulfillment of these conditions is left to the responsibility of the user.
Although the acquisition of raw images requires compatible hardware and software, recent smartphones tend to support this image format and the use of expensive cameras is no more needed. Moreover, many free tools are available to convert raw or tiff images to JPEG with the desired quality.

ASCII limitations The application requires messages that are encoded using the ASCII character encoding standard. Each character is represented using 7 bits. Neither plaintexts nor ciphertexts can use special characters that don't appear in the ASCII standard.

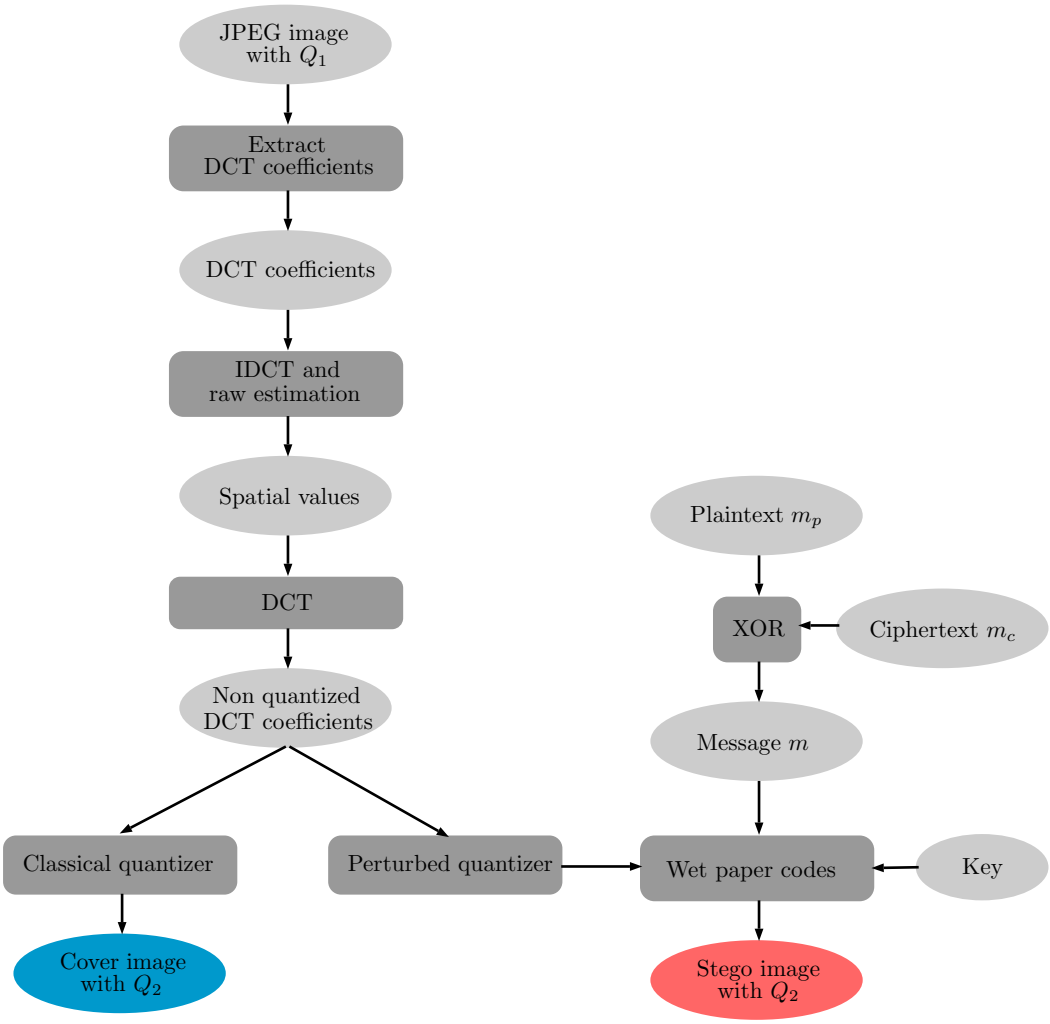


Figure 5.4: Global view of the embedding system.

Algorithm 3 WPC Encoding with additional $f(x_i)$

$\boldsymbol{m} \leftarrow preprocess(message)$
$\beta \leftarrow \lceil nr_2/k_{avg} \rceil$
$h \leftarrow \lceil log_2(nr_2/\beta) \rceil + 1$
$q \leftarrow |\boldsymbol{m}| + \beta h$
$\gamma \leftarrow 10$
$\boldsymbol{D} \leftarrow$ Using key, generate matrix with sufficiently many rows and $\lceil n/\beta \rceil$ columns
$\boldsymbol{\Delta} \leftarrow f(\boldsymbol{b})$ (5.1)
for $i = 0, i < \beta, i++$ **do**
  $\boldsymbol{b}^{(i)} \leftarrow$ take next n_i bits from $\boldsymbol{b}$
  $k_i \leftarrow count\_contributing\_elements(\boldsymbol{b}^{(i)})$ (4.14)
  if $i < \beta - 1$ **then**
    $q_i \leftarrow \min(\lceil k_i(q+10)/k \rceil, 2^h - 1, |\boldsymbol{m}|)$
  else
    $q_i \leftarrow q$
  end if
  $[index, \boldsymbol{\Delta}^{(i)}_{asc}] \leftarrow sort\_ascending(\boldsymbol{\Delta}^{(i)})$
  Require: $\delta > \gamma$
  if $k_i - q_i > \delta$ **then**
    $k_i^{old} \leftarrow k_i$
    $k_i \leftarrow q_i + \gamma$
    Set elements of $\boldsymbol{b}^{(i)}[index(k_i \rightarrow end)]$ to non contributing multiples
  end if
  if $i < \beta - 1$ **then**
    $\boldsymbol{m^{(i)}} \leftarrow$ next q_i bits of $\boldsymbol{m}$
  else
    Prepend q_β using h bits to $\boldsymbol{m}$
    $\boldsymbol{m^{(\beta)}} \leftarrow \boldsymbol{m}$
  end if
  $\boldsymbol{D}^{(i)} \leftarrow$ submatrix $\boldsymbol{D}[0 : q_i - 1, 0 : n_i - 1]$
  while $\boldsymbol{v}^{(i)}$ is empty **and** q_i was not decremented more than γ times **do**
    $\boldsymbol{H^{(i)}} \leftarrow$ submatrix of $\boldsymbol{D}^{(i)}$ with q_i rows and the k_i columns at positions of changeable elements of $\boldsymbol{b}$
    $\boldsymbol{v}^{(i)} \leftarrow solve\_in\_GF2(\boldsymbol{H^{(i)}v^{(i)} = m^{(i)} - D^{(i)}b^{(i)}})$ (4.5)
    if $\boldsymbol{v}^{(i)}$ is empty **and** $i < \beta - 1$ **then**
      $q_i \leftarrow q_i - 1$
    else
      Exit with failure
    end if
  end while
  $\boldsymbol{b'^{(i)}} \leftarrow \boldsymbol{v}^{(i)} + \boldsymbol{b}^{(i)}$
  Append q_i using h bits to $\boldsymbol{m}$
  Remove $\boldsymbol{m}^{(i)}$ from $\boldsymbol{m}$
  $q \leftarrow q - q_i$
  $k \leftarrow k - k_i$
end for

Chapter 6

Steganalysis

The modified WPC steganographic scheme proposed in the previous chapter is tested in order to evaluate its level of security. This chapter contains the results of all experiments along with some observations.

First, in order to reduce the complexity, features are extracted from each cover image and its corresponding stego image. The features that were chosen for this work of steganlaysis are described in section 6.3. Then, all feature vectors are fed in pairs to an ensemble classifier, which principle has been described in section 3.2.4. Finally, the performances are compared using ROC curves and their detection accuracy ρ, which have been defined in section 3.2.3.

6.1 Data acquisition

6.1.1 Image dataset

The steganographic system requires as input only images that have been compressed to JPEG with quality factor $Q_1 = 85$. Since JPEG images commonly found on the web are compressed with a lower quality factor, a database has been built from a set of raw images. The RAISE image dataset [13] is a collection of high-resolution raw images guaranteed to be uncompressed and never processed. The images capture various scenes such as nature scenery, buildings, indoors, close-ups, etc.

Each image was cropped to the size 640×480 and converted to grayscale. The choice of cropping instead of resizing has been made to minimize the number of modifications. Then, the images were compressed with JPEG quality factor 85. Finally, all images that looked too empty were discarded: the RAISE image database contains very large images and the cropping can produce a patch of sky, snow, or sea, which is too flat.

6.1.2 Plaintext and ciphertext datasets

One of the motivations of this work is to use the wet paper codes stegosystem jointly with perturbed quantization in real conditions which means embedding non-random bitstreams. The following experiments are done with English sentences. With the proposed processing of messages, each image embedding requires a pair of plaintext and ciphertext. Moreover, for a given capacity of embedding, each image has a different number of changeable elements, thus the texts need to be adapted for every embedding. For a testing setup of 1000 images, a set of 2000 different texts is needed. This large dataset is built by appending the summaries of random articles picked on Wikipedia.

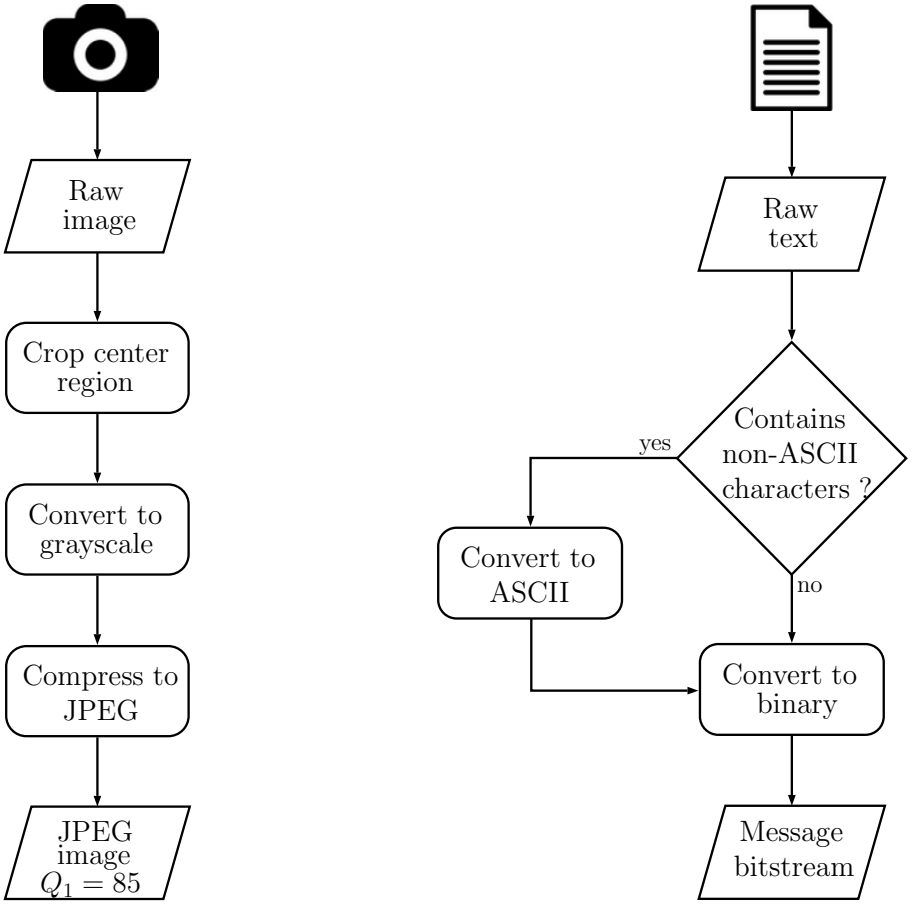


Figure 6.1: Image and message preprocessing before embedding.

6.2 Key considerations

When communicating with the use of the WPC scheme, three elements are involved: the image, the shared key and the secret message. It is possible for the concerned parties to change each element for every new communication or, for the sake of simplicity, to keep the same parameters for all messages. The second option

is undoubtedly very dangerous for the security of the stegosystem and should be discouraged. Each and every different secret message shall be concealed within a different image.

Keeping the same image If the sender decides to use more than once the same image for different messages, he exposes the stegosystem to attacks. An attacker will not only easily detect the presence of covert messages, but he will also be able to know which DCT elements were modified. With this information, learning which Q_1 and Q_2 where used is straightforward: the stegosystem is public and so is the computation of contributing pairs. Testing all pairs of quality factors to know which ones produces these positions is a fast operation. As a result, the attacker knows where the message bits are hidden in subsequent stego images and can now focus on trying to extract the message.

Keeping the same key In order for the stegosystem to be practical, the sender and the recipient share only one secret key. Communicating a secret key is done using another media and/or a cryptographic technique. If the key must be different for each embedding, the stegosystem will be dependent on the exchange of keys and will become troublesome to use.
The detection of communications using the same key is evaluated with an experiment in the next section.

Key size When the security of a system is investigated, the first element that is considered is the amount of information available at the attacker side; this is the case in cryptography as well as in steganography. The whole WPC stegosystem, including the algorithm, is known by all. Therefore, an attacker is able to intercept an image that is suspected of enclosing a covert message and then to compute the value β. With the value of β, it is possible to compute the size of each block as well as the number of bits h used to contain the last header q_β and this is achievable because n is obtained from the image and the parameters r_2 and k_{AVG} are publicly known.

The first step of decoding consists in getting the header from the last block, which position and size are known by the attacker. In the event that the size of the secret key is small, the attacker could achieve a simple brute-force instead of building a detector that needs training and therefore a large dataset. With a key size of $|k|$, there are $2^{|k|}$ different keys to try. Furthermore, as pointed out in section 4.2, extracting a message does not require complex computations and is a fast operation. If $|k|$ is small, the attacker may first try to brute-force the key to decode the whole message. He will skip a brute-force attempt if the elements uncovered don't correspond to any character encoding and pass to the next key.

Under those circumstances, the system that is built for this work does not accept keys that hold less than 128 bits. A key is stored in a simple text file that is passed as parameter to the program along with the other elements (plaintext, ciphertext, original image and capacity).

6.3 CC-PEV features

As the use of supervised learning is increasing in the field of blind steganalysis, new features are developed in order to increase the detection accuracy. These features are obtained from the images and can be spatial or extracted from the frequency domain. In the case of JPEG steganalysis, features from the DCT coefficients seem to offer more precise results [36].

The CC-PEV features consist in a merging of PEV-features with their Cartesian calibration. PEV features are composed of a first set of 23 DCT features and a reduced set of Markov features. The resulting feature vector PEV-274 has, as its name indicates, a dimensionality of 274 with 81 Markov features and 193 DCT features [36]. Calibration aims to estimate the histogram of the cover image from the stego image. The procedure consists in decompressing the JPEG stego image to the spatial domain using the IDCT, then, recompressing it with the same quality after cropping 4 pixels in both direction [27].

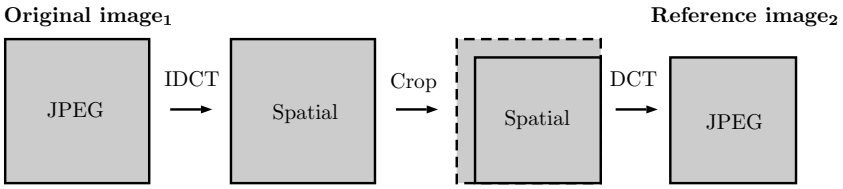


Figure 6.2: Calibration process: each calibrated feature $\boldsymbol{F}$ is obtained with $\boldsymbol{F}(\mathbf{image_2}) - \boldsymbol{F}(\mathbf{image_1})$ [27].

The calibration process is known to improve steganalysis and is used in this case with PEV-274 to double the number of features: the final CC-PEV vector contains 548 features.

6.4 Embedding rate

The embedding rate is expressed in terms of bits per non-zero DCT coefficient of the stego image (bpc) [19].

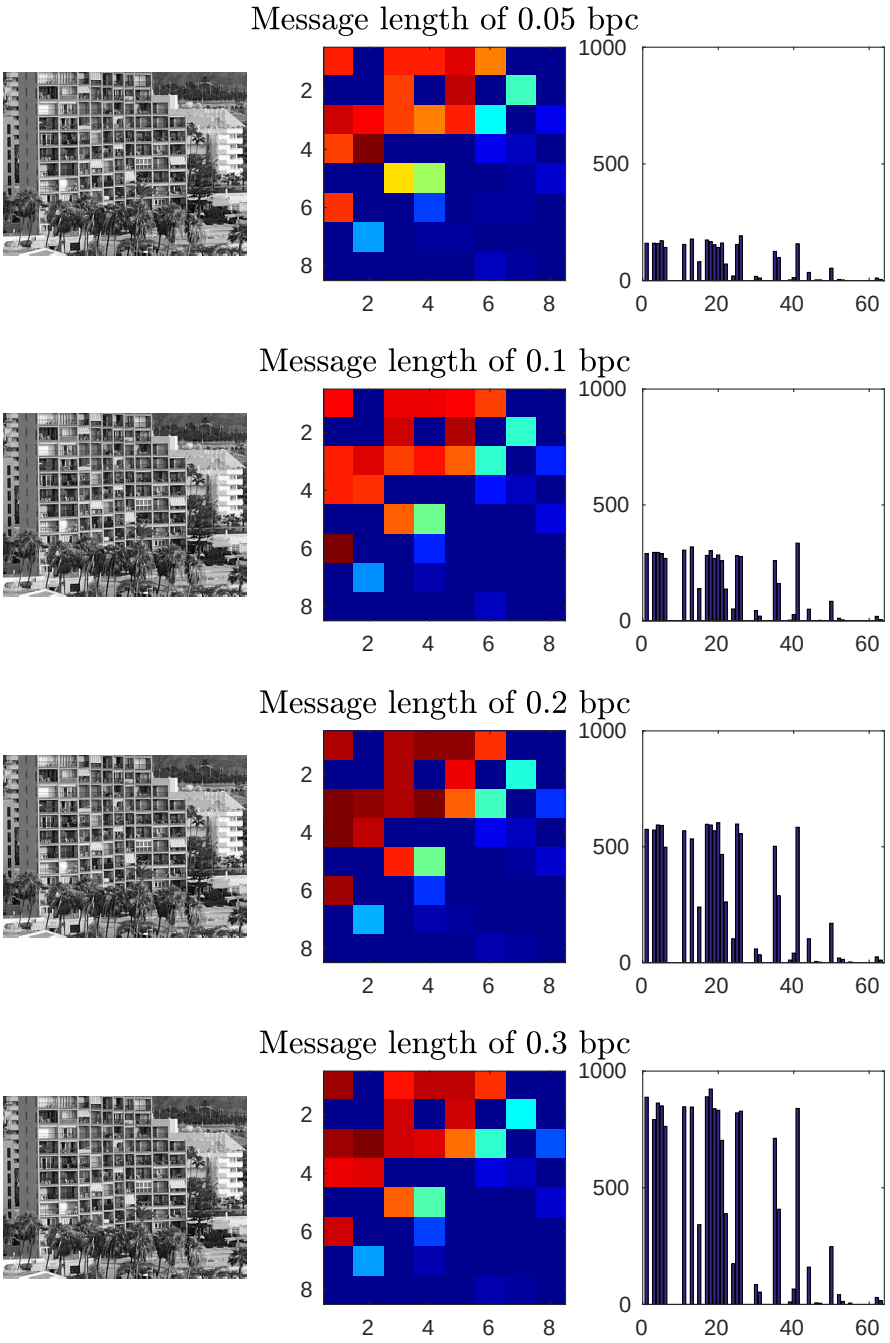


Figure 6.3: Embedding different message lengths. Left: the stego image. Center: positions of modified DCT coefficients of all 8×8 blocks. Locations with the most modifications are dark red while locations without modifications are in dark blue. Right: histogram of the number of elements changed per location in a 8×8 block.

6.5 Detection results

In this section, the modified WPC embedding scheme is evaluated using the ensemble classifier. This is done in three steps. First, the embedding of a considerable number of messages. Images from the large RAISE dataset are transformed as described in section 6.1.1. For each image embedding, the message size is computed for the corresponding bpc and a pair of plaintext and ciphertext is downloaded and saved after being processed as described in section 6.1.2. With these elements, a message is hidden in each image. The second step consists in extracting the CC-PEV features from the resulting cover and stego images in order to work on a smaller feature space. Finally, the set of all features is separated into two sets of the same size: the training set and the testing set. The training set is used to train the ensemble classifier described in section 3.2.4 and then the testing set is submitted to the classifier that will decide for each image if it is innocuous or carrying a covert message.

The purpose of the first set of experiments is to determine if the message preprocessing involving the XOR increases the detection rate. Two datasets are evaluated: 1000 pairs of cover and stego images containing messages from XORed English plaintexts and ciphertexts, and 1000 pairs of cover and stego images containing random bitstreams as in the experiments of [19]. The test is done first with 0.05 bpc, then with 0.1 bpc.

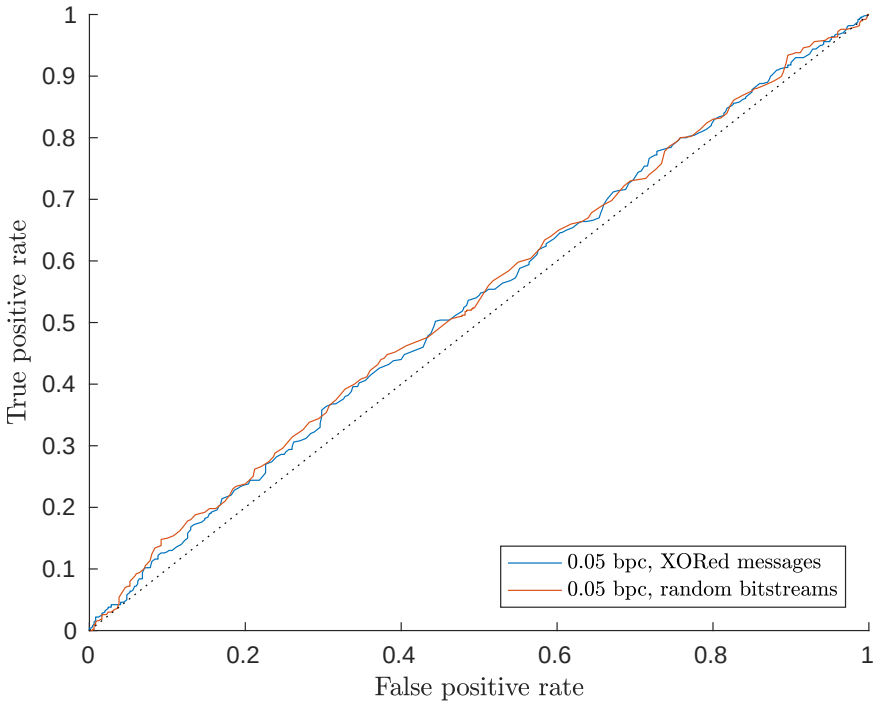


Figure 6.4: ROC curves of images embedded with XORed messages and random bitstreams with embedding rate of 0.05 bpc.

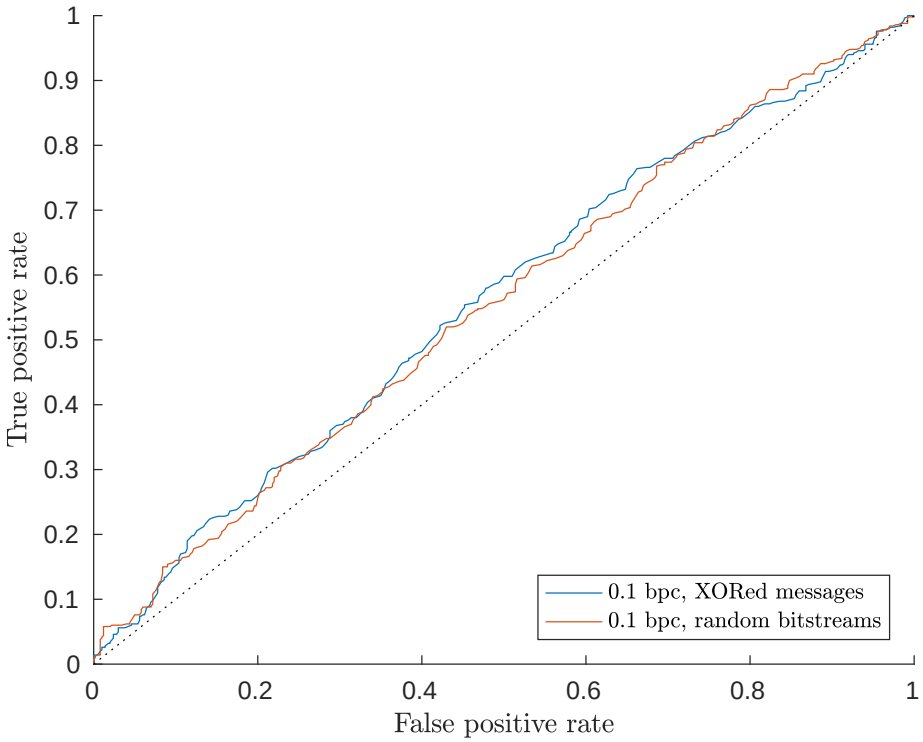


Figure 6.5: ROC curves of images embedded with XORed messages and random bitstreams with embedding rate of 0.1 bpc.

	ρ	
	XORed messages	Random bitstreams
0.05 bpc	0.0646	0.0738
0.1 bpc	0.1227	0.1092

Table 6.1: Detection reliability ρ for XORed messages and random bitstreams.

It can be observed that with 0.05 bpc, the measure of detectability for the system using the XOR shows a lower detection with 0.0646, while the system embedding random bitstreams scores higher with 0.0738. This observation is not valid in the case of 0.1 bpc, where images with random messages are less detected. However, the values of ρ can vary in a close range because at each classification, the ensemble classifier builds randomly the testing and training sets.

Next, the security of the modified version of the selection rule needs to be evaluated. Two series of tests are done: a first embedding of 1000 images with random messages and keys with the proposed selection rule, and a second series of 1000 embeddings using the classical selection rule where the WPC algorithm uses all available changeable elements.

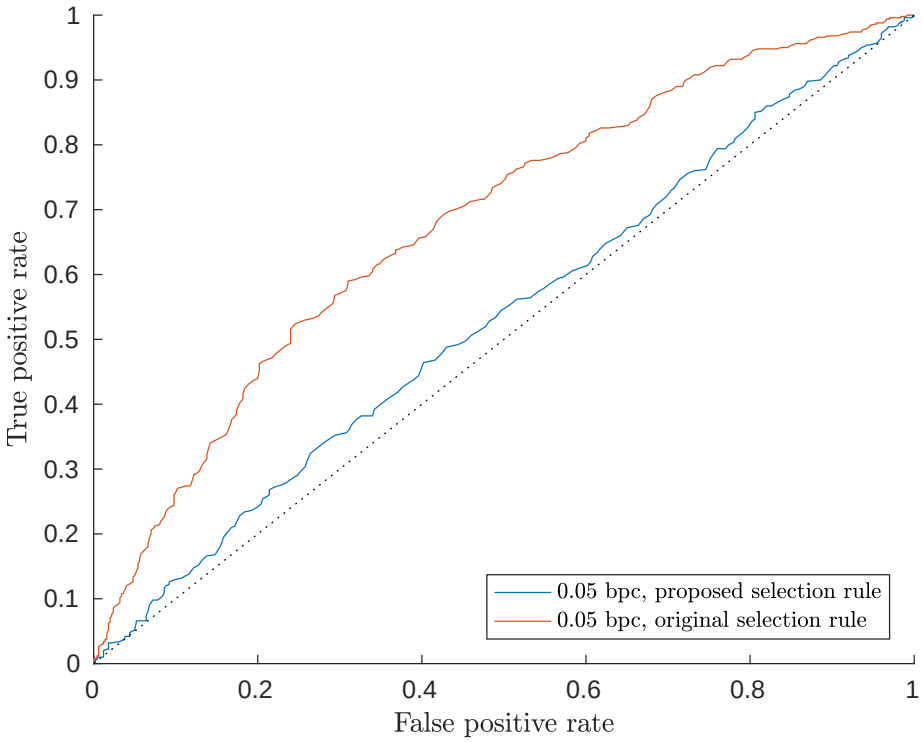


Figure 6.6: ROC curves of images embedded with the proposed selection rule and with the original selection rule.

	ρ	
	Proposed selection	Original selection
0.05 bpc	0.0624	0.3582

Table 6.2: Detection reliability ρ for original and proposed selection rules.

It can be observed that the system using the original selection rule has an increase of approximately 80% in the detection reliability, which means that the proposed system is less detectable.

A third series of tests is done in order to evaluate the proposed system with various message lengths. The stegosystem uses all elements explained in the previous chapter which include XORed messages and the selection rule using the additional $f(x_i)$. Once again, each test is a series of successful embedding of 1000 images for each bpc parameter. In all tests conducted, every embedding is verified such that if the decoding of the stego image does not provide the original covert message, the embedding is done again.

It is first expected by the stegosystem to be as much undetectable as the original one introduced in the work of [19]. Secondly, like any other stegosystem, it should be more vulnerable and more easily detected when the number of changes in the DCT coefficients increases due to longer messages.

bpc	0.005	0.1	0.2	0.3
$\boldsymbol{\rho}$	0.0648	0.1190	0.1941	0.2581

Table 6.3: Detection reliability ρ for PQ with the proposed selection and using XORed messages.

As expected, the probability that the classifier correctly identifies stego images and innocuous images depends on the length in bits of the secret message; stego images are more easily identified when the embedding rate is high.

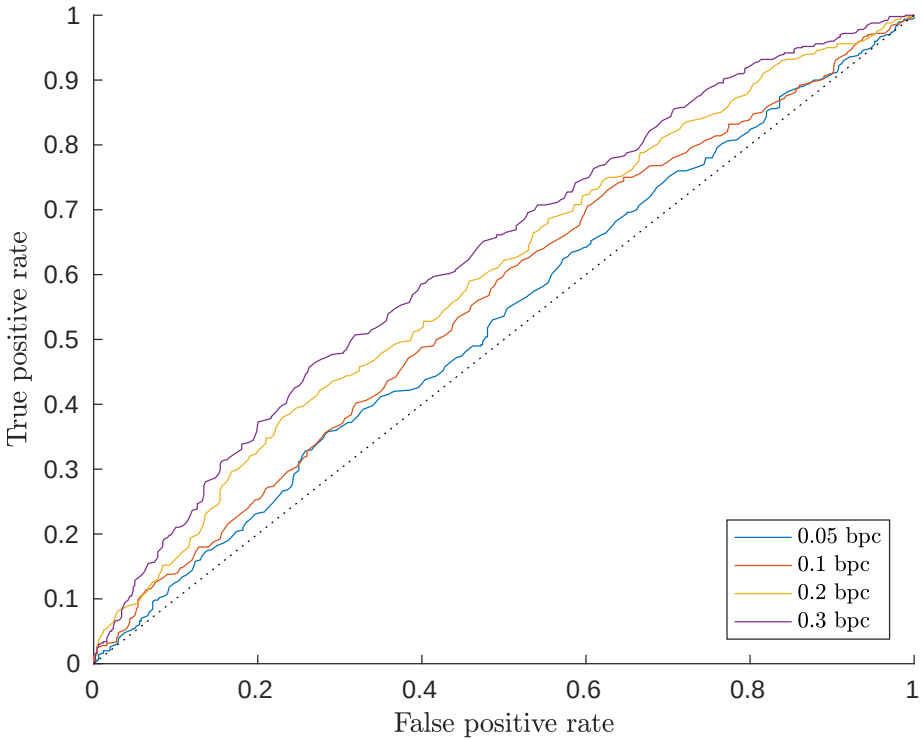


Figure 6.7: ROC curves of images embedded with XORed messages using $Q_1 = 85$ and $Q_2 = 70$ with the proposed selection rule.

In a practical scenario, the sender and the recipient(s) use a single secret key for all communications. The impact of this choice is illustrated with two series of embedding: a first set of different messages is embedded in different images using a unique key for each embedding and then the same scenario is performed but without changing the key.

	ρ	
	Different keys	Same key
0.05 bpc	0.0662	0.0724
0.1 bpc	0.1191	0.1328

Table 6.4: Detection reliability ρ when changing the key parameter with embedding rates of 0.05 and 0.1 bpc.

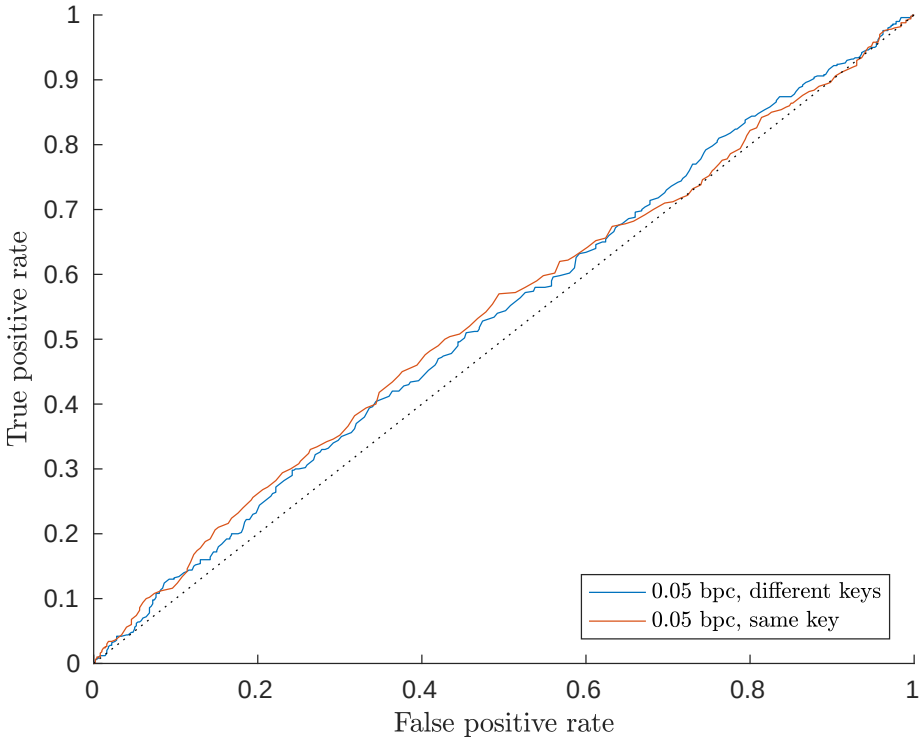


Figure 6.8: ROC curves of images embedded using different then identical keys.

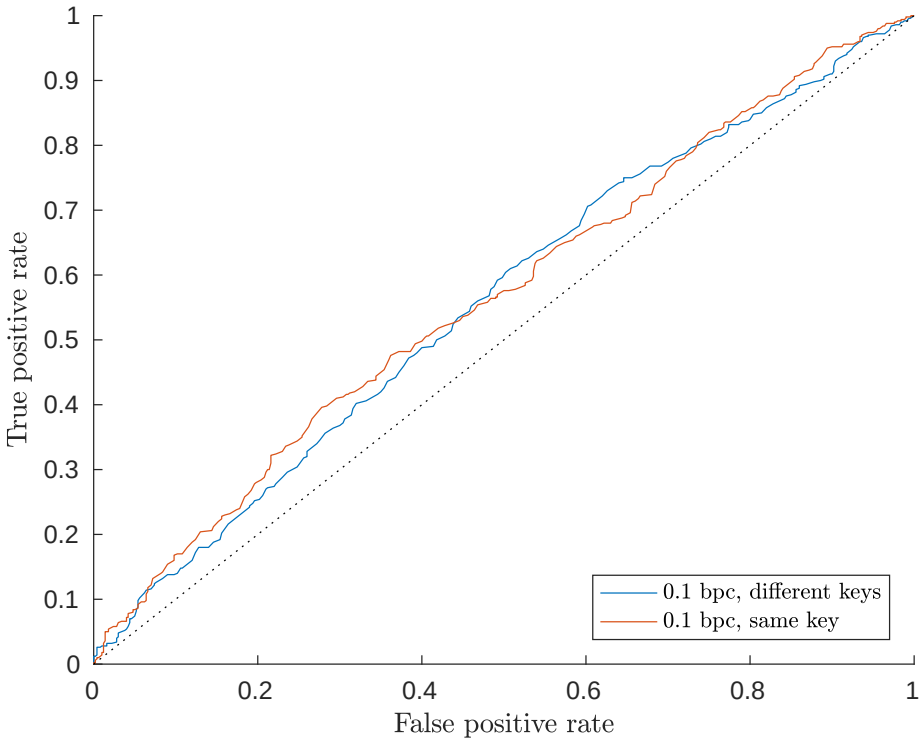


Figure 6.9: ROC curves of images embedded using different then identical keys.

Finally, all tests are grouped into a single figure with messages length of 0.05 bpc. For formal comparison, the ρ values are also reported below.

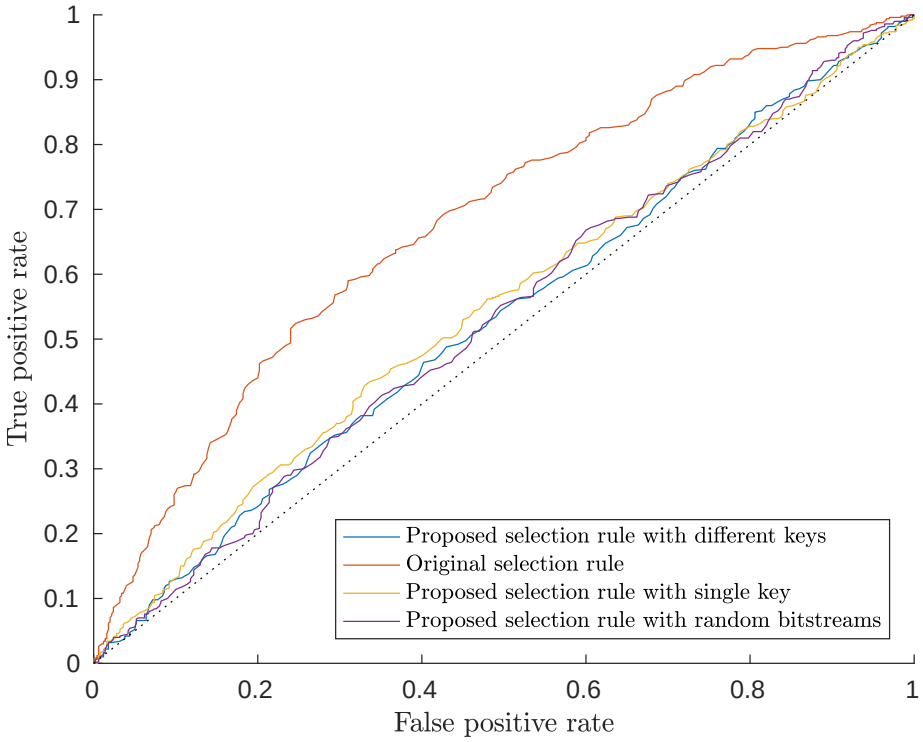


Figure 6.10: ROC curves for 0.05 bpc.

	ρ			
	Proposed selection	Original selection	Unique key	Random messages
0.05 bpc	0.0624	0.3583	0.0912	0.0654

Table 6.5: Detection reliability ρ for 0.05 bpc.

6.6 Results analysis

The first two Figures 6.4 and 6.5 reveal the favorable evidence that for the embedding rates tested, the ROC curves of messages that were XORed beforehand are close to the curves of messages consisting in random bits. It is important to recall that a classification with poor separation of the cover and stego objects yields a ROC curve close to the diagonal; these detectors are as bad as random guessing. Moreover, the blue curve crosses at multiple points the red curve, which makes the evaluation complicated. Therefore, for accurate decision making, using the reliability of detection ρ is preferred. In the figures cited above, the closeness of the two ROC curves is shown by comparing the reliability of each detection. As it can be seen in Table 6.1, for a given bpc, there is no major variation in the ρ values of the two schemes. Hence, introducing this randomization by XOR of the data does not increase the detection rate, especially when considering that the results of the ensemble classifier are prone to small variations because of the dynamic separation of the training dataset.

After reviewing the effects of the XOR on the detectability, the impact of the proposed selection rule must be determined. The embedding of messages of 0.05 bpc was run using the classical selection rule from the core paper [19] and then the proposed improved selection. The resulting ROC curves in Figure 6.6 show a large disparity in the performances of detection. With a detection reliability of $\rho = 0.3582$, the system using the original selection rule can be considered unsafe, providing a detectable communication channel. On the contrary, the system using the proposed selection rule is less detectable with the ensemble classifier achieving $\rho = 0.0624$. These results demonstrate that systems using the additional selection rule based on the function $f(x_i)$ grant more security.

With these results in mind, the next step was to examine the impact of the embedding of different message lengths on the detection. As it can be seen in Figure 6.7, the ROC curves evolve as expected and the number of stego images that are correctly detected increases with the message length.
When the message length is increased, the curve's distance to the diagonal increases too, meaning that the detector performs better. More formally, the Table 6.3 indicates that for bpc from 0.05 to 0.3, the reliability of detection grows linearly with the message length. In fact, the more bits are hidden, the more DCT coefficients are perturbed during the compression, which results in higher probability of detection. If the sender wants to increase the chances of its stego image being transferred undetected, he must minimize as much as possible the distortions introduced in the cover, thus, showing the trade-off between capacity and undetectability of the steganographic channel.

Reusability of keys is a crucial matter in any communication system designed for the purpose of hiding or encrypting information. The Kerckhoffs' principle states that the security of a system is solely dependent on the key. With the last exper-

iment, Figures 6.8 and 6.9 show that there are some changes in the detection rate when using a single key for all 1000 images. If a big difference between the two curves was observed, it would suggest that a dependence between the keys creates a vulnerability in the system. Using two different but related keys could then reveal the stegosystem. Despite an increasing of approximately 10% in the reliability of detection from the embedding with different keys to the embedding with a single key, the system still achieves good performances.
Users should freely decide to keep the same key file for all communications. Obviously, this practice does not provide forward secrecy given the fact that if at one point in time an attacker takes possession of the key, he will be able to read all previous messages.

Finally, the last experiment gathers all cases tested above in Figure 6.10. Using Table 6.5, it can be observed again that in comparison to the system with modification of the selection of contributing elements, the one with normal selection provides poor security. The other difference that can be noticed is the detection reliability of the system using always the same key for all messages that, with a value of $\rho = 0.0912$, has the biggest rate of detection between the three systems that implement the new selection. Systems using random bitstreams or XORed messages can be considered equivalent in term of detectability with values of ρ very close at 0.0624 and 0.0654.

6.7 Conclusions

The experiments and following analyses presented above indicate that the proposed modification of the Perturbed Quantization selection rule, along with the XOR achieved a good level of undetectability. Transforming the message with a XOR before embedding appears to be a sufficient operation to randomize the data without the use of a key or an encryption algorithm. Furthermore, the dynamic change of the selection of DCT coefficients using the function $f(x_i)$ provides not only faster embedding but also considerably less detectable systems. Choosing to keep for each block the most robust elements reduces the number of changes in the cover and the size of the system to solve at the encoder side.

It was observed during the tests that a stegosystem that uses the same key for every embedding is slightly more detectable than if different keys were used for each embedding. Key reusability is a core element of a system that aims to be practical for the users. Further works are required to study the real impact on the security of a stegosystem when all communications use the same single key.

In conclusion, the stegosystem that was implemented in this work has achieved better scores than the other systems that were tested.

Chapter 7

Conclusion

The increasing need of security and confidentiality was the main motivation behind this thesis. The purpose of this work was (1) to propose a functional steganographic application with (2) a message randomization by the XORing process that introduces duality between the message that is publicly sent and the hidden one without the use of an additional key and (3) testing the implications of the suggested improvement of the selection rule on the steganographic channel detectability.

The first step was to review in Chapter 2 the literature about major existing steganographic techniques. In order to understand what makes a system secure, the underlying theory of steganographic channel and security was explained in Chapter 3, which showed the importance of embedding while minimizing changes and distortions. Next, the Wet Paper Codes method was selected for this work with the Perturbed Quantization and both techniques were then detailed in Chapter 4. The proposed contribution to the scheme was presented in Chapter 5. With all these elements, the next step was to test the system and analyze the differences between different setups, which was done in Chapter 6.

The experiments showed the trade-off between capacity and embedding distortions, where stego images containing large messages are more detected. Among the systems that were tested in the previous chapter, the one using XORed messages and the improved selection of changeable elements showed the best results. The ensemble classifier appeared to detect the stego images embedded with this system with a reliability of only ~ 0.11 for 0.1 bpc. However, the experiments showed that using the same key for all communications has an impact on the detection. Further works could find answers concerning the issue of key reusability. Moreover, it would be interesting to modify the scheme in order to include the parameter r_2 in the message as suggested previously. Making this parameter variable and hidden would hinder the attacker who would not be able to compute neither the number of blocks nor their sizes.

The field of steganography is vast and techniques are still being developed with the only concern of providing covert communications. There is a legitimate need for steganography which motivates the competition between steganographers and steganalysts.

Bibliography

[1] Crypto Law Survey. http://www.cryptolaw.org/. Accessed: 2017-09-04.

[2] D. Alleysson, S. Susstrunk, and J. Herault. Linear Demosaicing Inspired by the Human Visual System. *Trans. Img. Proc.*, 14(4):439–449, 2005.

[3] Ross Anderson and Fabien Petitcolas. On The Li of Steganography. *IEEE Journal of Selected Areas in Communications*, 16:474–481, 1998.

[4] European Machine Vision Association. EMVA 1288 - Standard for Characterization of Image Sensors and Cameras. volume 1288, 2010.

[5] Patrick Bas. Natural Steganography: cover-source switching for better steganography. *arXiv:1607.07824 [cs]*, July 2016.

[6] Patrick Bas, Tomáš Filler, and Tomáš Pevný. "Break our steganographic system": The Ins and Outs of Organizing BOSS. In *Proceedings of the 13th International Conference on Information Hiding*, pages 59–70, Berlin, Heidelberg, 2011. Springer-Verlag.

[7] Christian Cachin. An information-theoretic model for steganography. *Information and Computation*, 192(1):41–56, 2004.

[8] Guillermo Calderón-Meza. Fitting the noise of a CCD camera to a theoretical probability distribution. Technical report, 2007.

[9] Bradley S. Carlson. Comparison of modern CCD and CMOS image sensor technologies and systems for low resolution imaging. In *ResearchGate*, volume 1, pages 171–176 vol.1, February 2002.

[10] Chi-Kwong Chan and L. M. Cheng. Hiding data in images by simple LSB substitution. *Pattern Recognition*, 37(3):469–474, March 2004.

[11] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. Digital image steganography: Survey and analysis of current methods. *Signal Processing*, 90(3):727–752, March 2010.

[12] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[13] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. RAISE: A Raw Images Dataset for Digital Image Forensics. In *Proceedings of the 6th ACM Multimedia Systems Conference*, MMSys '15, pages 219–224, New York, NY, USA, 2015. ACM.

[14] Jessica Fridrich. Digital image forensics. *IEEE Signal Processing Magazine*, 26(2):26–37, March 2009.

[15] Jessica Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[16] Jessica Fridrich, M. Goljan, P. Lisonek, and D. Soukal. Writing on wet paper. *IEEE Transactions on Signal Processing*, 53(10):3923–3935, October 2005.

[17] Jessica Fridrich and Miroslav Goljan. Practical Steganalysis of Digital Images - State of the Art. In *In Proceedings of SPIE*, pages 1–13, 2002.

[18] Jessica Fridrich, Miroslav Goljan, and Rui Du. Reliable Detection of LSB Steganography in Color and Grayscale Images. In *Proceedings of the 2001 Workshop on Multimedia and Security: New Challenges*, pages 27–30, New York, NY, USA, 2001. ACM.

[19] Jessica Fridrich, Miroslav Goljan, and David Soukal. Perturbed Quantization Steganography with Wet Paper Codes. In *Proceedings of the 2004 Workshop on Multimedia and Security*, pages 4–15, New York, NY, USA, 2004. ACM.

[20] Jessica Fridrich, Miroslav Goljan, and David Soukal. Efficient Wet Paper Codes. In *Proceedings of the 7th International Conference on Information Hiding*, pages 204–218, Berlin, Heidelberg, 2005. Springer-Verlag.

[21] Jessica J. Fridrich, Miroslav Goljan, and Dorin Hogea. Steganalysis of JPEG Images: Breaking the F5 Algorithm. In *Revised Papers from the 5th International Workshop on Information Hiding*, pages 310–323, London, UK, UK, 2003. Springer-Verlag.

[22] F. Galvan, G. Puglisi, A. R. Bruna, and S. Battiato. First Quantization Matrix Estimation From Double Compressed JPEG Images. *IEEE Transactions on Information Forensics and Security*, 9(8):1299–1310, August 2014.

[23] Jeremiah J. Harmsen and William A. Pearlman. Steganalysis of Additive Noise Modelable Information Hiding. *ResearchGate*, 5020, February 2003.

[24] IJG. Independent JPEG Group. http://www.ijg.org/. Accessed: 2017-08-17.

[25] Stefan Katzenbeisser and Fabien A. Petitcolas. *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, Inc., Norwood, MA, USA, 1st edition, 2000.

[26] Stefan Katzenbeisser and Fabien A. P. Petitcolas. Defining Security in Steganographic Systems. In *ResearchGate*, volume 4675, January 2002.

[27] Jan Kodovský and Jessica Fridrich. Calibration Revisited. In *Proceedings of the 11th ACM Workshop on Multimedia and Security*, MM&Sec '09, pages 63–74, New York, NY, USA, 2009. ACM.

[28] Jan Kodovský and Jessica Fridrich. Steganalysis in high dimensions: fusing classifiers built on random subspaces. volume 7880, page 78800L. International Society for Optics and Photonics, February 2011.

[29] Jan Kodovsky, Jessica Fridrich, and Vojtěch Holub. Ensemble Classifiers for Steganalysis of Digital Media. *Trans. Info. For. Sec.*, 7(2):432–444, April 2012.

[30] Kwangsoo Lee, Andreas Westfeld, and Sangjin Lee. Category Attack for LSB Steganalysis of JPEG Images. In *Proceedings of the 5th International Conference on Digital Watermarking*, pages 35–48, Berlin, Heidelberg, 2006. Springer-Verlag.

[31] Xin Li, Bahadir Gunturk, and Lei Zhang. Image demosaicing: a systematic survey. In William A. Pearlman, John W. Woods, and Ligang Lu, editors, *Proceedings of SPIE*, volume 6822, page 68221J, January 2008.

[32] Pierre Magnan. Detection of visible photons in CCD and CMOS: A comparative view. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 504(1–3):199–212, 2003.

[33] Pierre Moulin and Ying Wang. New Results on Steganographic Capacity. *Proceedings of the Conference on Information Sciences and Systems*, 2004.

[34] Pierre Moulin and Ying Wang. *Perfectly Secure Steganography: Capacity, Error Exponents, and Code Constructions.* 2007.

[35] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. *Information Hiding – A Survey.* 1999.

[36] Tomáš Pevný and Jessica Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In *ResearchGate*, volume 6505, pages 3 1–3 14, San Jose, CA, February 2007. editors. DOI: http://dx.doi.org/10.1117/12.696774.

[37] T. Pevny, P. Bas, and J. Fridrich. Steganalysis by Subtractive Pixel Adjacency Matrix. *IEEE Transactions on Information Forensics and Security*, 5(2):215–224, June 2010.

[38] Inc Quantum Scientific Imaging. Understanding CCD Read Noise. http://qsimaging.com/ccd_noise.html, September 2016. Accessed: 2016-09-05.

[39] Gustavus J. Simmons. The Prisoners' Problem and the Subliminal Channel. In David Chaum, editor, *Advances in Cryptology*, pages 51–67. Springer US, 1984.

[40] Douglas R. Stinson. *Cryptography: Theory and Practice, Third Edition*. CRC Press, November 2005.

[41] Gregory K. Wallace. The JPEG Still Picture Compression Standard. *Commun. ACM*, 34(4):30–44, 1991.

[42] Andreas Westfeld. F5—A Steganographic Algorithm. In Ira S. Moskowitz, editor, *Information Hiding*, pages 289–302. Springer Berlin Heidelberg, April 2001.

[43] Andreas Westfeld and Andreas Pfitzmann. Attacks on Steganographic Systems - Breaking the Steganographic Utilities EzStego. In *Jsteg, Steganos, and S-Tools - and Some Lessons Learned," Lecture Notes in Computer Science*, pages 61–75. Springer-Verlag, 2000.

[44] Timothy York. Fundamentals of Image Sensor Performance, October 2016. Accessed: 2016-10-06.

[45] Tao Zhang and Xijian Ping. A Fast and Effective Steganalytic Technique Against JSteg-like Algorithms. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, SAC '03, pages 307–311, New York, NY, USA, 2003. ACM.