Image Authentication Scheme and its Hardware Architecture Using Watermarking and Visual Cryptography

by

Ángel Hernández Joaquín

A thesis submitted in partial fulfillment of the requirement for the degree of

Master of Science in Computer Science

at

National Institute of Astrophysics, Optics and Electronics
August, 2017
Tonantzintla, Puebla

Adviser:

PhD. René Armando Cumplido Parra
Computer Science Department
INAOE

Contents

List of Figures

List of Tables

Acknowledgements

I would first like to thank my advisor PhD. René Armando Cumplido Parra for his wonderful guidance, support and generosity. It has been a privilege to work under his supervision, always showing me the right direction, but allowing me to walk the path myself.

Special thanks to PhD. Claudia Feregrino Uribe for her valuable contribution to the development of this work.

Thanks a lot to CONACyT, the Mexican Council for Science and Technology, for having sponsored my graduate studies, and the development of this research, through the grant 702608.

Abstract

There is a tendency to protect digital content from unauthorized use. The importance of protecting images is that it seeks to authenticate said content with the help of a certain entity against unauthorized use. A proper method is required to mark large number of images produced in certain applications. This work proposes to solve the problem of unauthorized use of natural images by using watermarking techniques and visual cryptography. Efficient hardware architecture to implement the proposed method was also be explored. The principal contribution of this work is that the proposed scheme can reach copyright protection and authentication at same time. According to the experimental results, the proposed watermarking scheme obtains similarly results in terms of imperceptibility and robustness, but embedding more information than others schemes.

Chapter 1

Introduction

With the development of the transmission technologies and the Internet technologies, the digital media industry began to flourish. [CISCO, 2016] [Boland et al., 1995] Originator's rights and publisher's reputation are affected by the condition that various infringement behaviors, particularly video and image products, have occurred. As a means of information security protection, digital watermarking technology can be used to protect the copyright of digital products effectively. Therefore, digital watermarking technology has drawn the attention of the researchers.

Image watermarking is an efficient solution for authentication and copyright protection of images in popular communication environments like the Internet, which is susceptible to illegal usages [Singh et al., 2014]. Digital watermarking describes a method and technology that hides information. It is the process of embedding a piece of digital information into any multimedia data. In some watermarking schemes, a watermarked image contains an image or some other information embedded into the image that may be visible or invisible. The quality of the watermarking scheme largely depends upon the choice of the watermark structure and insertion strategy.

Given algorithm characteristic that work in the frequency domain, these themselves to its parallelization and even to its implementation in hardware [Cho and Lee, 1990] [Sava et al., 1997]. Hardware-based watermark schemes offer real-time processing, where watermarks are embedded at the very moment the image is captured. It is beneficial in applications such as real-time transmissions, video authentication, and security [ElAraby et al., 2010] [Joshi et al., 2011].

1.1 Problem Statement

With the rapid growth of Internet technologies, digital media industry began to flourish. Originators' rights and publishers' reputation are affected by the condition that various infringement behaviors, particularly the pirated video and image products, have been occurred. As a means of information security protection, digital watermarking technology can be used to protect the copyright of digital products effectively. Therefore, digital watermarking technology has drawn researchers attention. Digital watermarking provides a promising way of protecting multimedia data from illegal manipulation and duplication.

1.2 Motivation

Half Internet traffic corresponds to multimedia content. With the great and easy transmission of digital multimedia contents, the copyright protection has been receiving an increasing attention to protect multimedia content, it is a relevant theme for industries as music, cinema, television, books, software, etc. Image and video are the most popular multimedia objects which are shared easily over the network. The research aims to increase hiding information to protect digital multimedia, for example Watermarks (to hide copyright messages) and digital fingerprint (to hide serial numbers or some set characteristics to authenticate a digital content). The principal idea is to use that information against to unauthorized users.

Hardware implementation is essential for low power, real-time performance, high reliability, low-cost applications, and also for easy integration with existing consumer electronic devices. For example, watermarking chips can be integrated with any existing digital image camera. The hardware modules can also be integrated with a JPEG codec [Tsai and Lu, 2001] or systems for video authentication [Mohanty et al., 2007, Roy et al., 2013] . Successively, the JPEG codec can be part of a scanner a digital camera or any other multimedia device so that the digitized images are watermarked right at the capture time.

1.3 Summary of objectives

1.3.1 Principal Objective

This work proposes a method for images authentication and their hardware architecture. This method must be efficient to insert information and to take advantage of frequency domain characteristics to obtain a good performance and to have superior performance compared to the software version.

1.3.2 Specific Objectives

- To select watermarking and visual cryptography algorithms for authentication, considering approaches reviewed in literature.

- To define the best implementation strategy in hardware.

- To evaluate the architecture regarding processing and hardware resources used.

1.4 Dissertation overview

The following chapters of the dissertation describe the main components of the design and the fundamental principles. The remainder of the document is organized as follows:

Chapter 2 This chapter describes thesis background, it shows the basic concepts used during this research work.

Chapter 3 This chapter describes the principal related works as robust, fragile and dual watermark image schemes.

Chapter 4 This chapter describes the proposed watermark image scheme. The chapter details the internal processes than the scheme performs: DWT transform, shares and fragile watermark generation, embedding robust and fragile watermark, detection, watermark extraction and authentication process.

Chapter 5 This chapter describes details of the hardware implementation. It describes changes respect to software application and directives used.

Chapter 6 This chapter presents experimental results of proposed scheme, also the comparison against another robust, fragile and dual watermarking schemes.

Chapter 7 This chapter describes the conclusion, a review of objectives and the future work.

Chapter 2

Background

This section describes the main features of watermarking techniques, Visual Cryptography Schemes (VCS) and hardware watermarking with the aim of providing an introductory explanation of both techniques. This chapter is not intended to provide a complete description of all features of these research fields since a large number of books (see for example [Wang et al., 2009], [Cimato and Yang, 2011]) and several hundreds of papers have been published in recent years. Interested readers can look to some interesting surveys on those topics. [Hartung and Kutter, 1999, Ateniese et al., 1996a]

2.1 Watermarking

A watermark is a digital code embedded into a digital media (images, sound, and texts), usually containing different kinds of information about the owner or creator and the data destination. In most cases, a watermark is composed of another image or logo which can be directly related to image owner; the relationship between the watermark and the owner can be assessed by storing the watermark at a Trusted Authority (TA) which can intervene in a case of dispute. The usual application of watermarks is to detect copyright infringements and to be used as a proof in case of a dispute between the owner or the legitimate destination and the malicious user. To prevent attacks, the watermarks should be robust enough to avoid the intentional or accidental removal and should not introduce disturbing effect on the original data. On the other in many cases, only the selected receiver should be enabled to detect and manipulate the embedded watermark.

2.1.1 Applications

Watermarking Techniques are usually used as a method to protect Intellectual Property Rights (IPR). If a suspected image is examined and the embedded watermark detected, then a follow-up action can be started against the illegitimate use of the image. A typical scenario is the one where the image creator makes its images available online. But to avoid misuse or false ownership attribution, the user embeds a watermark inside the images; in this case, everyone can download the image, but in a case of dispute, the owner will be able to show the presence of the watermark inside the disputed image and claim its ownership.

Watermarking techniques have also been used for other goals, such as data authentication, data monitoring, and tracking. In the first case a fragile watermark is embedded into the original data so that any manipulation occurred during the data transmission can be detected; indeed a fragile watermark has the characteristic of being very sensitive to slight modification of the embedding image. In the second case, watermarks are used by monitoring systems to automatically detect the owners of broadcast data and pay the due royalties to them.

2.1.2 Requirements

Digital watermarking schemes are typically based on two phases, the embedding phase where the data of the watermark are merged with the data of the original image, and the extracting phase, where a watermarked image is examined, and the inverse procedure is applied to retrieve the watermark. These are the basic requirements that a watermarking scheme should [Cox et al., 2002, Coatrieux et al., 2000, Ni et al., 2008] :

Imperceptibility The watermark should be perceptually invisible and when embedded should not introduce too much distortion into the original image.

Robustness It refers to the ability of the embedded message to overcome the insertion problem such as alteration in the pixels and information loss. The embedded watermark should be extractable and identifiable after various intentional or occasional attacks are performed. These include common signal processing operations and geometric distortions, such as blurring, JPEG compression, noising, sharpening, scaling, rotation, cropping, and so on. The watermark should resist against intentional attacks to remove it as well as introduced noise.

Security Only the legitimate owner should be able to extract and modify the embedded watermark. The security should depend only on keeping the key secret, while the watermarking algorithm should be public.

Blindness The original image is not needed to verify the existence and extract the watermark in the test image. In this case, the copyright owner is not required to utilize extra disk space to store original images. The blindness property is very useful in practical schemes.

Multiple watermarking Sometimes the possibility of inserting multiple watermarks inside the original data is requested to trace the distribution of digital images. However, the possibility of crossing a latter watermark over a first watermark should be avoided, and in general, multiple watermarking schemes are complex and try to overcome this weakness.

Unambiguity The embedded watermark should be verifiable without ambiguity and the ownership of the image correctly and unambiguously determined. The problem of confusing the ownership by simply appending an illegal watermark to the watermarked image, and the consequent possibility to have multiple claims of ownership (also called the deadlock problem, counterfeit attack, or invertibility attack)

2.2 Classification

There are several types of watermarks; each one has unique characteristics that can be useful depending on the case in which they are used.

2.2.1 Robust Watermarks

These watermarks cannot be broken easily as they withstand many signal processing attacks. The robust watermark should remain intact permanently in the embedded signal such that attempts to remove or destroy the robust watermark will degrade or even may destroy the quality of the image. Many applications require watermarks to be detected in images that may have been altered after embedding. Watermarks designed to survive legitimate and everyday usage of content are referred to as *robust* watermarks. Robust Watermarks are designed to resist any attempt by an adversary to thwart their intended purpose. General methods for achieving high robustness will be presented.

2.2.2 Fragile Watermarks

Most methods currently proposed for providing image authentication are based on a fragile watermark in opposition to robust watermark classically used for copyright protection. The basic idea underlying these techniques is to insert a particular watermark (generally independent of the image data) so that any attempt to alter the content of an image will also alter the watermark itself (Figure 2.1). Therefore, the authentication process consists of locating watermark distortions to detect the regions of the image that have been tampered with. The major drawback of these approaches is that it is difficult to distinguish between malicious and non-malicious attacks (e.g., most fragile methods consider a lossy compressed image as a tampered image, whereas the semantic of the image is unchanged) A fragile watermark is simply a mark likely to become undetectable after an image is modified in any way. Until now, seeking instead to design robust watermarks that can survive many forms of distortion. However, fragility can be an advantage for authentication purposes. If a very fragile is detected in an image, it is possible to infer that image has probably not been altered since the watermark was embedded. At least, it is unlikely the image has been accidentally altered.

2.2.3 Semi-fragile watermarks

A semi-fragile watermark is another type of authentication watermark. The alterations on the images can occur unintentionally or can be implanted intentionally. Unintentional or innocent alterations are usually result from such diverse facts as bit errors during transmission and storage, or signal processing operations such as filtering, contrast enhancement, sharpening, and compression. On the other hand, intentional or malicious alterations, are assumed to be due to an explicit forgery attempt by a pirate with the explicit purpose of changing the contents of an image.

Semi-fragile watermarks are more robust than fragile watermarks and less sensitive to classical user modifications such as JPEG compression.

Some semi-fragile watermarking schemes use the domain of Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT). Usually, these schemes use the chosen transform domain as the media to embed and extract watermarks and they use extracted watermarks to authenticate digital images and localize the tampered areas if possible. [Qi and Xin, 2011]

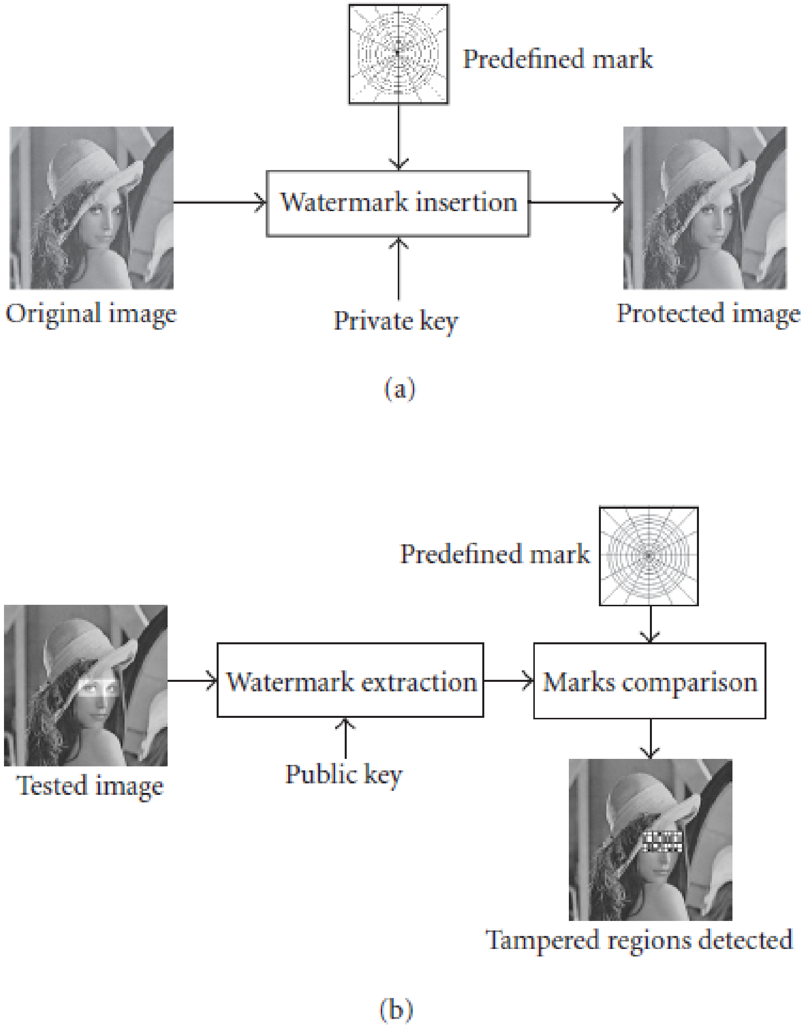


Figure 2.1: Fragile watermark scheme: (a) Image security. (b) Authenticity verification

2.3 Performance Evaluation Metrics

People who work with watermarking schemes needs some way to evaluate and compare them. These people interested in applying watermarking to an application need to identify the system that are most appropriate. They need measures by which to verify algorithm improvements.

2.3.1 Metrics to Evaluate Imperceptibility

The quality and fidelity of an image are in function of imperceptibility of the mark. Fidelity represents the difference between watermarked image and original image, and the quality represents the watermark's acceptability. The best metric to evaluate the quality and fidelity of a watermarked image are the human senses. For example, a sight to visual media, hearing to audio. A metric used to evaluate imperceptibility is done through the use of some experiments, for example using the Two-alternative forced choice test. This test consists in showing to a set of observers two options, in this case, the original digital image and the watermarked image. Each observer must select which image looks better, after many tests with various images. In this work a statistic analysis is performed to determinate if the watermarked image has an acceptable quality respect to original image. Some metrics use the selection of alternatives [Cox et al., 2002], they provide precise metrics to perceptibility. However, there are subjective, expensive, hard to repeat and they can not be automated. Otherwise, there are automated metrics to measure the distortion of an embedded watermark. These are easy to implement and are not subjective. The distortion caused by watermark embedding process can be defined as the difference or distance between original media and the watermarked media.

The principal objectives metrics to measure distortion between two images, specifically in images [Eskicioglu and Fisher, 1995][Saffor et al., 2001], (where X is the original image of size M and X' is the watermarked image of size N) are:

1. Mean Squared Error (MSE): it is the easiest distortion measure to calculate, defined as:

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (X(i,j) - X'(i,j))$$

   To get a good watermark imperceptibility, MSE should be as small as possible.

2. Root-Mean-Square Error (RMSE): this distortion measure is the square root of MSE; It is used to set more weight to MSEs smaller. To get a good watermark imperceptibility, RMSE should be as small as possible.

$$RMSE = \sqrt{MSE}$$

3. Signal-to-Noise Ratio (SNR): It is the most popular distortion measure, it is calculated as follows:

$$SNR = 10 \log_{10} \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (X'(i,j))^2}{\sum_{i=1}^{M} \sum_{j=1}^{N} (X(i,j)) - (X'(i,j))}$$

   With a higher SNR value, the watermark is less noticeable, the measure unity is expressed in decibels.

4. Peak Signal-to-Noise Ratio (PSNR): Similar to SNR, a higher PSNR value means a better watermark imperceptibility

$$PSNR = 10 \log_{10} \frac{MAX_I^2}{MSE} = 20 \log_{10} \frac{MAX_I}{\sqrt{MSE}}$$

Where MAX is the value maximum of a pixel. For example if we are comparing 8 bit images, MAX takes the value of 255. Values greater than 30 dB are admissible.

5. Mean Absolute Error (MAE): It is the maximum value of the absolute values of the differences of the two elements

$$PSNR = 10 \log_{10} \frac{MAX_I^2}{MSE} = 20 \log_{10} \frac{MAX_I}{\sqrt{MSE}}$$

A low value corresponds to a better approximation between the two elements

6. Normalized cross-correlation (NC): This metric measures the similarity between original media and the watermarked media, the values then it can get are between 0 and 1. These values indicate the similarity percentage between two images. A higher value corresponds to a better image quality.

$$NC = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} X(i,j) X^0(i,j)}{\sum_{i=1}^{M} \sum_{j=1}^{N} X(i,j)^2}$$

2.3.2 Metrics to Evaluate Robustness

The robustness of the watermarking schemes can be evaluated using many types of distortions or modifications in the watermarked image; these attacks must be relevant for watermarking scheme applications. The robustness can be evaluated measuring the probability of extracting the watermark after an image has received an attack, that is, this work evaluates measuring the precision, it is the relation between the watermark extracted and the original embedded watermark.

To evaluate the robustness of the watermarks usually some very well known attack libraries are used as benchmarks: StirMark, unZign and Checkmark [Petitcolas et al., 1998, Pereira et al., 2001]. The Stirmark library includes several attacks, such as compression, geometric transformations, processing for signal enhancement, and noise addition. The unZign benchmark introduces local pixel jittering and is very efficient in attacking spatial domain watermarking techniques including estimation based attacks by considering prior information about the watermark. Checkmark benchmark library includes new removal attacks, such as maximum likelihood estimation attacks, maximum a posteriori (MAP) based attack, de-noising assuming low pass watermark, and other new geometric attacks. Watermarking schemes are considered robust if they can survive attacks contained in the library

2.4 Natural Images

This work deals with “natural images” . In the context of this work, a “natural image” is the result of taking an ordinary digital camera, pointing it somewhere in the world and pressing the shutter button. This is a definition than coincides with the majority of images taken in the world today, but it still is not a very good definition. Previous definition ignores many different aspects of acquiring digital images, lens effects, sensor, pre and post-processing of image. [Gonzalez et al., 2004, Liu et al., 2008]

The term natural images is used in many different contexts in the relevant literature [Barlow, 1961, Ruderman, 1997, Zhu and Mumford, 1997, Weiss and Freeman, 2007], but there is no agreed upon definition of natural images.

2.4.1 The Image Space

Natural images are an extremely diverse and complex entity. Because they are projections of the vast world around us, they embody many of the physical properties of this world. An explicit representation for such a complex thing may be very hard to find, and as such, natural images lend them selves to a statistical description. [Hyvärinen and Muszynski, 2008, Zoran, 2013]

A digital image is represented as a numerical array containing the intensity values of its picture elements, or "pixels", it will be of $N \times M$ pixels. Each pixel may be a grayscale value (a scalar) or a color value (usually an RGB).

To understand the sheer size of the possible space of images, consider a small 32×32 grayscale image where each pixel can be assigned one of 256 grayscale values. This space of all possible image for this small image comprises of 256^{1024} different images. An important question is: how would the points be distributed in this space? A majority of these images, will be complete garbage, most of them will not contain anything similar to objects we see in this world, just noise.

2.5 Visual Cryptography

Visual cryptography schemes allow the encoding of a secret image, consisting of black or white pixels, into n shares which are distributed to a set of n participants. [Cimato and Yang, 2011] The secret pixels are shared with techniques based on the intelligent subdivision of each secret pixel into a certain number of subpixels. Each share is then composed of black and white subpixels, which are printed near each other so that the human visual system averages their black/white contributions. White means transparent, so that the superposition of white pixels, let the color of the pixel contained in the other shares pass through.

The shares are such that only qualified subsets of participants can visually recover the secret image, but other subsets of participants, called forbidden sets, cannot gain any information about the secret image by examining their shares. The shares can be conveniently represented with an $n \times m$ matrix S where each row represents one

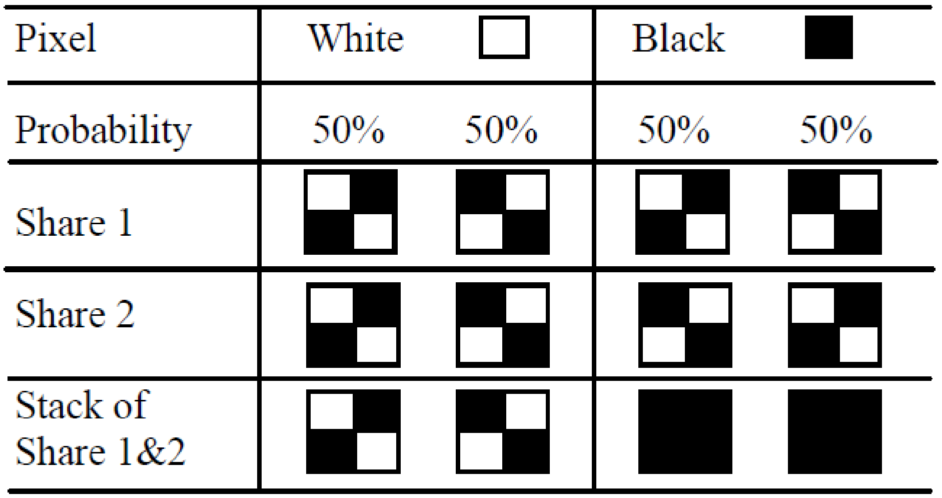


Figure 2.2: A simple Visual Cryptography Scheme (VSC)

share, i.e., m subpixels, and each element is either 0, for a white subpixel, or 1 for a black sub pixel. A matrix representing the shares is called distribution matrix.

To reconstruct the secret image, a group of participants stacks together their shares. The grey level of the combined share, obtained by stacking the transparencies $i_1, \ldots, i_s$, is proportional to the Hamming weight $w(V)$ of the m-vector $V = OR$ $(r_i 1, \ldots, r_i s)$, where $r_i 1, \ldots, r_i s$ are S rows associated to stacked transparencies. This gray level is interpreted by the visual system of the users as black or as white according to some contrast rule. Since each secret pixel is represented by m pixels in the shares, the reconstructed image will be bigger than the original one (depending on m and on actual positions of pixels, the image can also be distorted; a perfect square is a good choice for m because it avoids distortion).

Two parameters are essential for visual cryptography schemes: Pixel expansion, corresponding to the number of subpixels contained in each share (transparency) and contrast, which measures the "difference" between a black and a white pixel in the reconstructed image. In general, a scheme is characterized by other parameters: the number of participants n, the threshold k that determines whether a set of members is qualified to reconstruct the image, the contrast thresholds and h, which determine whether a reconstructed pixel is considered white or black.

This cryptographic paradigm was introduced by Naor and Shamir [Naor and Shamir, 1994]. They analyzed the case of (k, n)-threshold visual cryptography schemes, in which a black and white secret image is visible if and only if at least k transparencies among n are stacked together. The model by Naor and Shamir has been extended in [Ateniese et al., 1996b] to general access structures (an access structure is a specification of all qualified and forbidden subsets of participants), where general techniques to construct visual cryptography schemes for any access structure have been proposed.

To provide shares to the participants the dealer chooses uniformly at random a distribution matrix from a collection of matrices C_1, if the secret pixel is black, or from a collection of matrices C_0, if the secret pixel is white. Let report here the formal definition of a deterministic VCS:

Let $(\Gamma_{Qual}, \Gamma_{Forb})$ be an access structure on a set of n participants. Two collections (multisets) of $n \times m$ boolean matrices C and C_∞ constitute a visual cryptography scheme $(\Gamma_{Qual}, \Gamma_{Forb}, m)$-VCS if there exist the integers ℓ and h, $\ell < h$, such that:

1. Any (qualified) set $Q = i_1, i_2, \ldots, i_p \in \Gamma_{Qual}$ can recover the shared image by stacking their transparencies. Formally, for any $S \in C$, the "or" V of rows $i_1, i_2, \ldots, i_p$ satisfies $w(V) \leq \ell$; whereas, for any $S \in C_\infty$ it results that $w(V) \geq h$.

2. Any (forbidden) set $X = i_1, i_2, \ldots, i_p \in \Gamma_{Forb}$ has no information on the shared image. Formally, the two collections of $p \times m$ matrices Dt, with $t \in 0, 1$, obtained by restricting each $n \times m$ matrix in C_X to rows $i_1, i_2, \ldots, i_p$ are indistinguishable in the sense that they contain the same matrices with the same frequencies.

In many schemes, the collection C_0 (resp. C_1) consists of all the matrices that can be obtained by permuting all the columns of a matrix M_0 (resp. M_1). For such schemes, the matrices M_0 and M_1 are called the base matrices of the scheme. Base matrices constitute an efficient representation of the scheme. Indeed, the dealer has to store only the base matrices and in order to randomly choose a matrix from C_X he has to randomly choose a columns permutation of the basis matrix M_X.

The basis matrices M_0 and M_1 in the $(2, 2)$-VC scheme are:

$$M_0 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \quad M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

As shown in Table 2.1, if the original pixel is white, one of six combinations of share pixels is randomly created. Similarly, the possible share combination for black pixels is also shown. After stacking the shares with white transparent and black opaque, the original secret image will be revealed. Stacking can be viewed as mathematically "OR", where white is equivalent to "0" and black is equivalent to "1".

Table 2.1: Encoding pin (2, 2)-VCRG.

p	Probability	$r1$	$r2$	$r = r1 \otimes r2$
1	0.5	1	1	1
	0.5	0	0	0
0	0.5	1	0	0
	0.5	0	1	0

Visual cryptography based watermarking seems to provide some interesting opportunity to solve some of the problems related to the digital right management. Two or more shares can be generated to embed a share into Host Image and protect it.

2.6 Watermarking and Chaos

Chaotic systems are deterministic systems (predictable provided enough information is available) that are governed by nonlinear dynamics. These systems show deterministic behavior that is very sensitive to its initial conditions, in a way that the future results are uncorrelated and seem random. One rather intriguing category of digital watermarking is based on chaos theory and chaotic functions. In image processing, the chaotic functions that are used are two-dimensional and are known as chaotic maps. There are many such functions, but the most relevant for images is Arnold's cat map [Arnold, 2017]. An image is hit with a transformation that apparently randomizes the original organization of its pixels. However, if iterated enough times, the original image is recovered.

2.6.1 Chaotic map

Voyatzis and Pitas [Voyatzis and Pitas, 1996] first introduced chaos theory in digital watermarking and presented a watermarking scheme based on a two-dimensional chaotic function, called "Toral Automorphism". A cyclic chaotic function is applied to a square image to rearrange its pixels. After it is applied T times, where T is the period of the function and depends on the size of the image [Li and Xu, 2005], the pixels are found in their initial location. If (x, y) are the initial coordinates of a pixel of a $M \times M$ image, the outcome coordinates of the chaotic function (x, y) are given by

$$\begin{bmatrix} x\prime \\ y\prime \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \ell & \ell + 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} modM$$

where M denotes the width of the image and ℓ is an integer parameter that affects the period T of the chaotic function.

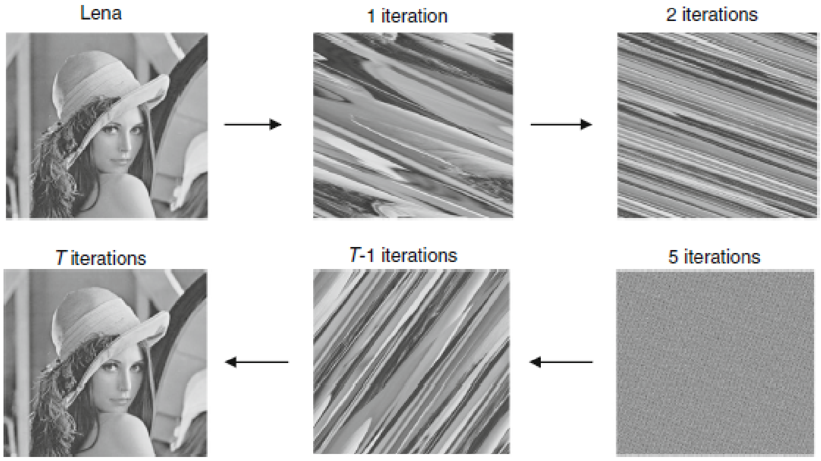


Figure 2.3: Application of chaotic map on "Lena" image

After transforming normal and chaotic images in the frequency domain (e.g., DCT), coefficients of the chaotic image will be more scattered with higher standard deviation as shown in [Wu and Shih, 2007]. Consequently, the number of coefficients with significant value will be higher in the chaotic image in comparison with the coefficients of the respective "normal" image.

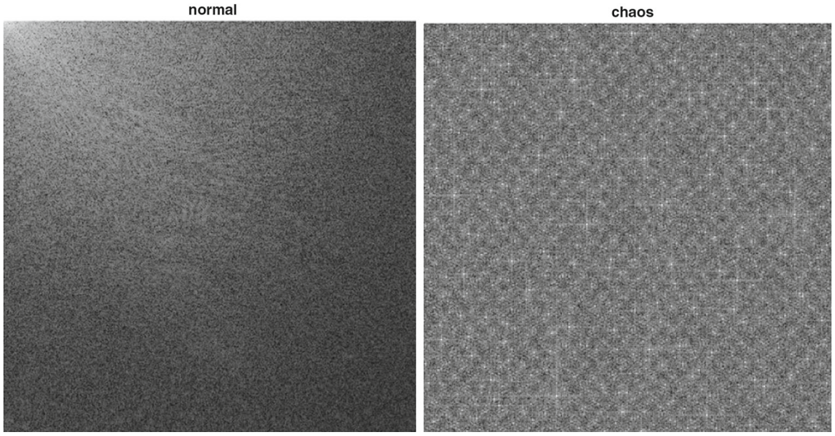


Figure 2.4: Power spectrum of DCT coefficients for "Lena" image (left) and chaotic "Lena" image (right), respectively. DCT coefficients are in logarithmic scale and have been normalized with regard to the DC coefficient

Therefore, the transformed chaotic image is richer in frequency content, a desirable property in watermarking, as there are more suitable candidate coefficients for manipulation and information embedding. [Mishra et al., 2014] In Fig. 2.5 , it is evident that the energy of chaotic image is spread to entire region of image as opposed to the typical image, where energy of image is concentrated to the upper left region of image. Comparing histograms of Fig. 2.5 , it is also obvious that in chaotic domain, there are more coefficients with significant value, which are more suitable for manipulation and information embedding.

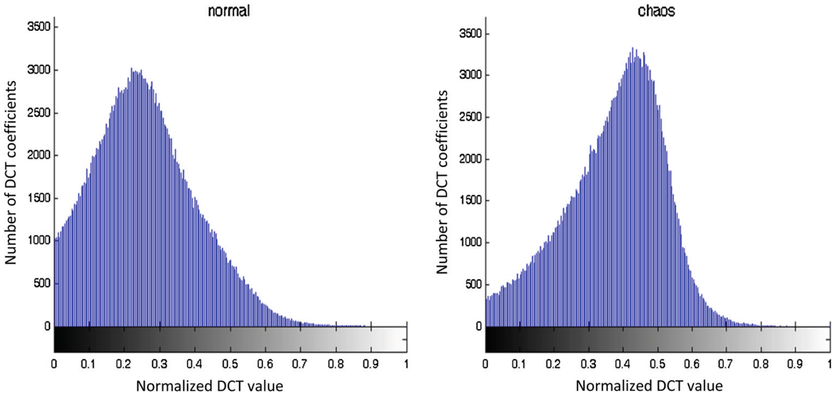


Figure 2.5: Histogram of DCT coefficients (in logarithmic scale and normalized with regard to the DC coefficient) for “Lena” image (left) and chaotic “Lena” image (right), respectively

2.7 Image Watermarking Embedding Domain

A watermark can be seen as a signal that needs to be embedded with the host data, such that the watermark signal is hidden in the obtained permutation, but it can be easily recovered later if the correct procedure is adopted, sometimes using a cryptographic key. There are three main classes of watermarking techniques [Cox et al., 2002]: spatial-domain, transform-domain and quantization domain based techniques.

In spatial domain based techniques, host image data, i.e., pixel values, are directly modified to hide corresponding bits of watermark image. The most common technique in this class is the Least Significant Bit (LSB) modification, where last bits of some selected positions of host image are filled with the watermark bits. [Liu et al., 2016b, Rawat and Raman, 2011] Spatial based techniques are easy to implement, achieve a good compromise as regards visual quality, but show weak robustness, since simple attacks based on geometric distortion of watermarked image can cause impossibility to recover embedded watermark.

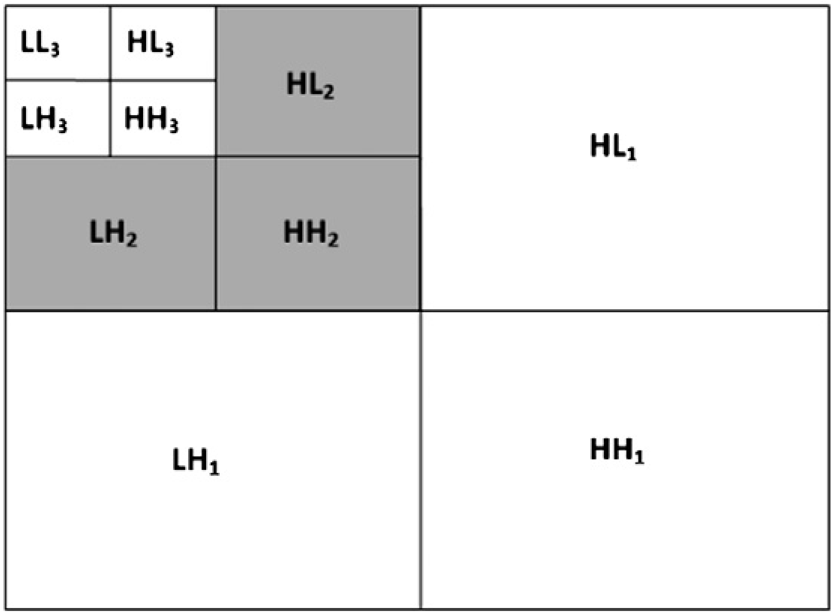


Figure 2.6: Wavelet decomposition on L = 3 resolution levels.

In transform-domain based techniques, raw data of host image are transformed into frequency components using some standard methods such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), or Discrete Wavelet Transform (DWT). Frequencies are then modified according to values of corresponding watermark bits without causing any distortion in original image.

Among the transformation domain based technique, DWT method has been used in several proposals because of its advantages. [Mishra et al., 2014, Singh et al., 2015, Xia et al., 1997] In Fig. 2.6 , a 2-scale wavelet transform is depicted. The image is first decomposed into four sub-bands, LL_1, LH_1,HL_1, and HH_1 (each having 1/4 size of the original image). The sub-bands denoted HH1, LH1,HL1 contain the finest scale detail wavelet coefficients, corresponding to the higher frequency detail information. Re-applying the wavelet transform process to the sub-band LL_1, the next coarser scale can be obtained, and the four novel sub-bands LL_2, LH_2,HL_2 and HH_2 can be obtained, each having 1/4 size of the LL_1 sub-band. The process could be iterated t times if further decomposition is needed. The low-frequency component is the sub-band LL1 which contains most of the information of the original image, while the sub-bands LH,HL, and HH represent horizontal, vertical and diagonal details, respectively. The transform-domain based technique can extract an invariant feature of the original image and show strong robustness against attacks based on geometrical distortion and compression.

The quantization-domain based techniques are based on the application of lossy image compression methods based, as on vector quantization (VQ). [Liu et al., 2016b] In such techniques, an encoder and a decoder are used in combination with a codebook. During the encoding, the input image vector is built by retrieving, for each input block, the index of the codeword which best represents the block. The vector containing the indexes can be sent to the decoder. The codewords can be retrieved, and the original image reconstructed using the same codebook and the indexes.

2.8 Watermarking and Visual Cryptography

In watermarking schemes, usually, the produced watermark is directly embedded into the image to be protected, to prevent abuses and illegitimate distribution of the image. When a visual cryptographic scheme is used in combination, usually the watermark is given as input to the VC scheme, obtaining some shares. One of the shares is then used as watermark, while the other ones will be stored and protected.

[Chourasia, 2013, Devi et al., 2012, Surekha and Swamy, 2012, Han et al., 2014] The typical scenario considered in combined watermarking VC based scheme includes a number of actors:

- The owner of the image who wants to mark its image and prevent not authorized use of the image
- A trusted authority (TA) who participates in the scheme and whose intervention can be requested to arbitrate the ownership of the image if a dispute occurs
- Finally, the adversary who wants to alter the image and its watermark and use it, cheating about the ownership of a stolen image.

2.8.1 Watermarking with (2,2) VC Scheme

Most of the schemes combining watermarking with visual cryptography are based on the use of a (2, 2) VC scheme. As mentioned in [Naor and Shamir, 1994], such VC schemes can be thought of as a private key cryptosystem. Indeed, the secret printed message is encoded into two random looking shares: one of the two shares can be freely distributed and used as a cipher-text, whereas the other share plays the role of a secret key. The original image is reconstructed by stacking together the two transparencies. This system recalls the one-time pad, as each page of ciphertext is decoded by using a different transparency. In combined watermarking schemes, the input image to the VC scheme is the watermark.

The proposed model is depicted in figures 2.7 and 2.8, where the embedding and the extraction phases of such kind of combined watermarking techniques are described [Ateniese et al., 1996b, Sencar and Memon, 2005, Li et al., 2006]. The owner of the image I, gives in input the watermark SI to one of the variants of the $(2,2)$ VC

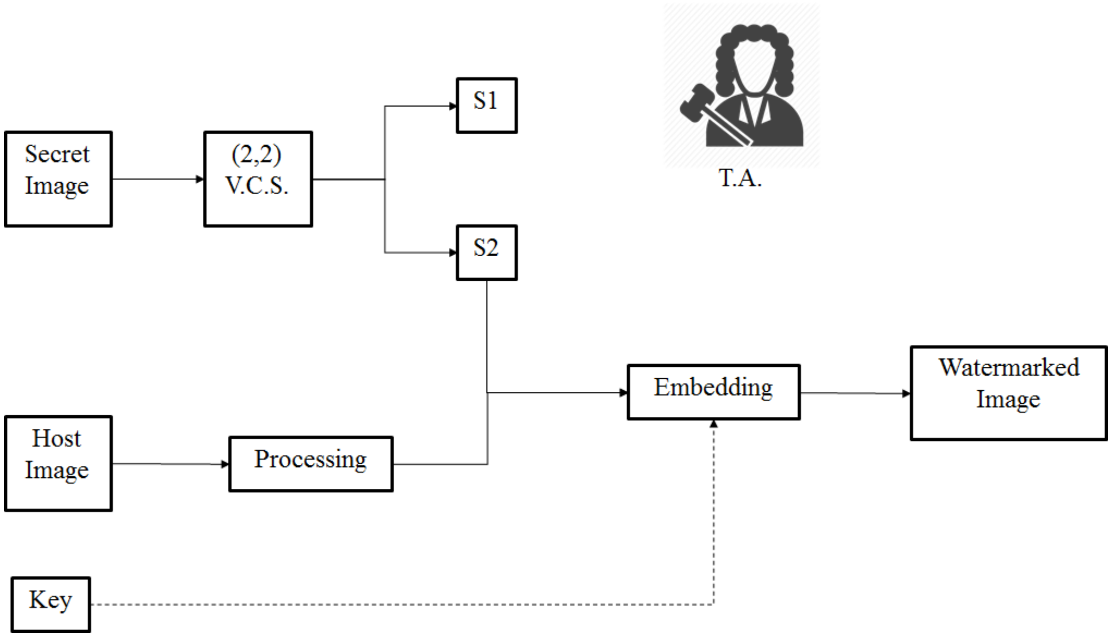


Figure 2.7: Embedding phase for watermarking combined with a $(2, 2)$ VC scheme

scheme previously described to obtain two watermarks $S1$ and $S2$, which appear as random images. One of the shares, $S1$ is then used as a key such that only the legitimate extractor can reconstruct the watermark and show it to a third party; in some cases, $S1$ is registered to a TA who can then resolve a dispute over the ownership of a claimed image. The second share, $S2$, is then embedded into the original image, performing an embedding operation that depends on the particular kind of watermark technique considered. Indeed, the original image I undergoes a processing phase, where some decomposition or some feature extraction is used (i.e. some frequency domain transform). During this phase, some schemes require the knowledge of a secret key K, needed for example to select the locations or the values of the original image which contain the watermark bits. At the end of the process, the watermarked image WI can be published or distributed for any legitimate use.

In the extraction phase, depicted in Fig. 2.8, the process above is inverted [Ateniese et al., 1996b, Sencar and Memon, 2005, Li et al., 2006]. The owner of the image, wanting to claim the ownership of a suspected image during a controversy with an

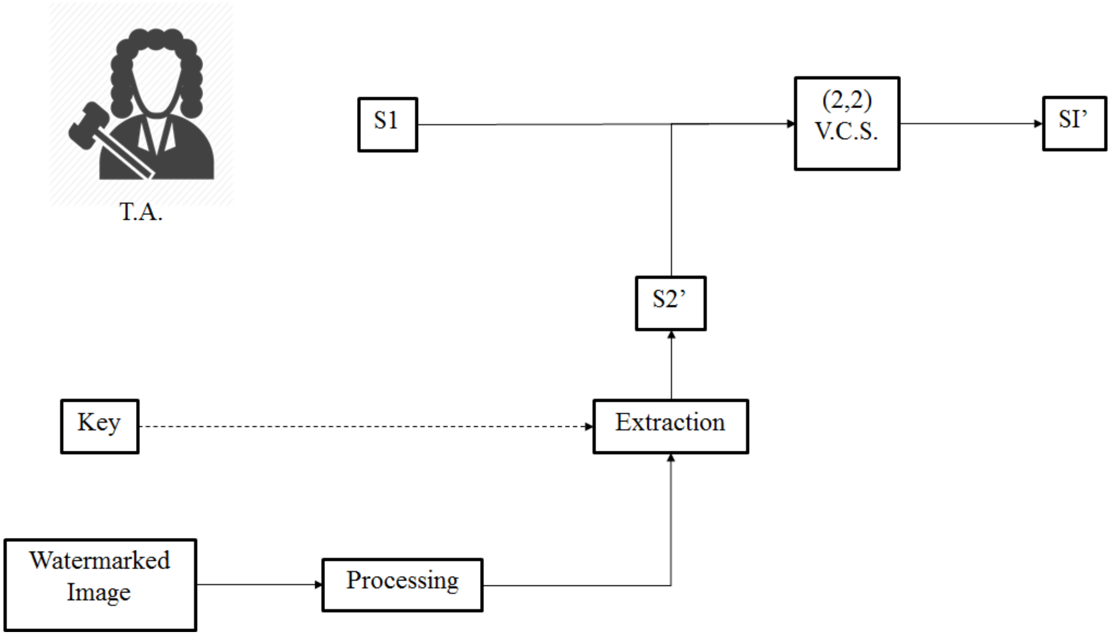


Figure 2.8: Extraction phase for watermarking combined with a $(2,2)$ VC scheme

adversary, can (using the secret key if necessary) extract the needed information from image WI to obtain embedded share. At the same time, the owner can use the second share $S1$, possibly involving the TA where the share has been stored, to reconstruct the original watermark. If the image was belonging to the claiming owner, the watermark W is equal or similar to the original watermark W, and the dispute is resolved. Several schemes respecting the structure above described have been proposed in the literature, each one introducing some variations in the way the VC scheme is used to generate the shares or the way that the image is processed and the watermark embedded into the original image.

2.9 Watermarking and Hardware

Implementing algorithms onto electronic circuits is a tedious task that involves operations scheduling. Whereas algorithms can theoretically be described by sequential operations, their implementations need better than sequential scheduling to take advantage of parallelism and improve latency. It brings signaling into the design to coordinate operations and manage concurrency problems. These problems have not been solved in processors that do not use parallelism at algorithm level but only at instruction level. In these cases, parallelism is not fully exploited. The frequency race driven by processor vendors shadowed the problem replacing operators' parallelism by faster sequential operators. However, parallelism remains possible and it will obviously bring tremendous gains in algorithms latencies. [Coussy and Morawiec, 2008] High-Level Synthesis (HLS) design is a kind of answer, and opens a wide door to designers.

High-Level Synthesis (HLS) [Martin and Smith, 2009] is a process of converting a behavioral/algorithmic description of an Intellectual Property (IP) core into its equivalent register transfer level counterpart consisting of several sub-steps like a compilation, transformation, scheduling, allocation, and binding. High-level synthesis, comprising of sub-tasks such as scheduling, allocation, binding, etc.

An IP (intellectual property) core is a block of logic or data that is used in making a field programmable gate array (FPGA) or application-specific integrated circuit (ASIC) for a product. IP cores are part essential elements of design reuse electronic design automation (EDA) industry trend towards repeated use of previously designed components. [Kahng et al., 1998]

[Abdel-Hamid et al., 2005, Martin and Smith, 2009, Poli et al., 2007, Memik et al., 2005, Ram et al., 2012] , plays a major role in designing digital IP cores. Embedding a watermark in the register allocation step during HLS is non-trivial as it may incur hardware overhead. Additionally, due to the existence of numerous competitive design solutions in the design space, choosing an appropriate solution to embed the watermark is highly complex. This is because each solution in the design space may result in different latency and hardware area. Selecting an arbitrary solution for embedding a watermark may result in a violation of user specified latency and area constraints for the design. Therefore, to have an optimal watermark scheme is crucial.

The process of embedding the watermark can be implemented on hardware or software platform. An watermarking scheme can reach a better performance, it helps to embed the watermark in real-time, which is considered to be more secure than software watermarking inserting additional complexity to overcome security gaps [Shoshan et al., 2008]. The software watermarking is an offline process where the original video/image is captured at the first stage, and subsequently, the watermark is embedded in the second stage. The approach introduces the certain delay between capturing the object and inserting the watermark.

2.9.1 Field Programmable Gate Array

An FPGA is a type of integrated circuit (IC) that can be programmed for different algorithms after fabrication. Modern FPGA devices consist of up to two million logic cells that can be configured to implement a variety of software algorithms. [Brown and Rose, 1996] Although the traditional FPGA design flow is more similar to a regular IC than a processor, an FPGA provides significant cost advantages in comparison to an IC development effort and offers the same level of performance in most cases. Another advantage of the FPGA when compared to the IC is its ability to be dynamically reconfigured. This process, which is the same as loading a program in a processor, can affect part or all of the resources available in the FPGA fabric. It is important to have a basic understanding of the available resources in the FPGA fabric and how they interact to execute a target application.

FPGA Architecture

The basic structure of an FPGA is composed of the following elements [Tocci, 2007] [Brown and Rose, 1996] [Xilinx, 2013] [Farooq et al., 2012] :

Look-up table (LUT) This element is the basic building block of an FPGA and is capable of implementing any logic function of N Boolean variables. Essentially, this element is a truth table in which different combinations of the inputs implement different functions to yield output values. The limit on the size of the truth table is N, where N represents the number of inputs to the LUT.

Flip-Flop (FF) is the basic storage unit within the FPGA fabric. This element is always paired with a LUT to assist in logic pipelining and data storage. The basic structure of a flip-flop includes a data input, clock input, clock enable, reset, and data output.

Wires These elements connect elements to one another.

Input/Output (I/O) pads These physically available ports get data in and out of the FPGA.

The combination of these elements results in the basic FPGA architecture shown in Figure 2.9. Although this structure is sufficient for the implementation of any algorithm, the efficiency of the resulting implementation is limited in terms of computational throughput, required resources, and achievable clock frequency.

Contemporary FPGA architectures incorporate the basic elements along with additional computational and data storage blocks that increase the computational density and efficiency of the device. These additional elements, which are discussed in the following sections, are:

Embedded memories for distributed data storage Phase-locked loops (PLLs) for driving the FPGA fabric at different clock rates High-speed serial transceivers Off-chip

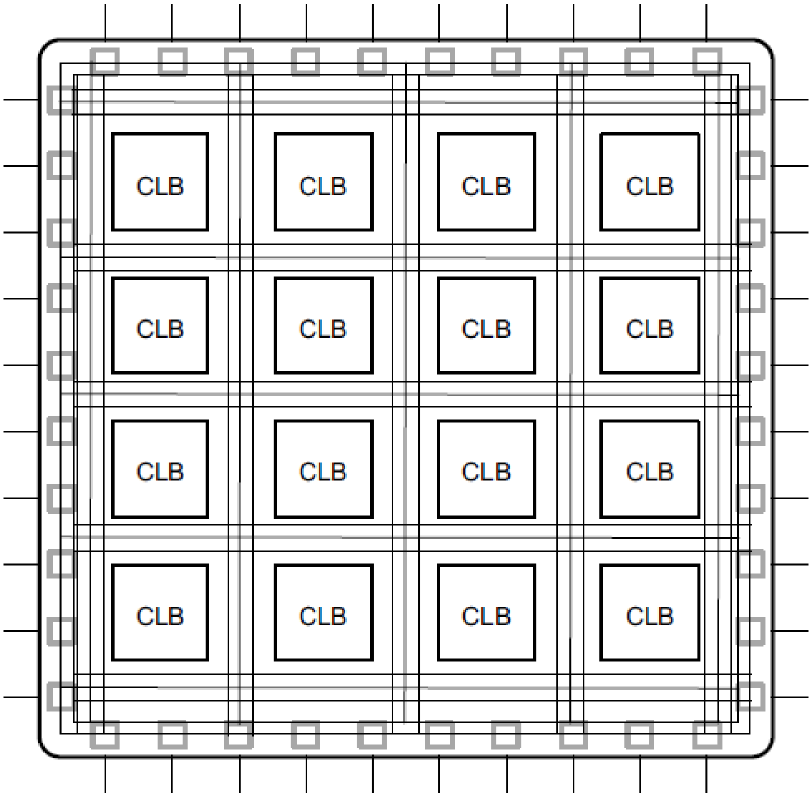


Figure 2.9: Basic FPGA Architecture [Tocci, 2007]

memory controllers Multiply-accumulate blocks The combination of these elements provides the FPGA with the flexibility to implement any software algorithm running on a processor and results in the contemporary FPGA architecture shown in Fig. 2.10

2.9.2 Vivado High-Level Synthesis

The Xilinx® Vivado® High-Level Synthesis (HLS) compiler provides a programming environment similar to those available for application development on both standard and specialized processors. [Xilinx, 2013] HLS shares key technology with processor compilers for the interpretation, analysis, and optimization of C/C++ programs. The main difference is in the execution target of the application. By targeting

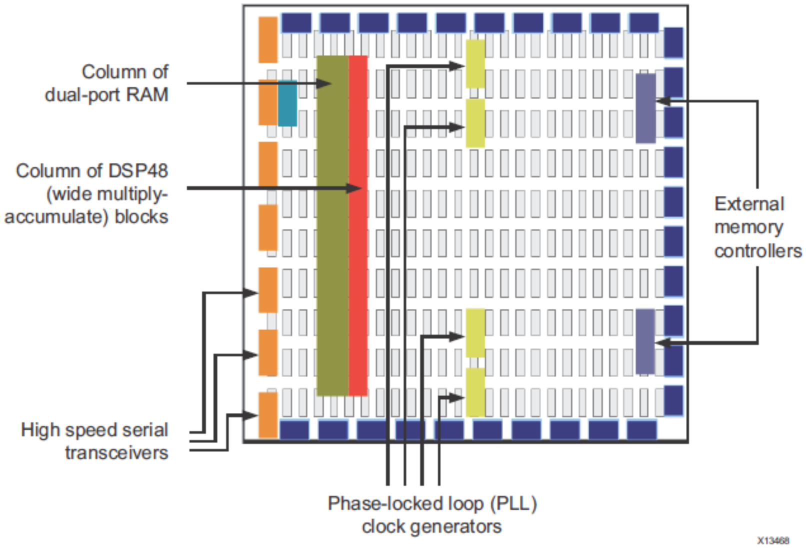


Figure 2.10: Contemporary FPGA Architecture [Farooq et al., 2012]

an FPGA as the execution fabric, HLS enables a software engineer to optimize code for throughout, power, and latency without the need to address the performance bottleneck of a single memory space and limited computational resources. This allows the implementation of computationally intensive software algorithms into actual products, not just functionality demonstrators.

Programming Model

Usually, software algorithms are typed in C/C++ or some other high-level language. It abstracts the details of the computing platform. These languages allow for quick iteration, incremental improvements, and code portability, which are critical to the software developer. [Gajski et al., 2012, Coussy and Morawiec, 2008]

The HLS compiler uses the same categories as any processor compiler to process application code. HLS analyzes all programs regarding: [Xilinx, 2013]

Operations Refer to both the arithmetic and logical components of an application that are involved in computing a result value.

Conditional statements Conditional statements are program control flow statements that are typically implemented as if, if-else, or case statements. These coding structures are an integral part of most algorithms and are fully supported by all compilers, including HLS.

Loops Loops are a common programming construct for expressing iterative computation. One common misconception is that loops are not supported when working with compilers like HLS. Although this might be true with early versions of compilers for FPGAs, HLS fully supports loops and can even do transformations that are beyond the capabilities of a standard processor compiler

Functions Functions are a programming hierarchy that can contain operators, loops, and other functions. The treatment of functions in both HLS and processor compilers is similar to that of loops.

Chapter 3

Related Work

The previous chapter exposes than the principal objective of a watermark scheme is protect a digital content against unauthorized use. There are some of watermarking schemes as robust, fragile, semi-fragile and hybrid. This chapter describes the principal schemes then uses these ways to embed information, these methods are important to select and determine the best methods to propose a new watermarking scheme.

3.1 General Overview

The following are the schemes that apply techniques of watermarking with visual cryptography in Table 3.1, the most important characteristics of these works are exposed.

An analysis of the schemes that have been performed, the robustness of the watermark against attacks and the imperceptibility are fundamental to consider in this analysis. In Table 3.2, it can be seen that approaches that use transform in the frequency domain, obtain better results in resistance to attacks.

Another interesting case that shows us the comparison of Table 3.2 is that the methods that use and / or combine frequency-domain methods, usually have greater tolerance to modifications, which can affect the adequate recovery of the mark of the watermark.

If a review is made of those applications that make watermarks on hardware implementations, the main focus is on videos (sequence of images), such as real-time transmission, video authentication, movie recording and video systems vigilance [ElAraby et al., 2010] [Joshi et al., 2011] . The implementation takes advantage of the characteristics of the techniques in the domain of the frequency [Lach et al., 1999], besides accelerating the flow applying pipeline techniques, with this, they obtain the required redemption for applications in real time [Ghosh et al., 2014b].

To create a new scheme, the most effective methods to insert information must be reviewed. The principal scheme used is a hybrid technique using one or more transformations as Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), Singular Value Decomposition (SVD) and Discrete Fourier Transform (DFT).

Some authors use a single technique to embed the watermark, for example [Preda and Vizireanu, 2015] , [Ali et al., 2014] , [Das et al., 2014] ,[Li and Qin, 2013] use DCT, [Zope-Chaudhari et al., 2015] uses DWT. In Watermarking Schemes and Hardware Implementations techniques as DCT and metadata [Roy et al., 2013] , DWT [Ghosh et al., 2014a] , spatial insertion [Lakshmi and Surekha, 2016] , [Jadhav et al., 2015] , fingerprinting [Mohanty et al., 2007] and AES [Singh and Lamba, 2015] are used.

Table 3.1: Watermarking Schemes and Visual Cryptography

Method	Characteristics	Image Host	Watermark Image
1. [Han et al., 2014]	DCT and Visual Cryptography Scheme VSC (2,2)	Color Image 1024 x 1024	Binary Image 128 x128
2. [Han et al., 2013]	DWT and VSC (2,2)	Color Image 256x256	Binary Image 75 x 75
3. [Sleit and Abusitta, 2008]	Generates verification information to validate	Bitmap	Binary Image
4. [Chourasia, 2013]	FFT or DCT and VSC (2,2)	Natural Images 225x225	Fingerprint 54x69
5. [Bekkouch and Faraoun, 2015]	DCT, DWT and SVD	Medical Images 256x256	Fingerprint
6. [Devi et al., 2012]	DCT, SVD and VSC	Color Image 512x512	Binary Image 106 x143
7. [Surekha and Swamy, 2012]	Iterative Watermark Embedding	Grayscale 512x512	Binary Image

Table 3.2: Common Image Attacks on State of the Art

	Attacks	**Scheme**							
		1.	2	3	4	5	6	7	%
Signal Processing	Compression	x	x		x	x	x	x	85
	Contrast				x				14
	Histogram Equalization							x	14
	Salt and Pepper	x			x	x		x	57
	Gaussian Noise						x		14
	Gaussian Blur	x	x		x		x		57
	Gamma Correction						x		14
	Filters: Gaussian, Median, Avgerage		x		x		x	x	57
Geometric	Scaling			x			x	x	43
	Rotation		x	x		x	x	x	71
	Cropping	x	x			x	x	x	71
	Pattern Insertion			x					14
	%	33	41	25	41	33	66	58	

Most authors use two or more techniques to embed information, for example [Singh et al., 2014], [Fazli and Moeini, 2016], [Khan et al., 2013] use DWT, DCT, SVD to process the host image before embedding the watermark. Similarly, [Makbol et al., 2016], [Mishra et al., 2014], [Singh et al., 2015], [Ji et al., 2015], [Hu et al., 2016], [Kumar et al., 2016] use DWT/DCT with SVD before to perform the insertion.

An interesting point is performing an image preprocessing. Using chaos schemes to preprocess the host image, it is possible to reach robustness and imperceptibility over the host image. Some authors as [Chrysochos et al., 2014], [Wang et al., 2015], [Zhu et al., 2013], [Khalili, 2015], [Behnia et al., 2014], [Liu et al., 2016a], [Lei et al., 2014], [Reyes et al., 2010], [Rawat and Raman, 2012] perform Chaotic map, Arnold Transform Map (ATM), firefly algorithm, visual cryptography to host image and/or watermark image, then they apply some transform domain and insert the watermark. After, they perform the inverse process to recover the original positions pixels.

3.2 Robust Schemes

Many applications require watermarks to be detected in images that may have been altered after embedding. Watermarks designed to survive legitimate and everyday usage of content are referred to as *robust* watermarks. Robust Watermarks are designed to resist any attempt by an adversary to thwart their intended purpose. General methods for achieving high robustness will be presented.

[Makbol et al., 2016] presented a DWT–SVD block-based image watermarking scheme. Several characteristics were employed to achieve high-level grades for the watermarking requirements and maintain the trade-off between them. Initially, blocking was used to divide the image into blocks. Then, only a portion of these blocks was selected to include the watermark; this selection ensured that the embedding process would affect specific regions of the image. The HVS characteristics of entropy and edge entropy were used to select the low informative blocks as the best embedding regions. The scheme employed the properties of a DWT and SVD. These methods aimed to provide high robustness by selecting the most robust regions with an emphasis on maintaining non-noticeable distortions, in other words, to maintain imperceptibility

3.3 Fragile Schemes

A fragile watermark is simply a mark likely to become undetectable after an image is modified in any way. Until now, seeking instead to design robust watermarks that can survive many forms of distortion. However, fragility can be an advantage for authentication purposes. If a very fragile is detected in an image, it has probably not been altered since the watermark was embedded. At least, it is unlikely the image has been accidentally altered.

[Rawat and Raman, 2011] presented a novel fragile watermarking scheme for image authentication and locating tampered regions. They use Chaotic maps to make the scheme highly secure. Since chaotic maps are sensitive to initial values, they are used as keys in their scheme. Extracting the right watermark is only possible if someone has correct keys. A person with wrong keys will not be able to forge the watermark. As in order to thwart counterfeiting attacks, it is essential to break pixel wise independence. Their proposed scheme employs chaotic maps to break the corresponding position relation between pixels in the watermarked image and the watermark. Experimental results show that their scheme has high fidelity and is capable of localizing modified regions in watermarked image.

3.4 Semi-fragile Schemes

[Preda, 2013] has proposed a new image authentication scheme using a semi-fragile watermark that can detect and locate malicious tampering in images. The embedding and extraction of the authentication watermark are done in the DWT domain. His technique achieves high image quality and high tampering detection resolution at a low watermark payload. Thanks to the random permutation of the wavelet coefficients before embedding, the watermark is protected against local attacks, like, for example, the VQ attack. The embedded watermark is also robust to mild to moderate JPEG compression by selecting the appropriate embedding parameters.

3.5 Hybrid Schemes

[Lu and Liao, 2001] propose a novel multi purpose watermarking scheme, in which robust and fragile watermarks are simultaneously embedded, for copyright protection and content authentication. By quantizing a host image's wavelet coefficients as masking threshold units (MTUs), two complementary watermarks are embedded using cocktail watermarking, and they can be blindly extracted without access to the host image. For image protection, the new scheme guarantees that no matter what kind of attack is encountered, at least one watermark can survive well. On the other hand, for image authentication, this approach can locate the part of the image that has been tampered with and tolerate some incidental processes that have been executed. The authors show that the performance of this multi purpose watermarking scheme is indeed superb concerning robustness and fragility with experimental results.

[Lin et al., 2009] present a novel dual watermarking mechanism for digital media that embeds a recognizable pattern into the spatial domain and an invisible logo into the frequency domain. Undoubtedly, visible watermarking is important for protecting online resources from unauthorized reproduction. Due to the visibility of embedded patterns, however, watermarked digital media are vulnerable to the inpainting attack and common signal processing operations. Utilizing hybrid strategies, simulation results show that the novel method can resist these attacks. In particular, the new mechanism allows legal subscribers to restore an unmarked image, whereas other dual watermarking schemes do not. This feature makes it suitable for protecting artistic and valuable media.

[Lusson et al., 2013] proposed a watermarking algorithm, inspired by steganography techniques, to hide the watermark so that it is undetectable and thus harder to remove or destroy. On the security aspect, the proposed hybrid algorithm was benchmarked against Stirmark, showing the excellent resilience of the ASCII watermark to the PSNR test, at all levels of attacks. Results show that the proposed hybrid watermarking technique is undetectable to visual inspection and steganalysis tools failed to detect it. The hybrid watermarking method can withstand levels of geometric and processing attacks, up to a point where the commercial value of the images tested would be lost.

[Liu et al., 2016b] propose a blind dual watermarking mechanism for color the authentication of images and copyright protection. The invisible, fragile, and robust watermarks are embedded into the spatial domain of the RGB color space and the frequency domain of the YCbCr color space. The major contribution of this work is that their scheme can achieve copyright protection and image authentication simultaneously, and the extraction of watermarks from the protected image can be processed blindly without the original host image and watermarks.

3.6 Digital Watermarking in Hardware Schemes

The process of embedding the watermark can be implemented on hardware or software platform. The hardware watermarking is a better approach because it helps to embed the watermark in real-time, which is considered to be more secure than software watermarking [Shoshan et al., 2008]. The software watermarking is an offline process where the original video/image is captured at the first stage, and subsequently, the watermark is embedded in the second stage. The approach introduces certain delay between capturing the object and inserting the watermark.

[Wong, 1998] presented a watermarking method used to embed the watermark in

Least Significant Bit (LSB) of each pixel of the image. The original image is separated in different blocks where LSB of each block is modified. Then, the watermark of each block is calculated with bit-wise EXOR operation using a hash function. The algorithm is extended by [Brunton and Zhao, 2005] who developed a real time video watermarking system using Graphics Processing Units (GPUs). They had provided the solution for video authentication and tamper localization. [Mathai et al., 2003] designed the video watermarking scheme known as Just Another Watermarking Scheme (JAWS) which uses shifting of reference pattern. The generated watermark is scaled down by a constant factor and then subsequently inserted in each frame of the video. The frames are stored using dual port memory during the process of watermarking. [Mohanty and Kougianos, 2011] illustrated the visible watermarking method for ownership verification applications. The method is useful for MPEG-4 compression that inserts the logo in a video stream. The hardware performance of the algorithm is also tested using Altera Cyclone II FPGA. [Roy et al., 2013] developed the high performance architecture for video authentication. MJPEG based video compression algorithm uses the concept of pipeline and parallel processing for the improvement of overall system performance. The algorithm posses an excellent robustness to withstand several potential attacks, including cropping and segment removing of the video sequence.

[Ziener and Teich, 2006] presented a power watermark technique for the FPGA cores. In this approach the signature is detected at the power supply pins of the FPGA. With this approach it is possible to read the watermark only with a given device without any other information from the vendor of the product. [Nie et al., 2013] proposed a hierarchical watermarking method for FPGA IP protection has been presented. The authors' embed the watermark into the netlist and bitstream of the design. In this way, an embedded watermark is propagated entirely through the design. [Ziener and Teich, 2008] proposed a power watermarking method that detects signature (watermark) at the power supply pins of the FPGA. They integrated the signature into functional parts of the watermarked core, and detected it from a voltage trace with high reliability.

[Joshi et al., 2015] presented a video watermarking based on H.264 coding standard. The algorithm is based on Integer DCT where fully parallel and pipeline architectures were designed for better speed and area performance respectively. The algorithm uses the scene change detection concept for robustness against temporal attacks. Later on, the same algorithm is implemented on FPGA for real time ownership verification application [Joshi et al., 2016]. The parts of the same watermark are inserted in various frames of a particular scene of the video to improve the performance. The algorithm is simulated by using MATLAB and then it is prototyped on the FPGA to verify the performance on hardware.

3.7 Remarks

The previous review from state of the art, there are a lot of watermark algorithms, like fragile, robust watermarks. One of the main reasons for using the frequency domain to insert a watermark is increase its imperceptibility, while maintaining its robustness characteristics. Some schemes use chaotic functions to preprocess an image and to get the best coefficients to embed a watermark.

There are two aspects to achieve: Authentication (to detect some tampering on host image) and copyright protection (to determine image owner). There are two interesting robust watermark schemes: [Singh et al., 2014] and [Liu et al., 2016a]. The first work is a non-blind scheme, but it has a good way to embed information using binary data (like a share) compression through a scale factor. The second work divides Wavelet coefficients into several 8×8-sized block, and, then, each coefficient of a block is quantized by directly dividing it by the corresponding value of luminance quantization table. An interesting authentication scheme is the proposed by [Preda, 2013], he uses high-frequency wavelet coefficients randomly permuted with a secret key to embed a fragile watermark.

This work proposes a scheme where two watermarks are embedded; a fragile to authentication and a robust to prevent not authorized usage. It is going to use Arnold Transform Map and a two resolution levels on 2D-DWT decomposition. Embedding the robust watermark on $LL2$ using the way to embed the watermark of [Singh et al., 2014], and performing a wavelet coefficients quantization as in [Liu et al., 2016a]. Simultaneously embedding the fragile watermark as in [Preda, 2013]. If there are two independent processes is very useful while design and implementing a hardware architecture, because there is no data dependency and this directly impacts the execution time. Using tools like Vivado High-Level Synthesis to create a hardware implementation is faster using a high-level programming language like C/C++. Processing images, performing arithmetic operations and improving hardware implementation regarding processing and hardware resources can be optimally archived though HLS.

Chapter 4

Proposed Scheme

This chapter describes the proposed copyright protection scheme in detail. The steps of dual watermark embedding, extraction and authentication phases of this proposed scheme are described in following subsections.

Considering the semi-fragile watermarking approach proposed by [Preda, 2013], that uses high frequencies for embedding fragile watermark. The present proposed method uses low frequencies to embedding a robust watermark using quantization proposed by [Liu et al., 2016b] and the Watermark method by [Singh et al., 2014] to reach a good imperceptibility and data compression , Figure 4.1. Both approaches can be used to embed two watermarks on the frequency domain and achieve robustness and imperceptibility.

4.1 Proposed Visual Cryptography Scheme

Shyu [Shyu, 2015] provides a method to generate a set of threshold (k, n)-VCRG (visual cryptograms of random grids) for sharing a secret image P among n participants. For encoding an image pixel P in $(2, 2)$-VCRG, the author provides a table,

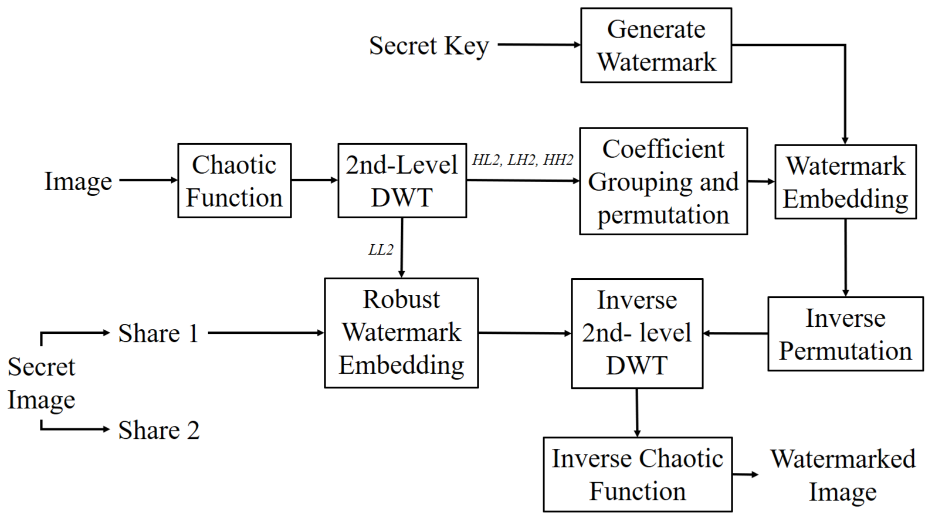


Figure 4.1: Proposed Watermark Embedding Scheme

this method processes pixels one by one. The proposed visual cryptography scheme encodes groups of 2 pixels. In contrast to the existing watermarking scheme based on Visual Cryptography (WVC), where one picture element is encoded at a time, the block (BVC) coding algorithm inputs a block of two pixels time. If the pixel of entry block is 00 (11), the algorithm selects one code column from the code table, by the secret key bit. When each code block containing dissimilar pixels, i.e., 01/10 is encountered, the coding algorithm increments a counter. Based on the counter value (even/odd), it selects either 00 or 11 coding columns, the secret key bit. The proposed Scheme use the concept of Block Visual Cryptography (BVC) to generate two non-expanded cipher shares from the watermark image. Also, since a block of two pixels is replaced every time with another block of two pixels, the size of the shares is unexpanded. Thus, it improves the quality of the reconstructed watermark.

Algorithm 1: Block Visual Cryptography based on Shyu [Shyu, 2015] scheme

```
function BVC ($B$);
Input   : Secret Binary Image $B$
Output: $Share1$, $Share2$
$EvenOddCounter \leftarrow 0$;
$key \leftarrow 0$;
$P \leftarrow zeros(2, 2)$;
for $i \leftarrow 1$ to $H$ do
    for $j \leftarrow 1$ to $W - 1$ do
        if $B[i][j] = B[i][j + 1]$ then
            $P \leftarrow EncodePixel(B[i][j], B[i][j + 1])$;
            $Share1[i][j] \leftarrow P[1][1]$;
            $Share1[i][j + 1] \leftarrow P[1][2]$;
            $Share2[i][j] \leftarrow P[2][1]$;
            $Share2[i][j + 1] \leftarrow P[2][2]$;
        else
            $EvenOddCounter = EvenOddCounter + 1$;
            $key = EvenOdd(EvenOddCounter)$;
            $P \leftarrow EncodePixel(key, key)$;
            $Share1[i][j] \leftarrow P[1][1]$;
            $Share1[i][j + 1] \leftarrow P[1][2]$;
            $Share2[i][j] \leftarrow P[2][1]$;
            $Share2[i][j + 1] \leftarrow P[2][2]$;
        end
    end
end
```

Table 4.1: Luminance quantization table

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

4.2 Embedding Scheme

This section, explains the proposed protection scheme in detail. Fig. 4.1 shows the proposed watermark scheme. The process of embedding the watermark is divided into two phases, i.e. 1) embedding the invisible robust watermark and 2) embedding fragile watermark.

4.2.1 Embedding Robust Watermark

1. Initially, the original RGB image is used, select the blue component and to perform Arnold's Transform several times. The blue component is extracted because the human eye has difficulty perceiving the changes in this component.

2. Secret Image is processed to generate two shares with Algorithm 1. A share is used to embed host image.

3. The 2nd level DWT is applied over the chaotic image obtained to get LL_2, LH_2, HL_2 and HH_2. sub-band LL_2 will used to embed robust watermark and sub-bands LH_2, HL_2 and HH_2 to embed fragile watermark

4. LL_2 sub-band is divided into several 8×8-sized blocks, and, then, each pixel of a block is quantized by directly dividing it by the corresponding value of the luminance quantization Table 4.1

5. To insert the robust watermark, The band LL_2 is used to embed the robust watermark as follow:

$$LL_2^w = LL_2 + \alpha I_w \tag{4.1}$$

I_w is the watermark's pixel value position

α is the watermark's scale factor. A high value improve the robustness, but the imperceptibility is.

QLL_2 is the coefficient quantized position

LL_2^w are the watermarked coefficients

4.2.2 Embedding Fragile Watermark

1. The selected wavelet coefficients are concatenated in a 1D vector C and randomly permuted using a secret key K into a new vector C' to embed fragile Watermark. This process assures that coefficients corresponding to the same spatial location will be separated in C'.

2. The sequence C' of permuted coefficients is divided into groups of d coefficients. Parameter d controls the watermark payload of the proposed scheme.

3. The watermark, a binary random sequence w, is generated using the secret key K and serves as the authentication code. w has the same length as the number of wavelet coefficient groups.

4. The weighted mean M_i of every group i of permuted wavelet coefficients is obtained using the following equation:

$$M_i = \sum_{j=1}^{d} (-1)^j |C_i(j)| \tag{4.2}$$

$c_i(j)$ denotes the jth coefficient of group i.

$(-1)^j$ is used to make the scheme more robust to common image processing operations

5. The watermarked mean m_i^w is obtained by quantization of m_i to the nearest even or odd quantization level according to the value of the corresponding watermark bit w_i, using the following equation:

$$m_i^w = \begin{cases} \lfloor m_i/Q \rfloor \cdot Q & \text{if } mod2(\lfloor m_i/Q \rfloor) = w_i \\ \lfloor m_i/Q \rfloor \cdot Q + Q & \text{if } mod2(\lfloor m_i/Q \rfloor) \neq w_i \end{cases}$$

6. For every group i of coefficients, the weighted mean mi is changed to the watermarked mean m_i^w by modifying the wavelet coefficient $c_{i,max}(j)$ with the highest absolute value. Because of the random permutation operation every group should have at least one coefficient with high absolute value. The updating of $c_{i,max}(j)$ is done as follows:

$$C'_{i,max}(j) = C_{i,max}(j) + (-1)^j \cdot sign(C_{i,max}(j)) \cdot (m_i^w - m_i) \tag{4.3}$$

$C'_{i,max}(j)$ is the watermarked coefficient.

$$sign(x) = \begin{cases} 1 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{4.4}$$

7. After updating the coefficients with the highest absolute value from every group, the inverse permutation is applied using the secret key K.

8. Finally, the Inverse 2D-DWT with watermarked coefficients is calculated to obtain the watermarked image.

4.3 Extraction Process

In the proposed extraction procedure, the robust watermark and the fragile watermark can be extracted separately for copyright detection and image authentication, respectively. The two different methods are described in detail below.

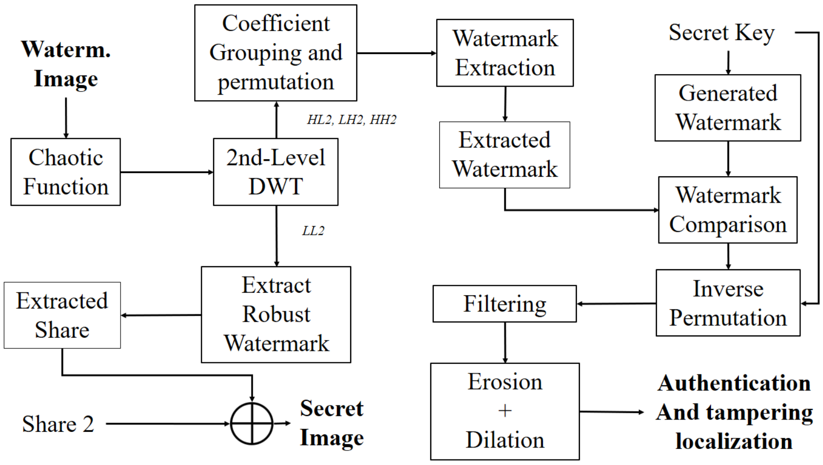


Figure 4.2: Extraction Process

1. The watermarked possibly tampered image is taken and its blue component is selected, extracted and transformed using Arnold Transform Map

2. Later the 2D-DWT decomposition is performed over the result to generate the LL_2, LH_2, HL_2, and HH_2 sub-bands.

3. The detail wavelet coefficients from the LH_2, HL_2 and HH_2 Wavelet sub-bands are selected to fragile watermark extraction and LL_2 to robust watermark.

4. Then, LL_2 sub-band is divided into several 8×8-sized blocks, and each pixel of host image block is quantized by directly dividing it by the corresponding value of the luminance quantization Table 4.1

5. After all of the blocks are quantized, a quantized LL ($Q_L L$) sub-band is generated. Then, the robust watermark is extracted by the following equation

$$W_w e = (LL_2 - Q_L L)\alpha \tag{4.5}$$

   $Q_L L$ is the quantized LL (Q-LL) sub-band generated

   α is the watermark's scale factor.

   LL_2 is the coefficient position

   $W_w e$ is the watermark extracted

6. To improve the quality of the reconstructed watermark, a logical XOR operation is used in combining share 1 and 2.

7. To extract the fragile watermark, the selected wavelet coefficients are concatenated. Using a secret key K, the same random permutation is performed.

8. The sequence of permuted coefficients is divided into groups of d coefficients.

9. The weighted mean m'_i of every group of coefficients is calculated.

10. A watermark bit wi is extracted from the weighted mean of every group of d coefficients using following equation:

$$W'_i = mod2(\lfloor m'_i/Q \rfloor)$$

4.4 Authentication Process

To perform image authentication the following steps need to be done:

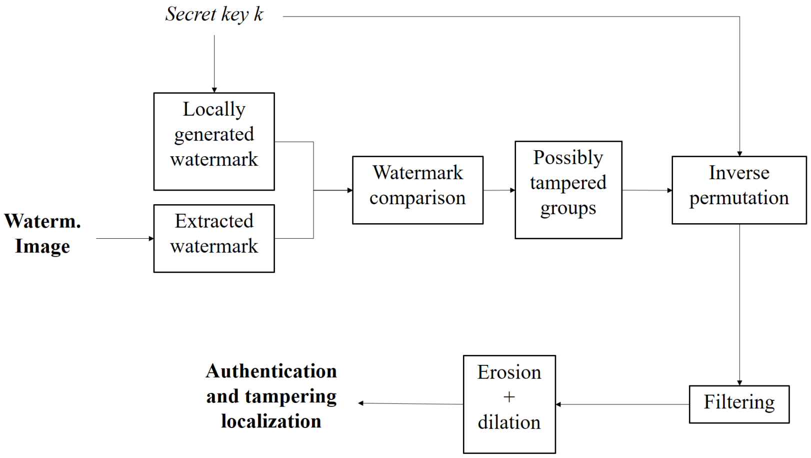


Figure 4.3: Block diagram of the image authentication.

1. The original watermark w is locally generated using the secret key K. If the extracted watermark w' matches the original one, the image is authentic. Otherwise, the following steps determine if the image was manipulated

2. If a bit of the extracted watermark w_i' does not match the original one w_i, all coefficients belonging to group i are flagged as tampered.

3. All coefficients are permuted back to their original position using the inverse permutation with the secret key. Coefficients flagged as potentially tampered should be now spread all over sub-bands. The tampered regions should have a high density of flagged coefficients. The other flagged coefficients should be isolated and distributed like random noise, they are false positives.

4. All the coefficients are permuted to their original position using the inverse permutation with the secret key. coefficients marked as altered are distributed by each of the sub-bands. Tampered regions have a high density of marked

coefficients. The other labeled coefficients must be isolated and distributed as random noise, They are false positives.

5. For a selected resolution level n, there are three sub-bands, LH_n, HL_n and HH_n, with flagged and non-flagged coefficients. It is necessary to create a binary authentication matrix A of the same size $(M/2^n) \times (N/2^n)$ as the sub-bands. $A(x, y) = 1$ if there is a flagged coefficient at the same position (x, y) in any of the HL, LH, or HH sub-bands.

6. Locations incorrectly flagged as tampered are further eliminated by successive erosion and dilation, using a disk of radius R pixels and a square of size $S \times S$ as structural elements. The **1** bits in the filtered and eroded authentication matrix should now correctly indicate the tampered locations.

7. The flagged positions in A are then mapped back to the spatial domain to indicate the actual tampered locations.

Parameter Q is the quantization step size and can be adapted for the sensitivity of the tampering detection, it defines the size of structural element used for erosion and dilation in step 5. To increase the sensitivity a larger Q must be used and will also decrease the probability of false alarms. A solution must be found for specific applications.

Chapter 5

Hardware Architecture

This section will describe hardware implementation details. Unlike the software version, this version has some particular details that must be taken into account for its correct operation.

5.1 Hardware Implementation

The single port memory IP core is generated using Vivado HLS, and generated “bin” file is loaded to the IP core. In the next step, the lifting based wavelet architecture converts spatial domain pixel values in the frequency domain coefficients. The low-frequency coefficients are used for robust watermark embedding and the high-frequency coefficients to embed the fragile watermark. The dual watermark is processed in parallel for pre-processing before embedding. Same way, the pixel values of the original watermark are loaded in single port memory IP core with the help of .bin file. Now, as per the embedding process, the coefficients of the low-frequency band are quantized in watermark embedding architecture, and the high-frequency bands are grouped and permuted to embed the fragile watermark. After the completion of watermarking, the watermarked coefficients are transformed by inverse lifting wavelet. These coefficients are again stored in internal memory. Now, the watermarked image is constructed in the host with stored coefficients. The main stages were adapted such as the chaotic function, discrete wavelet transform, insertion of watermarks. After performing an optimization of data types, cycles and conditionals and use of functions, different policies will be applied to optimize the use of FPGA resources. For this proposed schemes is necessary taking advantage of the little dependence of data for the insertion of both marks, having two parallel stages, with this it is possible reduce the time of execution and latency.

The proposed architecture consists of four modules; image pre-processing, robust watermark insertion, semi-fragile watermark insertion, and the image watermarked reconstruction module. In the pre-processing module, the chaotic function is applied to the image, then 2nd level DWT is applied over chaotic image obtained. At this point, watermark image is necessary, because the robust watermark and the semi-fragile will be inserted at the same time.

In the robust watermark module, low frequency coefficients and the watermark image are received. The watermark image will be coded according to the alpha parameter, while the coefficients are quantized in groups of 8x8. Once quantized a coefficient is marked using the corresponding encoded value. The semi-fragile watermark module follows a sequential flow as explained in chapter 4. When the semi-fragile and robust watermark module are completed, the next stage can be started.

Finally, watermarked coefficients of both modules are taken, the IDWT and the IATM are applied to get the marked image.

5.2 High Level Synthesis Implementation

To design an hardware architecture with Vivado High-Level Synthesis it is important to consider some aspects, using some directives as dynamic memory allocation,

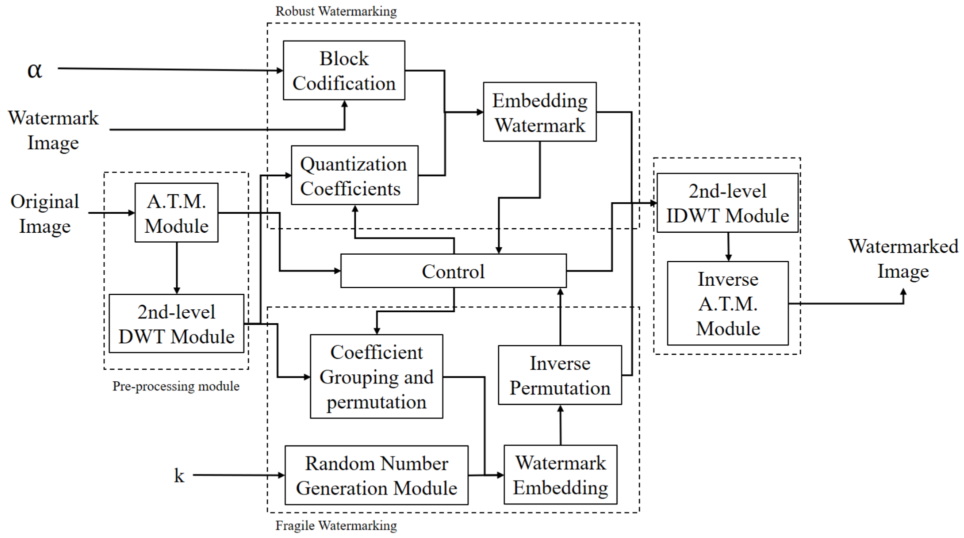


Figure 5.1: Architecture for Proposed Watermarking Scheme

indeterminate loops, write to file, random functions. Vivado HLS determines in which cycle operations should occur (scheduling), and which hardware units to use for each operation (binding), it performs HLS by obeying built in defaults user directives and constraints to override defaults, calculating delays and area using the specified technology/device.

5.2.1 Arnold Transform Map Implementation

This function is similar to software version, with the only difference being that it should initially define the number of iterations required, similarly the implementation of the inverse function, the only notable change is the unrolling done to the number of times to apply the Chaotic function.

5.2.2 Discrete Wavelet Transform Implementation

Several computation schedules have been proposed to implement the 2D DWT. In practical designs, the most commonly used computation schedules are the row–column (RC) the line-based (LB) and the block-based (BB). For this Wavelet implementation, the OpenCV library will be used as the reference, which includes the handling of DWT Haar, which is equal to that implemented in MATLAB. To apply the transform in its hardware implementation, it is essential to divide the task into four parts, given that the method obtain four sub-bands, the idea is to get the calculation separately and save it in an auxiliary variable to later repeat the process. For the second iteration of the wavelet, it is essential to reduce the dimension to the fourth part and apply the same technique previously performed. The previous description is represented on algorithm 2.

To obtain the inverse Wavelet Transform, the function must perform the process in such a way that, the process starts from the last step. In our case it is the second level of the wavelet transform, just like the normal process of the transform, the process must make the calculation for each one of the Sub-bands in parallel and save the result in an auxiliary matrix. Algorithm 3 represents the previous description to perform the inverse transform.

5.2.3 Watermark Embedding Implementation

For the insertion of watermarks, it is necessary consider that both marks do not depend on each other so they can be done in parallel, in addition to the way of inserting the watermark works in blocks, either semi-fragile or the robust one. All this lends itself to add directives after the optimization of the code.

The robust mark requires the quantization of the coefficients, and this operation can be done and then apply the insertion of the mark in the selected coefficient.

Algorithm 2: Implementation in hardware to get wavelet coefficients of an image.

```
function cvHaarWavelet $(src, dst)$;
Input  : Image Pixels $src$
Output: float $dst$
for $y \leftarrow 0;\ y < height;\ y \leftarrow y + 1$ do
    for $x \leftarrow 0;\ x < width;\ x \leftarrow x + 1$ do
        $dst[y][x] \leftarrow$
          $(src[2y][2x] + src[2y][2x+1] + src[2y+1][2x] + src[2y+1][2x+1])0.5;$
        $dst[y][x + width/2] \leftarrow$
          $(src[2y][2x] + src[2y+1][2x] - src[2y][2x+1] - src[2y+1][2x+1])0.5;$
        $dst[y + height/2][x] \leftarrow$
          $(src[2y][2x] + src[2y][2x+1] - src[2y+1][2x] - src[2y+1][2x+1])0.5;$
        $dst[y + height/2][x + width/2] \leftarrow$
          $(src[2y][2x] - src[2y][2x+1] - src[2y+1][2x] + src[2y+1][2x+1])0.5$ ;
    end
end
```

Algorithm 3: Implementation in hardware of Inverse Wavelet Transform.

```
function cvInvHaarWavelet ($src, dst$);
Input  : Wavelet Coefficients $src$
Output: float $dst$
for $y \leftarrow 0;\ y < height;\ y \leftarrow y + 1$ do
    for $x \leftarrow 0;\ x < width;\ x \leftarrow x + 1$ do
        $c \leftarrow src[y][x]$;
        $dh \leftarrow src[y][x + width]$;
        $dv \leftarrow src[y + height][+x]$;
        $dd \leftarrow src[y + height][x + width]$;
        $dst[2y][2x] \leftarrow 0.5 * (c + dh + dv + dd)$;
        $dst[2y][2x + 1] \leftarrow 0.5 * (c - dh + dv - dd)$;
        $dst[2y + 1][2x] \leftarrow 0.5 * (c + dh - dv - dd)$;
        $dst[2y + 1][2x + 1] \leftarrow 0.5 * (c - dh - dv + dd)$;
    end
end
```

Algorithm represents DWT implementation on C++.

$$x^{32} + x^22 + x^2 + 1 \tag{5.1}$$

The semi-fragile mark requires a pseudo random generator for the permutation of the coefficients and the generation of the semi-fragile watermark, in a hardware implementation, it is not possible to make use of the “ rand()” function which is imported by the “stdlib.h” library. For this reason, a shift registry with which are generated the necessary numbers for the permutation and semi-fragile watermark is required. The equation 5.1 and the algorithm 4 represent the behavior of the shift register.

Algorithm 4: Linear Feedback Shift Register (LFSR) used to generate random numbers in hardware.

```
function PseudoRandom ($seed, load$);
Input   : Seed $seed$ Flag to initialize $load$
Output: unsigned int $lfsr$
$staticapUint < 32 > lfsr$ ;
if $load = 1$ then
    $lfsr \leftarrow seed$;
end
$boolb_{32} \leftarrow lfsr.get_{bit}(32 - 32)$;
$boolb_{22} \leftarrow lfsr.get_{bit}(32 - 22)$;
$boolb_{2} \leftarrow lfsr.get_{bit}(32 - 2)$;
$boolb_{1} \leftarrow lfsr.get_{bit}(32 - 1)$;
$boolnew_bit \leftarrow b_{32} \wedge b_{22} \wedge b_2 \wedge b_1$;
$lfsr \leftarrow lfsr >> 1$;
$lfsr.set_bit(31, new_bit)$;
return $lfsr.ToUint()$;
```

5.2.4 Implementation Improvement

Vivado HLS has some ways to improve performance, and it has automatic (and default) optimizations, latency directives, pipelining to allow concurrent operations. Vivado HLS support techniques to remove performance bottlenecks, manipulating loops partitioning and reshaping arrays, optimizations are performed using directives. Vivado HLS will by default minimize latency throughput is prioritized above latency, Vivado HLS will automatically take advantage of the parallelism. It will schedule functions to start as soon as they can.

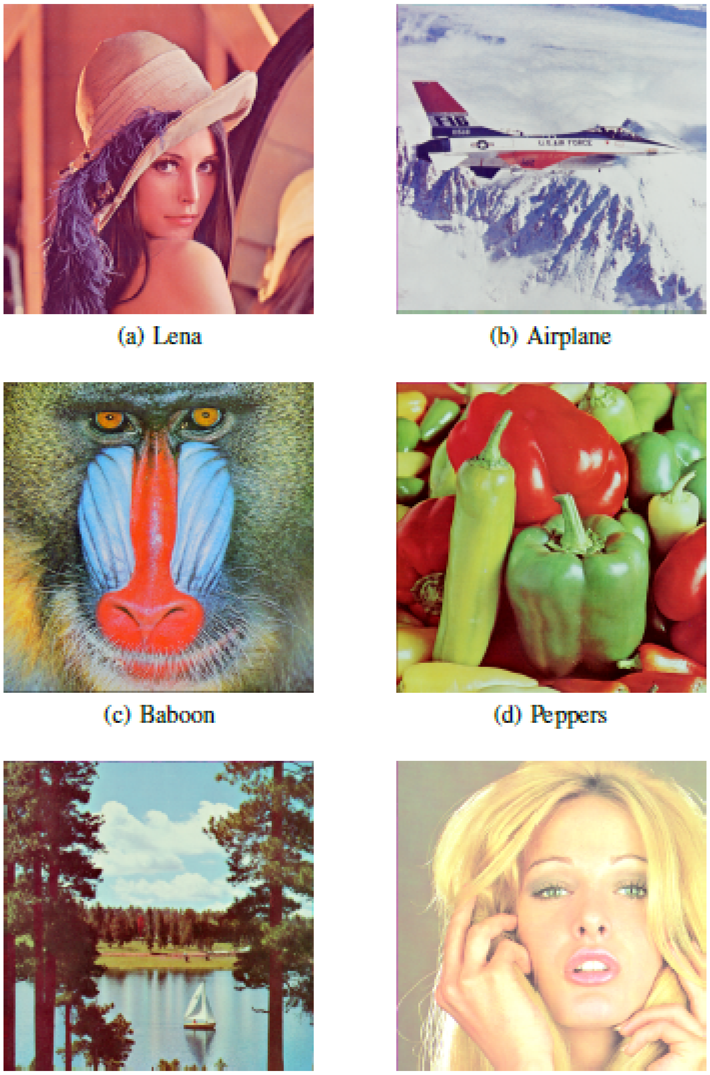
Directives like pipeline and unrolling were applied on Discrete Wavelet Transform and Inverse method. It is also used during the insertion of watermarks, the little dependence of data that exists benefits this directive. On the next chapter, the comparative between original implementation against apply directives over this project is exposed.

Chapter 6

Experimental Results

The purpose of this chapter is to present several experiments that show the effectiveness of the proposed scheme. According to the methods used it is expected that the results on the images will be highly effective against attacks such as compression, noise, low pass filtering, as well as detection and location of altered regions.

Performance results of imperceptibility, fragility, and robustness of the proposed dual watermarking scheme are presented in Subsections 6.1, 6.2, and 6.3, respectively. In Subsection 6.4, functionality comparisons with related studies are presented to demonstrate the superiority of the proposed scheme. The experiments were performed with eight commonly used color images, i.e., “Lena”, “Airplane”, “Baboon”, “Peppers”, “Lake” and “Tiffany” and images from the USC-SIPI database [of Southern-California, 1977]. All of the images have the same size, i.e. 512×512, as shown in Fig. 6.1. Although all eight images were tested during the experiment, most of the results presented in this section are based on “Lena” for succinct presentation. The robust watermark embedded in the test images was a 128×128-sized logo image. The embedded fragile watermark was a random binary bitstream and the parameter n was chosen as 2 in our experiments.



(a) Lena

(b) Airplane

(c) Baboon

(d) Peppers

(e) Lake

(f) Tiffany

Figure 6.1: Test Images

6.1 Results Imperceptibility

In the experiments, two image quality assessment (IQA) metrics (i.e. peak signal-to-noise ratio (PSNR) and structural similarity (SSIM)) were used to measure the imperceptibility performance of the dual watermarking scheme. The PSNR is a conventional IQA metric which operate directly on the pixel-based stage of the images.

The parameter α is a crucial parameter of the proposed dual watermarking scheme. A higher α can increase the imperceptibility of the embedded watermark, but it makes the watermarked image less robust against attacks. Therefore, the imperceptibility performance of the dual watermarking scheme is tested firstly with various α from value 1 to 16. The results of the IQA metrics of the proposed scheme with various α are shown in Fig. 6.2. The results of PSNR values in Fig. 6.2 show an inverse relationship with α, but all of the PSNR values are greater than 30 dB regardless of the α value. The average PSNR value of the eight dual watermarked images is highest (nearly 45dB) when α is 4, and it decreases to approximate 30 dB when α is 1. Moreover, there is very little difference between the PSNR values for the eight test images with the same α value, although each test image has different characteristics.

6.2 Results of Fragile Watermarking

The embedding and extraction of the authentication watermark is done in the DWT domain. This technique achieves high image quality and high tampering detection resolution at a low watermark payload.

Fig. 6.4 shows the refined authentication result after filtering and mathematical morphology operations. In Fig. 6.4 the authentication results before and after refinement using the following parameters: $n = 2$, $d = 8$ and $Q = 8$ are exposed.

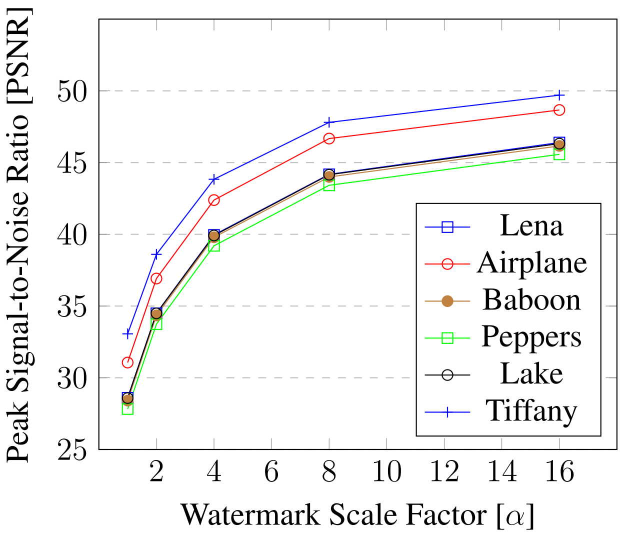


Figure 6.2: PSNR values of the dual watermarked images with various α

In this case, the detection resolution of the algorithm is $2n \times 2n = 4 \times 4$ pixels. In both cases, the proposed scheme detects the tampered regions correctly.

Next, the capacity of our approach to detect the tampering is tested. For this purpose, in the image in Fig. 6.4 shows, some malicious attacks were performed on the test image “Lena”, including adding an extra object, adding a fake watermark, and blurring Lena’s eyes. Fig. 6.4(b) shows the results of image authentication and tampered area localization. Fig. 6.4(c) shows the difference between an attacked image and an authenticated image. The results displayed in Figs. 6.4(b) and (c) show that the tampered area of the attacked image can be located accurately (nearly 100%), irrespective of the attack that is performed.

The experimental results also implied in the proposed scheme is indeed fragile enough to authenticate the attacked image and locate the tampered area accurately.

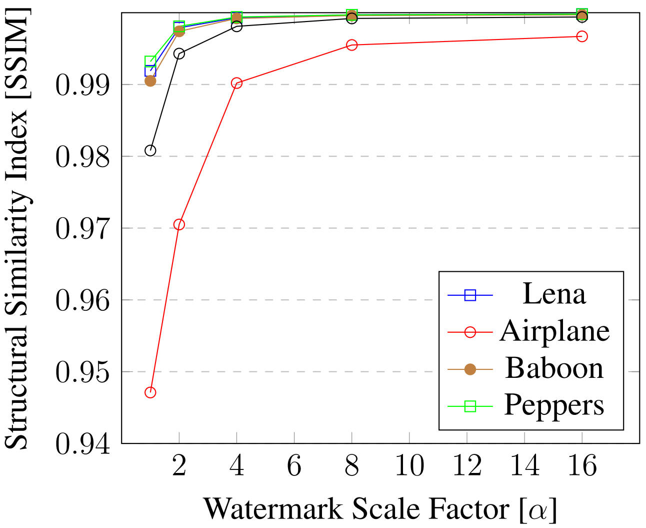


Figure 6.3: SSIM values of the dual watermarked images with various α

6.3 Results of Robust Watermarking

Robustness is a significant concern for copyright protection schemes. In this subsection, this work report the results of several signal processing attacks that were performed on the test image "Lena" to demonstrate the robustness of our watermarking scheme. The signal processing attacks that were performed in our experiments were adding an object, JPEG compression, salt and pepper, gaussian noise, brighten, darken, resizing, cropping, blurring, contrast, tone mapping, and twisting. After extracting the watermark, the well-known metric normalized correlation coefficient (NC) was computed using the original watermark and the extracted watermark to measure the robustness of the watermarking scheme. Robustness and security are two most important properties that a watermarking scheme should hold. Our scheme is robust, as after all the attacks the extracted watermarks are visually recognizable and all the extracted watermarks are very close to the original watermark.

Fig. 6.5 illustrates the images subjected to the above-mentioned attacks with given parameter and the visual inspection of the extracted watermark images. An ex-

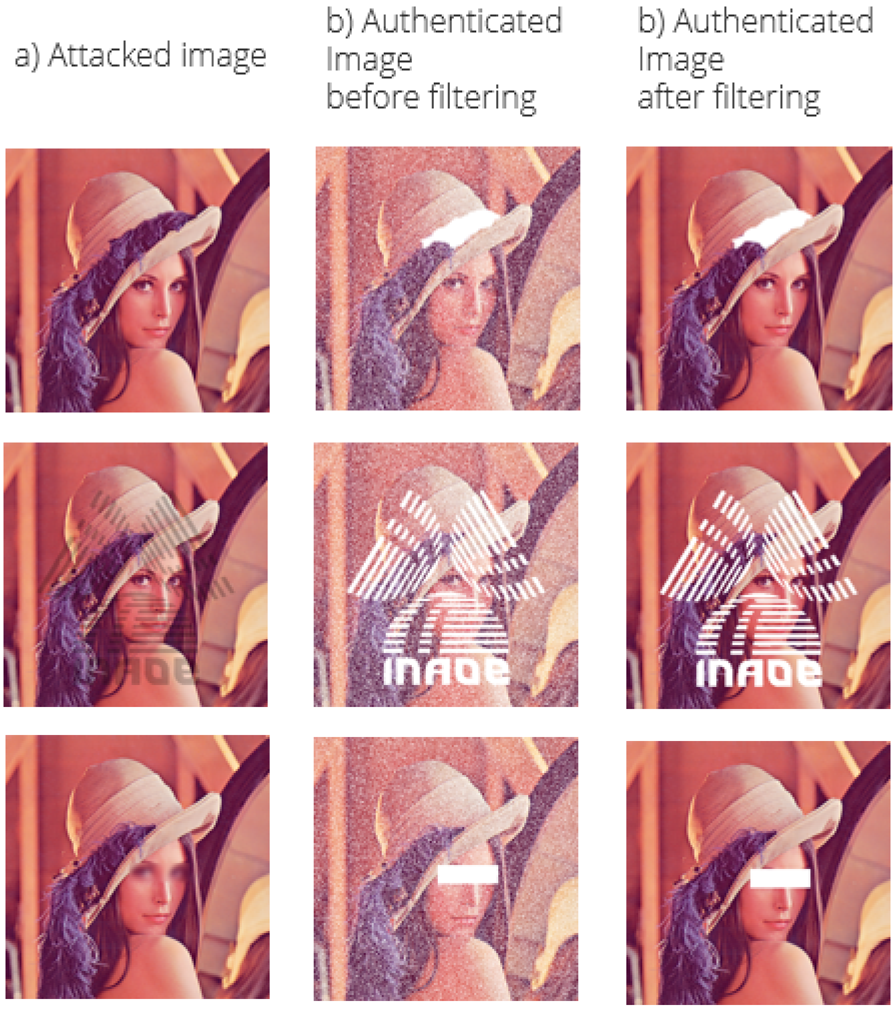

a) Attacked image
b) Authenticated Image before filtering
b) Authenticated Image after filtering

Figure 6.4: Attacked image and its corresponding authentication results

ception was that the extracted watermark image was not very satisfactory after performed the resizing attack on the test image (NC is 0.8652). This is because the contrast attack smoothed the detailed part of the watermarked image. The proposed robust watermarking scheme was highly correlated with the LL sub-band of watermarked image's B channel, and distorting it excessively would decrease the robustness of the watermark. Hence, It is necessary to define a higher value of parameter k in our robust watermarking scheme in order to resist resizing attacks. In general, the proposed scheme is robust against most malicious attacks, and the extracted watermark images are still recognizable by the human eye. This experimental result demonstrated that the proposed scheme can protect against the most common attacks and is beneficial in protecting the copyright of valuable images.

To highlight the robustness of the proposed scheme, the parameters of each attack were also tested and a quantitative comparison with other similar schemes was presented. Table 6.1 lists the IQA and NC results compared with similar robust watermarking schemes under some common attack types. All of compared robust watermarking schemes based on frequency domain except [Lusson et al., 2013] scheme, which is based on spatial domain. Therefore, [Lusson et al., 2013] scheme is not robust to most of the common attacks. Compared with [Su et al., 2013] scheme , both of our schemes can resist against the common attacks with outstanding NC results. However, the imperceptibility performance of [Su et al., 2013] scheme is satisfactory, especially for SSIM result. They embed the robust watermark by the modified U matrix of singular value decomposition (SVD) in original image, which would generate lots of obvious horizontal stripe in the watermarked image. Therefore, after considering robustness and imperceptibility performance of the watermarked image adaptively. As it can be seen from the above comparison, the overall imperceptibility and robustness performance of the proposed method outperform other similar schemes.

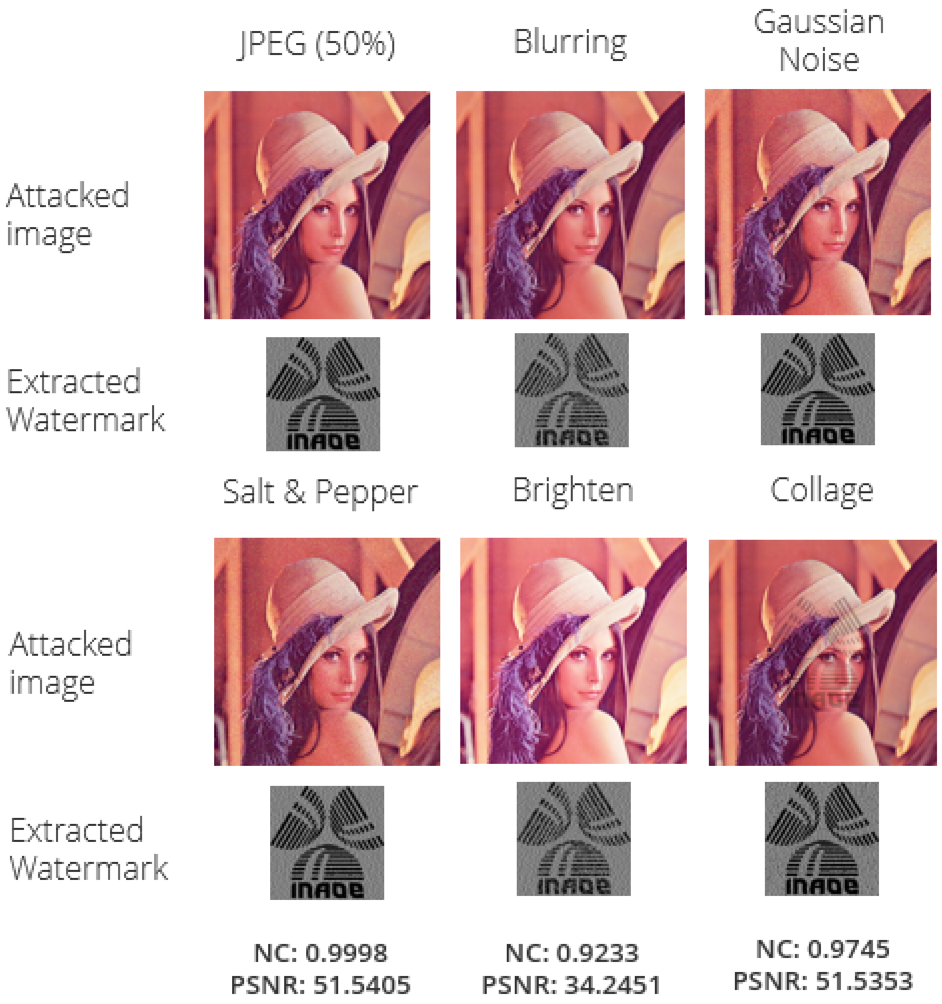


Figure 6.5: Results of Robust Watermark

Table 6.1: IQA and NC comparison results of related robust watermarking schemes part 1

Atacks	Parameters	Lusson et al	Su et al.	Liu et al	Proposed Scheme
		PSNR = 38.97 SSIM = 0.9793	PSNR = 36.30 SSIM = 0.9050	PSNR = 40.85 SSIM = 0.9814	PSNR = 40.57 SSMI = 0.9853
Salt & Pepper	0.01	0.9986	0.9952	0.9990	**0.9999**
	0.02	0.9946	0.9889	0.9981	**0.9999**
	0.04	0.9913	0.9847	0.9959	**0.9999**
	0.08	0.9898	0.9600	0.9893	**0.9998**
JPEG	90	0.9118	0.9988	0.9967	**0.9999**
	80	0.8976	0.9950	0.9842	**0.9988**
	70	0.8873	0.9907	0.9711	0.9891
	60	0.8844	0.9860	0.8687	0.9839
Blurring	0.1	0.9998	1	0.9997	0.998
	0.2	0.4436	0.9877	0.9514	0.9832
	0.3	0.3279	0.9393	0.8735	**0.9567**
Gaussian Noise	0.1	0.9664	0.9767	0.9664	**0.9823**
	0.3	0.9334	0.9003	0.9171	**0.9612**
	0.5	0.9064	0.8697	0.8735	**0.9148**
Tone mapping	Auto	0.0414	0.9999	0.9999	0.8039

Table 6.2: IQA and NC comparison results of related robust watermarking schemes part 2

Atacks	Parameters	Liu et al PSNR = 40.85 WI: 64x64	Proposed PSNR = 48.09 WI: 64x64	Liu et al PSNR =30.2976 WI: 128x128	Proposed PSNR = 40.57 WI : 128x128
Salt & Pepper	0.01	51.5974	60.9707	42.2929	52.8168
	0.02	51.5190	60.8696	42.7460	52.4891
	0.04	51.3668	60.5370	41.9735	52.4191
	0.08	50.9803	60.3754	40.8881	52.3902
JPEG	90	52.7883	60.0185	42.2312	52.9608
	80	50.7277	59.7916	41.1043	52.8967
	70	49.4110	59.3371	40.6306	52.5589
	60	39.814	58.5757	32.2173	52.3860
Blurring	0.1	51.2148	60.1284	42.2096	52.2818
	0.2	49.9401	59.6844	40.7805	49.89793
	0.3	34.5511	59.3436	26.9115	45.1855
Gaussian Noise	0.1	49.7543	59.3613	41.4029	49.4450
	0.3	44.8443	57.3481	32.5334	47.9616
	0.5	35.0657	53.5814	23.9042	41.6337
Tone Mapping	Auto	58.8954	37.4764	43.5766	28.0487

Table 6.3: PSNR comparison results of proposed scheme and [Liu et al., 2016b] robust watermarking schemes

Atacks	Parameters	Liu et al PSNR = 40.85	Proposed PSNR = 48.09	Liu et al PSNR =30.2976	Proposed PSNR = 40.57
		WI: 64x64	WI: 64x64	WI: 128x128	WI : 128x128
Salt & Pepper	0.01	51.5974	60.9707	42.2929	52.8168
	0.02	51.5190	60.8696	42.7460	52.4891
	0.04	51.3668	60.5370	41.9735	52.4191
	0.08	50.9803	60.3754	40.8881	52.3902
JPEG	90	52.7883	60.0185	42.2312	52.9608
	80	50.7277	59.7916	41.1043	52.8967
	70	49.4110	59.3371	40.6306	52.5589
	60	39.814	58.5757	32.2173	52.3860
Blurring	0.1	51.2148	60.1284	42.2096	52.2818
	0.2	49.9401	59.6844	40.7805	49.89793
	0.3	34.5511	59.3436	26.9115	45.1855
Gaussian Noise	0.1	49.7543	59.3613	41.4029	49.4450
	0.3	44.8443	57.3481	32.5334	47.9616
	0.5	35.0657	53.5814	23.9042	41.6337
Tone Mapping	Auto	58.8954	37.4764	43.5766	28.0487

6.4 Comparisons among Dual Watermarking Schemes

In this subsection, the superior performance of the proposed dual watermarking schemes is demonstrated by comparing its functionality with that of the related well-known dual watermarking schemes. Table 6.4 illustrates the different functionalities of the dual watermarking schemes.

Table 6.4: Comparasions of the Funcionalities of out Dual Watermarking Scheme and Related Works

Functionality	Lu and Liao (2001)	Lin et al. (2009)	Lusson et al (2013)	Liu et Al (2016)	Proposed Scheme
Dual watermarks	Fragile + Robust	Robust + Robust	Robust + Robust	Fragile + Robust	Fragile + Robust
Embedding domain	DWT + DWT	Spatial + DCT	Spatial + Spatial	Spatial + DWT	DWT + DWT
Visibility	Invisible + Invisible	Visible + Invisible	Invisible + Invisible	Invisible + Invisible	Invisible + Invisible
Blind extraction	Yes + No	Yes + Yes	Yes + No	Yes + Yes	Yes + Yes
Target image	Color	Grayscale	Color	Color	Color
PSNR	∼40 dB	∼30 dB	∼39 dB	∼ 40 dB	∼40 dB
Copyright protection	Yes	Yes	Yes	Yes	Yes
Image authentication	Yes	No	No	Yes	Yes

The major difference between [Lu and Liao, 2001] scheme and the proposed scheme present invisible hybrid watermarking for copyright protection and image authentication, whereas [Lin et al., 2009] scheme and [Lusson et al., 2013] scheme concentrate on intensive robust watermarking for copyright protection with no concern about image authentication. Therefore, after integrity of the protected image has been compromised, [Lin et al., 2009] scheme and [Lusson et al., 2013] scheme cannot detect the infringement. However, the infringement can be detected accurately by our proposed scheme and [Lu and Liao, 2001] scheme by using the image authentication scheme. With respect to copyright protection, all four of the dual watermarking schemes can resist multiple common attacks. However, in [Lusson et al., 2013] scheme, the original host image is required to extract the second robust watermark in RGB color space, which is a non-blind watermarking scheme. In terms of the PSNR value of the watermarked image, the proposed scheme achieved the optimal result (nearly 40 dB) among the four schemes that were evaluated and embedding more information on image host. This means that the proposed scheme is suitable for protecting valuable images without attracting the attention of malicious attackers.

6.5 Hardware Architecture Evaluation

In this section, the optimization performed on hardware implementation are evaluated. Version without applying any directive will be compared against the version optimized using the Vivado HLS tool. It will detail how they were used to obtain the optimization. For perform experiments, this experiment is using the same conditions defined at the beginning of this chapter: color images, Image Size 512 × 512, watermark image size 128 × 128. The test Board is Zinq ZedBoard Model “xc7z020clg484” using Vivado High-Level Synthesis 2016.4.

6.5.1 Latency

Table 6.5 represents original solution latencies. It is possible to observe latency is very high compared against Table 6.6, After applying pipeline and unrolling some loops it is possible reduce latencies considerably.

Table 6.5: Original solution latencies

Latency		Interval		Pipeline
min	max	min	max	Type
17512775	17512775	17512776	17512776	none

Table 6.6: Optimized solution latencies

Latency		Interval		Pipeline
min	max	min	max	Type
1287800	1287800	1287800	1287800	none

The tables 6.7 and 6.8 shows the latencies of each module, what can be seen is that the module for the insertion of semi-fragile and robust brands has the lowest latency, both in the original version and in the optimized version, this is due to the Little reliance on existing data. In contrast, processes with a lot of data dependence have a direct impact on the resulting latency, such as the chaotic function. The wavelet transform, on the other hand, being a process involving floating-point operations, has a certain impact on the final result, but the parallelization of the sub-bands calculation benefits a little by reducing latency.

Table 6.7: Detail Original solution latencies

	Latency		Interval	
Module	min	max	min	max
Watermarking	889634	889634	889634	889634
cvInvHaarWavelet	2787588	2787588	2787588	2787588
cvHaarWavelet	3263236	3263236	3263236	3263236
InverseArnoldTransfo	5253130	5253130	5253130	5253130
ArnoldTransformMap	5253130	5253130	5253130	5253130

Table 6.8: Detail Optimized solution latencies

	Latency		Interval	
Module	min	max	min	max
Watermarking	24625	24625	24625	24625
cvInvHaarWavelet	90270	90270	90270	90270
cvHaarWavelet	137835	137835	137835	137835
InverseArnoldTransfo	517535	517535	517535	517535
ArnoldTransformMap	517535	517535	517535	517535

6.5.2 Optimized solution Estimates

The tables 6.9 and 6.10 show the use of resources of test card. The usage of Flip-Flops, DSP modules, and LUTs increase considerably, this is because Vivado HLS makes use of more elements to insert control code for pipeline and to unroll. But this increase can be considered as available since the latencies obtained when applying the directives results in a shorter processing time, which implies that it is possible to process a greater amount of information, either in some elements or image size.

6.5.3 Comparison Between Hardware and Software Implementation

Most hardware implementations of watermarking schemes make their assessments based on the use of resources used. [Joshi et al.,] On the other hand, if it were a video-focused scheme, the comparisons are with the video standard that they handle, the type of design, chip statistics, among others. [Mohanty et al., 2007, Roy et al., 2013]

The concurrent software version was implemented in C ++ with the Visual C ++ 15.0 compiler, to get the times C ++ debugger was used and the execution time in the hardware implementation the execution time was determined using the Xilinx

Table 6.9: Utilization Estimates Original Solution

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	93
FIFO	-	-	-	-
Instance	64	58	12121	19743
Memory	128	-	-	-
Multiplexer	-	-	-	323
Register	-	-	137	-
Total	192	58	12258	20159
Available	280	220	106400	53200
Utilization	68	26	11	37

Table 6.10: Utilization Estimates Optimized solution

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	93
FIFO	-	-	-	-
Instance	64	69	22575	27794
Memory	128	-	-	-
Multiplexer	-	-	-	324
Register	-	-	137	-
Total	192	69	22712	28211
Available	280	220	106400	53200
Utilization	68	31	21	53

cosimulation tool. The tests were performed on a computer with a Core i7 6700, 16 GB RAM and Windows 10 operating system ver. 1607. Disk accesses are not being considered for this test.

To perform a comparison between the software version and the hardware version watermark tests will be done with images of different sizes: 256x256, 512x512 and 1024x1024 from the USC-SIPI database [of Southern-California, 1977]. The robust watermarks will be 64x64, 128x128 and 256x256 respectively. The average watermark time will be displayed for each image size and version of the implemented scheme. The additional parameters for the scheme are: $\alpha = 8$, $k = 31$ and $Q = 12$

Table 6.11: Comparison Between Hardware and Software Implementation

Image Size	Time (miliseconds)		Frames per Second	
	Software Ver.	Hardware Ver.	Software Ver.	Hardware Ver.
256x256	10.738	0. 3649	93.127	2740.47
512x512	39.828	1.2988	25.107	769.94
1024x1024	148.21	4. 7565	6.747	210.238

In the table 6.11 it is possible to observe the behavior in each resolution, this one is climbing in a ratio of four times concerning the previous one. Given these values you can estimate how many frames can be processed, it is possible to affirm that the hardware version is 30 times faster than the one implemented in software. If an implementation for video watermarking is necessary, the frames per second column can give us an idea of the method performance and the proposed hardware implementation.

6.5.4 Hardware Implementation Remarks

Performing hardware implementation and using the appropriate directives, a processing time considerably lower than in its software version can be reached. Recapitulating the data thrown by the synthesis, the watermark module is the one with the shortest execution time, but the modules of the Wavelet function and the chaotic function have the highest latencies. It is considered as future work to have an optimized version in time and hardware resources used.

Chapter 7

Conclusions

This chapter describes the conclusions reached in this research, as well as the possible future modifications or extensions that can be made to the proposed scheme.

7.1 Conclusions

This work presented a dual watermarking scheme for color (and grayscale) image authentication. The invisible fragile and robust watermarks are embedded into the frequency domain of the blue component of RGB color space. The robustness of the scheme is tested by performing various image processing attacks. The extracted secret image is visually recognizable after all attacks which prove the robustness of the method.

This work performed various attacks to measure its performance, robustness is given by scale factor, that is, a high value increases imperceptibility but the robustness is weak, a little value means a good robustness and imperceptibility decreases. This proposed scheme is good against JPEG compression, and it supports 40% compression concerning original image. Mostly, other schemes only report until 60%

compression.

Selected methods proposed are adaptable to various applications and types of images, It's possible to work with other color channel or information to hide.

The research carried out in this thesis is the proposal of a dual watermark method, which is useful in situations of authentication and protection of copyright. Remembering that the general objective is to design a method for the authentication of images and their hardware architecture. It sought that the proposed algorithm has a good performance and at the moment of implementing the architecture to have superior performance compared to the software version. We conclude that this achieved because our scheme allows:

- Use color and gray-scale images with a depth of 8 bits.
- The wavelet transform was used to insert both marks.
- The insertion of the mark allows impacting as little as possible the imperceptibility without compromising the robustness.
- It allows detecting altered regions in addition to supporting the most common attacks like JPEG and low pass filters.

Performing hardware implementation and using the appropriate directives, a processing time considerably lower than in its software version has been reached. Recapitulating the data thrown by the synthesis, the watermark module is the one with the shortest execution time, but the modules of the Wavelet function and the chaotic function have the highest latencies.

7.2 Contributions

The principal contribution of this work is that our proposed scheme can reach copyright protection and authentication at the same time. According to the experimental results, the proposed watermarking scheme obtains similar results regarding imperceptibility and robustness but embedding more information than others schemes.

This work can achieve copyright protection and image authentication simultaneously, and the extraction of watermarks from the protected image can be processed blindly without the original host image and watermarks.

7.3 Future Work

From the present work the following ideas are presented as possible future work.

- Improve pre-processing image phase: methods as Discrete Wavelet Transform, Arnold Transform Map, and theirs inverse functions, because these phases have a latency very high compared watermark module.
- Implement algorithm in various types of image format, the number of bits per pixel, channels, color schemes. Like YCbCr color space.
- We propose the analysis and estimation of the error rate, to guarantee the performance of the method theoretically and not only through experimentation.
- Implement the method in different image formats, bit depth per pixel, number of channels as well as their implementation in video; The input can be considered as a sequence of images.

List of Acronyms

ATM Arnold Transform Map.

IP Intellectual Property.

IPR Intellectual Property Rights.

MAE Mean Absolute Error.

MSE Mean Squared Error.

NC Normalized cross-correlation.

PSNR Peak Signal-to-Noise Ratio.

RMSE Root-Mean-Square Error.

SNR Signal-to-Noise Ratio.

TA Trusted Authority.

VCS Visual Cryptography Schemes.

Bibliography

[Abdel-Hamid et al., 2005] Abdel-Hamid, A. T., Tahar, S., and Aboulhamid, E. M. (2005). A public-key watermarking technique for ip designs. In *Design, Automation and Test in Europe, 2005. Proceedings*, pages 330–335. IEEE.

[Ali et al., 2014] Ali, M., Ahn, C. W., and Pant, M. (2014). A robust image watermarking technique using svd and differential evolution in dct domain. *Optik-International Journal for Light and Electron Optics*, 125(1):428–434.

[Arnold, 2017] Arnold, V. I. (2017). Problemes ergodiques de la mecanique classique.

[Ateniese et al., 1996a] Ateniese, G., Blundo, C., De Santis, A., and Stinson, D. R. (1996a). Constructions and bounds for visual cryptography. In *International Colloquium on Automata, Languages, and Programming*, pages 416–428. Springer.

[Ateniese et al., 1996b] Ateniese, G., Blundo, C., De Santis, A., and Stinson, D. R. (1996b). Visual cryptography for general access structures. *Information and Computation*, 129(2):86–106.

[Barlow, 1961] Barlow, H. B. (1961). Possible principles underlying the transformations of sensory messages.

[Behnia et al., 2014] Behnia, S., Ahadpour, S., and Ayubi, P. (2014). Design and implementation of coupled chaotic maps in watermarking. *Applied Soft Computing*, 21:481–490.

[Bekkouch and Faraoun, 2015] Bekkouch, S. and Faraoun, K. M. (2015). Robust and reversible image watermarking scheme using combined dct-dwt-svd transforms. *Journal of Information Processing Systems*, 11(3).

[Boland et al., 1995] Boland, F., O'Ruanaidh, J. J., and Dautzenberg, C. (1995). Watermarking digital images for copyright protection. In *Image Processing and its Applications, 1995., Fifth International Conference on*, pages 326–330. IET.

[Brown and Rose, 1996] Brown, S. and Rose, J. (1996). Architecture of fpgas and cplds: A tutorial. *IEEE Design and Test of Computers*, 13(2):42–57.

[Brunton and Zhao, 2005] Brunton, A. and Zhao, J. (2005). Real-time video watermarking on programmable graphics hardware. In *Electrical and Computer Engineering, 2005. Canadian Conference on*, pages 1312–1315. IEEE.

[Cho and Lee, 1990] Cho, N. I. and Lee, S. U. (1990). Dct algorithms for vlsi parallel implementations. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(1):121–127.

[Chourasia, 2013] Chourasia, J. (2013). Identification and authentication using visual cryptography based fingerprint watermarking over natural image. *CSI transactions on ICT*, 1(4):343–348.

[Chrysochos et al., 2014] Chrysochos, E., Fotopoulos, V., Xenos, M., and Skodras, A. N. (2014). Hybrid watermarking based on chaos and histogram modification. *Signal, Image and Video Processing*, 8(5):843–857.

[Cimato and Yang, 2011] Cimato, S. and Yang, C.-N. (2011). *Visual cryptography and secret image sharing.* CRC press.

[CISCO, 2016] CISCO (2016). The zettabyte era — trends and analysis – cisco.

[Coatrieux et al., 2000] Coatrieux, G., Maitre, H., Sankur, B., Rolland, Y., and Collorec, R. (2000). Relevance of watermarking in medical imaging. In *Information*

Technology Applications in Biomedicine, 2000. Proceedings. 2000 IEEE EMBS International Conference on, pages 250–255. IEEE.

[Coussy and Morawiec, 2008] Coussy, P. and Morawiec, A. (2008). *High-level synthesis: from algorithm to digital circuit*. Springer Science & Business Media.

[Cox et al., 2002] Cox, I. J., Miller, M. L., Bloom, J. A., and Honsinger, C. (2002). *Digital watermarking*, volume 1558607145. Springer.

[Das et al., 2014] Das, C., Panigrahi, S., Sharma, V. K., and Mahapatra, K. (2014). A novel blind robust image watermarking in dct domain using inter-block coefficient correlation. *AEU-International Journal of Electronics and Communications*, 68(3):244–253.

[Devi et al., 2012] Devi, B. P., Singh, K. M., and Roy, S. (2012). Dual image watermarking scheme based on singular value decomposition and visual cryptography in discrete wavelet transform. *International Journal of Computer Applications*, 50(12).

[ElAraby et al., 2010] ElAraby, W. S., Madian, A. H., Ashour, M. A., and Wahdan, A. M. (2010). Hardware realization of dc embedding video watermarking technique based on fpga. In *2010 International Conference on Microelectronics*, pages 463–466. IEEE.

[Eskicioglu and Fisher, 1995] Eskicioglu, A. M. and Fisher, P. S. (1995). Image quality measures and their performance. *IEEE Transactions on communications*, 43(12):2959–2965.

[Farooq et al., 2012] Farooq, U., Marrakchi, Z., and Mehrez, H. (2012). Fpga architectures: An overview. *Tree-based Heterogeneous FPGA Architectures*, pages 7–48.

[Fazli and Moeini, 2016] Fazli, S. and Moeini, M. (2016). A robust image watermarking method based on dwt, dct, and svd using a new technique for correction of main geometric attacks. *Optik-International Journal for Light and Electron Optics*, 127(2):964–972.

[Gajski et al., 2012] Gajski, D. D., Dutt, N. D., Wu, A. C., and Lin, S. Y. (2012). *High—Level Synthesis: Introduction to Chip and System Design*. Springer Science & Business Media.

[Gavini and Borra, 2014] Gavini, N. S. and Borra, S. (2014). Lossless watermarking technique for copyright protection of high resolution images. In *Region 10 Symposium, 2014 IEEE*, pages 73–78. IEEE.

[Ghosh et al., 2014a] Ghosh, S., Biswas, A., Maity, S. P., and Rahaman, H. (2014a). Design of a low complexity and fast hardware architecture for digital image watermarking in fwht domain on fpga. In *Electronic System Design (ISED), 2014 Fifth International Symposium on*, pages 68–72. IEEE.

[Ghosh et al., 2014b] Ghosh, S., Das, N., Das, S., Maity, S. P., and Rahaman, H. (2014b). Digital design and pipelined architecture for reversible watermarking based on difference expansion using fpga. In *Information Technology (ICIT), 2014 International Conference on*, pages 123–128. IEEE.

[Gonzalez et al., 2004] Gonzalez, R. C., Woods, R. E., and Eddins, S. L. (2004). Digital image using matlab processing. *Person Prentice Hall, Lexington.*

[Han et al., 2014] Han, Y., He, W., Ji, S., and Luo, Q. (2014). A digital watermarking algorithm of color image based on visual cryptography and discrete cosine transform. pages 525–530.

[Han et al., 2013] Han, Y., Shang, Y., and He, W. (2013). Dwt-domain dual watermarking algorithm of color image based on visual cryptography. pages 373–378.

[Hartung and Kutter, 1999] Hartung, F. and Kutter, M. (1999). Multimedia watermarking techniques. *Proceedings of the IEEE*, 87(7):1079–1107.

[Hu et al., 2016] Hu, H.-T., Chang, J.-R., and Hsu, L.-Y. (2016). Robust blind image watermarking by modulating the mean of partly sign-altered dct coefficients guided by human visual perception. *AEU-International Journal of Electronics and Communications*, 70(10):1374–1381.

[Hyvärinen and Muszynski, 2008] Hyvärinen, M. and Muszynski, L. (2008). Introduction. In *Terror and the Arts*, pages 1–22. Springer.

[Jadhav et al., 2015] Jadhav, A. A., Babar, R., and Gaikwad, M. (2015). Hardware implementation of digital watermarking system for real time captured image transmitting. In *Pervasive Computing (ICPC), 2015 International Conference on*, pages 1–4. IEEE.

[Ji et al., 2015] Ji, K., Lin, J., Li, H., Wang, A., and Tang, T. (2015). A dct and svd based watermarking technique to identify tag. *arXiv preprint arXiv:1502.02969*.

[Joshi et al.,] Joshi, A., Bapna, M., Malpani, A., Goyal, A. K., and Meena, M. Hardware implementation of image and video watermarking for ownership verification.

[Joshi et al., 2011] Joshi, A. M., Darji, A., and Mishra, V. (2011). Design and implementation of real-time image watermarking. In *Signal Processing, Communications and Computing (ICSPCC), 2011 IEEE International Conference on*, pages 1–5. IEEE.

[Joshi et al., 2015] Joshi, A. M., Mishra, V., and Patrikar, R. (2015). Design of real-time video watermarking based on integer dct for h. 264 encoder. *International Journal of Electronics*, 102(1):141–155.

[Joshi et al., 2016] Joshi, A. M., Mishra, V., and Patrikar, R. M. (2016). Fpga prototyping of video watermarking for ownership verification based on h. 264/avc. *Multimedia Tools and Applications*, 75(6):3121.

[Kahng et al., 1998] Kahng, A. B., Lach, J., Mangione-Smith, W. H., Mantik, S., Markov, I. L., Potkonjak, M., Tucker, P., Wang, H., and Wolfe, G. (1998). Watermarking techniques for intellectual property protection. In *Design Automation Conference, 1998. Proceedings*, pages 776–781. IEEE.

[Kaur and Sidhu, 2016] Kaur, G. and Sidhu, G. (2016). Image watermarking scheme using combined dct-dwt-svd transforms. *Imperial Journal of Interdisciplinary Research*, 2(9).

[Khalili, 2015] Khalili, M. (2015). Dct-arnold chaotic based watermarking using jpeg-ycbcr. *Optik-International Journal for Light and Electron Optics*, 126(23):4367–4371.

[Khan et al., 2013] Khan, M. I., Rahman, M., Sarker, M., Hasan, I., et al. (2013). Digital watermarking for image authenticationbased on combined dct, dwt and svd transformation. *arXiv preprint arXiv:1307.6328*.

[Kumar et al., 2016] Kumar, A., Agarwal, P., and Choudhary, A. (2016). A digital image watermarking technique using cascading of dct and biorthogonal wavelet transform. In *Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing*, pages 21–29. Springer.

[Lach et al., 1999] Lach, J., Mangione-Smith, W. H., and Potkonjak, M. (1999). Robust fpga intellectual property protection through multiple small watermarks. In *Design Automation Conference, 1999. Proceedings. 36th*, pages 831–836. IEEE.

[Lakshmi and Surekha, 2016] Lakshmi, H. and Surekha, B. (2016). Asynchronous implementation of reversible image watermarking using mousetrap pipelining. In *Advanced Computing (IACC), 2016 IEEE 6th International Conference on*, pages 529–533. IEEE.

[Lei et al., 2014] Lei, B., Tan, E.-L., Chen, S., Ni, D., Wang, T., and Lei, H. (2014). Reversible watermarking scheme for medical image based on differential evolution. *Expert Systems with Applications*, 41(7):3178–3188.

[Li and Xu, 2005] Li, B. and Xu, J.-w. (2005). Period of arnold transformation and its application in image scrambling. *Journal of Central South University of Technology*, 12(1):278–282.

[Li and Qin, 2013] Li, C. and Qin, Z. (2013). A blind digital image watermarking algorithm based on dct. In *Smart and Sustainable City 2013 (ICSSC 2013), IET International Conference on*, pages 446–448. IET.

[Li et al., 2006] Li, Q., Memon, N., and Sencar, H. T. (2006). Security issues in watermarking applications-a deeper look. In *Proceedings of the 4th ACM international workshop on Contents protection and security*, pages 23–28. ACM.

[Lin et al., 2009] Lin, P.-Y., Lee, J.-S., and Chang, C.-C. (2009). Dual digital watermarking for internet media based on hybrid strategies. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(8):1169–1177.

[Liu et al., 2008] Liu, C., Szeliski, R., Kang, S. B., Zitnick, C. L., and Freeman, W. T. (2008). Automatic estimation and removal of noise from a single image. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):299–314.

[Liu et al., 2016a] Liu, H., Xiao, D., Zhang, R., Zhang, Y., and Bai, S. (2016a). Robust and hierarchical watermarking of encrypted images based on compressive sensing. *Signal Processing: Image Communication*, 45:41–51.

[Liu et al., 2016b] Liu, X.-L., Lin, C.-C., and Yuan, S.-M. (2016b). Blind dual watermarking for color images' authentication and copyright protection. *IEEE Transactions on Circuits and Systems for Video Technology.*

[Lu and Liao, 2001] Lu, C.-S. and Liao, H.-Y. (2001). Multipurpose watermarking for image authentication and protection. *IEEE transactions on image processing*, 10(10):1579–1592.

[Lusson et al., 2013] Lusson, F., Bailey, K., Leeney, M., and Curran, K. (2013). A novel approach to digital watermarking, exploiting colour spaces. *Signal Processing*, 93(5):1268–1294.

[Makbol et al., 2016] Makbol, N. M., Khoo, B. E., and Rassem, T. H. (2016). Block-based discrete wavelet transform-singular value decomposition image watermarking scheme using human visual system characteristics. *IET Image Processing*, 10(1):34–52.

[Martin and Smith, 2009] Martin, G. and Smith, G. (2009). High-level synthesis: Past, present, and future. *IEEE Design & Test of Computers*, 26(4):18–25.

[Mathai et al., 2003] Mathai, N. J., Kundur, D., and Sheikholeslami, A. (2003). Hardware implementation perspectives of digital video watermarking algorithms. *IEEE Transactions on Signal Processing*, 51(4):925–938.

[Memik et al., 2005] Memik, S. O., Kastner, R., Bozorgzadeh, E., and Sarrafzadeh, M. (2005). A scheduling algorithm for optimization and early planning in high-level synthesis. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 10(1):33–57.

[Mishra et al., 2014] Mishra, A., Agarwal, C., Sharma, A., and Bedi, P. (2014). Optimized gray-scale image watermarking using dwt–svd and firefly algorithm. *Expert Systems with Applications*, 41(17):7858–7867.

[Mohanty and Kougianos, 2011] Mohanty, S. P. and Kougianos, E. (2011). Real-time perceptual watermarking architectures for video broadcasting. *Journal of Systems and Software*, 84(5):724–738.

[Mohanty et al., 2007] Mohanty, S. P., Kougianos, E., and Ranganathan, N. (2007). Vlsi architecture and chip for combined invisible robust and fragile watermarking. *IET Computers & Digital Techniques*, 1(5):600–611.

[Naor and Shamir, 1994] Naor, M. and Shamir, A. (1994). Visual cryptography. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 1–12. Springer.

[Ni et al., 2008] Ni, R., Ruan, Q., and Zhao, Y. (2008). Pinpoint authentication watermarking based on a chaotic system. *Forensic Science International*, 179(1):54–62.

[Nie et al., 2013] Nie, T., Zhou, L., and Li, Y. (2013). Hierarchical watermarking method for fpga ip protection. *IETE Technical Review*, 30(5):367–374.

[of Southern-California, 1977] of Southern-California, U. (1977). Sipi data base. sipi.usc.edu/database/.

[Pereira et al., 2001] Pereira, S., Voloshynovskiy, S., Madueno, M., Marchand-Maillet, S., and Pun, T. (2001). Second generation benchmarking and application oriented evaluation. In *International Workshop on Information Hiding*, pages 340–353. Springer.

[Petitcolas et al., 1998] Petitcolas, F. A., Anderson, R. J., and Kuhn, M. G. (1998). Attacks on copyright marking systems. In *International workshop on information hiding*, pages 218–238. Springer.

[Poli et al., 2007] Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimization. *Swarm intelligence*, 1(1):33–57.

[Preda and Vizireanu, 2015] Preda, R. and Vizireanu, D. (2015). Watermarking-based image authentication robust to jpeg compression. *Electronics Letters*, 51(23):1873–1875.

[Preda, 2013] Preda, R. O. (2013). Semi-fragile watermarking for image authentication with sensitive tamper localization in the wavelet domain. *Measurement*, 46(1):367–373.

[Qi and Xin, 2011] Qi, X. and Xin, X. (2011). A quantization-based semi-fragile watermarking scheme for image content authentication. *Journal of visual communication and image representation*, 22(2):187–200.

[Ram et al., 2012] Ram, D., Bhuvaneswari, M., and Prabhu, S. S. (2012). A novel framework for applying multiobjective ga and pso based approaches for simultaneous area, delay, and power optimization in high level synthesis of datapaths. *VLSI design*, 2012:2.

[Rawat and Raman, 2011] Rawat, S. and Raman, B. (2011). A chaotic system based fragile watermarking scheme for image tamper detection. *AEU-International Journal of Electronics and Communications*, 65(10):840–847.

[Rawat and Raman, 2012] Rawat, S. and Raman, B. (2012). A blind watermarking algorithm based on fractional fourier transform and visual cryptography. *Signal Processing*, 92(6):1480–1491.

[Reyes et al., 2010] Reyes, R., Cruz, C., Nakano-Miyatake, M., and Perez-Meana, H. (2010). Digital video watermarking in dwt domain using chaotic mixtures. *IEEE Latin America Transactions*, 8(3):304–310.

[Roy et al., 2013] Roy, S. D., Li, X., Shoshan, Y., Fish, A., and Yadid-Pecht, O. (2013). Hardware implementation of a digital watermarking system for video authentication. *IEEE transactions on circuits and systems for video technology*, 23(2):289–301.

[Ruderman, 1997] Ruderman, D. L. (1997). Origins of scaling in natural images. *Vision research*, 37(23):3385–3398.

[Saffor et al., 2001] Saffor, A., Ramli, A. R., and Ng, K.-H. (2001). A comparative study of image compression between jpeg and wavelet. *Malaysian Journal of computer science*, 14(1):39–45.

[Sava et al., 1997] Sava, H., Fleury, M., Downton, A., and Clark, A. (1997). Parallel pipeline implementation of wavelet transforms. *IEE Proceedings-Vision, Image and Signal Processing*, 144(6):355–360.

[Sencar and Memon, 2005] Sencar, H. T. and Memon, N. (2005). Watermarking and ownership problem: a revisit. In *Proceedings of the 5th ACM workshop on Digital rights management*, pages 93–101. ACM.

[Shoshan et al., 2008] Shoshan, Y., Fish, A., Li, X., Jullien, G., and Yadid-Pecht, O. (2008). Vlsi watermark implementations and applications.

[Shyu, 2015] Shyu, S. J. (2015). Visual cryptograms of random grids for threshold access structures. *Theoretical Computer Science*, 565:30–49.

[Singh et al., 2014] Singh, A. K., Dave, M., and Mohan, A. (2014). Hybrid technique for robust and imperceptible image watermarking in dwt–dct–svd domain. *National Academy Science Letters*, 37(4):351–358.

[Singh et al., 2015] Singh, A. K., Dave, M., and Mohan, A. (2015). Hybrid technique for robust and imperceptible multiple watermarking using medical images. *Multimedia Tools and Applications*, pages 1–21.

[Singh and Lamba, 2015] Singh, G. and Lamba, M. S. (2015). Efficient hardware implementation of image watermarking using dwt and aes algorithm. In *Systems Conference (NSC), 2015 39th National*, pages 1–6. IEEE.

[Sleit and Abusitta, 2008] Sleit, A. and Abusitta, A. (2008). A visual cryptography based watermark technology for individual and group images. *Systemics, Cybernetics And Informatics*, 5(2):24–32.

[Su et al., 2013] Su, Q., Niu, Y., Zou, H., and Liu, X. (2013). A blind dual color images watermarking based on singular value decomposition. *Applied Mathematics and Computation*, 219(16):8455–8466.

[Surekha and Swamy, 2012] Surekha, B. and Swamy, G. (2012). A semi-blind image watermarking based on discrete wavelet transform and secret sharing. In *Communication, Information & Computing Technology (ICCICT), 2012 International Conference on*, pages 1–5. IEEE.

[Tocci, 2007] Tocci, R. J. (2007). *Digital systems : principles and applications.* Pearson Prentice Hall.

[Tsai and Lu, 2001] Tsai, T. and Lu, C. (2001). A systems level design for embedded watermark technique using dsc systems. In *Proceedings of the IEEE international workshop on intelligent signal processing and communication systems.*

[Voyatzis and Pitas, 1996] Voyatzis, G. and Pitas, I. (1996). Applications of toral automorphisms in image watermarking. In *Image Processing, 1996. Proceedings., International Conference on*, volume 2, pages 237–240. IEEE.

[Wang et al., 2015] Wang, B., Zhou, S., Zheng, X., Zhou, C., Dong, J., and Zhao, L. (2015). Image watermarking using chaotic map and dna coding. *Optik-International Journal for Light and Electron Optics*, 126(24):4846–4851.

[Wang et al., 2009] Wang, F.-H., Pan, J.-S., and Jain, L. C. (2009). *Innovations in digital watermarking techniques*, volume 232. Springer.

[Weiss and Freeman, 2007] Weiss, Y. and Freeman, W. T. (2007). What makes a good model of natural images? In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.

[Wong, 1998] Wong, P. W. (1998). A public key watermark for image verification and authentication. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 455–459. IEEE.

[Wu and Shih, 2007] Wu, Y.-T. and Shih, F. Y. (2007). Digital watermarking based on chaotic map and reference register. *Pattern Recognition*, 40(12):3753–3763.

[Xia et al., 1997] Xia, X.-G., Boncelet, C. G., and Arce, G. R. (1997). A multiresolution watermark for digital images. In *Image Processing, 1997. Proceedings., International Conference on*, volume 1, pages 548–551. IEEE.

[Xilinx, 2013] Xilinx (2013). Introduction to fpga design with vivado high-level synthesis. Accessed June 2016.

[Zhu et al., 2013] Zhu, P., Jia, F., and Zhang, J. (2013). A copyright protection watermarking algorithm for remote sensing image based on binary image watermark. *Optik-International Journal for Light and Electron Optics*, 124(20):4177–4181.

[Zhu and Mumford, 1997] Zhu, S. C. and Mumford, D. (1997). Prior learning and gibbs reaction-diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1236–1250.

[Ziener and Teich, 2006] Ziener, D. and Teich, J. (2006). Fpga core watermarking based on power signature analysis. In *Field Programmable Technology, 2006. FPT 2006. IEEE International Conference on*, pages 205–212. IEEE.

[Ziener and Teich, 2008] Ziener, D. and Teich, J. (2008). Power signature watermarking of ip cores for fpgas. *Journal of Signal Processing Systems*, 51(1):123–136.

[Zope-Chaudhari et al., 2015] Zope-Chaudhari, S., Venkatachalam, P., and Buddhiraju, K. M. (2015). Secure dissemination and protection of multispectral images using crypto-watermarking. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(11):5388–5394.

[Zoran, 2013] Zoran, D. (2013). *Natural Image Statistics for Human and Computer Vision*. PhD thesis, Hebrew University of Jerusalem.