Fundamentals of Digital Watermarking

A Thesis Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Shon Patil

August 2014

Fundamentals of Digital Watermarking

_____

Shon Patil

APPROVED:

_____

Dr. Ernst Leiss, Chairman

_____

Dr. Olin Johnson

_____

Dr. Andrew Török

_____

Dr. Dan Wells, Dean, College of Natural Sciences & Mathematics

Fundamentals of Digital Watermarking

An Abstract of a Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Masters of Science

By

Shon Deepak Patil

August 2014

Abstract

Theft has always been a problem throughout the ages. Piracy is on the rise and encryption is not enough to counteract the theft of virtual property. Digital watermarking presents another concept that may help in the prevention and damage control of piracy. Before examining digital watermarking, first the history of watermarking and steganography is examined. After briefly going over watermarking in general, we examine the applications of digital watermarking. We then look at models of digital watermarking as a method of communication or with respect to its orientation in space. Understanding the two broad models of digital watermarking will allow proper understanding of basic message coding. Finally, we delve into the realm of watermark security to show how they can be defeated. Digital watermarking is not as developed as cryptography and possesses critical flaws. Digital watermarking, utilized on its own, is not a sufficient deterrent. A watermark may be resistant to one form of attack while being vulnerable to another. However, there is still potential for digital watermarking and it will see more development in the future.

Contents

1. Introduction

1.1 Information Hiding, Steganography, and Watermarking

Historically, people have always sought to protect their valuables from theft. Security is especially important in the Age of Information. Digital watermarking is a way to ensure that the integrity of the media is protected. The term digital watermark was first used by Andrew Tirkel and Charles Osborne in 1992 [1]. "Digital watermarking is the practice of imperceptibly altering a Work to embed a message about that Work" [2]. A concept that is similar to digital watermarking is steganography. "Steganography is the practice of undetectably altering a Work to embed a secret message" [2]. Steganography seeks to hide messages with media while digital watermarking seeks to support the media in question.

1.2 History of Watermarking

While paper was invented in China about a thousand years ago, paper watermarks were first utilized in Italy in the year 1282. The paper watermarks were created by using thin wire patterns in the paper mold. The wire patterns would produce a slightly thinner area which was translucent. There were many reasons as to why the watermarks were created. The watermarks could have been used for some mystical sign or merely as decoration. On the other hand, there were more practical reasons as to why paper watermarks were used. Watermarks were used to provide information as to how and where the paper was produced.

By the eighteenth century, watermarks in Europe and America had a much more functional use. Around this time, watermarks provided an anti-counterfeiting measure on currency and important documents. Watermarks were also used to indicate when the paper was created and the size of the sheets. The term *watermark* would appear to come about towards the end of the eighteenth century. However, the term is a misnomer in that water is not used to create the watermark. The term probably came about because of how water would make the paper more translucent whenever it meets paper. Regarding any concept of security, there is a conflict between two groups of individuals. There are those that want to protect the paper's integrity, and then there are others that wish to override that same protection. In an effort to thwart counterfeiters, William Congreve developed a method to make color watermarks. The technique involved the use of dyed material to be inserted in the middle of the paper making process. While the watermarks that were created using this technique were hard to duplicate, they were also hard to implement. William Henry Smith developed another method of creating watermarks by replacing the thin wire patterns used in earlier processes. Instead of using thin wire patters, the paper mold had shallow relief sculpture that created various shades of grey on the paper. [3]

1.3 History of Steganography

The first and often cited example of steganography is a story from Herodotus. [4] In the story, the master sends a slave to the city of Miletus with a secret message that is tattooed on the scalp of the slave. The slave would then

grow his hair back after tattooing. When the hair was grown back, the slave would journey to Miletus to meet the city's regent. Aristagoras, the city's regent, shaved the slave's head to reveal the tattoo that was underneath. The message that was tattooed on the scalp encouraged Aristagoras to revolt against the Persian king. Herodotus also documents the story of Demeratus. Demeratus alerted Sparta of a planned invasion by the Persian king Xerxes. The way Demeratus alerted the Spartans was to scrape off all the wax from a wooden tablet, write the warning on the wooden tablet, and then apply a fresh coat of wax to the wooden tablet so that it would appear to be an blank wax tablet. Aeneas the Tactician [5] developed improved methods in the field of steganography. He would hide messages in small items such as women's earrings. He also proposed using pigeons to carry messages from one place to another. He even came up with ways to hide a message within a text. For example, he described altering the height of letter strokes in a text or would mark certain letters with small holes. The method by which one would hide messages within the text itself is called linguistic steganography, or acrostic. Acrostic was one of the more popular methods of hiding messages in the ancient world. One famous example of acrostic would be *Amorosa visione* by Giovanni Boccaccio [6]. Acrostic became more advanced with the help of Cardan (1501-1576), famous for Cardan's Grille. The letters in the message would not appear as legible sentences. Rather, the letters would form a jumble of letters that didn't make sense to anyone. The message would be

revealed to the viewer by placing a special mask over the text. The mask was an example of a secret key that was agreed upon between the involved parties.

As a precursor to modern steganography, Francois Bacon [7] used italic or normal font as the basis of binary representations of the hidden message. Five letters could hold five bits and would thus represent one letter of the hidden message. A modern version of this technique was created by Brassil, et al. [8]. They shifted text up or down a small fraction of an inch to produce an altered product. The shift is not usually perceivable to most people and could survive photocopying. Another idea that played an important role in wars between the nineteenth and twentieth century was first proposed by Brewster in 1857. His suggestion was to miniaturize messages to such a degree that they would resemble ink splotches or specs of dirt [9]. This technology was made possible by French photographer Dragon during the Franco-Prussian War. Because of this technology, microscopic messages could be hidden just about anywhere. Dirt under the fingernails, blemishes on one's face, or splotches on one's ear could be a microscopic message that was hidden in plain sight. During World War I, Germans would use the "microdots" and hide them in the corners of postcards. These "microdots" could hold up to one page of text or could contain photographs. A good example of great thinking under pressure and steganography would be when Commander Jeremiah Denton was detained by North Vietnamese captors. Commander Denton would be paraded around to show the world that there are people willing to defect to North Vietnam.

However, while being forced to smile and do whatever was asked of him by his captors, Commander Denton would blink his eyes in Morse code to spell out the word "TORTURE". [10] Thus, while the North Vietnamese would claim that there are soldiers willing to defect to their cause, foreign powers could point out that a code was being used to inform them that not all was well.

With the wide spread use of the internet came a big jump in the use of steganography. The digitization of various media and the accelerated expanse of computer networks have provided an ample playground for individuals to pass around hidden messages. In fact, it is possible for individuals to purchase software that will hide messages for them. One such program, Steganos, can be found online at http://www.steganos.com, along with other products of similar interests. It stands to reason that if such products are available to the general public, that other groups would start playing this clandestine game as well. In fact, it would not be hard to believe that terrorists would utilize such technology to coordinate criminal activities. As a result of possible criminal activity, there have been advances in steganalysis to meet the challenges that the advances in steganography have presented.

1.4 Importance of Digital Watermarking

With an increased concern in copyright protection comes an increased interest in digital watermarking. The internet, for the most part, is a user friendly place where people are interested in downloading pictures, music, and videos. The internet provides an efficient delivery system that is relatively inexpensive.

Acquiring various media via the internet requires a fraction of the time it would take to go to a physical store to purchase said media. Also, when one purchases media over the internet, one would only need virtual space to store the media in question as opposed to storing it on a shelf or wherever such media might be placed. Conversely, such ready availability provides people with the possibility of copyright violations.

If one were to visit any store that specializes in technology, one can acquire a plethora of digital recording devises. Back when the average customer could only acquire analog recording devices at great cost, the quality of such recordings was lacking and did not compare to the quality of the original. Conversely, the ready availability of digital recording devices can produce a duplicate with little loss in quality. The combination of these digital recording devices and the internet has provided individuals with the opportunity to rapidly distribute copyrighted material without appropriate compensation to the appropriate owners. Ergo, owners of various media are interested in technologies that are able to provide adequate protection for their product.

The technology that media owners applied to protect their content is cryptography. Since cryptography was used, this is the most common method for protection as well as the most developed. The collection of files would be encrypted using an encryption key. The files would then be distributed to paying customers. Finally, the customer would use a decryption key, provided by the

distributer, to access the set of files. The risk of someone acquiring the set of encrypted files is considered acceptable, provided that the decryption key is only available to paying customers. However, what is to stop the paying customer from distributing the set of files once it has been decrypted? Once the paying customer acquires the decryption key, that customer can then distribute the set of files at will via the internet. In other words, while cryptography can protect files from interception, the technology will not protect files from the end user.

Digital watermarking is a technology that might be able to protect a set of files from illegal distribution even after a legitimate customer purchases said product. Digital watermarking can protect a product after the customer acquires the product since the digital watermark is never removed. An ideal digital watermark should be able to survive most transformations. Since digital watermarking has the potential to limit the distribution of the product in question, it has been considered for use in many copy protection and copyright protection applications. The difference between copy protection and copyright protection is that copy protection will limit copying while copyright protection will provide information as to who is the content owner. There are other aspects to digital watermarking, but copy and copyright protection provide the bulk of interest in this particular field.

1.5 Importance of Steganography

Electronic communication is subject to interception and intervention, especially during the Information Age. When it comes to issues of security and

privacy, most people's first thought would turn towards encryption. In most cases, only the intended recipient would be able to decrypt the message. The idea is that even if someone intercepted an encrypted message, the message would be totally unintelligible. The field of cryptography is a well-developed field backed by a systematic mathematical foundation. On the other hand, sending an encrypted message is a blatant show that the message was meant to only be shared between specific parties. Steganography presents a way to covertly transfer a message between intended parties with no one else knowing about it. Steganography will allow someone to hide messages in innocuous objects in order to avoid detection. The ability to hide messages can be very valuable in areas where an encrypted message may draw unwanted attention.

Just as cryptanalysis is used to counteract cryptography, so too is steganalysis used to counteract steganography. In order to develop any good security scheme, one must spend time and effort trying to break said scheme. The need for steganalysis becomes more pronounced when a group suspects another group would have reason to transmit messages covertly. As such, some have suspected, but would be hard pressed to prove, that criminals will use steganography to coordinate their criminal activities.

2. Applications

A mere definition of digital watermarking is not enough to fully explain the concept. To even begin to understand what digital watermarking may be, one would have to understand why it is in use in the first place. One may have seen digital watermarks on certain pictures on the internet or maybe on some streaming movies. These digital watermarks are usually viewed as innocuous items on media or may not be noticeable at all.

2.1 Broadcast Monitoring

Consider an average advertising company that wants to place its ads on a certain television network. The television company will offer the advertising company prices depending on how long the commercial will air, how frequent the commercial will air, what time slot would the advertising company like, etc. The companies will come to an agreement and the commercial will go on the air based on the details on the agreement. However, what if the television company violates the agreement? What if the commercial was supposed to show up seven different times and only shows up for six of the agreed upon times? How could the advertising company be sure that the television company has kept its end of the bargain? One method, albeit a low-tech one, is to hire observers to watch the television of specific channels and times to make sure that the television company is honoring their end of the bargain. If one was worried about human error, the

advertising company could hire more people to provide as backup in case someone reported a false negative or positive. However, hiring more people will only take away from the profit margins. The advertising company would not like a manual solution and would quickly look for a more automated one. Passive monitoring will simulate the efforts of human observers watching the various broadcasts, but will do so more reliably and at a lower cost. Active monitoring will depend on associated information that broadcasted along with the content.

2.2 Owner Identification

Consider a specialty photography company that would like to publish one of its media. One of its artists is finished with a photo session and is reviewing all of the shots that were taken. After separating the good shots from the bad, the photographer will submit the work to the magazine publisher for distribution. The objective is to have people view the work of the company and to prevent others from using the company's work for their own commercial purpose. A solution to the problem as to who owns a piece of art is to place a trademark or a copyright notice at the edge of the art so as to not distract the viewer. However, what were to happen if the photo was cropped and the copyright notice was removed from the picture? Regardless of the intent of the person who takes off the copyright notice, other people might begin to use the new altered photo without any caution. The scenario just described matches the famous case of a photograph of Lena Sjööblom [11]. This photo is the most common photo when it comes to image processing. It would appear in classrooms, scientific journals,

and imaging labs. Many professors are ignorant of the fact that the photo is owned by Playboy, and thus is a copyright violation. Another way to prevent people from using the photograph without permission is to place a blatant copyright notice on the face of the picture. This copyright notice would not be placed off to the side of the picture so people can ignore it. The copyright notice would be placed in the center of the photo and large enough to cover most of the photograph. No one could possibly remove the copyright information through cropping and still have a salvageable image. However, the copyright information would ruin the aesthetic value of the photograph and would thus ruin the appeal.

The problem of copyright infringement is even more apparent when it comes to audio media. Music companies would manufacture music CDs and place copyright notices on the face of the CD. However, what were to happen if people were to copy the contents of the CD? There would be a digital representation of the contents of the CD located on the hard disk on the computer. From there, all one would have to do is distribute the music via the internet for fast distribution.

A possible solution to either of the problems presented is digital watermarking. The digital watermark can be both invisible and inseparable from the original work of interest. People with digital watermarking detectors should be able to detect a digital watermark from an image even though the image is cropped. Digimarc's digital watermarking detector is an example of a

commercially distributed detector that is available to the general public [12]. The digital watermarking detector is bundled with Adobe's image processing program called Photoshop. The digital watermarking detector will access an online central database and will use the digital watermark found on the image to find contact information for the image's owner.

2.3 Proof of Ownership

Suppose an artist wishes to post a work of art on a website; the artist will attach a copyright notice on the side of the picture and then upload it to the website. What would prevent someone from altering the image and replacing the previous copyright with a new one? One method the artist could use is to register the work of art. The artist could register the work of art at the United States Copyright Office. The image would be archived alongside with information pertaining to the rightful owner of the image. Any poor and struggling artist could register each piece of work for a fee and there would be no dispute as to who created a certain image. The down side is that not many poor and struggling artists have the resources to commit for each of their works of art. Without the necessary funds, another person could claim the work as his own by registering the images and paying the same fees. This would lead to the possibility where the artist who created the image could be taken to court for copyright infringement. The artist would need to provide evidence to the court that would acquit him of any wrongdoing. Negatives of photographs and examples of early drafts of the work of art in question would have to be submitted into evidence so that the court

may reach an outcome concerning the issue. However, the person who is willing to take the issue to court could also take the time to fabricate similar evidence as well. A person could fabricate negatives or make rough drafts to show that he created the art in question. Worse still, if the image was created digitally, there might not even be any proof to present to the court.

Digital watermarking could provide a way for the artist to protect an image from copyright infringement. All the court would have to do is place the image in a digital watermarking detector. The detector would produce the artist's digital watermark and would provide proof of ownership. However, a counterfeiter could also place his digital watermark within the image and pass it off as his own. To solve the problem of the counterfeiter placing a digital watermark in the image to pass off as his own, a different approach is necessary. Instead of the artist trying to directly prove that the image belongs to him, he could prove that one image is based off of another image. In other words, the artist has the original version of the image from which another image is produced. It is somewhat similar to having the negatives of an image with which one can use to produce a finished product.

2.4 Transaction Tracking

Large media companies would like to sell to the public on a massive scale so that they could maximize their profits. Many CDs, DVDs, and other media formats are widely distributed to the world at large. What would happen if someone were to find out that there are illegal copies of a certain media that are

available for download on the internet? How would anyone know who was responsible for the leak? One solution is to place a unique digital watermark on each piece of media. If an illegal copy was widely distributed on the internet, all one would have to do is obtain a copy and look at the unique digital watermark.

2.5 Content Authentication

There are commercially available applications that will allow one to alter the appearance of an image. Anyone could put in or take out any object in a photograph. This presents a problem if the photograph is to be presented as evidence to a court of law. Digital watermarking presents a solution by introducing the concept of an authentication mark. An authentication mark is a signature that is directly embedded into the image. The idea behind embedding an authentication mark inside of the image is to determine if someone tampered with the image in any way [13]. With respect to content authentication, a fragile digital watermark would be preferable compared to that of a robust digital watermark. Robust digital watermarks are designed to resist change for the purpose of identification and ownership. Fragile digital watermarks work in such a way that if the image is modified in any way, the authentication mark will be destroyed and would provide evidence that the image was subjected to tampering. Furthermore, depending on the construction of the authentication mark, one might be able to discover how the image was altered. For example, suppose the image was divided into even sections. Each section of the image has an authentication mark embedded into it. If the image was disturbed in any way, it would be

possible to find out which areas of the image were subjected to alteration. A person with some skill in image manipulation would gain some insight as to how the image was manipulated. An example of where this type of localized authentication would be useful would be in a police investigation. Suppose the police receive a surveillance video that has been edited by someone. If the images in the video have been embedded with a usual authentication mark, all that would be discovered is that the video has been tampered with. If the images in the video were embedded with the localized authentication scheme, the police would figure out which parts of the video were altered. At first glance, that might not make much difference considering that a person might have been edited out of the video. However, what if the image was altered so that one form of identification appears as something else? For example, if the parts of the video were edited so that only the license plate of the car was changed, then the police would know what to look for.

Lossy compression is another alteration that can be detected by authorization marks. Most lossy compression algorithms will leave small changes in the digital watermark that might be detected [14]. However, sometimes lossy compression is not a concern and a more robust digital watermark is needed. That is not to say that we need a robust digital watermark, but rather that we need some middle ground between robust and fragile. One such category that exists is called *semi-fragile*. A semi-fragile digital watermark will survive minor changes, like lossy compression, but will be disturbed by major alterations.

2.6 Copy Control

While it is useful to have countermeasures in place, should media become compromised, it is also useful to have prevention measures as well. Encryption is the most common method used to defeat unauthorized copying. There are a few methods that one can use in order to defeat encryption. One method is to decrypt the media without access to the encryption key. The person trying to compromise the encryption would have to systematically cycle through a list of keys until the right one is chosen. However, this method is impractical in most cases since it could take many years to get to the correct key. Another solution would be to reverse-engineer the hardware or software used to decrypt the encrypted media. It is possible to analyze a system with the express purpose of finding decryption keys. Finally, the last method of obtaining the decryption key is to pay for it. For example, someone could simply record television and satellite broadcasts once they have been decrypted. Then, all one would have to do is distribute the media over the internet. The main goal is to distribute media to paying customers and prevent non-customers from ever viewing the media in question. Because digital watermarks are present within the media itself, it might provide a solution to copy control. If digital watermark detectors are supplied alongside media players, the recording capabilities of said players could be overridden if the digital watermark was configured to a setting that restricted copying. However, the problem with adding a digital watermark detector to media players with a recording capability is that the digital watermark detector will decrease the amount of sales for that

particular product. If a customer has a choice between a recorder that has the ability to make illegal copies of various media and a recorder that will prevent any illegal activity, the recorder without the digital watermark detector would be the preferable choice.

The direct approach is to pass a law stating that digital watermark detectors have to be included in all media players. The approach is unreliable at best since passing the law would be very difficult. Rather than taking the law creation approach, the digital watermarking detectors could be bundled with a new technology. For example, the patent license for CSS encryption is bundled with a digital watermark detector. Any manufacturer that would want to use the CSS technology would have to incorporate a digital watermark detector into the media player, or else the technology would not work. However, it is still possible for a manufacturer to not want to incorporate the technology into their media players and still come out with a legal product. Devices that use the technology are called *compliant* and those that do not are called *non-compliant*.

To counter the issue of non-compliant devices on the market, the notion of *playback control* is introduced [15]. If someone uses a non-compliant player to record media, the copy has a digital watermark attached to it. The player will make a decision whether the media being played is either a copy or an original based on the watermark attached to the media.

2.7 Device Control

Digital watermarks have the ability to add value rather than restrict use. A recent example of device control is the unique identifier that printed alongside of advertisements, tickets, packaging, etc. [16]. After scanning the image via the phone's camera, the software will use the unique identifier to direct the web browser to the corresponding web site.

3. Models of Watermarking

3.1 Communication

3.1.1Components of Communication Systems

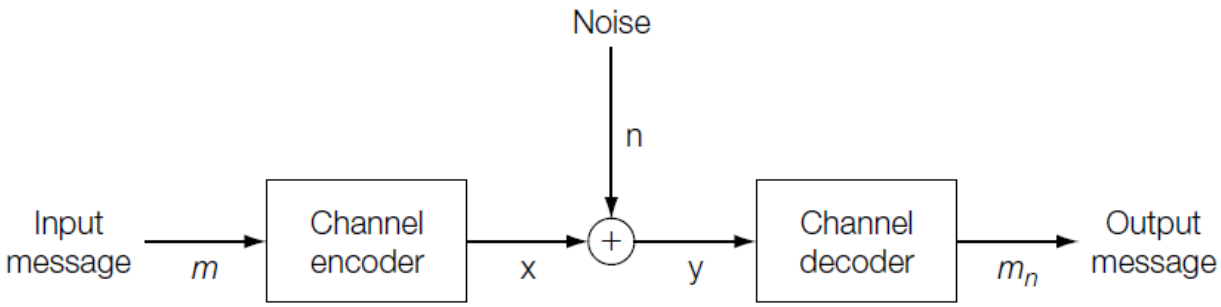

Figure 3.1: Standard model of a communications system [17].

Figure 3.1 illustrates the communication process. Let m be a message to be transmitted across a communication channel. The message m is prepared for transmittion by the channel encoder. The channel encoder is a function that accepts the message m as input and generates a code word x as output based on a

set of signals that can be transmitted over a given channel. The encode is comprised of two parts: a source encoder and a modulator. The source encoder is a function that takes the message m and provides output as a sequence of symbols derived from a given alphabet. Then the modulator is a function that takes the sequence of symbols created by the source encoder as input and outputs a physical signal that can be transmitted over a communications channel (i.e., modulate amplitude, frequency, or phase of a physical carrier signal for radio transmittion).

The output of the channel encoder depends on the type of transmittion channel. For the purpose of generalization, let x be a sequence of real values ($x = [x_1, x_2, ..., x_N]$) that are quantized to some arbitrary high precision. Assume that the range of feasible signals is limited by power constraint:

$$\sum_{t} (\mathbf{x}[t])^2 \leq p,$$

where p is a constant that limits power.

The signal x is then sent over a given transmission channel to a reciever. Assuming that the transmission channel is noisy, the signal x will then be transformed into y due to additive noise. The transmission channel is assumed to add a random noise signal n to the sent signal x.

The received signal y is then processed by a channel decoder. The channel decoder is a function that will accept y as input and attempts to correct the transmisson errors due to noise. The function takes y and then creates a set of

messages mn. The decoder is generally a many-to-one function so that even noisy messages are properly decoded. The probability that there are errors in a decoded message is negligible, provided that the channel code is appropiate for the channel in question.

3.1.2 Classes of Transmission Channels

When designing a communication system for the model system in question, it is assumed that the transmission channel is fixed. Because the transmission channel is fixed, it is not possible to create or alter the noise generation that occurs between transmission points. The channel is charactized by the condition probablity distribution, $P_{y|x}(y)$, which provides the probablity of recieving the altered message y provided that the origional message x is the transmitted signal.

The classification of differnet transmission channels depends on the type of noise function that is applied to the signal and how the noise is applied to said signal. In the model above, the channel had an additive noise channel where the received signal y is the sum of the original message x and the random noise signal n. The random noise signal may or may not be independent of the signal being modified. One of the simplest channel to analyze is the Gaussian channel. The Gaussian channel has a noise signal, n[i], where each element is independent of a Normal distribution with zero mean and some variance.

It is important to note that there are several non-additive channels. One of the most important ones is called a fading channel. A fading channel creates a

time variance in the strength of a signal [18]. The fading channel can be expressed as a product of the orignal message and a variable whose values range between zero and one and may either vary with the passage of time or each time the channel is utilized. The channel may also include an additive noise component.

3.1.3 Secure Transmission

Besides the considerations of the design of a channel, one must also be concerned with the security of a channel. The two types of attack that can be made against a channel are active and passive. Passive attacks read transmissions between senders and receivers. Active attacks will either disable communication or transmit messages. For example, both forms of attacks can be found regarding military communication. A passive attack will monitor all enemy communications while an active attack will attemp to jam said communcation attempt. Two defenses against passive and active attacks are cryptography and spread sprectrum communications.

In cryptography, a key is used to encode a message. Then, the ciphertext is transmitted as usual. The ciphertext is recieved and is decoded using the same or derived key in order to revel the original message. This can be seen in figure 3.2.

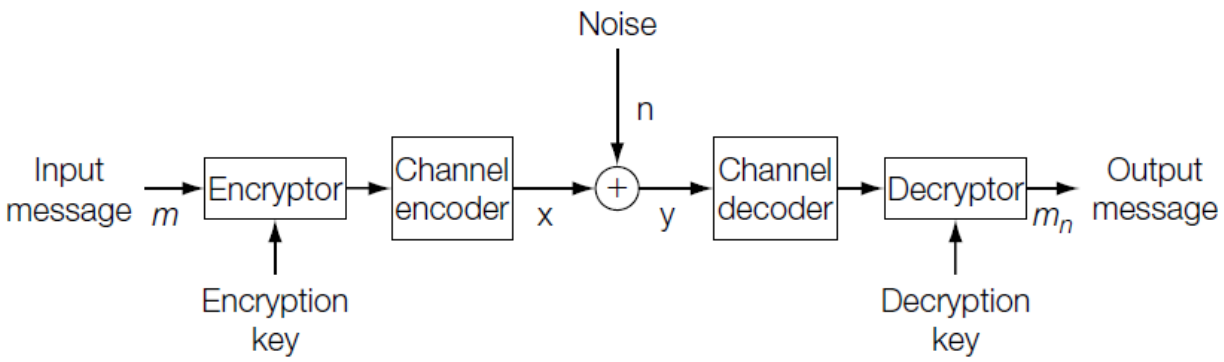

Figure 3.2: Standard model of a communications channel with encryption [18].

There are two types of cryptography in regards to utilization. The first type is used to deter passive attacks where the enemy has unauthorized use of the channel. The second type is used to deter active attacks that try to create unauthorized messages. Cryptography will not prevent the enemy from knowing that a message is being transmitted. Furthermore, cryptography will not prevent the enemy from attacking the message and thus prevent it from being delievered. Signal jamming, which prevents a given message from being delievered between two or more parties, can be countered by spread spectrum communications. Modulation, altering the properties of a carrier signal, is applied to a secret code and sends the signal across a wider bandwidth [19]. This secret code can be thought of as a type of key that is used to encode and decode a message. This can be seen in figure 3.3.

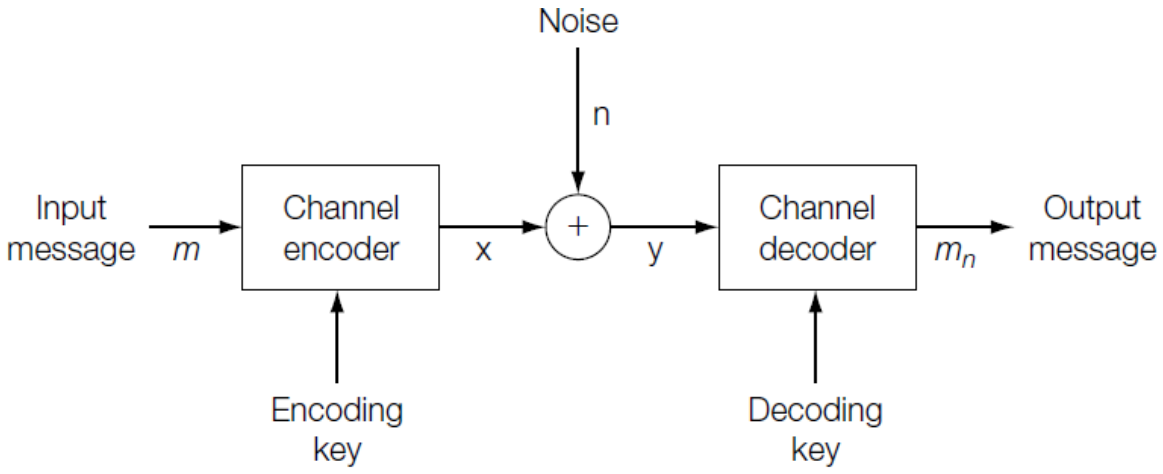

Figure 3.3: Standard model of a communications channel with key-based channel coding [20].

Frequency hopping is one of the simplest and easiest technologies using spread spectrum communication practices [21]. Frequency hopping will transmit a portion of the original message over a given number of different frequencies. In order for the transmitter to properly send the message to the receiver, an agreed upon key is used to form a pattern of hops from one frequency to the next one. The absense of a key will prevent the enemy from using either passive or active attacks against the entire transmission.

Spread spectrum communications increases the probablity of the signal reaching its destination while cryptography will ensure the privacy of the message in question. The two technologies complement one another and are often used in tandum. With respect to the OSI model, spread sprectrum is in the transportation layer while cryptography is in the messaging layer [22].

3.2 Communication-based Models of Watermarking

Watermarking is a type of communication where the embedder attaches a watermark to the media in question for the receiver. It is then possible to fit digital watermarking to the model of communications thus illustrated. The next three sections will presents three ways to treat watermarking as communication with regards to how the cover work is incorporated. The basic model considers the cover work to be noise under the given paradigm. The second method still considers the cover work as noise, but the channel encoder considers it as side information. The third method regards the cover work as a second message that must be tranmitted along side of the watermark using multiplexing.

3.2.1 Basic Model

Figures 3.4 and 3.5 show one way where watermarking can be applied to the given model of communication shown in figure 3.3. Figure 3.4 shows an informed detector while figure 3.5 shows a blind detector. The basic model views watermarking as the transmission channel where the watermark message is sent, while the cover work is a part of said channel.

Independent of the choice to use either an informed or blind detector, embedding a watermark into a given medium can be broken down into two parts. First, the message is processed into an added pattern, w_a, that has the same type and dimensions as the cover work, c_0. Then the added pattern, w_a, is added to the cover work, c_0, to produce a watermarked work, cw [23]. In visual media, the embedder will generate a two-dimensional pattern that is the same size as the

picture or movie. In audio media, an audio signal is produced.

In many practical examples, the creation of a added pattern is a multi-step process. Generally, the process starts with the creation of one or more reference patterns w_{r0}, w_{r1}, ... These reference patterns are predefined and are possibly dependent on a key. After the reference patterns are generated, they are then combined to form a pattern that will encode the message to be transmitted, or a message pattern wm. The message pattern wm is then altered to the appropiate size to produce the added pattern w_a. The added pattern is then added to the cover work to produce the final product. This method is the blind embedder approach since the embedder is independent of the cover work.

After the watermarked work is generated, it is assume that the work in question will undergo a process where noise is added. Processes that might alter the work in question will be either intentionally harmful and/or a normal alteration. Such examples include compression and decompression, image and audio alterations, etc. Since the effects of processing depend on the type of watermark applied to the work in question, it is easier to simplify the model from the view of effects of application of additive noise.

An informed watermark detector uses a two step process. The first step is to take the unwatermarked work and subtract it from the watermarked work, cwn, thus leaving a noisy watermark pattern, wn. Next, the noisy watermark pattern, wn, is decoded with a watermark decoder with the use of a watermark key. Note that the watermarked work is the sum of the original work and the noisy

watermark pattern. Removing the original work only leaves a noisy watermark pattern that has been modified by the noise process. Ergo, the ignoring the cover work, the watermark encoder, the noise process, and the watermark decoder all form a system similar to the transmission channel model that was explained previously in figure 3.3.

Keep in mind that there are more advanced watermark detection processes that do not rely on the given unwatermarked cover work in question. Typically, some function dependent on the original cover work, c_0, (i.e., a data-reducing function) is utilized by the detector to eliminate the metaphorical noise effect that is simulated by the addition of the cover work in the embedder. An example of this can be seen in Cox et al. [24] . where a small portion of the DCT coefficients from the original work is used in the detection process.

While the informed watermark detector process uses the original cover work in order to produce the noisy watermark pattern, the blind watermark detector does not. The blind watermark detector cannot use the same methods as the informed watermark detector where the original work is removed from the watermarked work before decoding. Using the established model shown in figure 3.3, the current situation can be seen from the point where the added pattern experiences distortion from the noise, or the cover work and the actual noise signal. The received pattern is seen as a corrupted version of the added pattern and the detector is seen as the channel decoder.

It is ideal to maximize the probablity of having the recieved message be

the same as the sent one in applications that require robustness (i.e., copy control or transaction tracking). This is the same goal as in traditional communication systems. There is a difference between the two systems. The focus of authentication is to learn how or whether a work has been changed after a watermark was embedded while a transmission chanel's purpose is to communicate. Since the goal of the two systems differ, the models shown in figures 3.4 and 3.5 are not typically used to illustrate authentication systems.

The model in figure 3.5 can be used to illustrate a basic example of how a blind watermark detector works within the watermarking system.

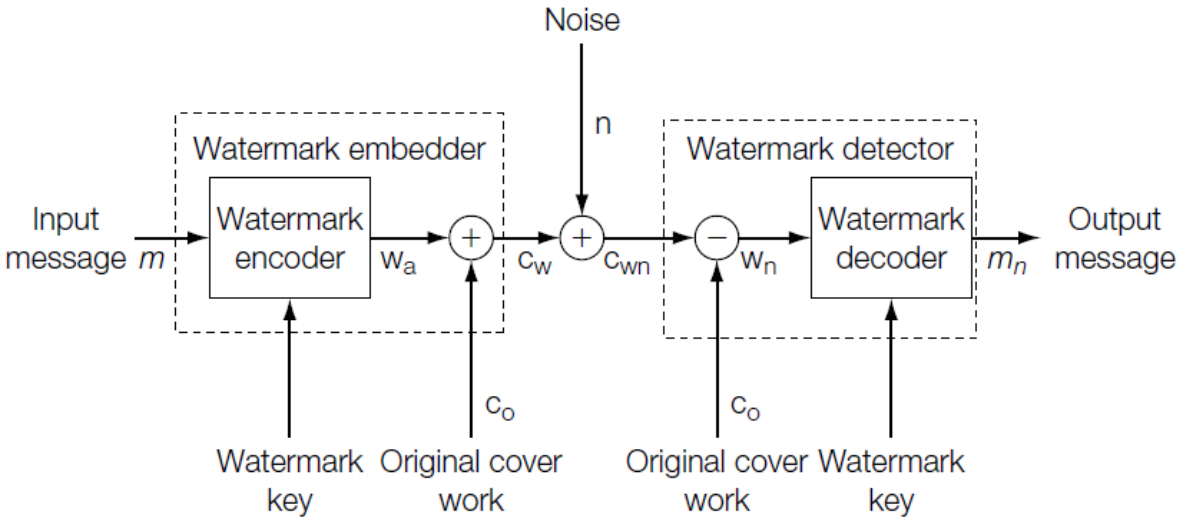

Figure 3.4: Watermarking system with a simple informed detector mapped into a communications model [25].

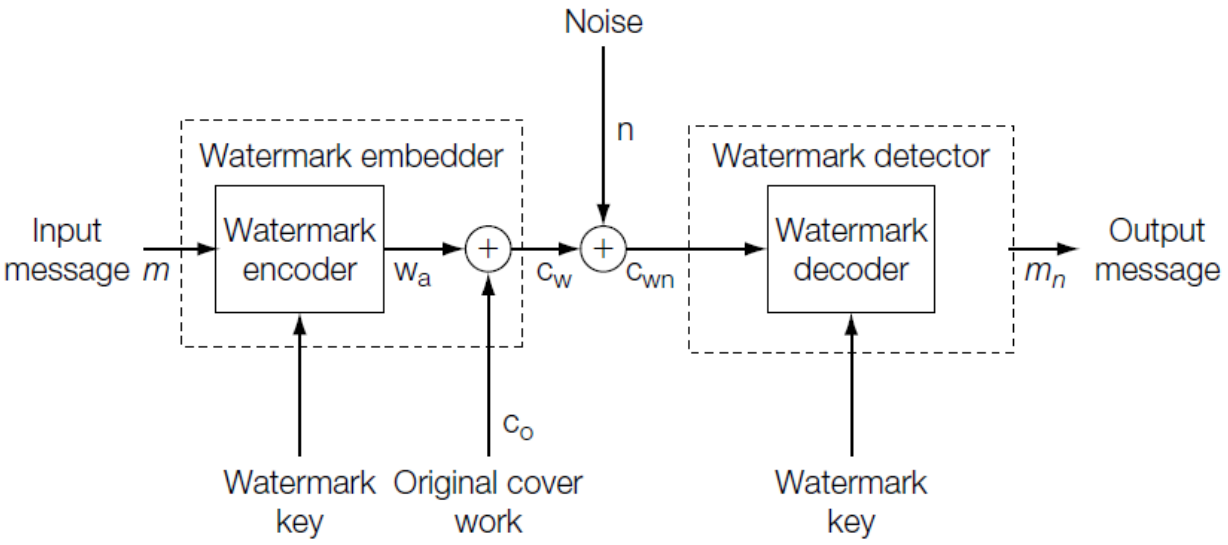

Figure 3.5: Watermarking system with blind detector mapped into a communications model [25].

3.2.2 Watermarking as Communications with Side Information at the Transmitter

While some knowledge regarding robust watermarking with blind detectors may be obtained by using the model in figure 3.5 as a reference, this model does not include all potential embedding algorithms, since it forces the encoded watermark to be independent of the cover work. Since the unwatermarked cover work, c_0, is clearly known to the embedder, there is no need for this restriction. Far more efficacious embedding algorithms may be created when the watermark encoder is permitted to examine c_0 before encoding the additional pattern w_a. Figure 3.6 illustrates a watermarking model that permits w_a to rely on c_0. The model is nearly a copy of Figure 3.5, with one difference, that c_0 is offered as an added input to the watermark encoder. This means that the

alteration permits the embedder to set cw to any wanted value by allowing w_a to equal cw minus c_0. When it is kept in consideration that the cover work is part of the noise process (c_0 + n) in the transmission channel, the new model emerges as a representation of a system of communications with auxiliary information at the transmitter, which was first researched by Shannon [26]. In summation, the embedder has the capacity to exploit some information regarding the channel noise, particularly c_0 itself. After Shannon opened debate on the topic, multiple authors have researched communications with auxiliary information [27][28][29]. It has been ascertained that with various types of channels it is of no concern if the auxiliary information is retrievable by the receiver, the transmitter, or both. Its interference is negligible. In recent times, some researchers have started applying the examples learned regarding communications with auxiliary information to watermarking. This topic will be explored in detail in Chapter 5. At the moment, the potential of the approach is exemplified by altering the blind embedder/linear correlation detector (E_BLIND/D_LC) system to be 100% effective.

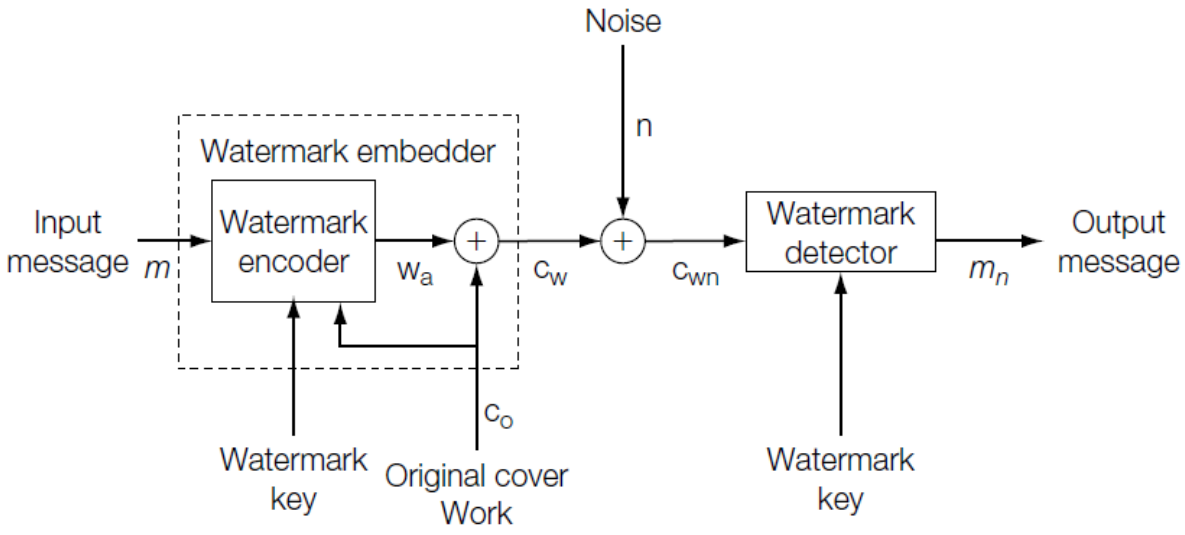

Figure 3.6: Watermarking as communications with side information at the transmitter [30].

3.2.3 Watermarking as Multiplexed Communications

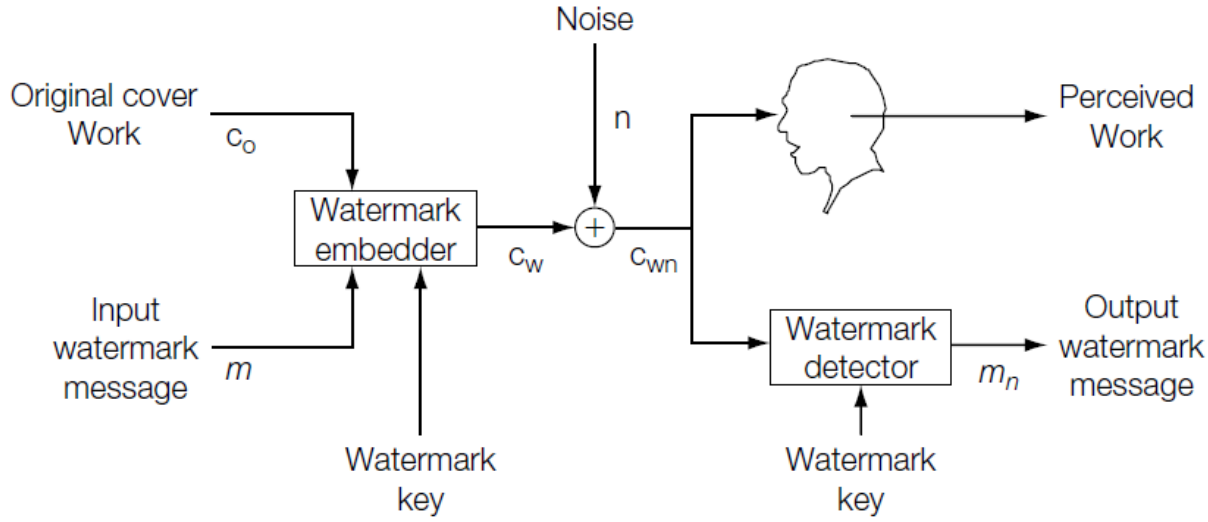

Figure 3.7: Watermarking as simultaneous communications of two messages [31].

Figure 3.7 displays an alternate illustration of watermarking as communications. In this case, the cover work is no longer recognized as integral to the transmission channel. Instead, it serves as a secondary messenger to be transmitted while coupled with watermarking message cw. Meanwhile, messages c_0 and m are recognized and then decoded by both a watermarking detector and a human user.

The watermarking embedder synchronizes m and c_0 into a signal known as cw. This synchronization is akin to the transmission of several messages over one line in conventional communications by either code-division, frequency-division, or time-division. However, there is a distinguishing factor to take into consideration: conventional communications use the same fundamental technology for various messages, which are distinguished by one parameter such as time or frequency. For the sake of comparison, the two messages are distinguished by both watermark detection and by a human end user. This is similar to when frequency-division is used for one message and spread spectrum is used for another [31].

Following the passing of the signal through the transmission channel, it is then viewed by either a human or a watermark detector. When viewing c_{wn}, the person will probably observe something resembling the original Work, minus the watermark's interference. During the detection of the watermark in cwn the original watermark message is likely to be the original watermark's message with

no interference from the cover Work. An informed watermark detector receives a function of the cover Work, or the original cover Work, as a secondary input. This image of watermarking stresses the importance of the equilibrium between the watermark and the cover Work. One such example of the aforementioned equilibrium is observable in watermarking literature is in the two varying uses of the term *signal-to-noise ratio* (SNR).

Regarding fidelity, "signal" means the cover work, while "noise" indicates the watermark. However, regarding effectiveness and robustness, "signal" indicates the watermark, while "noise" is the cover work (and/or any related distortions). The former is known as the document-to-noise (DNR), while the latter is referred to as the watermark-to-noise ratio (WNR). Knowing which meaning is intended is commonly derived from contextual connotations. The equilibrium in Figure 3.12 insinuates that problems in transmitting m should have a proportionate relationship with problems in transmitting c_0. In this example, the informed embedding algorithm E_FIXED_LC can have the problem of effectiveness in blind embedding by scrutinizing the cover Work anterior to the designing of the additional pattern. The embedder calculates the interference between the reference pattern and the cover Work by computing c_0 times wr. The embedder then modifies the amplitude of the additional pattern to adjust by switching the value of □. Another similar phenomenon is when the system is improved when faced with problems associated with fidelity. It is improved by using a perceptual model to scrutinize how the watermark interferes with the

Work. W_a is then adjusted to reduce this interference to the fullest capacity possible.

3.3 Geometric Models of Watermarking

3.3.1 Distributions and Regions in Media Space

Works could be considered as points in a *media space* composed of *N* dimensions. *N* depicts the total amount of samples needed to represent each Work. For monochromatic images, let this be known as the number of pixels. In images with red, green, and blue, *N* would be the number of pixels multiplied by three. For constant temporal content it is presumed that the watermark is integrated in a fixed-length signal segment with time sampled content. Therefore, when it pertains to audio, *N* would be the total amount of frames in a segment times the number of pixels per frame times three, if the video is in color.. Because the emphasis here is on digital content, every sample is bounded and quantized. One example would be that each pixel value of a grayscale image with 8 bits has a value between 0 and 255. There are many possible combinations that can be formed from a grid of pixels. The points that are outside the bounds, or between the lattice points, do not relate to digital format representations of works. While the quantization step size in usually relatively small, and the bounds large, this is usually ignored and it is usually presumed that media space is constant. In other words, all points in the space, including those off the lattice, relate to realizable works. The following section will regard each of the regions within media space, and various distributions of probability over media space that are

relevant when examining watermarking systems.

3.3.1.1 Distribution of Unwatermarked Works

Various works have varying probabilities of entering into a watermark detector or embedder. Audio is more probable to embed watermarks in the actual music than in static. Video usually embeds in images of natural scenes rather than the video "snow." Each type of media contains unique statistical distributions to take into consideration [32][33][34][35]. During the evaluation of the attributes of a watermarking system, for example false effectiveness and the probability of false positives, it is crucial to create a framework with the *a priori* distribution of content in mind. The two ways to express this are either a probability distribution, concerning the lattice points representing digital works, or a probability density function regarding all media space [36]. A large variety of statistical models exists regarding the distribution of unwatermarked content. The most sparse models presume an elliptical Gaussian fit to the distribution. An example of such is instituted in the derivation of the E_BLIND/D_WHITE watermarking system. Laplacian or generalized Gaussian distributions can create more accurate models of most media. These are used in the examination of lossy compression. While more elaborate models attempt to illustrate content as the result of unpredictable parametric processes, such a framework will not be discussed presently.

The application dependence of unwatermarked content distribution is worth noting. This multitude of varying distributions for different types of

Works can lead to imperfections. If one were to measure a watermark detector's false positive using a single distribution, and utilize the detector in an application where the unwatermarked Works are extracted from a varying distribution, then the estimate would not be entirely correct. This problem can have deep implications in applications that need very low false positive rates, such as copy control.

3.3.1.2 Region of Acceptable Fidelity

Visualize the original image c_0 as having only one pixel modified by a single increment of brightness. The modified image that results from this augmentation looks like the original but has a different vector. Obviously, several such images exist, visualize a region around c_0 where every vector relates to an image that is virtually identical to c_0. If c_0 is a signal made of audio, correspondingly minute alterations will result in virtually identical sounds which also form a media space region. The region of media space vectors that are virtually identical to cover work c_0 is known as the *region of acceptable fidelity*. It is very hard to pinpoint the exact region of acceptable fidelity around a particular work, due to the fact that not enough is known about human perception. The region is usually estimated by putting a threshold of a measurement of perceptual distance. One example would be to utilize the mean squared error (MSE) as an unsophisticated perceptual distance metric. This is known as [37]

$$D_{\text{mse}}(\mathbf{c_1}, \mathbf{c_2}) = \frac{1}{N}\sum_{i}^{N}(\mathbf{c_1}[i] - \mathbf{c_2}[i])^2,$$

where c_1 and c_2 are *N*-vectors (*N*-dimensional vectors) in media space. When limitation T_mse is placed on this function, an *N*-dimensional ball (*N*-ball) of radius √N_t_mse is the region of acceptable fidelity.

The MSE function fails to be a highly effective practical predictor of the perceived differences between works [38]. Suppose c_1 is an image and c_2 is an alteration of c_1 that is shifted slightly to the left. The images c_1 and c_2 will be perceptually the same, but the MSE might be large. So, MSE does not factor in visual tracking. A visual example is illustrated in figures 3.8 and 3.9, where the distances between the watermarked and unwatermarked images are 16.4 and 16.3, respectively. However, it is clear to see that the watermarked image is worse than the unwatermarked image.

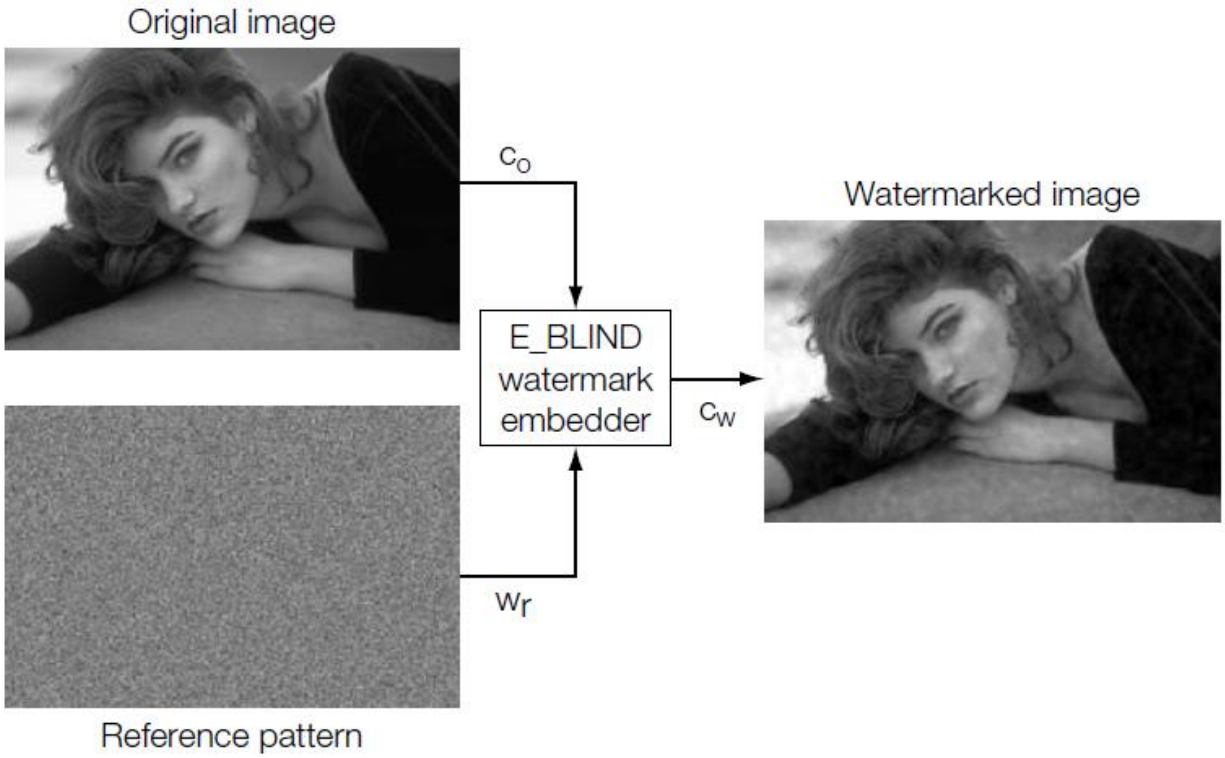

Fig 3.8 Results of the blind embedding algorithm, E_BLIND, with a reference pattern made from uniformly distributed noise [39].

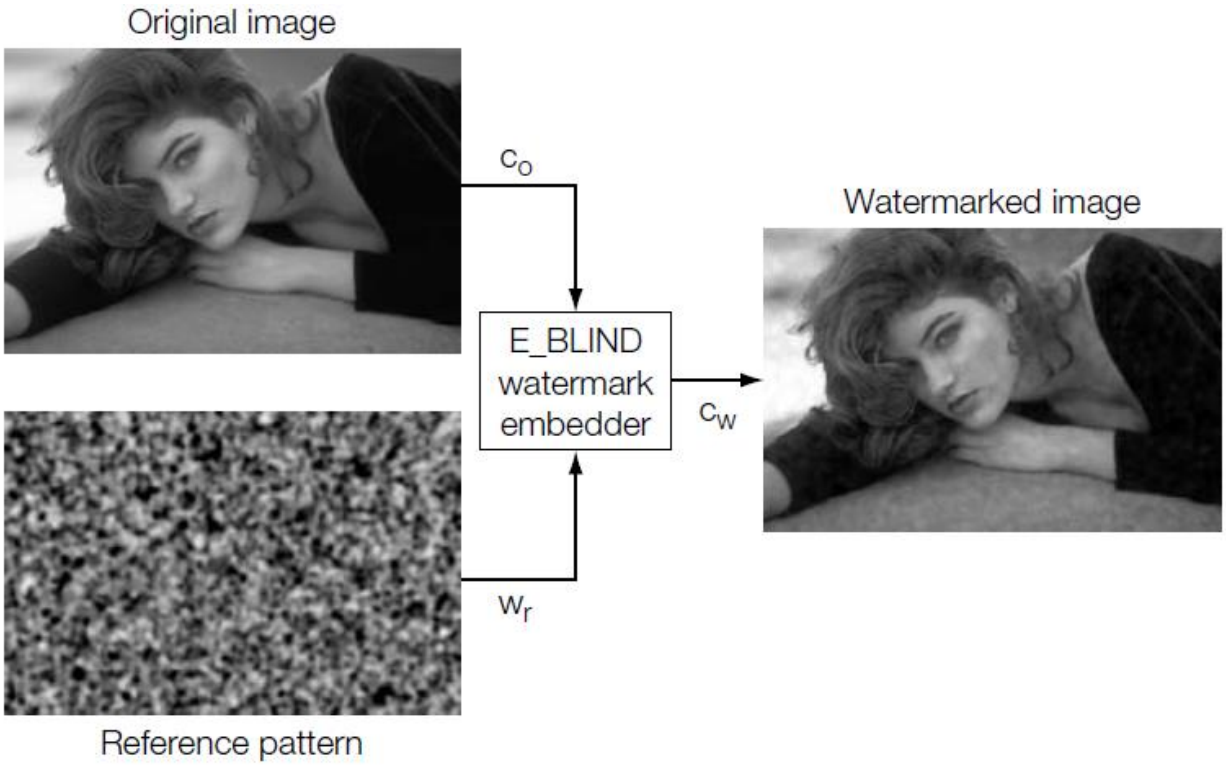

Fig 3.9 Results of the blind embedding algorithm, E_BLIND, with a reference pattern made from low-pass filtering the uniformly distributed noise of figure 3.8 [39].

There are some perceptually asymmetric distance functions where the two arguments have slightly different interpretations. Normally, the first argument is interpreted as the original work and the second as the distorted work. An example of a common asymmetric distance function is based on the reciprocal of the SNR [37]:

$$D_{\text{snr}}(\mathbf{c_1}, \mathbf{c_2}) = \frac{\sum_{i}^{N}(\mathbf{c_2}[i] - \mathbf{c_1}[i])^2}{\sum_{i}^{N}\mathbf{c_1}[i]^2}.$$

The argument c_1 is the signal and c_2 is the noisy version. The distance gauges how noisy c_2 is in comparison to c_1.

There are more complex versions, for both images and audio, that provides better predictions of human judgment. Such functions use the unit of just noticeable difference (JND) to measure perceptual distance.

3.3.1.3 Detection Region

Given a message, m, and a watermark key, k, the detection region is the set of works in the media space to be decoded by the detector that have the encoded message. The detection region, similar to the region of fidelity, is usually defined by the detection measure, a threshold of a measure of commonalities between the input of the detector and the pattern that encodes the message, m.

The detection measure, in the detection algorithm D_LC, is a linear correlation, $z_{lc}(\mathbf{c}, \mathbf{w_r})$. In order to find the shape of the detection regions for this detector, it is noteworthy to observe that the linear correlation between the work received and the reference pattern, $c \cdot w_r/N$, is equal to the product of their distances and the cosine of the angle between said lengths, divided by N. Since w_r is a constant, the measure is the same as finding the perpendicular projections

of the N-vector c onto the N-vector w_r. All the points that are larger than the value τ_{lc} are going to be on one side of the plane perpendicular to w_r, which is the detection region for m=1. Likewise, the detection region for m=0 is set for all the points on the side of the plane gauged by $-\tau_{lc}$.

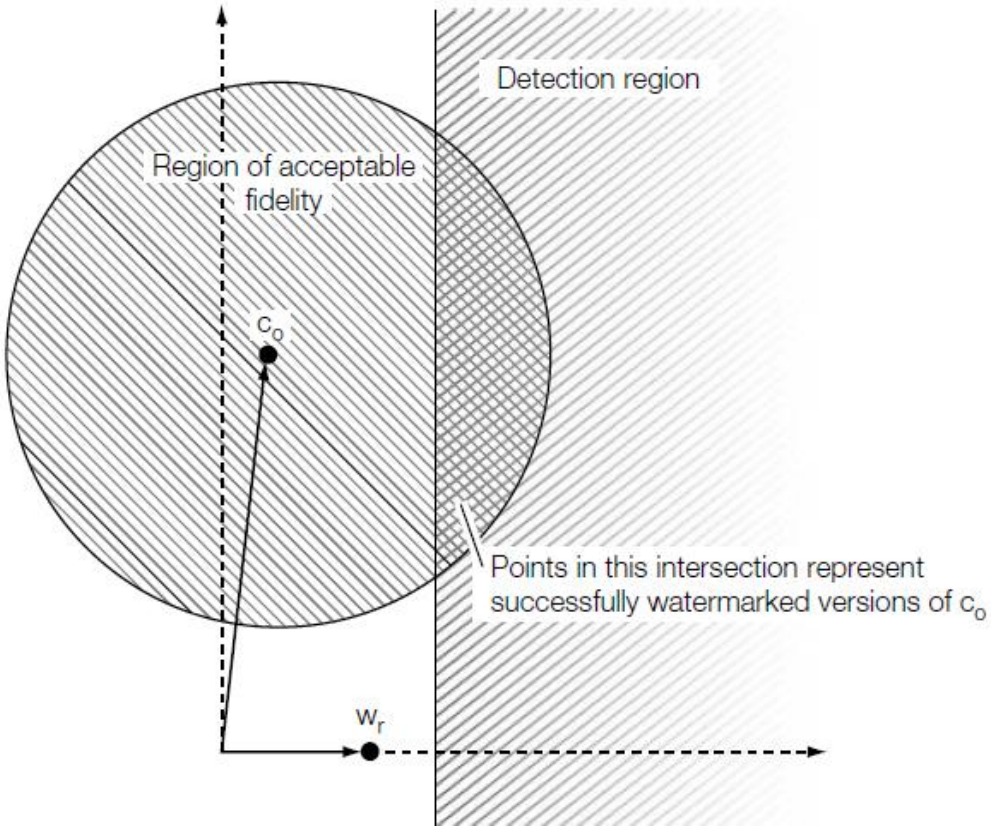

Figure 3.10: The region of acceptable fidelity and the detection region for a watermarking system [40].

Once a watermark message is placed inside of a work, the watermarked work lies on the intersection of the region of acceptable fidelity and the detection region. Figure 3.10 shows this using a region of acceptable fidelity based on MSE and a detection region based on linear correlation between the work

received and the reference pattern, w_r. Figure 3.10 displays the 2-D slice of the media space that has the two vectors c_0 and w_r, where w_r lies on the horizontal axis and c_0 lies on the vertical axis (because randomly selected vectors in high-dimension space tend to be perpendicular). An N-ball is the region of acceptable fidelity where the intersection with the diagram is a 2-ball. The detection region plane intersects the diagram on a line perpendicular to w_r. All the points in the region of acceptable fidelity and everything to the right of the planar edge of the detection region correspond to versions of c_0 that are in the acceptable range of fidelity will make the detector report positive to the presence of a watermark.

3.3.1.4 Embedding Distribution or Region

The watermark embedder is generally a deterministic function that maps a given work, message, and maybe a key into the same watermarked work, c_w. Since the original works are picked randomly from a group of unwatermarked works, the output can be seen as random. Thus, the probability that watermarked work, c_w, will be the output is the same as the probability that an original work is picked from a group of unwatermarked works. However, if several unwatermarked works can produce c_w, the probability of c_w is the sum of the probabilities of the works in question. This is called the embedding distribution.

Some algorithms define the embedding distribution where every point has a nonzero probability. Ignoring the effects of clipping and rounding, this is the exact case of the embedding distribution seen in the E_BLIND image watermarking algorithm. All possible images, regardless of being outside the

detection region, can eventually be placed in the detection region by using the E_BLIND embedder on another image, as seen in figure 3.11. These algorithms are less than perfect in that there exist a nonzero probability where the embedder will produce a work that is outside the detection region.

There are algorithms that can only produce a small set of outputs. The E_FIXED_LC algorithm will take a given reference pattern, message, and embedding strength, β, and only produce images that lie on a fixed plane, as seen in figure 3.12. It is prudent, with a system such as this, to consider the embedding region, the set of all possible outputs of the embedder, for a given message. The system is guaranteed to work if and only if the embedding region is inside of the detection region.

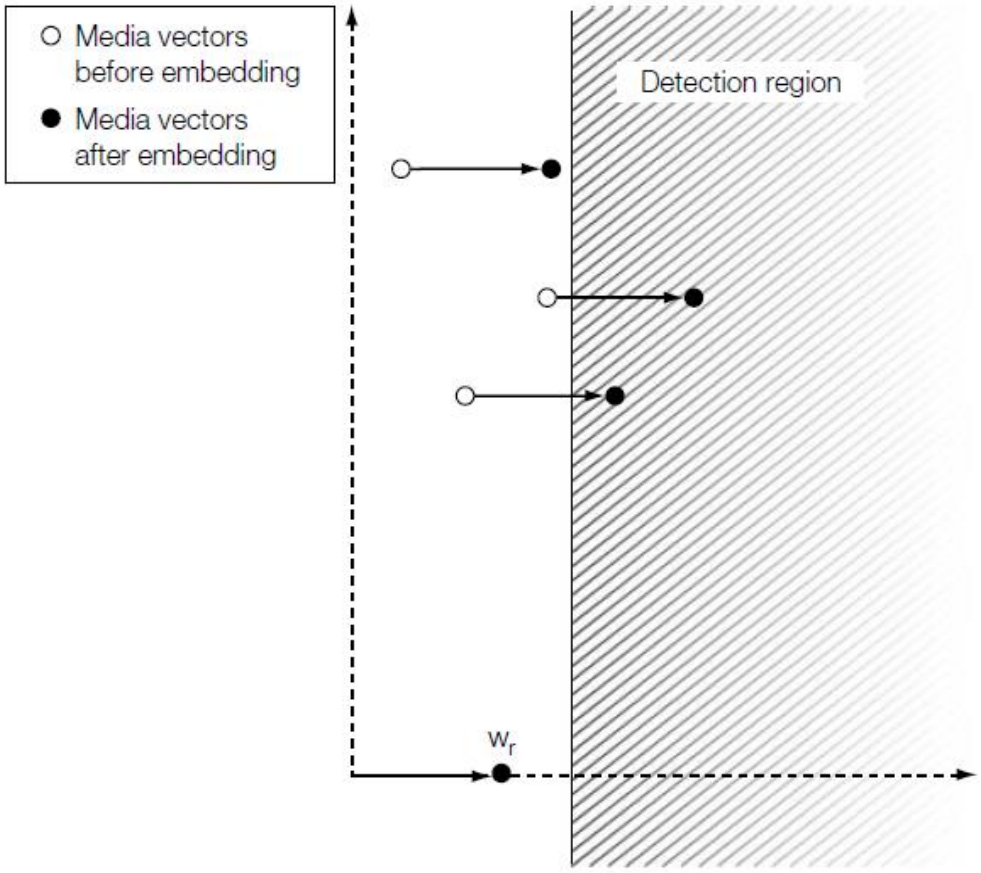

Figure 3.11: The effect of the E_BLIND embedding algorithm [41].

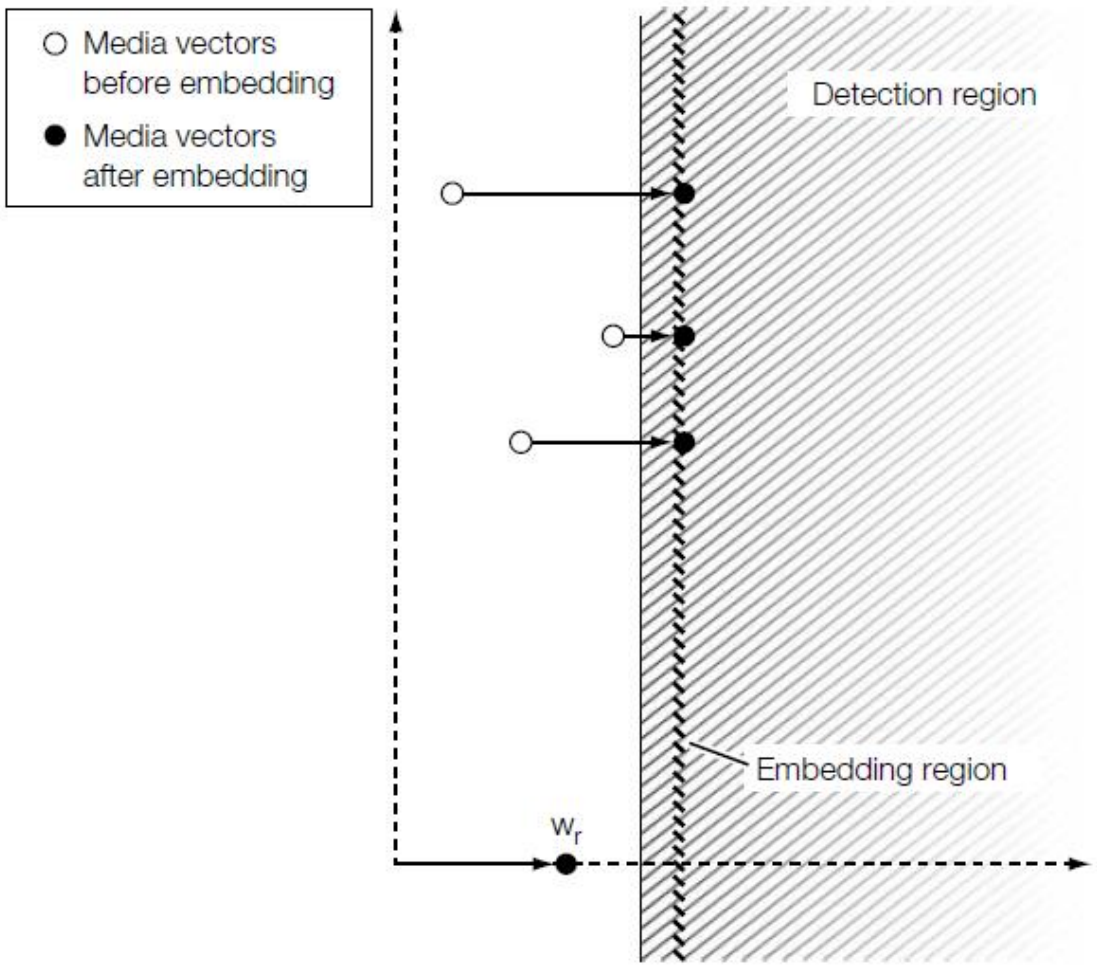

Figure 3.12: The effect of the E_FIXED_LC embedding algorithm [42].

3.3.1.5 Distortion Distribution

To evaluate the aftermath of attacks on watermarked works, it is necessary to know the probability of getting a distorted work, c_{wn}, where the undistorted watermarked work was c_w. This distortion distribution around c_w is the same type of distribution used to describe transmission channels in communication theory.

It is assumed that the distortion distribution is possible to be modeled as

additive Gaussian noise [41]. While this assumption makes analysis simple, it is not an accurate representation of what really happens. Not many of the results act like Gaussian noise nor are they random. Normally, the content will be affected by distortions (i.e., lossy compression, filtering, etc.) Such manipulations are examples of deterministic functions, so the "noise" depends on the content.

Suppose there was an easy way to crop a work. If an image was cropped, the columns and rows of pixels around the edges would be set to black. It is expected for these distortions to happen in many applications, so cwn needs to have a nonzero probability in the distortion distribution around any work. It is important to know that cwn and cw are far away from each other in media space and the points between the two works have a low probability of happening. For example, it is not likely for pixels near the right edge of a given image to have half the brightness of the other pixels. Thus, the distortion distribution is multimodal (many modes), which is different from the results of a Gaussian noise process (one mode) [42].

3.3.2 Marking Spaces

Simple watermarking systems find embedding and detection regions in media space easily. Such is not the case in more complex systems. It is useful to consider part of the system as projecting or distorting media space into a marking space, while the rest of the system can be considered as a simple watermarking system working in said marking space [42].

Usually, detectors function with a clear notion of marking space. Figure

3.13 shows a two-step process of such a detector of extracting and decoding. Watermark extraction uses some method to obtain a vector, in marking space, which may have fewer dimensions than the original. This vector is called an extracted mark. It is then necessary to determine if the extracted mark has a watermark and then decode the embedded message. Normally, the extracted mark is compared against predefined reference marks in order to decode the message. One can think of this step as a basic detector working on vectors in marking space.

Usually, watermark embedders are not created with an explicit notion of marking space. If such functionality is needed, it would involve a three-step process as seen in figure 3.14. First, map the unwatermarked work to marking space. Next, select a new vector in marking space close to the extracted mark so that it might be detected. The difference between the new vector and the extracted mark is called the added mark. Finally, invert the extraction process so that the new vector is projected into media space to get the watermarked work. The purpose is to get a work that will have the new vector as the extracted mark. If marking and media space have the same dimensions, the projection is straightforward. Conversely, if the marking space has less dimensionality, each point in marking space will map to many points in media space. More than one work will produce the same new vector as the extracted mark. While the ideal case is to pick the one visually closest to the original work, what usually happens is that the algorithm will choose one that is close enough.

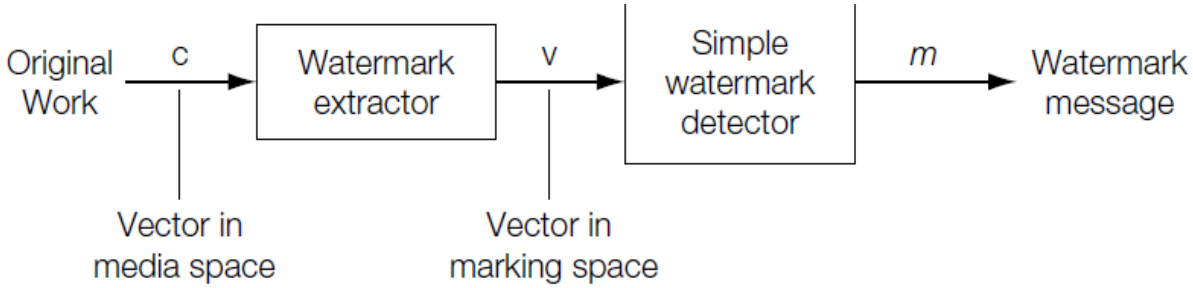

Figure 3.13: General two-step outline of a watermark detector [43].

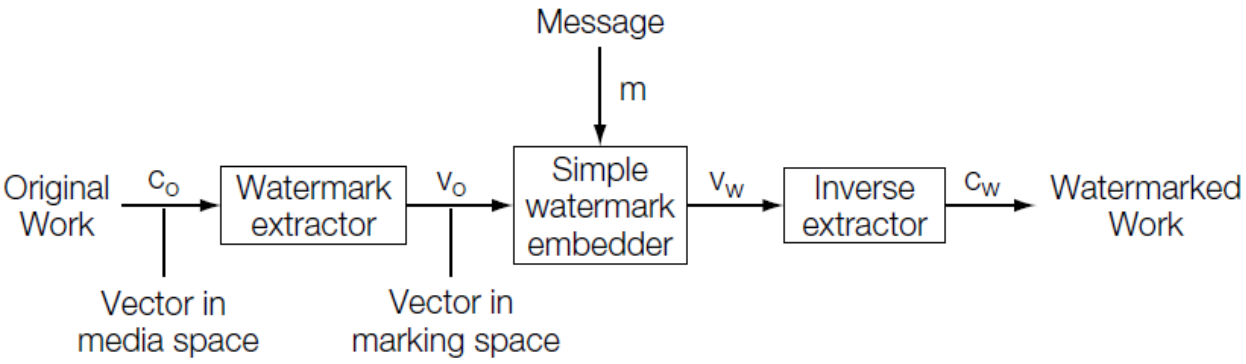

Figure 3.14: General three-step outline of a watermark embedder [43].

One of the purposes of the extraction function seen in systems illustrated in figures 3.13 and 3.14 is to make the embedding and detection process more efficient. Another purpose is to provide a simply way to distribute unwatermarked works, the region of fidelity, and/or the distortion distribution where there is improved performance in simple algorithms. For instance, finding the average of groups of independent samples makes it possible to get a marking space where the distribution of unwatermarked works is more Gaussian, as per the central limit theorem [44]. If one were to apply a frequency transform and then scale the terms by visually determined constants, one could get a marking space where the region of acceptable fidelity is more spherical. Furthermore,

compensating for geometric and temporal distortions will make it possible to get a marking space where the distortion distribution does not have many modes.

3.4 Modeling Watermark Detection by Correlation

3.4.1 Linear Correlation

The linear correlation is the average of the inner product of the elements of between two vectors, c and w_r [45]:

$$z_{lc}(\mathbf{v}, \mathbf{w_r}) = \frac{1}{N} \sum_i \mathbf{v}[i]\mathbf{w_r}[i].$$

It is typical to determine if there is a transmitted signal, w_r, and a received signal, v, in communication by finding the linear correlation and then comparing against a threshold. This method is called matched filtering and it is an efficient way to find signals when additive, white Gaussian noise is present [46].

Geometric Interpretation

The detection region that is produced in matched filtering is the set of all points on one side of a hyperplane, as seen in figure 3.11. The hyperplane is orthogonal to the reference mark while the distance of said hyperplane from the origin is based on the detection threshold. To understand the robustness of the detection region, it is important to note that in the presence of high dimensionality that vectors gained from a white Gaussian distribution are perpendicular to the reference mark. So, if a vector becomes corrupted by the application of additive white Gaussian noise, the noise is usually parallel to the edge of the detection region.

Equivalent Detection Methods

Generally speaking, an algorithm uses linear correlation if it finds the linear function of the samples in a work and then compares said function against a detection threshold. Bender et al. [47] presents an image watermarking detection algorithm where pixels are separated into two groups. The difference between the sums of each group produces a detection statistic, which is compared with the detection threshold to find a watermark. This is the same as correlating the image with a pattern that has 1s and -1s. The pattern has a 1 or a -1 for every pixel that is added or subtracted to the detection statistic, respectively. Koch and Zhao [48] present a different type of image watermarking detection algorithm using a discrete cosine transform (DCT). The DCT is applied to every 8x8 block within the image where coefficients are grouped together in ordered pairs. Each pair is encoded with one bit of the watermark, depending if the first coefficient in the ordered pair is bigger than the second. In other words, the first coefficient might get a 1 if it is larger and a 0 if it is smaller. A pattern must be defined for each bit in order to use this algorithm with linear correlation. Set all coefficient of the block DCT to 0, save for the pair used to encode the bit (which will be set to {1,-1}). Processing the block DCT of one of the patterns against the block DCT of the image will produce a sign that will provide information as to whether the first coefficient is larger than the second. Since DCT is a linear transform, the same result is obtained by correlating the pattern with the image in the spatial domain.

3.4.2 Normalized Correlation

Normalized Correlation is the sum of the normalized inner product between two vectors [49]:

$$\tilde{\mathbf{v}} = \frac{\mathbf{v}}{|\mathbf{v}|}$$

$$\tilde{\mathbf{w}}_r = \frac{\mathbf{w}_r}{|\mathbf{w}_r|}$$

$$z_{nc}(\mathbf{v}, \mathbf{w}_r) = \sum \tilde{\mathbf{v}}[i]\tilde{\mathbf{w}}_r[i].$$

Normalized correlation differs from linear correlation in that there is no dependence on the magnitude on the corresponding vectors extracted from the given work. This is more robust than linear correlation when it comes to dealing with simple changes (i.e., increasing the brightness of pictures).

Geometric Interpretation

The detection region obtained from a threshold in normalized correlation is different from that found in linear correlation. Linear correlation's detection region has all the points on one side of the hyperplane while normalized correlation's detection region will have a conical shape. This is because the inner product of two vectors is the same as the product of their Euclidian lengths and the cosine of the angle between them [49]:

$$\mathbf{v} \cdot \mathbf{w}_r = |\mathbf{v}||\mathbf{w}_r| \cos(\theta),$$

Naturally, the normalized correlation between the two vectors is the cosine of the

angle. Thus, applying a threshold to the normalized correlation is the same as applying it to the angle between the vectors. This produces [49]:

$$\frac{\mathbf{v} \cdot \mathbf{w_r}}{|\mathbf{v}||\mathbf{w_r}|} > \tau_{\text{nc}} \quad \Leftrightarrow \quad \theta < \tau_\theta,$$

where [49]

$$\tau_\theta = \cos^{-1}(\tau_{\text{nc}}).$$

This results in an N-dimensional cone as the detection region for the reference vector, w_r. The subtended angle of the N-dimensional cone is 2□□□□

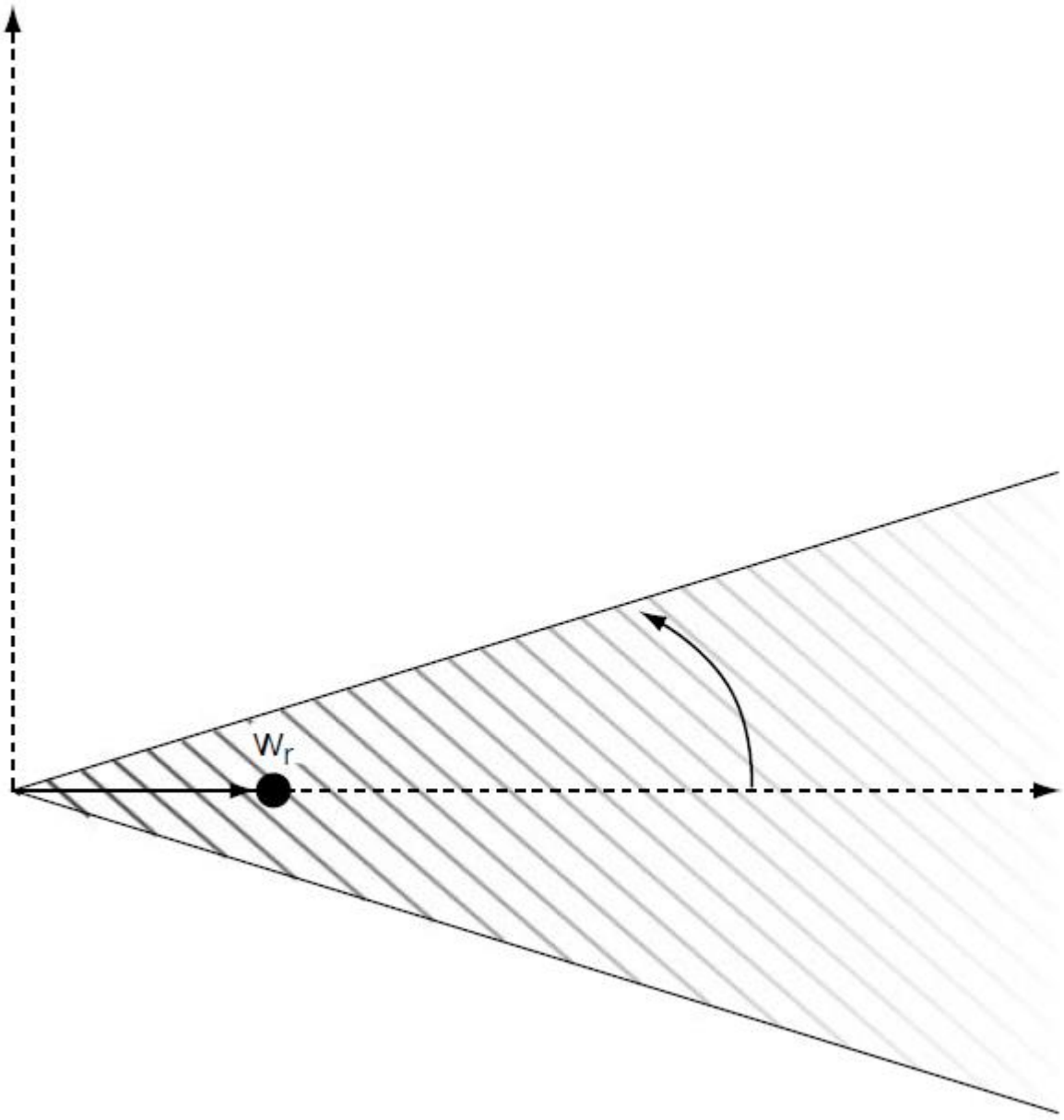

Figure 3.15 Detection region derived from thresholding normalized correlation [50].

Figure 3.15 shows the detection region of the reference vector, wr, with threshold □nc. This figure shows a part of marking space, with the x-axis on the

reference vector and the y-axis is an random direction. The shaded region is all of the points on the plane that will be detected as having the reference mark. A high threshold will produce a small cone while a low threshold will produce a large cone.

Equivalent Detection Methods

Using linear correlation as a detection measure, but scaling the threshold by the magnitude of the extracted mark, is similar to using normalized correlation. So, if the correlation is divided by the extracted vector's magnitude, that would yield [52]:

$$z_1(\mathbf{v}, \mathbf{w_r}) = \frac{\mathbf{v} \cdot \mathbf{w_r}}{|\mathbf{v}|}.$$

Note that the only difference between this and normalized correlation is that the result is not divided by the magnitude of the reference mark. However, reference marks typically have a constant magnitude. So, the difference between z_1 and z_{nc} is a multiplicative constant.

4. Basic Message Coding

4.1 Mapping Messages into Message Vectors

The trouble of watermarking many messages stems from the problem of mapping between messages and watermarking vectors. Traditional communication systems have a similar mapping between messages and signals which is done on two levels. Source coding will map messages into a string of symbols and modulation will map the symbols into signals. Watermarking can follow a similar method where modulation will map from symbols and watermarking vectors [51].

4.1.1 Direct Message Coding

The direct approach to message coding is where a unique, preexisting message mark is substituted for each message. Given a set of messages M, where the number of message are |M|, let W be the set of |M| message marks that will be mapped from a message. W[m] is the message mark linked with the message m, where $m \in M$. The watermark embedder places the message mark W[m] into the work. The process is known as direct coding [51].

The detector calculates the detection values for all |M| message marks and then chooses the message that corresponds with the message mark that has the largest detection value. If the value is lower than the threshold, then there is no

watermark. Otherwise, there is a watermark and the detector produces the index as the detected message. This process is known as maximum likelihood detection[51].

While the message marks should have good characteristics typically sought in watermarking (fidelity, robustness, etc.), one should consider the probability of one message being similar to another. If the message mark is corrupted before detection, the decoder will choose a different message [51]. Thus, it is important to have code separation, where the marks are apart from one another in marking space.

If linear or normalized correlation is used for the detection metric, code separation will rely on having low correlation between the message marks. The best case is to have negative correlation, where embedding a message mark will lower the correlation between the watermarked works and the other message marks [51]. This would lower the chances that another message mark will be selected.

In a system that has two messages, negative correlation occurs when one message mark is the negative of the other. The correlation of the two message marks is -1 and there is maximal separation. In a system with three message marks, the angle will need to be one hundred twenty degrees, as seen in figure 4.1. The figure shows three message marks on a two-dimensional plane in marking space and the corresponding linear correlation detection regions for a

threshold. Generally, the issue with making the ideal set of |M| N-dimensional message marks is the same the issue of placing |M| points on a given N-dimensional sphere, or N-sphere, where all points are equidistant.

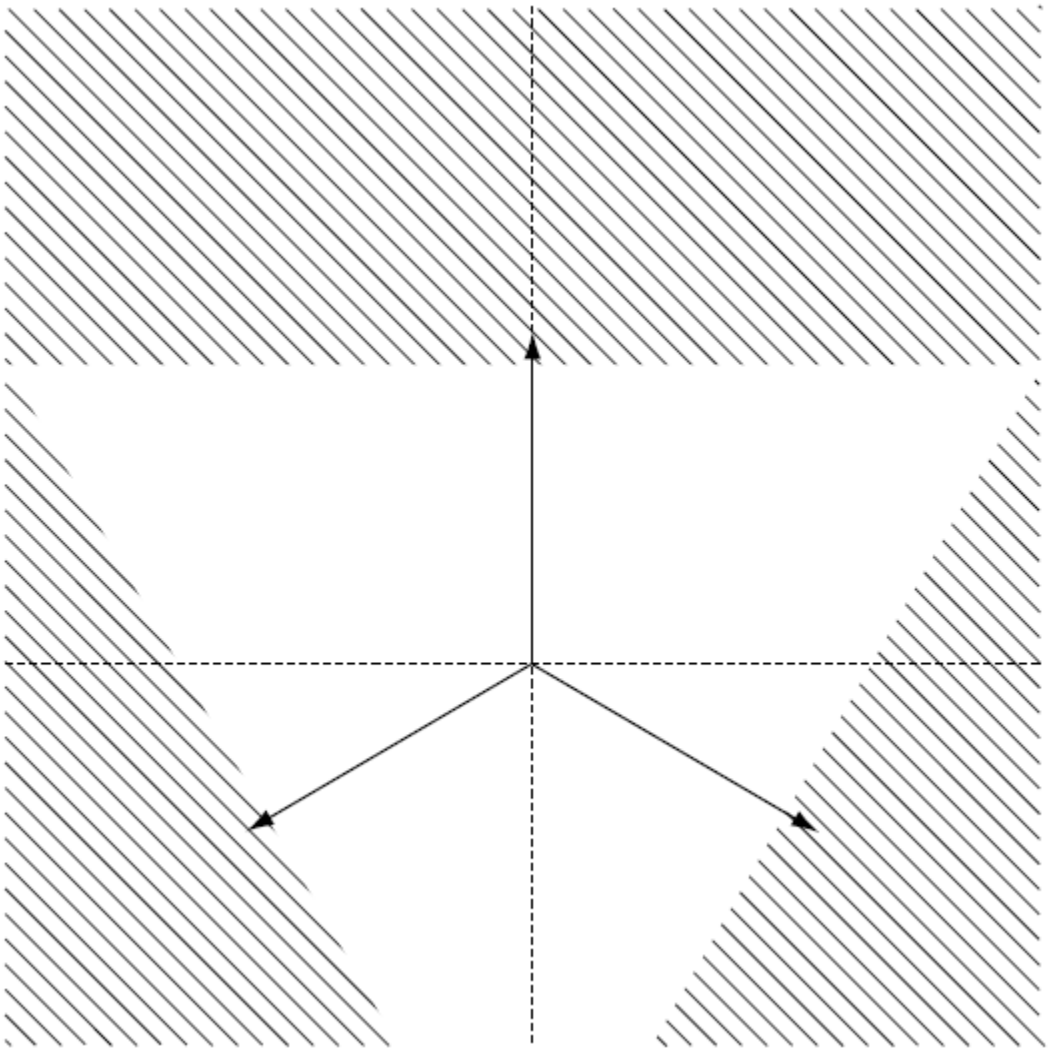

Figure 4.1: Optimal arrangement of detection regions for three messages using linear correlation detection [53].

If the number of messages is larger than the dimension of the marking space, random generated codes will have good code separation [54]. Figure 4.2 shows the result of an experiment, where ten thousand three-message codes were randomly generated in three-dimensional space [51]. Each of the message marks were chosen from an independent and identically distributed Gaussian distribution. Figure 4.2 is the histogram of the average angle of all pairs of message vector in all codes. The most typical angles were close to one hundred twenty degrees. In greater dimensions, where N is greater than three, even more angles will tend towards one hundred twenty degrees which will produce a smoother curve.

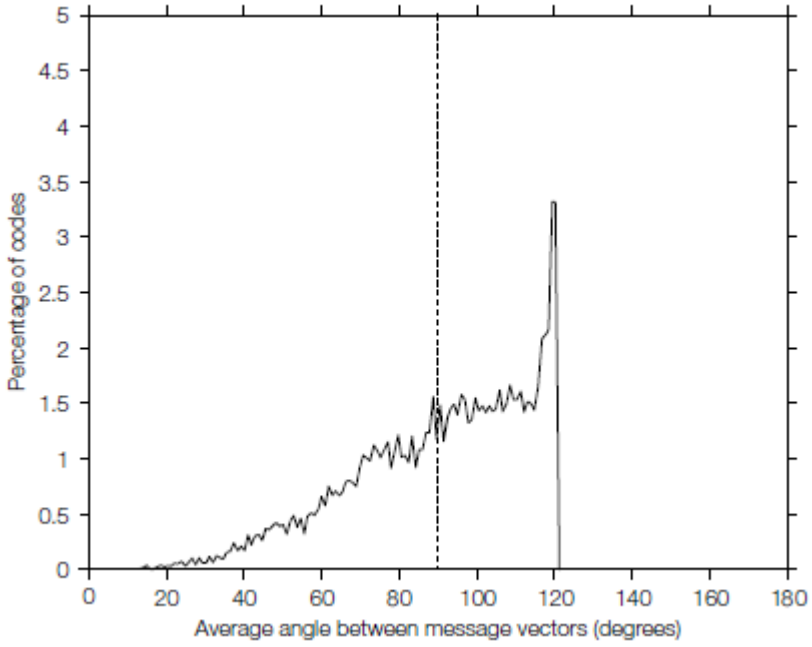

Figure 4.2: Distribution of average angles between three-message vectors in randomly generated three-dimensional codes [55].

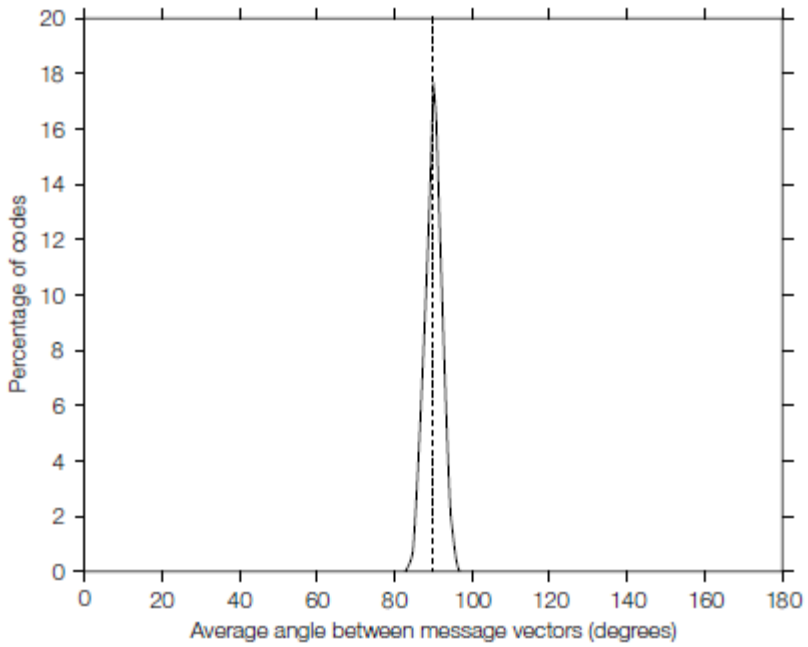

Figure 4.3: Distribution of average angles between three-message vectors in randomly generated 256-dimensional codes [55].

Conversely, if the number of messages is less than the dimension of marking space, the randomly generated message vectors will likely be perpendicular to each other. This can be seen in figure 4.3, where the results of a similar experiment used to generate figure 4.2, and the exception being that the three-message codes are created in a space with two hundred fifty six dimensions. The results show that the most typical angle was ninety degrees.

One feature of perpendicular message marks is that multiple messages can be embedded in a work, provided that the system is based on linear correlation. In such a system, since each message mark will not impact the correlation of other message marks, adding more marks will not affect the detection of previously

placed marks. This is shown in figure 4.4; where two perpendicular message vectors, having zero correlation, are seen in a marking space with of dimension two. The plane has three detection regions: an area for the first message, and area for the second message, and an area for both messages.

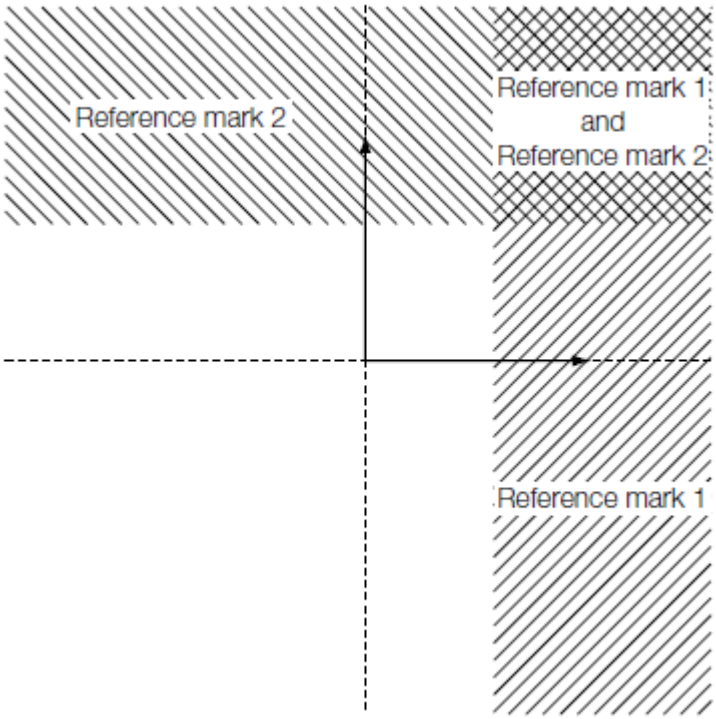

Figure 4.4: Detection regions for orthogonal watermarks [56].

4.1.2 Multi-symbol Message Coding

While direct message coding is effective, it will not scale well. If 16 bits of information are to be encoded, the detector will have to store and compare 65536 different marks, thus increasing the computation time need for a zero-bit watermark by the same amount. Having 100 bits in the same manner would be inconceivable. The solution to this problem is by substituting every message with

a sequence of symbols, where each symbol is an element of an alphabet, A. Every sequence of symbols can then be embedded and detected with its corresponding reference mark.

All messages are represented with the sequence L, where each element is found in an alphabet of length, |A|. Thus, there are $|A|^L$ distinct messages in such a system. The detection process scales better with this method. If |A| = 4 and L = 8, then $|A|^L$ = 65,536. This is the equivalent of 16 bits of information. This new method requires comparing the eight symbols with the four reference marks, which would yield 32 comparisons. This amount is much less than the 65536 comparisons needed in the direct approach [57].

The most basic way to embed a sequence of symbols is to embed a reference mark for each symbol, independent of the other selections. While previous systems would place L reference marks to the work, a more general alternative is to have the embedder process the sequence of symbols into one message mark.

4.1.2.1 Time- and Space-Division Multiplexing

The direct approach would be to split the work into separate regions, with regards to either space or time, and embed a reference mark for a symbol in each region. Thus, the message mark is formed by joining various reference marks. For instance, in order to embed four symbols in an image of size w x h, a reference mark of size w/2 x h/2 would be used. This is an example of space-

division multiplexing. Similarly, embedding a sequence of eight symbols in an audio clip of length l samples would need reference marks of length l/8. This is an example of time-division multiplexing [57].

4.1.2.2 Frequency-Division Multiplexing

A work could also be separated into different bands in the frequency domain and then embedded with a reference mark for each symbol. So, the message mark is made by adding multiple reference marks of different frequencies. This is called frequency-division multiplexing [57].

One way to perform frequency-division multiplexing is to have a watermark extraction function that has a frequency transform and an embedder and a detector that have the same structure as the ones that were covered in the marking space section. A simple watermark embedder will separate the input into pieces and then place a symbol in each. So, the embedder can be the same one used for a system that uses time- or space- multiplexing [58].

4.1.2.3 Code-Division Multiplexing

Since there are multiple reference marks that are independent of one another, there would be no effect on other marks in a system that uses linear correlation. This leads to something similar to code-division multiplexing that is seen in spread spectrum communications. Let a message be a sequence of L symbols selected from an alphabet of size |A|, the set of L x |A| reference marks is W_{AL}. Every mark maps to a given symbol at a given index in the sequence. Let

the reference mark $W_{AL}[i,s]$ be the symbol s at location i. The message mark is produced by adding the corresponding reference marks. Figure 4.5 shows a set of reference marks, say 8x8, that will be used in an image watermarking system, where the system uses a marking space produced by the average of the supposed 8 x 8 blocks. The table of supposed 8 x 8 marks, $W_{AL}[1\ldots5,1..4]$, can be used to represent a sequence of five symbols from an alphabet of size four. The figure displays the encoding process of the symbol sequence 3, 1, 4, 4, 2:

$$\mathbf{w}_m = W_{AL}[1, 3] + W_{AL}[2, 1] + W_{AL}[3, 4] + W_{AL}[4, 4] + W_{AL}[5, 2],$$

with the sum w_m being the message mark.

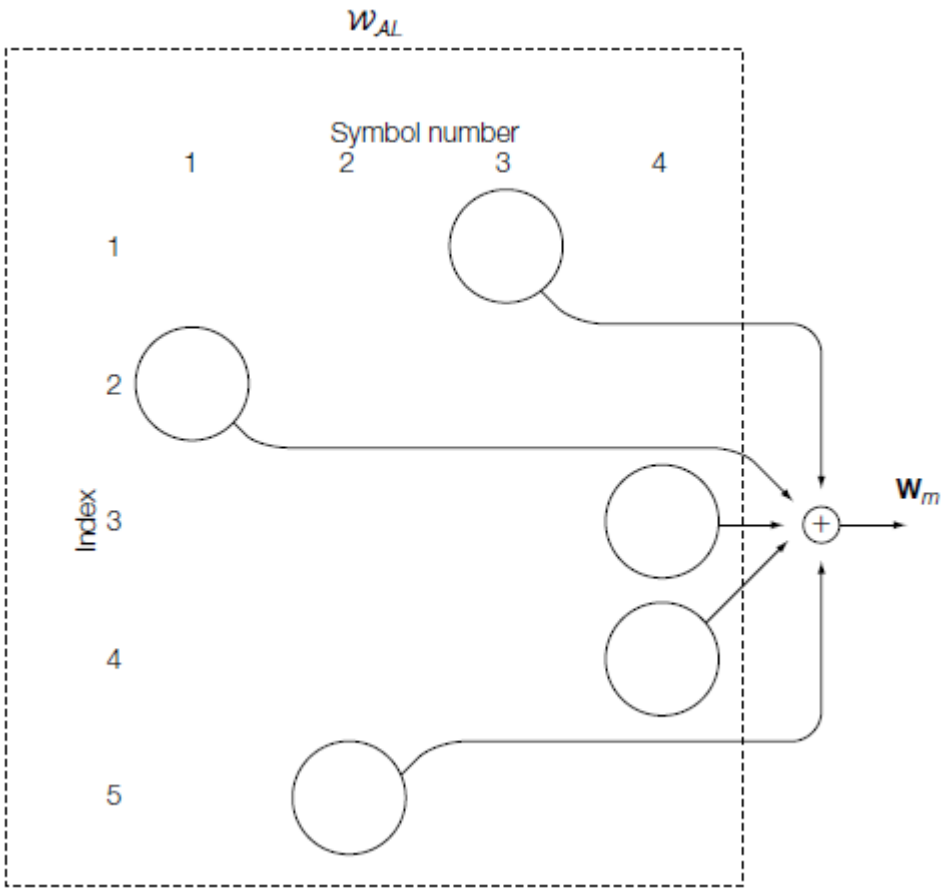

Figure 4.5: Code-division multiplexing using 8x8 reference marks. The rows correspond with the position in the symbol sequence while the columns [59].

All reference marks added to w_m should be nearly perpendicular. Every mark symbolizes a different symbol position where any two marks can be added together only if there is a corresponding symbol in a different position. If, for example, W_{AL} [1, 3] and W_{AL} [2, 1] could be added together in a message mark, but W_{AL}[1, 3] and W_{AL}[1,1] would not. To ensure that the marks are at least nearly perpendicular for any two symbols a and b (even if a=b), $W_{AL}[i, a] \cdot W_{AL}[j, b] \approx 0$ if i = j. Furthermore, the symbols cannot be confused with one another. So,

if a is not equal to b, the two reference marks $W_{AL}[i, a]$ and $W_{AL}[i, b]$ should be clearly different [58].

There exist some cases where there is a concern of registering a work to prevent temporal delay or geometric translation. In these cases, there is a need for the reference marks to be uncorrelated to one another and to their shifted counterparts. In other words, low cross-correlations are necessary for the marks for different symbol positions. If that need is not fulfilled, temporal or geometric shift could cause confusion with regards to reference marks with respect to the symbol and the position. In the case of one-dimensional temporal shifting, sets of code vectors that have low cross-correlations have been developed. Examples of such sets include gold-sequences and m-sequences. Such sets are illustrated in Sarwate and Pursley [60]. Geometric shifting is less studied, but there are attempts to extend to two-dimensional patterns [61][62][63][64][65].

4.1.2.4 Equivalence of Code-Division Multiplexing to Other Approaches

Time-, space-, and frequency-division multiplexing are not typical cases of code-division multiplexing. If a message of length L=8 is processed for embedding in an audio clip utilizing time-division multiplexing, the method to do so would be to concatenate eight short reference marks. Conversely, each of the reference marks could be added to the entire message mark if they were padded with zeroes, as seen in figure 4.6. The figure shows a sequence of eight short reference marks that could be embedded into an audio signal, symbolized as a

sequence of eight symbols. It also shows eight full-length reference marks where the sum matches the prior signal. The full-length marks are perpendicular to one another, where only one of them is nonzero with respect to any given element.

Figure 4.7 shows space-division multiplexing, where a sequencing system has each reference mark have nonzero values in 1/L elements and $W_{AL}[i, s]$ is the shifted version of $W_{AL}[j,s]$ if i-=j. Frequency-division multiplexing can be applied by having patterns that are band limited to the frequency domain to be converted to the temporal or spatial domains. If the transform is linear, the patterns that do not overlap in the frequency domain, but may overlap in the other domains, will have zero correlation in time and space.

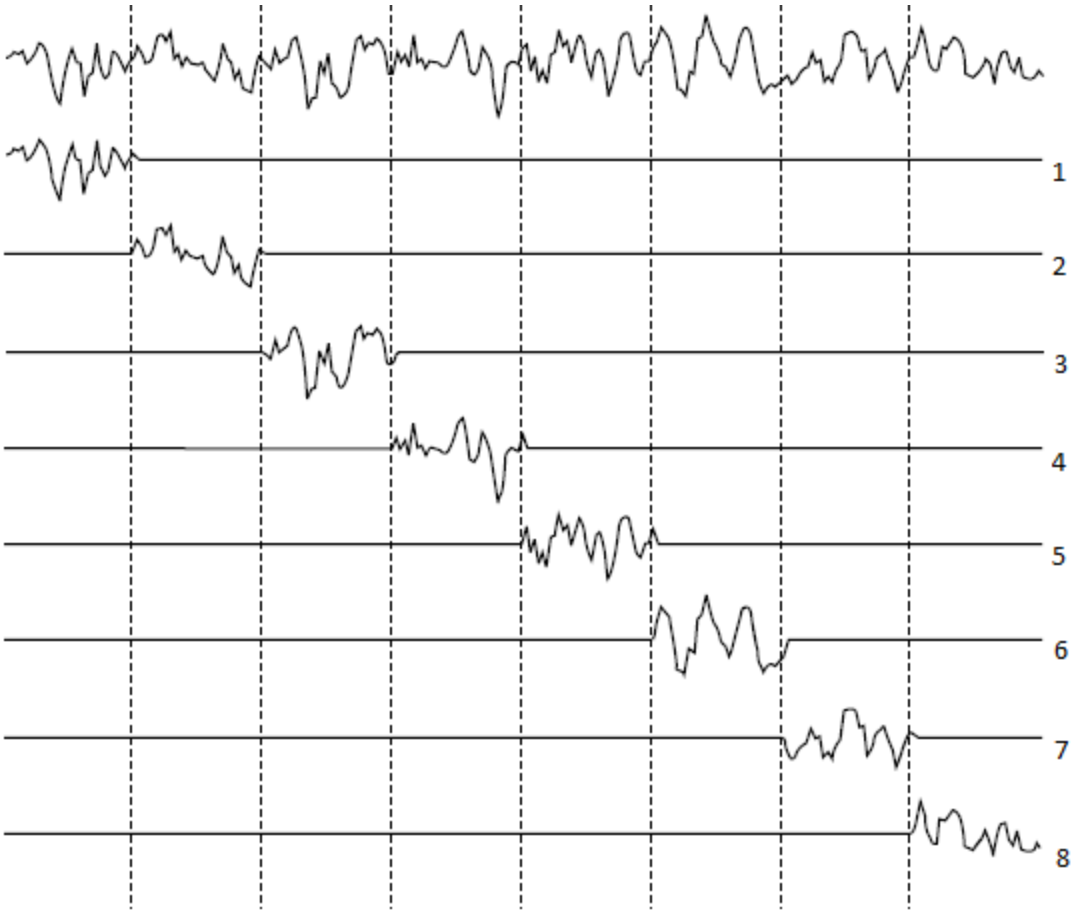

Figure 4.6: Interpretation of time-division multiplexing as code-division multiplexing [66].

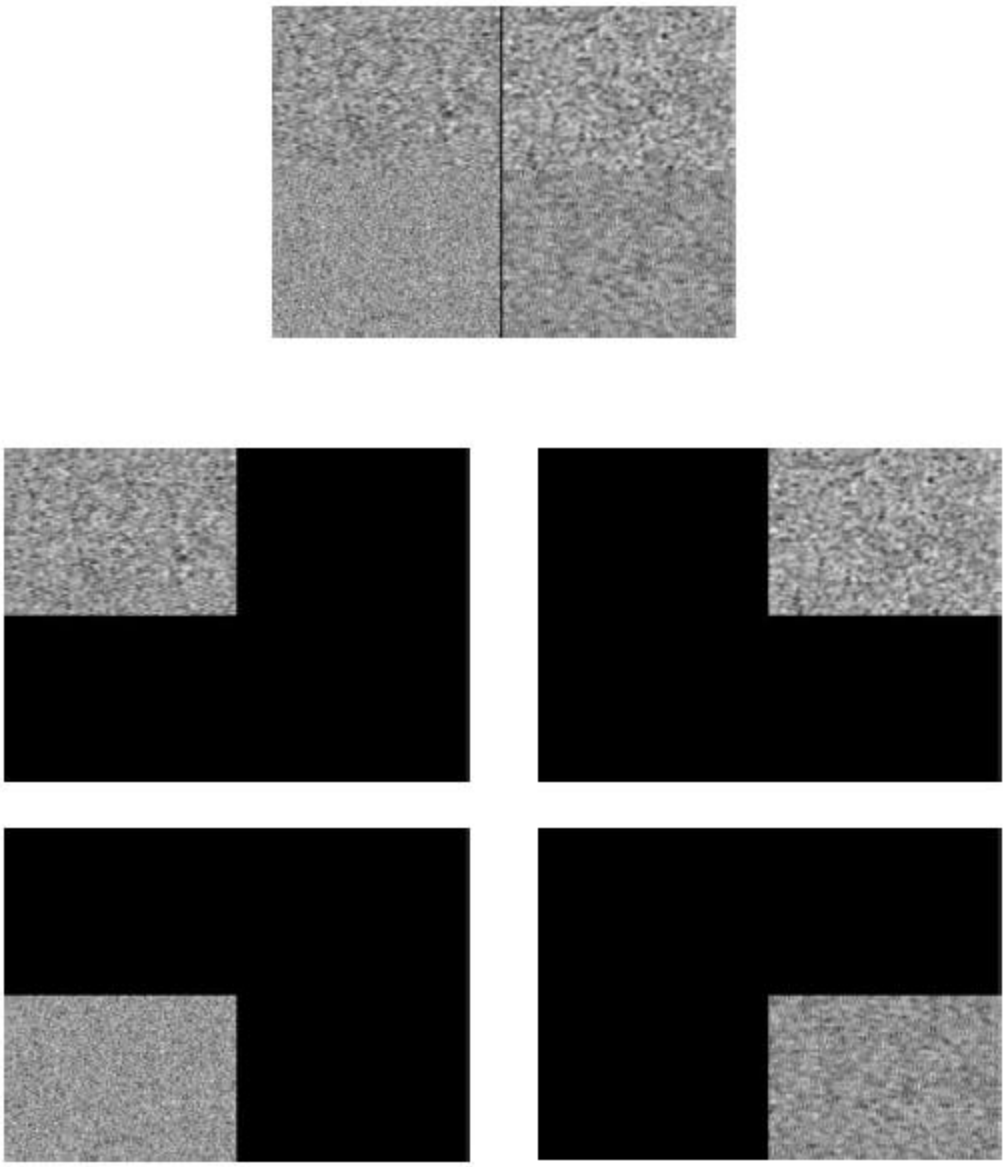

Figure 4.7: Interpretation of space-division multiplexing as code-division multiplexing [67].

4.2 Error Correction Coding

The use of code-division modulation will result in some of the vectors having poor code separation. The concept of error correction codes attempts to

fix this issue. Trellis codes and Viterbi decoding are examples of error correction code [68].

4.2.1 The Problem with Simple Multi-symbol Messages

Any set of message marks is valid for direct message coding. Therefore, the angle between any pair of message marks will be the largest possible one. Conversely, code separation in multi-symbol systems depends on the methods used in source coding and modulation. Some systems where all sequences of symbols represent a unique message and then modulated with code division (i.e., E_SIMPLE_8 or D_SIMPLE_8), there exists a lower limit on the dot product which results in an upper limit on the angle between message marks [69].

Consider a system where the size of the alphabet, |A|, is four with a message length, L, of three. Let

$$\mathbf{w}_{312} = \mathcal{W}_{AL}[1, 3] + \mathcal{W}_{AL}[2, 1] + \mathcal{W}_{AL}[3, 2]$$

be a message mark to be embedded in a work that represents the symbol sequence (3, 1, 2) and

$$\mathbf{w}_{314} = \mathcal{W}_{AL}[1, 3] + \mathcal{W}_{AL}[2, 1] + \mathcal{W}_{AL}[3, 4].$$

be a message mark that encodes the sequence (3, 1, 4). The dot product of w312 and w314 yields:

$$\begin{aligned}\mathbf{w}_{312}\cdot\mathbf{w}_{314} &= \left(\mathcal{W}_{AL}[1,3]+\mathcal{W}_{AL}[2,1]+\mathcal{W}_{AL}[3,2]\right)\\ &\quad\cdot\left(\mathcal{W}_{AL}[1,3]+\mathcal{W}_{AL}[2,1]+\mathcal{W}_{AL}[3,4]\right)\\ &= \mathcal{W}_{AL}[1,3]\cdot\left(\mathcal{W}_{AL}[1,3]+\mathcal{W}_{AL}[2,1]+\mathcal{W}_{AL}[3,4]\right)\\ &\quad+\mathcal{W}_{AL}[2,1]\cdot\left(\mathcal{W}_{AL}[1,3]+\mathcal{W}_{AL}[2,1]+\mathcal{W}_{AL}[3,4]\right)\\ &\quad+\mathcal{W}_{AL}[3,2]\cdot\left(\mathcal{W}_{AL}[1,3]+\mathcal{W}_{AL}[2,1]+\mathcal{W}_{AL}[3,4]\right).\end{aligned}$$

Since all marks in one location are perpendicular to all other marks, the equation is simplified to

$$\begin{aligned}\mathbf{w}_{312}\cdot\mathbf{w}_{314} &= \mathcal{W}_{AL}[1,3]\cdot\mathcal{W}_{AL}[1,3]+\mathcal{W}_{AL}[2,1]\cdot\mathcal{W}_{AL}[2,1]\\ &\quad+\mathcal{W}_{AL}[3,2]\cdot\mathcal{W}_{AL}[3,4].\end{aligned}$$

Assuming all marks are normalized to have unit variance, where $W_{AL}[1, 3] \cdot W_{AL}[1, 3]$ and $W_{AL}[2, 1] \cdot W_{AL}[2, 1]$ are equal to N (in N-dimension space) and $W_{AL}[3, 2] \cdot W_{AL}[3, 4]$ has a lower bound of –N, $w_{312} \cdot w_{314} \geq N$ and the dot product of the closest messages will not be lower than this.

Generally, the smallest result of a dot product of two message marks that have *h* different symbols is N(L – 2h). As L increases, the message marks of the closest pair will become more similar.

4.2.2 The Idea of Error Correction Codes

There is a solution to the problem where a source coding system does not have every possible sequence of symbols that corresponds to a message. Code words are sequences that do refer to a message and corrupted code words are seen as those that do not. Having a mapping from messages to code words would

make it possible to make a decoder that can find the closest code word of a given a corrupted sequence. This is the concept of an error correcting code [70].

Error correcting codes are designed by lengthening the symbol sequences. For instance, let there be a set of 16 different messages. Each message can be represented as a sequence of 4 bits. The encoder for this system can take the sequence as input and produce a longer sequence. If the longer sequence is seven bits, there would be 2^7 possible 7-bit words with only 16 different code words. The code words could be defined so that at the beginning of one, the system could flip at least three bits to get a code word that encodes a different message. If there is a corrupted sequence, the decoder will look for the closest code word that differs from it with the fewest amounts of bits. This is still possible if only one bit was flipped.

If there is a 7-bit (L = 7) code that is encoding a 4-bit message while making sure that the code words of all pairs are different by at least three bits (h = 3), the largest dot product possible between two code words is N(L – 2h) = N(7 – 2 X 3) = N. This provides better performance compared to a system without error correction, where the messages are four bits (L = 4) and can only differ in one bit (h =1), which has a maximum dot product of N(L-2h) = N(4-2) = 2N.

An atypical application of error correction would increase the size of the alphabet. In the example of 4-bit messages, the sequence of symbols may come from an alphabet of size four as opposed to a binary alphabet. Essentially, this is

the same as increasing the length of the message. The only part that is different is how to devise later modulation.

There are a variety of error correcting codes. Hamming code [71] is a basic example of error correcting codes that guarantees that every pair of coded messages will differ by at least three bits and will correct single-bit errors. BCH [71] and trellis codes [72] are more complex examples of error correction codes that will correct more errors. Some turbo codes [73] provide the best performance and a few researchers have used them to encode watermark messages [74][75].

There are other errors to consider besides symbol error correction when using a coding method. Hamming codes handle random errors well while BCH is better suited for burst errors (errors that occur in groups of sequential symbols). The one thing they all have in common is that they are all modulated, so all codes will produce well-separated messages marks.

4.3 Detecting Multisymbol Watermarks

While the previously discussed watermarking systems can hold more data, there is no established method to detect the presence of a watermark. Meaning, while there are ways to map a work into a sequence of symbols, is the given work watermarked? Presented herein are ways to determine if a work is watermarked.

Direct message encoding provides a straightforward method to find the presence of a watermark. The detector calculates the detection value for all messages. The largest value is assumed to be relevant to the embedded

watermark. Therefore, if the largest value passes the threshold, the detector reports the presence of the corresponding watermark. The shape of the detection region depends on the method used to calculate the detection value.

Multisymbol systems are more complicated than direct message encoding. The detector calculates multiple detection values that supposedly mirror the presence of symbols. Comparing multiple values against a threshold will produce different detection regions. Another method uses a detection test where the detection values are not used.

There are three ways to find the presence of a watermark. The first method separates all messages into two groups: valid and invalid. Valid messages are messages that might be embedded and invalid messages are those that will never be embedded. If the detector recognizes an invalid message, then it reports the presence of no watermark. The second method uses linear correlation to determine if all the symbols needed to encode a message are indeed embedded. The third method uses normalized correlation to determine if the most probable message mark is indeed embedded. One should also consider the possibility of a false positive in each method [76].

4.3.1 Detection by Looking for Valid Messages

The direct way to determine the presence of a watermark is to assume that there exist only a few possible symbol sequences that could represent legitimate messages.

4.3.1.1 Application

Suppose there is a system that can embed 2^{16} different messages. Each message is represented with a series of sixteen bits and then joined with a 9-bit checksum created by the sum of the first and second half of the message. So, only one in 2^9, or 512, possible series of bits is actually legal. The detector will first decode the 25-bit message. Then, the first eight bits are added to the second eight bits and then compared to the last nine bits. There is a watermark present if there is a match.

The example presented is the proposed watermarking method for systems that have high data payload [75][77]. From the results of the example, it is assumed that systems with high payloads will have only a few legitimate messages and are thus presented with no dispute [78][79]. Suppose a watermark encodes a text as a string of one hundred ASCII characters. Since most permutations would not produce anything meaningful, it can be assumed that any work that has a meaningful string would have a watermark.

4.3.1.2 False Positive Probability

It is easy to estimate the probability of a false positive if the watermark detector can tell the difference between a valid and an invalid message. Assume that all messages are have equal probability to be found in an unwatermarked work. This is the case in binary systems if there is a fifty percent chance for a bit to hold the value ‘1’ and if there is no correlation between the bits. The probability of a false positive is the fraction of possible messages that are invalid.

With respect to the previous example of 16-bit messages that have 9-bit checksums, the probability of a false positive is 1/512. If the application in question needs a lower false positive probability, the symbol sequences must be longer.

4.3.2 Detection by Detecting Individual Symbols

Another way to detect the presence of a multisymbol watermark of length L is to test for each of the symbols separately. If the symbol's detection value is greater than the threshold, then there is a watermark present.

4.3.2.1 Application

This method is valid when the detection values are linear correlations, but not when they are normalized correlations. In a system that utilizes linear correlation, this method produces detection regions that are defined by the intersection of hyperplanes. Figure 4.8 shows the four detection regions of a 2-bit binary system. The x and y-axis are the reference vectors for bit 1 (w_{r1}) and 0 (w_{r0}), respectively. The four possible messages are: 00, 01, 10, and 11. Each of the messages have their own different detection region and any work that is outside of all the regions is considered to have no message.

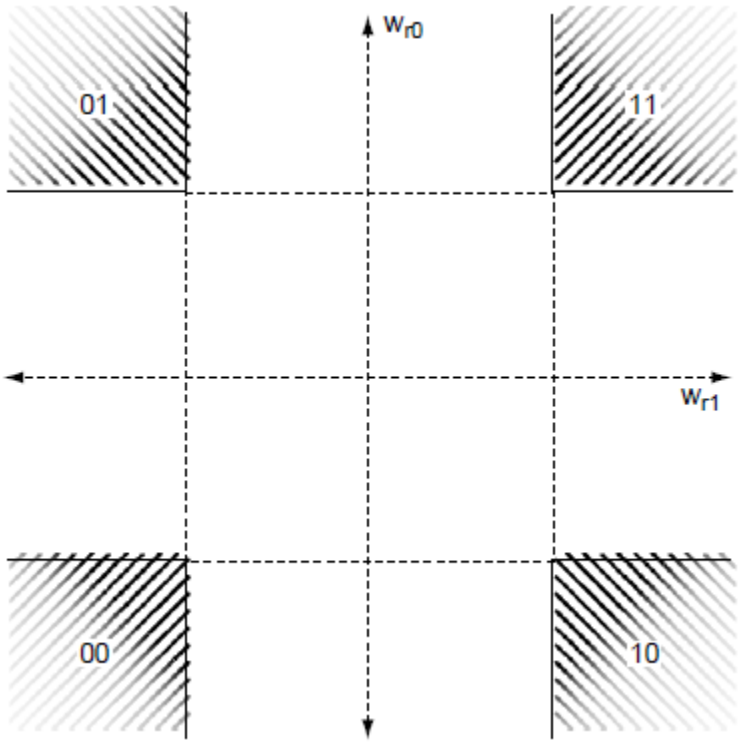

Figure 4.8: Detection regions for four possible messages encoded with two orthogonal watermark patterns [80].

4.3.2.2 False Positive Probability

The false positive probability can be calculated from the probability that a given reference mark produces a correlation that is greater than the threshold, P_{fp0}. Using P_{fp0}, it is possible to approximate the probability, P_{fp1}, where the most probable symbol at a given sequence location has a greater detection value than the threshold. P_{fp1} can then be used to approximate the probability of the most probable symbols in all sequence locations are greater than the threshold. This is the probability of a false positive, P_{fp}.

In order to calculate the estimate of P_{fp1}, first note that the detector performs |A| correlations on each of the L locations in the sequence of symbols and then chooses the symbol with the greatest value to compare it against the threshold. P_{fp1} is the probability that there exists a correlation that is greater than the threshold when there is no watermark. Assuming all the |A| marks for a given location are unique, P_{fp1} is about equal to the sum of the probabilities and thus:

$$P_{fp1} \approx |\mathcal{A}| P_{fp0}.$$

Using P_{fp1}, one can now calculate the false positive probability. The detector reports a positive if all L max correlations are greater than the threshold. The probability that all of the max correlations for each location, assuming independence, are greater than the threshold on an unwatermarked work is:

$$P_{fp} = P_{fp1}^{L} \approx \left(|\mathcal{A}| P_{fp0}\right)^{L}.$$

If the size of the alphabet is small in comparison to the message sequence length, L, then P_{fp} is smaller than P_{fp0}. As the sequence length gets bigger, the detection threshold can get smaller and still have the same probability of a false positive [81].

4.3.3 Detection by Comparing against Quantized Vectors

The previous methods assume that presence of perpendicular reference marks does not manipulate every mark's linear correlation detection value, which is not the case if the detector uses normalized correlation. The previous methods

will not work since additional patterns decrease the detection value for all the others.

Let there be two orthogonal reference marks, w_{r1} and w_{r2}. The vector taken out of the work with w_{r1} embedded might be $v_1 = v_0 + w_{r1}$. The normalized correlation with respect to w_{r1} would be:

$$z_{nc}(\mathbf{v}_1, \mathbf{w}_{r1}) = \frac{(\mathbf{v}_o + \mathbf{w}_{r1}) \cdot \mathbf{w}_{r1}}{|\mathbf{v}_o + \mathbf{w}_{r1}||\mathbf{w}_{r1}|} = \frac{\mathbf{v}_o \cdot \mathbf{w}_{r1} + \mathbf{w}_{r1} \cdot \mathbf{w}_{r1}}{|\mathbf{v}_o + \mathbf{w}_{r1}||\mathbf{w}_{r1}|}.$$

If w_{r2} is added, $v_2 = v_0 + w_{r1} + w_{r2}$, the normalized correlation would be:

$$z_{nc}(\mathbf{v}_2, \mathbf{w}_{r1}) = \frac{(\mathbf{v}_o + \mathbf{w}_r 1 + \mathbf{w}_{r2}) \cdot \mathbf{w}_{r1}}{|\mathbf{v}_o + \mathbf{w}_{r1} + \mathbf{w}_{r2}||\mathbf{w}_{r1}|} = \frac{\mathbf{v}_o \cdot \mathbf{w}_{r1} + \mathbf{w}_{r1} \cdot \mathbf{w}_{r1}}{|\mathbf{v}_o + \mathbf{w}_{r1} + \mathbf{w}_{r2}||\mathbf{w}_{r1}|}.$$

The numerators are the same. Since w_{r1} and w_{r2} are perpendicular, $|v_o + w_{r1} + w_{r2}| > |v_o + w_{r1}|$, so $z_{nc}(v_2,w_{r1}) < z_{nc}(v_1,w_{r1})$. The detection value of wr1 is reduced by magnitude of w_{r2} being embedded. Thus, w_{r2} gets stronger as w_{r1} gets weaker.

How would the detection values of w_{r1} and w_{r2} compare against the detection threshold? Let $z_{nc}(v,w_{r1})$ and $z_{nc}(v,w_{r2})$ be the two detection values. A watermark is present if the two detection values are greater than the threshold.

Really, all that is needed is to compare the lower of the two detection values against the threshold:

$$\min(z_{nc}(\mathbf{v}, \mathbf{w}_{r1}), z_{nc}(\mathbf{v}, \mathbf{w}_{r2})) > \tau_{nc}.$$

Since increasing one detection value would decrease the other, the largest value of the minimum is when the two detection values are equal to one another. Therefore, the largest value of the minimum, where $|w_{r1}| = |w_{r2}| = K$ and $v = w_{r1} + w_{r2}$ would be [82]:

$$\begin{aligned} z_{\min} &= \frac{\mathbf{w}_{r1} \cdot \mathbf{w}_{r1}}{|\mathbf{w}_{r1} + \mathbf{w}_{r2}||\mathbf{w}_{r1}|} \\ &= \frac{K^2}{(\sqrt{2}K)K} \\ &= \frac{1}{\sqrt{2}}. \end{aligned}$$

No detections are possible if the threshold is larger than $\sqrt{2}/2$. If there are L orthogonal patterns to be embedded, the best cases for the minimum for all the L normalized correlations would be $\sqrt{L}/L$.

Figure 4.9 is a geometric representation of the normalized correlation limitation. It has four different detection regions for the individual reference marks in the binary system. The threshold is large enough so that there is no overlap in detection regions. Under this condition, having two marks of equal strength would render them both to be undetectable.

A different test for detecting a watermark can be used to circumvent the problem. Instead of identifying the presence of a mark by evaluating each symbol, just identify the most probable message mark and then test for that message mark. This method is a form of vector quantization.

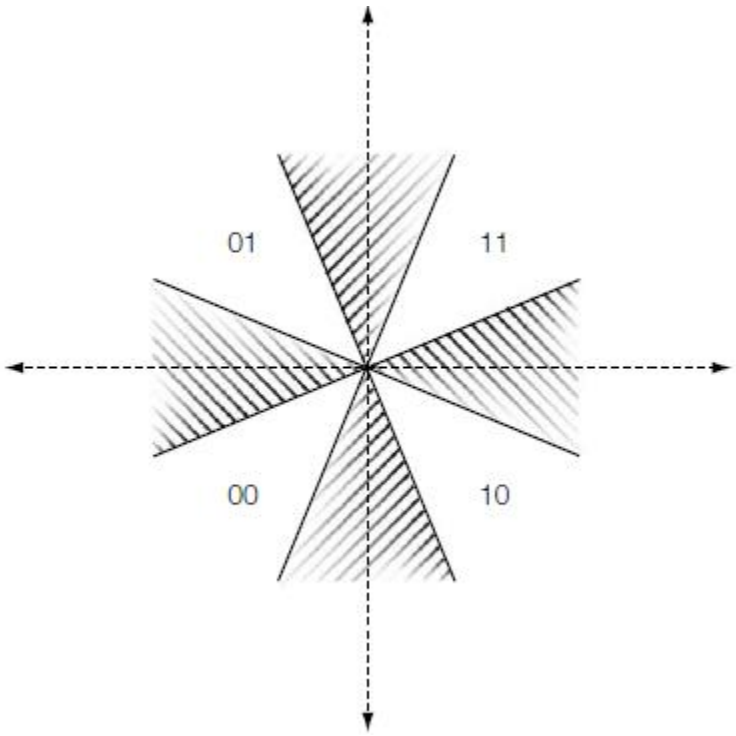

Figure 4.9: A geometric interpretation of the upper bound on normalized correlations for four reference marks in a binary system [83].

4.3.3.1 Application

The most common method decodes the extracted vector, v, which is the first step for finding the most probable message mark and then testing to see if it is present. The decoded vector produces a sequence of symbols, m[1] … m[L]. Next, encode the decoded message. The resulting message mark is the

summation of the most probable message marks. More advance methods would have some form of error correction in the decoding and encoding steps.

The encoding step produces the message mark, w_m, which can be placed in a work that embeds the message m. The next step is to perform a test to see if w_m is present. Normalized correlation can be performed between the extracted vector, v, and the message mark, w_m. Finally, compare the results against a threshold. This produces detection regions than are displayed in figure 4.10.

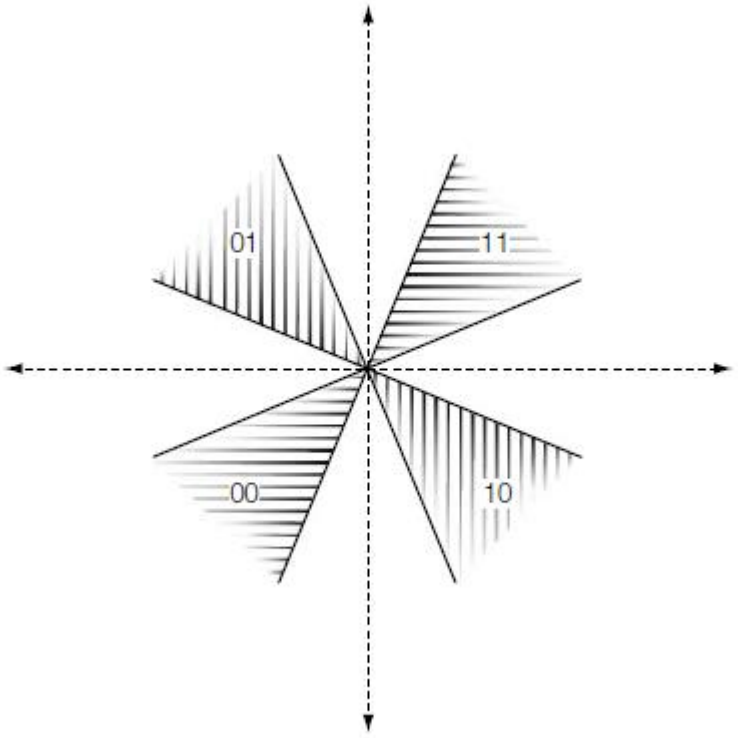

Figure 4.10: The reencoding method for four reference marks [84].

4.3.3.2 False Positive Probability

Generally speaking, the probability of a false positive has an upper bound of the product between the number of possible messages, |M|, and the probability of any one message producing a false positive, P_{fp0}. There is a tight bound when there is no overlap in the detection regions.

To illustrate this, consider a system that uses normalized correlation in its detection method. The detection region for each message is an N-cone. If the subtended angle of the cones is less than the minimum angle between the two message vectors, then none of the cones overlap. The change of getting a false positive in such a system is the sum of the false positive probabilities in |M| different messages. If the probabilities of the |M| different messages are equal, then total false positive probability of the system is $P_{fp} = |M|P_{fp0}$.

5. Watermark Security

5.1 Security Requirements

Depending on the requirements of an application, the security requirements can vary. Sometimes a watermark need not be secure since there is no reason to modify said watermark. A device control watermark is an example of an application that only adds to the content and does not affect the security. If someone alters the device control watermark, the content would behave incorrectly and the content will not be presented correctly. The only safeguard the watermark must have is to protect against normal processing.

For those applications that do need security, the forms of attack are very different. Sometimes the types of safeguards used can differ from various methods of the same application. For instance, consider a copy-control system that prevents people from copying media that they are not supposed to copy. One way to implement this is to prevent watermarks from being removed. Another system could allow people to copy media if there is no watermark present. Thus, the second method would need a way to prevent people from embedding watermarks into the media.

The level of security can vary between different applications based on the expected level of expertise of the attacker. Any military or national government

systems would have a more complex watermark compared to that of one meant to prevent children from seeing inappropriate material.

Before choosing a security model, one must first examine the limitations that applications place on the alteration of watermarks. Certain people might be prevented from adding, detecting, and/or subtracting watermarks. In each type of restriction, there are variations of attack methods that one may use. For instance, while an attacker may be prevented from detecting a watermark, the person may detect and decode or merely attempt to detect the presence of a watermark without trying to read the message. Finally, one must examine the expected skill level of the attacker with respect to the security of the watermark.

5.1.1 Restricting Watermark Operations

Every system of watermarking has individuals that are able to add, detect, and/or subtract watermarks while others are restricted from doing some, if not all, of these actions. A secure watermark is needed in order to have the necessary restrictions. Consider the following scenarios:

1) Alice places a watermark in her radio commercials before sending them off to a number of radio stations. Alice observes the radio stations and logs the broadcast of her commercials with a watermark detector. Later, she will match the logs with the invoices she will receive to identify any false charges. However, Bob is in charge of one of the radio stations and would like to air another commercial instead. He still wants to charge

Alice for air time so he places Alice's watermark in his ad and then places it on the air. Alice's watermark is detected and so Alice believes that her ad went on the air.

2) Alice owns a watermarking service that will add an identification watermark in the owner's images that will be distributed over the internet. There is also a report that will inform the customers of any instance of their watermark that is found on the web. The customers can use this information to locate any unauthorized use of their work. However, Bob has a web crawler that can detect Alice's watermark and lures away Alice's customers with a cheaper service. He can afford to do this since there is no cost to him for placing the watermarks in the images.

3) Alice has a movie studio where she can place a copy-control watermark in the movies she wants to distribute. She assumes that all digital recorders that are able to copy her movies have watermark detectors and can refuse to copy her movie. However, Bob is able to design a device that will remove the copy-control watermark. Using this device, Bob may now produce illegal copies of the movies.

Bob defeats Alice in every scenario by performing unauthorized actions. The first scenario shows an example of unauthorized embedding, or a forgery attack, by adding a watermark that only Alice is authorized to do. The second scenario illustrates an unauthorized detection, or a passive attack, by detecting watermarks

only Alice should be able to detect. Finally, the last scenario is an example of an unauthorized removal by eliminating the watermark that no one should be able to perform. Each of these scenarios frames the different categories of attack with each having its own set of challenges to safeguard against.

The reason as to why to prevent each kind of attack depends on the application. It is important to note the security requirements of an application by finding out who has access to perform which action. Given an application, one could divide a group of users and then assign a set of permissions for each. The resulting permissions table will then determine what type of security is needed in a watermarking system.

Table 5.1 illustrates the permissions needed for different groups of people for each of the three scenarios previously described. Some of the entries in the table show that whether or not a group of people have access to certain permission is unimportant to the functionality of the watermarking system. It should be noted that it may be difficult to deny a group access to one operation while granting it access to another. For example, it is arguable as to whether a system can be constructed that allows people to add a watermark without providing the ability to remove said watermark. If it is not possible, the ability to add watermarks should be denied.

	Embed	Detect	Remove
Broadcast Monitoring			
Advertiser	Y	Y	–
Broadcaster	N	N	–
Public	N	N	–
Web Reporting			
Marking Service	Y	Y	–
Reporting Service	–	Y	–
Public	N	N	N
Copy Control			
Content Provider	Y	Y	–
Public	–	Y	N

Table 5.1: Operations table for the presented scenarios. "Y" = "must be allowed", "N" = "must be prevented, and "-" = "system will work regardless if operation is allowed" [85].

5.1.2 Public and Private Watermarking

Table 5.2 shows two types of permission groups: private and public. In both cases, one can find two distinct groups of people: trusted and public. Trusted people are assumed to be the ones that stand to benefit from the watermark and the public is assumed to be the attackers on the system. In private watermarking, the public is not allowed to access the watermarks. Public watermarking, however, allows the public to see the watermarks only. Public and private watermarking describe the security requirements of an application.

	Embed	Detect	Remove
Private Watermarking			
Trusted	Y	Y	–
Public	N	N	N
Public Watermarking			
Trusted	Y	Y	–
Public	N	Y	N

Table 5.2: Operational table for private and public watermarking [86].

An example of a private system is one that uses informed detectors, provided that the public does not have access to the work that is needed for informed detection. However, it is possible for a public watermarking system where the public has access to the work in question. Suppose there is a web site that provides pictures. Each of these original pictures comes with a number of versions, which were altered by an image-processing technique. This scenario shows an informed detector that is used by a public watermarking system.

An example of a public system is one that uses blind detectors (but without keys), provided that the public knows about the detection algorithm. However, this can be a private watermarking system if the algorithm is kept a secret. Ensuring security in such a system would be difficult.

Finally, the systems that uses blind detection (but with keys) can be used in both public and private watermarking systems. The different being is how the keys are distributed between the trusted and public parties.

5.1.3 Categories of Attack

Tables 5.1 and 5.2 provide a general sense of who is allowed to perform what action. Between the three actions of embedding, detecting, and removing watermarks, there are many variations that have only a few changes. It is possible for an attacker to perform part of an unauthorized action even if the whole action is denied. For example, consider watermark detection. Detection is made up of two steps: detecting and decoding. Some systems might make it easy for an attack to detect the presence of a watermark but can be difficult to decode. This can be a serious vulnerability depending on the application. It is necessary to examine the three categories of actions into the different types of attacks.

5.1.3.1 Unauthorized Embedding

The ultimate embedding attack is where the attacker is able to make and then embed an original message. For example, suppose Alice has a watermarking service. However, she only charges for embedding watermarks and gives away the web-monitoring software for free. Bob wants to use this software but does not want to pay Alice to embed his identification watermark. Bob must create his own watermark that signifies that the item of interest is his and he must build a tool to place the watermark said item. This attack can be prevented using cryptographic techniques.

A partial embedding attack happens when the attacker obtains an authentic watermark, instead of creating one, and then places this watermark into media. Consider the first scenario where Bob wants to place Alice's watermark in his

advertisement. If the watermarking system has serious flaws, Bob can find out the pattern Alice uses to place her watermark in her advertisements. Then, he can copy said pattern into his advertisements without needing to know how the watermark was encoded. This type of attack is known as the copy attack [87].

5.1.3.2 Unauthorized Detection

Some applications are concerned with preventing people from decoding watermarks. For example, suppose a hospital has a policy of adding a watermark to its patients' X-rays. In order to maintain patient confidentiality, potential attackers should be prevented from decoding the watermark. This can be done with cryptographic techniques.

In other applications, an attacker might simply be satisfied in knowing if a watermark is present, regardless if he can decode it or not. While this is more of a consideration for steganography rather than watermarking, the possibility of a watermarking system remains where it can be compromised by an attacker, whose focus is to detect the presence of a watermark. Being able to find watermarks can provide an advantage in trying to remove them. Therefore, a watermark must have security against unauthorized detection before being guarded against unauthorized removal.

The third type of detection occurs in between the last two detection scenarios. The attacker is able to understand the differences between watermarks, but is unable to decode them. So, if there are two watermarked works, the

attacker can determine if the two works use the same or different watermarks. This only becomes a problem if the attacker can determine the markings without decoding them. For instance, suppose Bob wants to steal Alice's customers by finding their watermarks on the web and offer a cheaper price than Alice's watermarking service. Alice can try to prevent this by using a code for linking watermarks with paying customers, thus preventing Bob from decoding the watermarks. However, if Bob can tell if two watermarked works contain the same mark, he can still take customers from Alice. All Bob would have to do is ask Alice's clients to provide him a sample of the watermarked works. He can then search the Web for any work that has the same watermark. Bob can then steal Alice's business without decoding her watermark [88].

5.1.3.3 Unauthorized Removal

A watermark is essentially removed from media if it can no longer be detected. The most extreme case of removal would be to restore the altered media to its original form. Preventing an attacker from obtaining the original work is important for watermarking applications that require security against unauthorized removal. However, simply restricting access to the original is not enough.

Most of the time, an attacker will alter the watermarked media so that it looks like the original media but the watermark detector will not be able to detect the watermark. The original media is but one of many examples that fits this description. Therefore, preventing the attacker from acquiring the original media is but a small step in securing against removal.

An attacker can render a watermark unusable without severely damaging the watermarked work. The attack is similar, but not necessarily the same, as the embedding process of placing a watermark on media. Furthermore, the requirements necessary for this kind of attack are usually not as strict as those that are needed to embed them [89][90]. This is a comparison of the integrity of the watermarked media versus the attacked media.

The attacker can be satisfied with an attacked work if the results of an attack fall within a certain range. The range that is designated as a successful attack is complete removal of the distortion of the watermark where the original detector is not complex enough to detect it. An elimination attack removes a watermark while a masking attack obfuscates the presence of a watermark [91].

To illustrate a masking attack, suppose there is a watermarking system for images that is unable to detect a watermark if the image was rotated slightly. The watermark is effectively removed for the purposes of detection. However, a smarter watermark detector would be able to restore the image and then proceed to its normal detection process.

The elimination attack is very different from the masking attack. The elimination attack tries to predict the pattern of the watermark and then remove it from the watermarked media. The results of the attack may be a close approximation, but not an exact copy, of the original work.

While it is clear that the elimination and masking attacks are two separate methods, the distinction is less clear in media space. Both the rotation attack (masking) and the estimate-and-subtract attack (elimination) relocate data from the areas the detector is interested in to outside the areas of consideration. An inferior detector can be replaced with one that includes a search algorithm to look for watermarks that are rotated. The question becomes then, why can't every attack be countered in this manner?

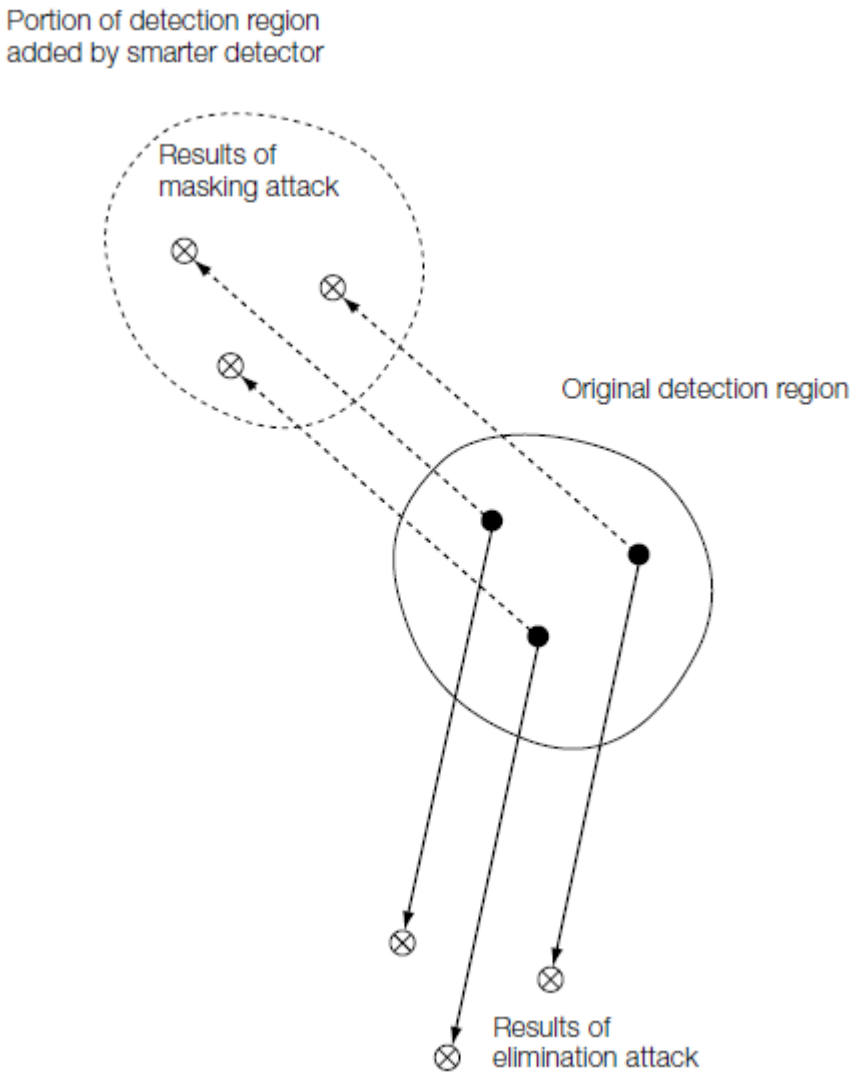

Figure 5.1: Picture of elimination and masking attacks [92].

There are attacks that cannot be countered by simply changing the detector. Attacks that recover the original media are the most glaring example. Other attacks can be found by comparing the location in media space of attacked and unwatermarked works. Works affected by the rotation attack have the same characteristics that the embedding process bestowed. In other words, these

characteristics are very unlikely to show up in unwatermarked works. So, if the detection area is widened to include areas around the attacked work, the possibility of a false positive will not be significantly increased.

Conversely, an elimination attack can have characteristics to works that have no watermark. Increasing the detection area to include areas around the attacked work will significantly increase the possibility of a false positive.

Eliminating attacks can be defined as a process that moves watermarked media into a realm where unwatermarked works are probable and masking attacks can move watermarked media into a realm where unwatermarked works are improbable. An illustration of the difference between masking and elimatination attacks is seen in figure 10.2.

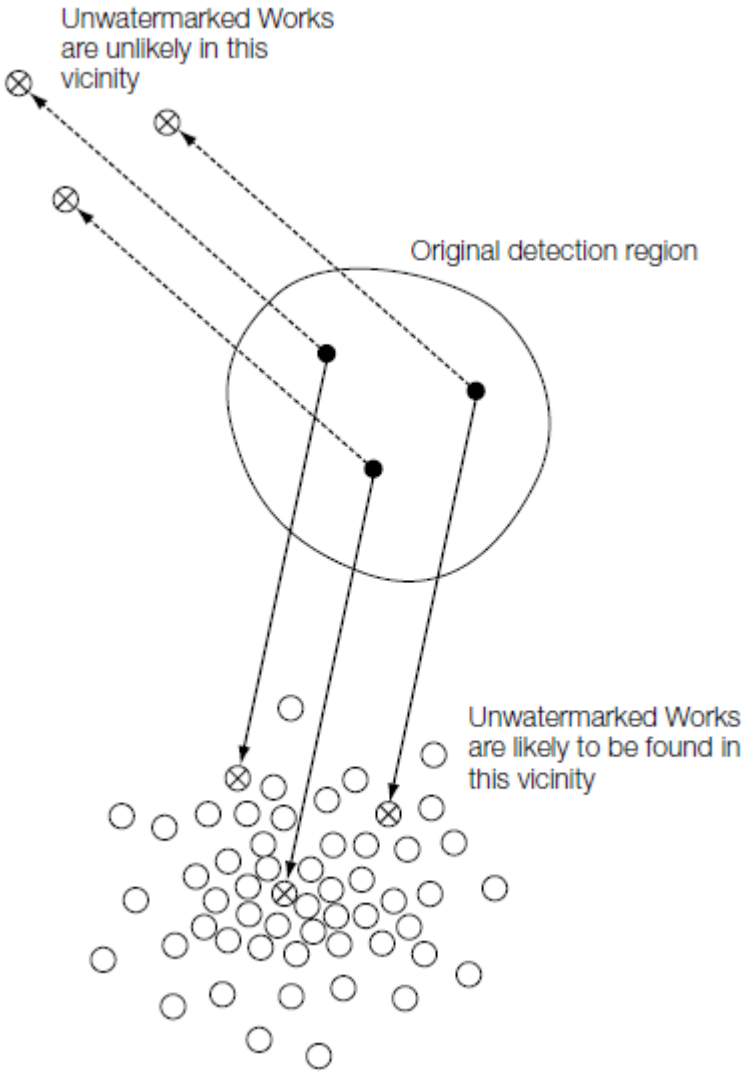

Figure 10.2: Picture showing the difference between elimination and masking attacks [93].

The differences between elimination and masking attacks are important to applications that can change the behavior of the detector after an attack is uncovered. For instance, only the owner would need to detect the watermark in a transaction-tracking program. If an attacker tries to use a masking attack, the

owner has a chance to upgrade to a smarter detector so that the distorted watermark can be noticed. However, if the attacker uses an elimination attack, the owner has little chance of detecting the watermark. Therefore, masking attacks are less serious than elimination attacks when it comes to the possibility of upgrading a detector.

5.1.3.4 System Attacks

Not all attacks are directed against the watermark. The attacker may choose to attack the system without engaging in any of the discussed unauthorized actions. Exploitation in how the watermarks are used, instead of the vulnerabilities of a watermark, is called a system attack. Consider a copy-control application where every recording device has a chip that can detect a watermark. All the attacker would have to do is removed the chip from the recording device to produce illegal copies. Thus, the security of the watermark is rendered trivial.

5.1.4 Assumptions about the Adversary

In order to examine the capability of a watermarking technology, one must makes some assumptions about attackers that will try to exploit the system. Does the attacker have knowledge about algorithm used to make the watermark? What tools can the attacker use to compromise the watermark? Does the attacker have access to the detector?

5.1.4.1 If the Attacker Knows Nothing

The basic case is to assume that the attacker knows nothing about the algorithm and has no tools. The only knowledge the attacker has is the general

knowledge about all watermarking systems and common exploits. Consider an attacker that suspects the presence of a watermark in an item of media. The attacker might try to remove the watermark using image distortion techniques in order mask the presence of a watermark. This is the approach used in the Stirmark program [94], which is successful most of the time.

5.1.4.2 If the Attacker Has More Than One Watermarked Work

When the attacker obtains a set of watermarked works, it is possible to find an exploit without knowledge of the algorithm. These types of attacks, based off of a collection of watermarked works, are called collusion attacks [91].

One type of collusion attack tries to decipher the algorithm by examining different works with the same watermark. A basic example is where the attacker takes the average over a set of works. By taking the average over the set of works, provided that the set all have the same watermark, the result would yield an approximation to the original pattern. The attack could then subtract the pattern to eliminate the watermark from the watermarked works.

An alternative to the first type of collusion attack can be performed on redundant watermarks that are tiled throughout the work. If the same watermark is used in all of the tiles, the attacker could consider each section as a different piece of work and perform the same averaging attack. This attack has proven to be successful against the Secure Digital Music Initiative (SDMI), an audio watermarking system [95].

Another type of collusion attack reverses the first scenario. Instead of having different works with the same watermark, the attacker will use the same work with different watermarks. The objective here is to form some combination of the set of works in order to form the original work. The basic example is to take the average of the works with different watermarks to reduce in impact of each watermark. There are more advanced techniques that can use a smaller set [96].

Boneh and Shaw [97] posed the problem of the second scenario of collusion-secure codes. Assume that the work is sent out to *n* people with a unique code word in each. The code is defined to be c-secure so that if c people conspire with one another, it is highly probably that the resulting work will reveal the identification of one of the conspirators.

It is possible to reveal one of the conspirators if parts of the code word are identical. When the attackers compare their works with one another, the unique parts remain unaffected and are thus unaffected by the attack. Provided that these unaffected portions of the code word contain ample information, one of the attackers can be identified. Other people have studied c-secure codes [98][99][100][101] and have created variations [102].

Cayre et al. [103][104] consider the follow cases where the attacker is in possession of the watermarked works:

1) Watermarked Only Attack – the attacker has a set of watermarked works.

2) Known Message Attack – the attacker has a set of watermarked works and the corresponding messages.

3) Known Original Attack – the attacker has a set of watermarked works and the corresponding originals. The purpose of this case is to obtain information about the secret key to unlock other watermarked works.

5.1.4.3 If the Attacker Knows the Algorithms

It is not safe to assume that an attacker knows nothing about the algorithm in regards to systems that require high security. It can be difficult to keep an algorithm secret. Also, if the algorithm is kept a secret, only a small amount of people will have access to it to find the security flaws.

Kerckoffs' assumption [105] presumes that it is safer to assume that an attacker possesses knowledge about the algorithm, with the exception of the secret keys. Cryptographers not only assume that the attacker has knowledge about the algorithm; they will actually ensure that the algorithm in question is made known to the public. At the very least, fellow cryptographers will be able to find flaws in the system.

There is a possibility than an attacker will be able to find vulnerabilities in the detection scheme to exploit. For instance, the attacker could find a distortion

technique that can trick the detector, thus having a successful masking attack. Furthermore, knowledge of the algorithm could have the attacker discover the secrets of a certain watermark. Consider an algorithm that outputs a tiled watermark pattern. If it is known that the detection process uses the averages of tiles, it is possible for an attacker to discover the tile sizes and approximate the watermark. The attacker may then perform unauthorized embedding, detection, or removal of the watermark.

5.1.4.4 If the Attacker Has a Detector

Thus far, it was assumed that the attacker had no special tools coupled with whatever knowledge about the system to be exploited. However, if the assumption is that the attacker must have certain permissions to perform some actions, it can be assumed that the attacker has tools require for those actions. The most interesting case is where the attacker is allowed to detect a watermark, but not be able to remove them. For this to happen, the attacker must have access to a detector.

If at the very least the attacker has no knowledge about the algorithm, possession of the detector presents an advantage in the attack attempt. One can assume that the detector is a closed system where the attacker can provide modified media as input and examine the output to see if the result is in the detection region. The attack can make systematic changes to the media, process the current iteration, and possibly discover how the detector works.

A more troublesome case is where the attacker has access to the detector and possesses knowledge of its workings. The attacker knows about the algorithm and the keys needed for the detection process. There is no known counter to this case at the time of writing. However, there are a few who are trying to use asymmetric-key cryptography to resolve this issue (which will be presented in section 5.2.4).

5.2 Watermark Security and Cryptography

It is important to note how encryption and watermarking relate to one another. In most cases, the problems of unauthorized embedding and detection are similar to the problems in cryptography and can be solved using cryptographic tools. However, unauthorized removal has no direct translation to the realm of cryptography, thus there is no obvious cryptographic solution.

5.2.1 The Analogy between Watermarking and Cryptography

The embedding and detection functions in watermarking are sometimes similar to encryption and decryption, respectively. Symmetric-key cryptography has an encryption function, $E_k(.)$, that takes a key, K, and a message, m, to produce an encrypted message, m_c.

$$m_c = E_K(m).$$

The encrypted message can be decrypted to produce m by using the decryption function, $D_k(.)$, and the key, K.

$$m = D_K(m_c).$$

Watermarking has a similar setup. There is an embedding function, e(.), that takes a message, m, and the original work, c_0, and produces the modified work, c_w. Furthermore, there is a detection function, d(.), that can take the modified work, c_w, to output m. The mapping of modified works and messages, in most cases, is controlled by the watermark key, K. In informed detection, however, the detection key can include a function that accepts the original work as input. In other words, informed detection can be considered to have a unique key in each case. The following equations describe most watermarking systems and are similar to the two previous equations regarding encryption.

$$\mathbf{c_w} = \mathcal{E}_K(\mathbf{c_o}, m)$$

$$m = \mathcal{D}_K(\mathbf{c_w}),$$

5.2.2 Preventing Unauthorized Detection

Consider the issue of confidential communication between two or more parties. With respect to watermarking, confidential communication needs to prevent the detection and decoding of the watermark message. Therefore, it is preferable to have a watermarking system that can guard against this type of attack.

Sometimes, it may be impractical to implement a watermarking system that thoroughly guards against detection and decoding of watermarking messages. In most cases, it is ideal to have a large amount of unique keys in order to prevent

against unauthorized detection and decoding. If there isn't a large amount of unique keys, brute force can be used identify the right key (assuming that the watermarking algorithm is known). The issue with any system, however, is that one will have to compromise between requirements that may conflict with one another. The result might be a small set of unique keys.

If the problem of a small set of unique keys is unavoidable, the problem of unauthorized decoding can be fixed with cryptography. In applying cryptography, the message is encrypted before being embedded, and is then decrypted after being detected. This system will require two keys: the watermark key, K_w, and the encryption key, K_c. The system can be represented as

$$\mathbf{c_w} = \mathcal{E}_{K_w}(\mathbf{c_o}, m_c) = \mathcal{E}_{K_w}(\mathbf{c_o}, E_{K_c}(m)).$$

The system for the detector is a reverse process:

$$m = D_{K_c}(\mathcal{D}_{K_w}(\mathbf{c_w})).$$

This whole process is shown in figure 5.3. The encryption layer hides the message while the watermarking layer hides the symbols.

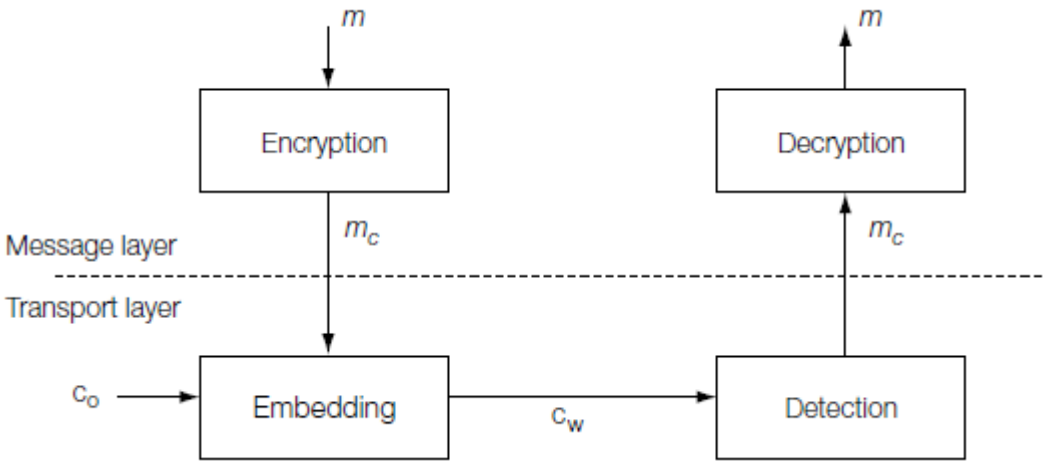

Figure 5.3: Two-layered watermarking system [106].

The communication between the encryption and watermarking layer is similar to the message and transport layer in the OSI (Open Systems Interconnection) model [107]. The message layer determines what messages are transmitted over a network and the transport layer makes sure the messages are not corrupted during transmission. With respect to the watermarking-encryption model, the encryption is part of the message layer while the watermarking is part of the transport layer.

While the encryption layer will prevent an attacker from decoding the message, it can sometimes prevent the detection of a message as well. If a system is supposed to notice the difference between valid and invalid messages, it would be difficult to find a valid message without decoding first. Therefore, an attacker that has a watermark key, but does not have a cipher key, will not find a watermark [108][109].

Most watermarking systems will not prevent the detection of an encrypted message. That is the case in systems that detect messages by comparing a detection statistic and a threshold. It is also the case where detection can be handled through other means. If a watermark is embedded in the least significant bit, there are remnant statistical artifacts found in the histogram. These anomalies can be used to tell the difference between watermarked and unwatermarked works, regardless if the marks blend in with the noise [110][111].

Referring back to the message and transport layer, stopping unauthorized detection without decoding is a problem in the transport layer. The attacker knows that a message is being transmitted. Conversely, encryption is being handled in the message layer and is only concerned with protecting the message [112]. Therefore, it is improbable that unauthorized detection will be prevented directly with cryptographic tools.

5.2.3 Preventing Unauthorized Embedding

The problem with unauthorized embedding is similar to the problem in cryptography with verifying the sender. Depending on the size of the messages, this problem can be solved using either asymmetric-keys or cryptographic signatures. The most direct approach to the problem of unauthorized embedding is to use either of the cryptographic tools in the message layer. So, before embedding, the message should be encrypted with an asymmetric-key or a cryptographic signature.

The direct approach will prevent an attack where the attacker creates a message and then embeds it into the media. Despite the possibility of the attacker having full knowledge of the watermarking algorithm and key, K_w, it is not possible to encrypt the message or make a cryptographic signature without knowledge of the encryption key. If the private encryption key is kept a secret, two parties can safely transmit messages with the use of public and private keys.

The direct approach, however, does not handle an attack where the attacker finds a valid message and then places it in a work. Copy attacks are examples where the attacker takes a valid watermark from legitimate works and copies it to unwatermarked works. The private key, or cryptographic signature, is copied over with the watermark and is then properly decoded. The illegitimate work will appear legitimate.

The receiving party will view the message as valid, but it will be matched with the wrong work. Watermarks are supposed to have information about the media in question, thus the message is incomplete without the proper work. Suppose there is a picture of a sausage pizza, there is a watermark that is embedded that says, "Send food". To guard against a copy attack, one could specify the food item. A future attempt could have the same picture of the sausage pizza with the watermark, "Send cheeseburger". There must be a way to validate the whole watermark message as it relates to the work in question.

There are different ways of adding a description of the work in the message. One could add the work to the message before making the cryptographic signature. In this way, the message and the signature are embedded. The attack would fail because this process modifies the work and would thus invalidate the signature.

An alternative to using the whole work would be to use only a portion of the work, like the lowest-frequency components. If the process is designed to not change this part of the work, the signature is still valid after the work is watermarked. The watermark is embedded using the following steps [113]:

1) Make a description of the work based on immutable information.

2) Make a one-way hash composed of the watermark and the description.

3) Encrypt the hash with a private key to produce a cryptographic signature.

4) Place watermark and signature in work with an algorithm that does not alter the description made in the first step.

The receiver would detect and decode the watermark using the following steps [113]:

1) Detect and decode the watermark to get the message and the cryptographic signature

2) Make the same description that the sender made

3) Make a one-way hash of the watermark and the description

4) Decode the signature using the sender's public key

5) Compare the decoded signature with the hash of the message and description. If there is a match, then it is legitimate.

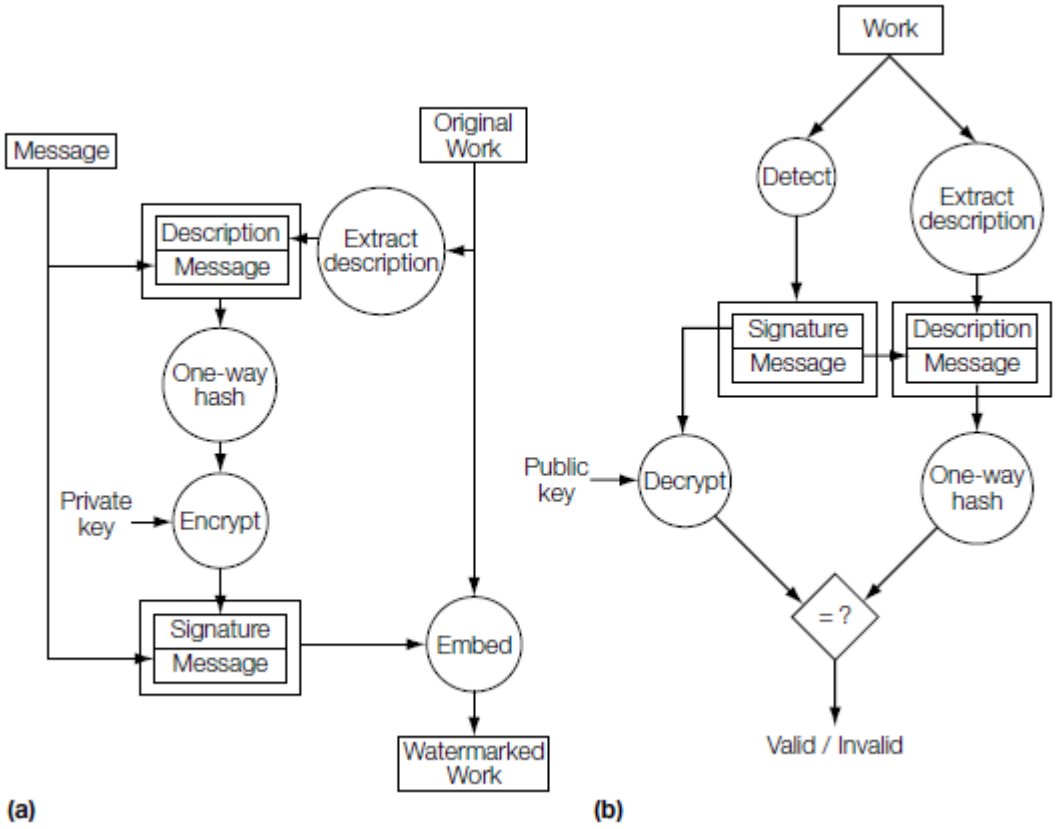

Figure 5.4: A way to link watermarks to their cover works [114].

The problem with using this method is if the work is expected to degrade between the embedding and detection steps. Therefore, the description in the watermark must be robust in order to prevent it from changing in even the

slightest way. A change in the description would change the one-way hash, and the signature will become invalid. To counter this problem, the sender will still create the description of the work to be used in the cryptographic signature. This time, however, the sender also embeds the description along with the message and the signature. The receiver can then get the precise description of the work that the sender used to make the signature. The receiver will then use the signature to authenticate any information that came with the work. The next step is to make an inexact comparison of the embedded description and the description of the work. One could find the correlation of the two descriptions and compare it against a threshold. This solution allows the sent description to be slightly different from what is received without making the signature invalid.

Both methods of including a description must involve the most significant features of the work. Specifically, two different works must have different descriptions. If the descriptions do not differ, an attacker could copy a watermark from a valid work to the intended target. The attacker may then change the target so that the description would match the valid work, from which the watermark was obtained from, and form a valid signature.

The systems that connect a watermark to the work reveal vulnerabilities in the "network layer" of the watermarking system. The model is important because each layer is independent of the others, if there is a clearly defined interface. In the context of the network model, the need for a connection between the message

and the work relies on the message (cryptographic) layer being well suited for the transport (watermarking) layer. The "message layer" will need information about how the message will be encrypted and how the watermarking algorithm selects which properties of the work will remain unaffected in regards to embedding.

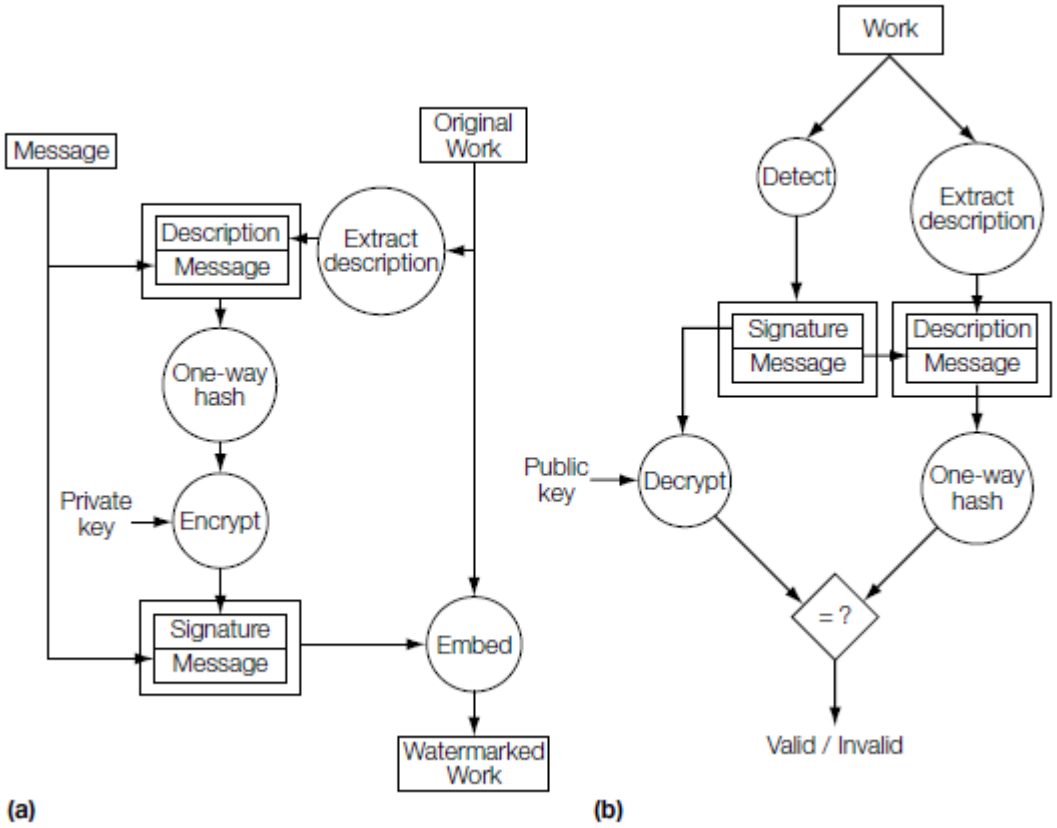

Figure 5.5: A different way to link watermarks to their cover works [115].

5.2.4 Preventing Unauthorized Removal

The previous network model does not address the case of unauthorized removal, since there is no direct analogy between unauthorized removal and most cryptographic problems [116]. With regards to the network model, unauthorized

removal would occur in the “transport layer”, where the attacker stops the message from being received.

Spread spectrum techniques, rather than cryptographic ones, are used to prevent unauthorized removal. Consider the problem of a secure military channel; spread spectrum techniques ensure the delivery of the message.

Spread spectrum sends the signal over many channels rather than what is required to transmit. The signal spread is based on an agreed upon key between two or more parties [19]. Jamming spread spectrum signals become difficult and the case of unauthorized detection is unlikely.

Spread spectrum communications used by the military appear to the enemy as background noise. Under those conditions, it is hard to tell what is or is not a transmission. In the case of watermarking, the embedded patterns can be created so that they have the same characteristics as the work. This makes it difficult for an attacker to detect the presence of a watermark and helps keep fidelity in the work [117]. Su and Girod [118] determined that watermarks that have a power spectrum that matches the work are most secure against Wiener filtering attacks.

Jamming is not a practical form of attack when it comes to spread spectrum communication. Spread spectrum spreads the signal over a large range and most attackers do not have the resources to transmit over this range. Likewise, spread spectrum applied to watermarking prevents the attacker from

adding enough noise to the work in order to nullify the watermark without damaging the fidelity.

Spread spectrum is used when an attacker is denied the permissions of embedding or decoding. However, there are systems, even in copy control, where it must be assumed that the attacker has permission to detect watermarks. In such a case, spread spectrum cannot guarantee secure transmissions. While some hypothesize that such a system is naturally insecure, there is no proof.

One method that is currently under investigation as to whether a watermark can become secure is by making something similar to the asymmetric-key system. In other words, the watermark embedder uses a different key than the one of the watermark detector. It is assumed that merely having the detection key is not enough to let an attacker remove a watermark. While there are a few proposed asymmetric-key watermarking systems, which either have different keys for embedding and detecting or just for embedding [119][120][121][122], there is evidence to suggest that they can prevent removal. Furthermore, it is possible that these systems are vulnerable to sensitivity analysis.

The purpose of asymmetric-key watermarking systems is comparable to asymmetric-key encryption systems. Asymmetric-key cryptography has two different descriptions of a mapping from cleartexts to ciphertexts that comes from two different keys. Asymmetric-key watermarking is similar in that the purpose is to make two descriptions of a mapping from works to embedded messages.

One description, based on the embedding key, provides a map from messages to works while the other description, based on the detection key, provides a map from works to messages [123].

There are some differences between asymmetric-key watermarking and asymmetric-key cryptography that make the analogy imperfect. Watermarking allows a many-to-one mapping between works and messages. Conversely, asymmetric-key cryptography allows only a one-to-one mapping between cleartext and ciphertext. Furthermore, the set of works that have the same message from mapping (detection region of the message) must be grouped in such a way as to provide robust watermarks. So, if the mapping of a watermarked work to a message is changed slightly, the altered work should still be able to provide the same message. Conversely, in asymmetric-key encryption, a small change in the cleartext will have a significant change in the ciphertext and vice versa. This can be seen in figure 5.6.

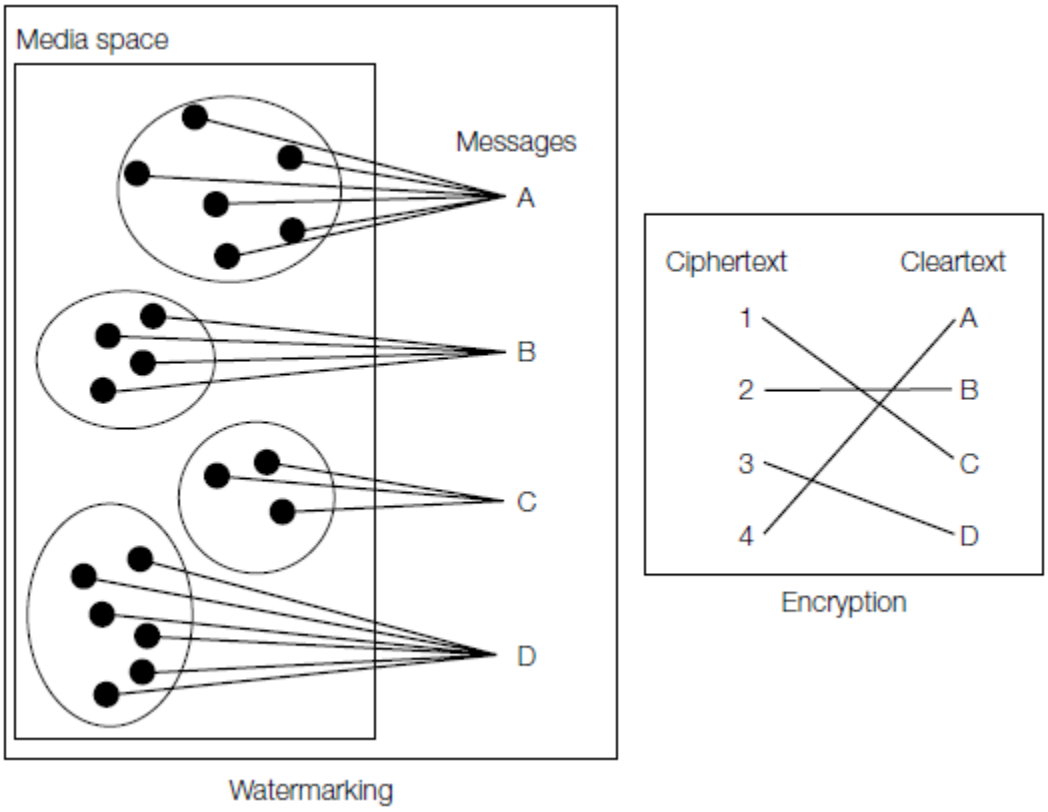

Figure 5.6: Picture showing the difference between watermark and cryptographic mappings [124].

Asymmetric-key watermarking can also be regarded with respect to a shape (detection region of a message) when creating two different descriptions, in lieu of mapping. Embedded key descriptions would provide an easy way to find a point in a shape (watermarked work) that is near to some point outside of the shape (original work). Detection keys would provide an easy way to find a point in a shape, but provide no easy way to find an outside point near the shape. Whether such a pair of descriptions can be created would have to take the details of the system into consideration.

Asymmetric-key watermarking may not be the only way to prevent an attacker, with knowledge of the detection algorithms and keys, from performing an unauthorized removal. Consider a system where the detection region is different from the embedding region. In such a system, there would be a simple embedding region, making it easy to locate an inside point near an outside point, and a complex detection region, where any description would provide some difficulty in locating an outside point close to an inside point. If the detection region encompasses the embedding region, embedding and detection would be easy while removal would be difficult, even with information about the embedding and detecting keys. Such a system would protect against removal without the need for secure keys.

5.3 Some Significant Known Attacks

5.3.1 Scrambling Attacks

Scrambling attacks target the system as a whole where samples of work are scrambled before reaching the detector and then descrambled. The complexity of the scrambling can range from an ordered permutation to a pseudo-random scrambling of sample values, depending on the detection algorithm. For instance, permuting an 8x8 grid will not defeat the D_BLK_CC detector, but will defeat the D_LC detector. The only constant is the scrambling can be visually inverted so that something close to the original work will appear.

The mosaic attack is a common form of the scrambling attack. The process breaks the image into small enough pieces so that the watermark detector

fails to notice the presence of a watermark [125]. These pieces are then placed on a table so that the pieces are close together. Visually speaking, the image is similar to the original work. This method is used to defeat web-crawling detectors, where the scrambling is merely breaking up the image into small pieces and then letting the browser perform the descrambling.

The mosaic attack is easy to perform since most browsers can descramble the image. Generally speaking, anyone benefitting from scrambling attacks will need a descrambling device or program. Suppose a video recording device has a copy-control system. The owner of the video recording device might wish to have a scrambler and a descrambler in order to overcome the copy-control system. The scrambler will manipulate the input of the video recorder so that the watermark will become undetectable. Playback of the scrambled video is sent through a descrambler so that normal viewing is possible. While such a device would be considered illegal, some can argue that the device can prevent children from watching adult content.

5.3.2 Pathological Distortions

A watermark must be robust enough for any process that keeps the fidelity of the work in order to be protected against unauthorized removal. While a normal process requires that a protected watermark be robust, an unlikely process could be used. Anything that keeps the fidelity of the work can be used by an attacker to defeat the detector with either masking or elimination techniques. The

two categories of pathological distortions are geometric/temporal (attacking synchronization) and noise removal distortions.

5.3.2.1 Synchronization Attacks

Most watermarking methods are vulnerable to synchronization. An attacker will try to mask a watermark by disrupting synchronization. Basic examples of this are delay and time scaling for audio and video, and rotation, scaling, and translation for images and video. These distortions are applied in such a way that they alter over time or space. Pitch-preserving scaling and sample removal in audio, and shearing, horizontal reflection, and column or line removal in images are more complex examples of distortions. There are even more complicated distortion processes like nonlinear warping of images. The StirMark program [126] is a tool that benchmarks how a certain watermark is resilient to a number of distortions.

5.3.2.2 Linear Filtering and Noise Removal Attacks

An attacker can also use linear filtering to remove a watermark. Suppose there is a watermark that has significant energy in the high frequencies. Using a low-pass filter will degrade the signal. Furthermore, any watermarking system that make makes the watermark look like noise will suffer from the use of noise-removal techniques.

Su and Girod [118] show that certain watermarking systems are vulnerable to Wiener filtering, which is an efficient linear-filtering/noise-removal attack. It

can be argued that Wiener filter is at worst a linear, shift-invariant process provided that the watermark is independent of the work, both the watermark and the work are composed from zero-mean Normal distributions, and that the linear correlation be the detection statistic.

It was shown that the watermark's protection can improve against Wiener filtering attacks if the power spectrum of the watermark was a scaled version of the original work's power spectrum, thus

$$|\mathbf{W_a}|^2 = \frac{\sigma^2_{\mathbf{w_a}}}{\sigma^2_{\mathbf{c_o}}}|\mathbf{C_o}|^2,$$

where $|c_0|^2$ is the power spectrum of the cover work, $|w_a|^2$ is the power spectrum of the watermark, and σ^2_{wa} and σ^2_{c0} are the variances of the distributions from which the watermark and the work are composed from, respectively. An attacker will have difficulty if the watermark and work are similar.

5.3.3 Copy Attacks

A copy attack is a form of unauthorized embedding where the attacker copies the watermark from a legitimate work and places the copy into an unwatermarked work.

Kutter et al. [127] introduce the concept of the copy attack. Provided a legitimate watermarked work, c_{1w}, and an unwatermarked work, c_2, the method removes the watermark from c_{1w} which will produce something close to the

original, c_1'. While the recommendation is to use a nonlinear noise-reduction filter, any method that produces the original work will be adequate. Next, obtain an estimate of the watermark by subtracting the watermarked work by the estimate produced in the first step:

$$\hat{\mathbf{w}}_a = \mathbf{c}_{1w} - \hat{\mathbf{c}}_1.$$

Finally, the result from the second step is added to an unwatermarked work to produce:

$$\mathbf{c}_{2w} = \mathbf{c}_2 + \hat{\mathbf{w}}_a.$$

The application of this method was used successfully against commercial image watermarking systems and would be a sufficient attack against the E_BLIND/D_LC and E_BLK_BLIND/D_BLK_CC example systems.

The method presented by Kutter et al. [127] needs close approximations of the original work, c_1'. While the obvious counter to this attack is to make sure that such an approximation is not possible, there are other ways of performing the copy attack without needing to create the original (depending on the watermarking system). For example, if the watermark is in the least significant (LSB) bit plane of the work, c_{1w}, creating an approximate is not possible since the values in the LSB plane were most likely random. However, copying the watermark becomes a trivial matter since all one would have to do is duplicate all the values of the LSB plane into the target work, c_2. Therefore, while making an

approximate of the original work was nearly impossible, a copy attack was still feasible. A more complex example is found in Holliman and Memon [128].

The use of cryptographic signatures that link the watermark to the work is a possible counter to the copy attack. Even if the attacker is successful in copying the watermark, the detector will calculate that the watermark does not belong with the target work.

5.3.4 Ambiguity Attacks

Ambiguity attacks create an illusion where a target work appears to have a watermark in it when that is not the case. The attacker can use this method in order to claim ownership of a target work. While this is a form of unauthorized embedding, this is also considered a system attack.

5.3.4.1 Ambiguity Attacks with Informed Detection

Craver et al. [129] describe the ambiguity attack as a method that is used against systems that have an informed detector. The attacker makes a fake watermark from a randomly generated pattern. Then, the pattern is subtracted from the watermarked work to produce a fake original work. While the pattern has characteristics different from the watermark, it is similar to the difference between the watermarked work and the fake original. Assuming the attacker's pattern is different from legitimate watermark, the difference between the actual and fake original will be similar to the fake watermark. Therefore, the owner and the attacker can make arguable claims to the ownership of the work. The owner

can argue that the legitimate watermark is detected in the watermarked work and the fake original. Conversely, the attacker can argue that the fake watermark is detected in the watermarked work and the true original.

5.3.4.2 Ambiguity Attacks with Blind Detection

In the case of a system that uses blind detection, the ambiguity attack creates a fake watermark that mimics a noise signal but possesses a high correlation with the watermarked work. The fake watermark is constructed by obtaining and distorting some features of the watermarked work. The fake watermark is likely to be found in the original work since; by definition, it is found in the watermarked work. The attacker subtracts the watermarked work from the fake watermark to make a fake original and then isolates the fakes in a protected area.

5.3.4.3 Countering Ambiguity Attacks

While it is not possible to prevent an attacker from creating a fake original and watermark, an owner can protect the true original by using a watermarking technique that prevents forgery. The owner may then present better evidence than the attacker's.

Invertibility [129] is the vulnerability that is being exploited by the attacker. A watermarking system is invertible if inverse of the embedding can be practically calculated. The inverse of the embedding method is a function that takes the watermarked work, c_d, as input and produces a fake original work, c_f,

and a fake watermark, w_f, so that the embedding function, E(.), can place the fake watermark into the fake original, producing the watermarked work. Given an embedding function, E(w,c), an inverse function can be created:

$$\mathcal{E}^{-1}(\mathbf{w_f},\mathbf{c_d}) = \mathbf{c_f}$$

so that

$$\mathcal{E}(\mathbf{w_f},\mathbf{c_f}) = \mathbf{c_d}.$$

E_BLIND and E_BLK_BLIND are embedding functions that add a noise pattern, thus their inverses simply subtracts those patterns.

Ambiguity attacks are not possible with noninvertible embedding algorithms. One way to make a non-invertible embedder is to generate a watermark based on the contents of the work. Using one-way hash functions in this method will prevent the attacker from creating a fake original for the purpose of creating a fake watermark. Suppose a watermark is created from a pseudo-random noise generator from the hash of the original. The attacker could find a random watermark that has a high correlation with the watermarked work. However, the random watermark did not use the generator and would not work. It is unlikely that an attacker can find a watermark with the same characteristic as the generator and has a high correlation with the watermarked work.

5.3.5 Sensitivity Analysis Attacks

Sensitivity analysis attacks are a form of unauthorized removal where the attacker possesses a black-box detector. The detector is used to find a way from the watermarked work to the edge of the detection region. It is assumed that the way can be estimated by the normal of the surface of the detection region and that the normal is mostly consistent.

There are three steps to the sensitivity analysis attack for a linear correlation detection region, illustrated in figure 5.7. First, find a work that is close to the edge of the detection region. This work does not need to be visually similar to the original. There are many ways to find the edge of the detection region by manipulating the watermarked work. Some examples include decreasing the amplitude of the contrast or volume, changing the samples with the mean value of the work, or making a linear combination of the watermarked work and a different unwatermarked work. In each case, one could increase the distortion until the watermark cannot be detected. The next step is to estimate the normal to the surface of the detection region. One way to calculate this estimate is to use an iterative technique. The last step is to adjust and subtract the normal from the target work.

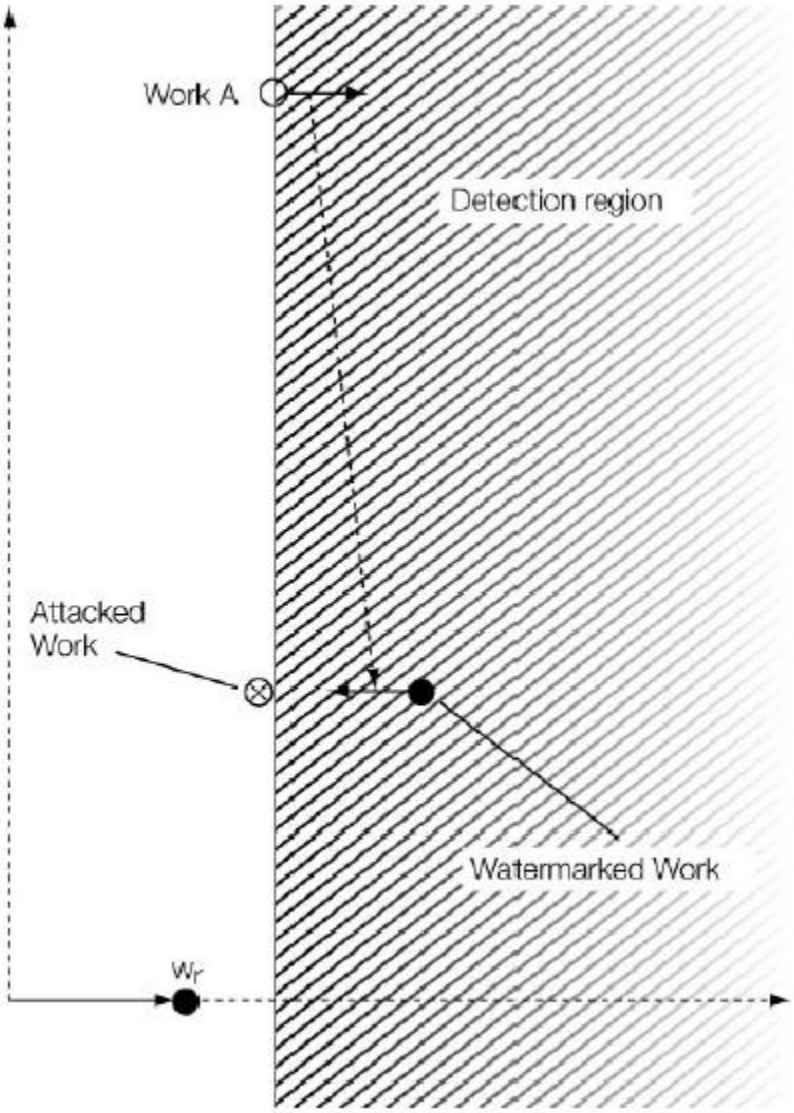

Figure 5.7: Picture showing the three steps to the sensitivity analysis attack [130].

There are two proposals on calculating the normal. Linnartz and van Dijk [131] calculate the vector by evaluating the effect of N N-dimensional modification vectors. Each vector is applied to the work until the detector fails to notice the presence of a watermark. The normal of the surface is then calculated

as the sum of the adjusted vectors, where each vector is altered by the negative of the respective scale factor.

Kalker el al. [132] calculate the normal iteratively. The process adds a random vector in each of the iterations and then checks to see if the detector can identify the presence of a watermark. A positive detection will add the vector to the estimate of the normal. If there is no watermark, the random variable is subtracted from the estimate.

A successful attack relies on the premise that the normal can be used to find a way outside of the detection region. This is true for detection regions that have thresholds on linear and normalized correlation. Failure would occur if the normal at a point on the surface revealed little information about the way to a short path. Thus, creating detection region with this property is of great interest.

5.3.6 Gradient Descent Attacks

The gradient descent attack differs from the sensitivity analysis attack in that rather than evaluating the final binary decision the attacker will have access to a detector than can report the detection values. At each of the iterations of modifying the target work, the attacker will evaluate the detection value in order to estimate the gradient of the detection statistic. The assumption is that the direction of steepest decent is the way outside of the detection region.

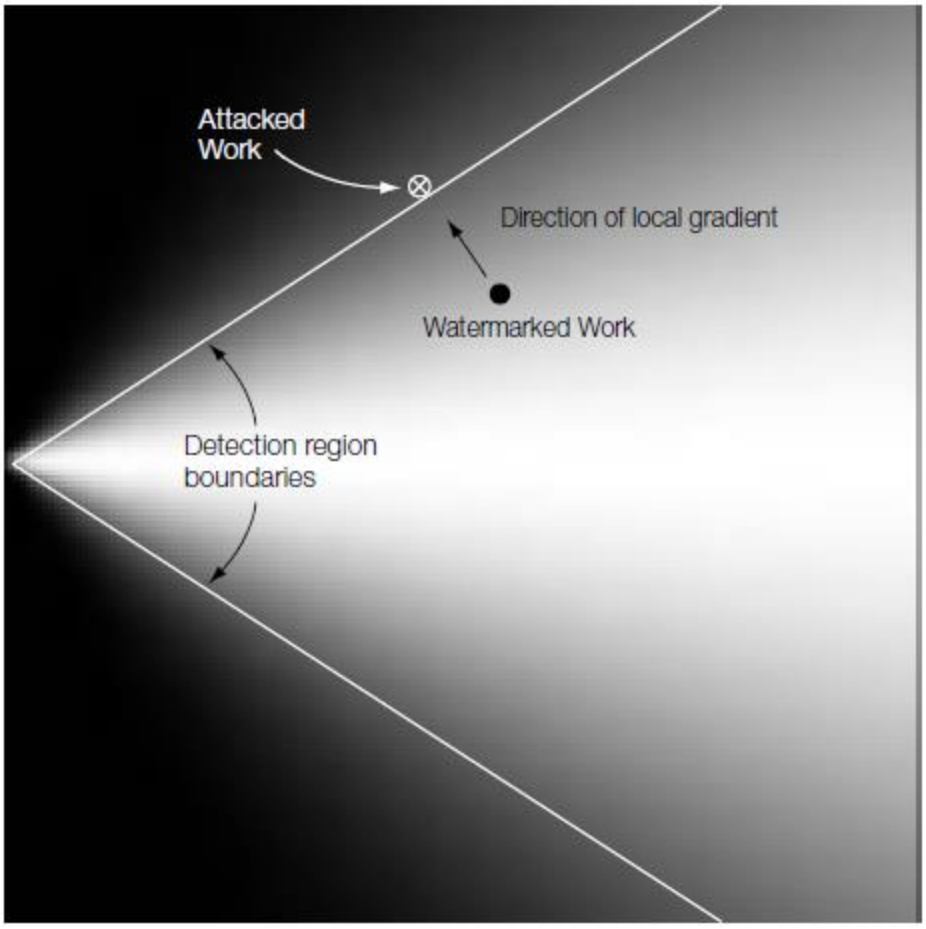

Figure 5.8: The local gradient used to find a short path out of the detection region [133].

Figure 5.8 shows the gradient descent attack where the detector uses normalized correlation statistic. It shows that the reference pattern is a vector that lies along the x-axis. The white lines are the edges of the detection region. The shaded region is the normalize correlation at each point, ranging from high/white to low/black.

Any search method may be used to find the local gradient of steepest decent, in a watermarked work. The work will steadily traverse along the path in each of the iterations until it is outside the detection region.

Success depends on the assumption that the local gradient will show the way to the edge of the detection region, which is the case for many detection statistics like linear and normalized correlation. Therefore, preventing the attack from happening relies on the detection statistic not be monotonically decreased to the edge of the detection region.

6. Conclusion

Basic implementation of digital watermarking is insufficient when it comes to safe-guarding media. Even existing schemes are vulnerable to attack since a single watermark would only be able to address certain types of problems while being vulnerable to others. Despite the drawbacks of the relatively new concept of digital watermarking, the field has potential. Digital watermarking is not as developed as cryptography and has room for improvement. In the future, digital watermarking may see more use once people realize that cryptography is simply not enough.

7. Appendix – Algorithms

E_BLIND/D_LC [134]

Let w_r be a reference pattern. Let w_m be a message pattern that encodes a message m. The embedder scales the w_m with □□to form the added pattern, w_a. The value □□is a scalar that balances between visibility and robustness of the watermark. The algorithm computes the following:

$$\mathbf{w}_m = \begin{cases} \mathbf{w}_r \text{ if } m = 1 \\ -\mathbf{w}_r \text{ if } m = 0 \end{cases}$$

$$\mathbf{w}_a = \alpha \mathbf{w}_m$$

$$\mathbf{c}_w = \mathbf{c}_o + \mathbf{w}_a.$$

Detecting the watermark requires finding the signal w_r with c_o and n present. To do this, compute the linear correlation between the received image, c, and w_r:

$$z_{lc}(\mathbf{c}, \mathbf{w}_r) = \frac{1}{N}\mathbf{c} \cdot \mathbf{w}_r = \frac{1}{N}\sum_{x,y} \mathbf{c}[x,y]\,\mathbf{w}_r[x,y],$$

where x and y are the pixel locations and N are the total number of pixels. If c = c_o + w_a + n, then:

$$z_{lc}(\mathbf{c}, \mathbf{w}_r) = \frac{1}{N}(\mathbf{c}_o \cdot \mathbf{w}_r + \mathbf{w}_a \cdot \mathbf{w}_r + \mathbf{n} \cdot \mathbf{w}_r).$$

Compare the result against a threshold, □$_{lc}$, to determine the presence of a watermark:

E_FIXED_LC/D_LC [135]

This system alters the embedding strength, □□□to guarantee the work is embedded. The embedding strength is calculated by first calculating $c_o \cdot w_m$ / N.

The goal is to guarantee that the magnitude of the detection strength is some constant greater than the detection threshold. This constant is the sum of the detection threshold, □$_{|c}$, and an embedding strength parameter,□□, which is greater than zero. The magnitude of the detection value is:

$$z_{|c}(\mathbf{c_w}, \mathbf{w}_m) = \frac{1}{N}(\mathbf{c_o} \cdot \mathbf{w}_m + \mathbf{w_a} \cdot \mathbf{w}_m),$$

where w_m is a message pattern and w_a = □w_m. Substituting □$_{|c}$ + □□for $z_{|c}(c_w,w_m)$ and solving for □□will yield:

$$\alpha = \frac{N(\tau_{|c} + \beta) - \mathbf{c_o} \cdot \mathbf{w}_m}{\mathbf{w}_m \cdot \mathbf{w}_m}.$$

After finding □□□follow the rest of the steps for E_BLIND.

E_BLIND/D_WHITE [136]

To find the whitening filter, use a covariance matrix where:

$$\mathbf{R}[i_1, i_2] = q^{|x_1 - x_2| + |y_1 - y_2|}$$

for every pair of pixels, i_1 and i_2, where (x_k, y_k) is the position of pixel i_k, and q is some constant between .90 and .99. Substituting the resulting covariance matrix into the probability of getting c_o:

$$P(\mathbf{c_o}) = \frac{1}{(\sqrt{2\pi})^N \sqrt{\det(\mathbf{R})}} \exp\left(-\frac{(\mathbf{c_o} - \mu_{\mathbf{c_o}})^T \mathbf{R}^{-1} (\mathbf{c_o} - \mu_{\mathbf{c_o}})}{2}\right),$$

produces an elliptical Gaussian distribution, where R is a covariance matrix.

To find the whitening filter, calculate the square root of the inverse of R. The resulting matrix will be g_{wh}. The central row produces the filter, f_{wh}.

The detector is similar to the D_LC algorithm, but convolves the image, c, and the normalized reference mark, w_r, by f_{wh} before computing the linear correlation. The detector computes:

$$z_{wh}(\mathbf{c},\mathbf{w_r}) = \frac{1}{N}(\mathbf{f}_{wh} * \mathbf{c}) \cdot \left(\frac{\mathbf{f}_{wh} * \mathbf{w_r}}{s_{wh}}\right),$$

where c is the input image, w_r is the reference pattern, and s_{wh} is the sample standard deviation of $w_r * f_{wh}$. The resulting value is then compared against a threshold to produce the following:

E_SIMPLE_8/D_SIMPLE_8 [137]

Let m be a message where the i^{th} bit in the string is mapped into $W_{AL}[i,1]$ if it is a 1, and $W_{AL}[i,0]$ if it is a 0. Let w_{ri} be a single reference pattern for each bit at location, i. At any location, i, $W_{AL}[i,1] = w_{ri}$ and $W_{AL}[i,0] = -w_{ri}$. Each reference pattern is pseudo-randomly generated using a seed, drawn from independent and identically distributed (i.i.d.) Gaussian distributions. Each pattern is then normalized to have zero mean. The message pattern, w_m, is normalized to have unit variance. The message pattern, w_m, encodes a message, m, by:

$$\mathbf{w}_{mi} = \begin{cases} \mathbf{w}_{ri} & \text{if } m[i] = 1 \\ -\mathbf{w}_{ri} & \text{if } m[i] = 0 \end{cases}$$

$$\mathbf{w}_{tmp} = \sum_i \mathbf{w}_{mi}$$

$$\mathbf{w}_m = \frac{\mathbf{w}_{tmp}}{s_{\mathbf{w}_{tmp}}},$$

where w_{tmp} is the message pattern before normalization and s_{tmp} is the sample standard deviation. The E_SIMPLE_8 embedder calculates the message pattern, w_m, and then embeds using blind embedding. The watermarked image, c_w, is calculated by:

$$\mathbf{c}_w = \mathbf{c}_o + \alpha \mathbf{c}_m,$$

where c_o is the original image and □□is the magnitude input by the user.

The D_SIMPLE_8 detector correlates the received image, c, against each of the reference patterns. The sign of each correlation is used to determine the most likely value for each bit. The detector will not discriminate against any

image, watermarked or not. Therefore, a non-watermarked message processed through the detector will produce a random message.

E_BLK_BLIND/D_BLK_CC [138]

The detection algorithm has two steps:

1) Extract mark, v, from received work, c.

2) Use a detection algorithm to find a watermark in mark, v.

In order to extracting the watermark from an image, the image is first divided into 8 x 8 blocks. Then, the blocks are averaged into an array of 64 values. Therefore, the mark, v, is calculated by:

$$\mathbf{v}[i,j] = \mathcal{X}(\mathbf{c}) = \frac{1}{B}\sum_{x=0}^{w/8}\sum_{y=0}^{h/8}\mathbf{c}[8x+i, 8y+j],$$

where i and j are indices with values in the range of zero to eight, w and h are the width and height of the image, respectively, and B is the total number of blocks within the image.

To detect a watermark in the mark, v, a comparison is made with a predefined reference mark. If linear correlation is used, the detection algorithm will be the same as the D_LC algorithm. An alteration to linear correlation, the correlation coefficient, allows for more robustness, with respect to brightness and contrast, because the means are subtracted from the two vectors and then normalized. Thus, all the values in a vector will be unaffected if a constant is either added or multiplied. The correlation coefficient is defined as:

$$z_{cc}(\mathbf{v}, \mathbf{w_r}) = \frac{\tilde{\mathbf{v}} \cdot \tilde{\mathbf{w}}_r}{\sqrt{(\tilde{\mathbf{v}} \cdot \tilde{\mathbf{v}})(\tilde{\mathbf{w}}_r \cdot \tilde{\mathbf{w}}_r)}},$$

where $\tilde{v} = (v - \overline{v})$ and $\overline{v}$ = mean value of v (the same goes for w_r). Also,

$$\tilde{\mathbf{v}} \cdot \tilde{\mathbf{w}}_r = \sum_{x=0}^{7}\sum_{y=0}^{7}\tilde{\mathbf{v}}[x,y]\,\tilde{\mathbf{w}}_r[x,y].$$

The correlation coefficient can be seen, geometrically, as the cosine of the angle formed between two vectors. Therefore:

$$-1 \le z_{cc}(\mathbf{v}, \mathbf{w_r}) \le 1.$$

The D_BLK_CC outputs:

$$m_n = \begin{cases} 1 & \text{if } z_{cc}(\mathbf{v}, \mathbf{w_r}) > \tau_{cc} \\ \textit{no watermark} & \text{if } -\tau_{cc} \le z_{cc}(\mathbf{v}, \mathbf{w_r}) \le \tau_{cc} \\ 0 & \text{if } z_{cc}(\mathbf{v}, \mathbf{w_r}) < -\tau_{cc}, \end{cases}$$

where τ_{cc} is a constant threshold.

The E_BLK_BLIND embedder has three steps:

1) Extract a mark, v_o, from the original work, c_o.
2) Choose a vector, v_w, in marking space that is close to the extracted mark but in detection space.
3) Project v_w into media space to obtain the watermarked work, c_w.

The extraction process is the same as the detector. Divide the image in to 8 x 8 blocks; find the average of each block, then make a vector in 64-dimensional marking space.

Watermarks are embedded using a blind embedding algorithm like E_BLIND. The added mark, w_a, is equal to αw_m, where $w_m = w_r$ if m = 1 and $-w_r$ if m = 0. The result is the vector v_w, which is equal to $v_o + w_a$.

Finally, projecting v_w into media space requires finding c_w, which is visually similar to c_o. One way to do this is to:

$$\mathbf{c_w}[x, y] = \mathbf{c_o}[x, y] + (\mathbf{v_w}[x \bmod 8, y \bmod 8] - \mathbf{v_o}[x \bmod 8, y \bmod 8]),$$

where mod is the modulo operator.

Works Cited

[1] Tirkel, A. Z., Rankin, G. A., van Schyndel, R. M., Ho, W. J., Mee, N. R. A., & Osborne, C.F., Electronic Watermark. In Digital Image Computing, Technology and Applications (DICTA'93), p. 666-673, Macquarie University, Sidney, 1993.

[2] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T. (2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 2

[3] Hunter, D. *Handmade Paper and Its Watermarks: A Bibliography*. B. Franklin, New York, 1967.

[4] Herodotus. *The Histories*. Penguin Books, London, 1996. Translated by Aubrey de S´elincourt.

[5] Tacticius, A. *How to Survive Under Siege/Aineas the Tactician*. Clarendon Ancient History Series, 1990.

[6] Wilkins, E. H. *A History of Italian Literature*. Oxford University Press, London, 1954.

[7] Bacon, F. *Of the Advancement and Proficiencies of Learning or the Partitions of Sciences*, volume VI. Leon Lichfield, Oxford, for R. Young and E. Forest, 1640.

[8] Brassil, J., Low, S., Maxemchuk,N., & O'Gorman, L. Electronic marking and identification techniques to discourage document copying. In *Proc. IEEE Infocom'94*, volume 3, p.1278–1287, 1994.

[9] Stevens, G. W. W. *Microphotography and Photofabrication at Extreme Resolutions*. Chapman&Hall, London, 1968.

[10] McFADDEN, R. D. Jeremiah A. Denton Jr., 89, Dies; With Blinks, Vietnam P.O.W. Told of Torture. In http://www.nytimes.com. Retrieved April 9, 2014, from http://www.nytimes.com/2014/03/29/us/politics/jeremiah-a-denton-jr-war-hero-and-senator-dies-at-89.html

[11] The Lenna Story - www.lenna.org. In http://www.cs.cmu.edu. Retrieved July 20, 2012, from http://www.cs.cmu.edu/~chuck/lennapg/lenna.shtml

[12] Digimarc for Images – Professional Edition. In digimarc.com. Retrieved April 9, 2014, from https://dfi.digimarc.com/landingPage/campaign/brandedproduct.html?_kk=digimarc%20photoshop&_kt=59840e91-847b-49f2-a0b2-a532d51a8322&gclid=CPP31M6g0r0CFa07MgodkjAAbw

[13] Adaptive Two-Level Watermarking for Binary Document Images. Muharemagic, E. In acm.org. Retrieved April 9, 2014, from http://dl.acm.org/citation.cfm?id=1048675

[14] Digimarc for Images: Best Practice Guide (Chroma + Classic Edition). In digimarc.com. Retrieved April 9, 2014, from https://www.digimarc.com/docs/dfi-pdf/digimarc-for-images-digital-watermarking-best-practices-guide.pdf

[15] Seitz, J. (2005). *Digital Watermarking for Digital Media*. P.7. Information Science Publishing, Hershey, PA, 2005

[16] QR Code Features. In qrcode.com. Retrieved April 9, 2014, from http://archive.is/20120915040047/http://www.qrcode.com/en/qrfeature.html

[17] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 63

[18] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 65

[19] Pickholtz, R. L., Schilling, D. L., & Milstein, L. B. Theory of Spread-spectrum Communications—a Tutorial. *IEEE Trans. on Communications*, 30(5):855–884, 1982.

[20] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 66

[21] Markey, H. K. & Antheil, G. Secret Communication System. Technical Report 2,292,387, *United States Patent*, 1942.

[22] The OSI Model's Seven Layers Defined and Functions Explained. In Microsoft.com. Retrieved April 9, 2014, from http://support.microsoft.com/kb/103884

[23] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 67

[24] Cox, I. J., Kilian, J., Leighton, F. T., & Shamoon, T. Secure Spread Spectrum Watermarking for Multimedia. IEEE Trans. on Image Processing, 6(12):1673–1687, 1997.

[25] Cox, I. J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 68

[26] Shannon, C. E. Channels with Side Information at the Transmitter. IBM Journal of Research and Development, 2(4): 289–293, 1958.

[27] Costa, M. Writing on Dirty Paper. IEEE Trans. Inform. Theory, 29:439–441, 1983.

[28] Heegard, C. & El-Gamal, A. On the Capacity of Computer Memory with Defects. IEEE Trans. on Inform. Theory, 29:731–739, 1983.

[29] Wyner, A. D. & Ziff, J. The Rate Distortion Function for Source Coding with Side Information at the Decoder. IEEE Trans. Inform. Theory, 22:1–10, 1976.

[30] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 76

[31] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 79

[32] Field, D.J. Relations between the Statistics of Natural Images and the Response Properties of Cortical Cells. J. Opt. Soc. Am. A, 4(12):2379–2394, 1987.

[33] Ruderman, D. L. & Bialek, W. Statistics of Natural Images: Scaling in the Woods. Physics Review Letters, 73:814–817, 1994.

[34] Simoncelli, E. P. Statistical Models for Images: Compression, Restoration and Synthesis. Thirty-First Asilomar Conf. on Signals, Systems, and Computers, p. 673–678, 1997.

[35] Wainwright, M. J. & Simoncelli, E.P. Scale Mixtures of Gaussians and the Statistics of Natural Images. In S. A. Solla, T. K. Leen, and K. R. M¨uller, editors, Advances in Neural Information Processing Systems 12, p. 855–861. MIT Press, 2000.

[36] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 82

[37] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 83

[38] Jayant, N., Johnston, J., & Safranek, R. Signal Compression Based on Models of Human Perception. Proc IEEE, 81(10): 1385 - 1422, 1993.

[39] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 73

[40] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 84

[41] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 86

[42] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 87

[43] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 88

[44] The Central Limit Theorem. In http://www.math.uah.edu. Retrieved April 24, 2014, from http://www.math.uah.edu/stat/sample/CLT.html

[45] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 96

[46] Helstrom, C. W. *Statistical Theory of Signal Detection*. Pergamon Press, 1960.

[47] Bender, W., Gruhl, D., Morimoto, N., & Lu, A. Techniques for data hiding. IBM Systems Journal, 35(3/4):313–336, 1996.

[48] Koch, E. & Zhao, J. Towards Robust and Hidden Image Copyright Labeling. In IEEE Workshop on Nonlinear Signal and Image Processing, 1174:185-206, 1995.

[49] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 98

[50] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 99

[51] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, pp. 106-107

[52] Cox, I. J., Kilian, J., Leighton, F. T., & Shamoon, T. Secure Spread Spectrum Watermarking for Multimedia. IEEE Trans. on Image Processing, 6(12):1673–1687, 1997.

[53] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 108

[54] Cover, T. M. & Thomas, J. A. Elements of Information Theory. John Wiley & Sons, 1991.

[55] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 109

[56] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 110

[57] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 111

[58] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 112

[59] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 113

[60] Sarwate, D. V. & Pursley, M. B. Cross-Correlation Properties of Pseudo-Random and Related Sequences. Proc. of the IEEE, 688(5):593–619, 1980.

[61] Delannay, D. & Macq, B. Generalized 2D Cyclic Patterns for Secret Watermark Generation. In IEEE International Conference on Image Processing, volume 2, p. 77–79, 2000.

[62] MacWilliams, F. J. & Sloane, N. J. A. Pseudo-Random Sequences and Arrays. Proc. of the IEEE, 64(12):1715–1729, 1976.

[63] van Schyndel, R. G., Tirkel, A. Z., & Osborne, C. F. A Digital Watermark. In Int. Conf. on Image Processing, volume 2, pp. 86–90. IEEE, 1994.

[64] van Schyndel, R. G., Tirkel, A. Z., & Svalbe, I. D. Key Independent Watermark Detection. In Proc. International Conference on Multimedia Computing and Systems, pp. 580–585. IEEE, 1999.

[65] van Schyndel, R. G., Tirkel, A. Z., Svalbe, I. D., Hall, T. E., & Osborne, C. F. Algebraic Construction of a New Class of Quasi-Orthogonal Arrays in Steganography. In Proc. of SPIE on Security and Watermarking of Multimedia Contents, volume 3657, pp. 354–364, 1999.

[66] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 114

[67] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 115

[68] Lecture 8 – Error Correcting Codes. In http://www.ee.duke.edu/. Retrieved April 24, 2014, from http://people.ee.duke.edu/~mbrooke/ECE283/2004_Fall/Lecture_Material/Lecture_8_Error_Correcting_Codes.ppt

[69] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 117

[70] Error-Correcting Code. In http://mathworld.wolfram.com . Retrieved April 24, 2014, from http://mathworld.wolfram.com/Error-CorrectingCode.html

[71] Rorabaugh, C. B. *Error Coding Cookbook: Practical C/C++ Routines and Recipes for Error Detection and Correction*. McGraw-Hill, 1996.

[72] Viterbi, A. J. CDMA: *Principles of Spread Spectrum Communications.* Addison Wesley Longman Inc., 1995.

[73] Berrou, C. & Glavieux, A. Near Optimum Error Correcting and Decoding: Turbo-codes. IEEE Trans. on Communications, 44(10):1261–1271, 1996.

[74] Eggers, J. J., Su, J. K., & Girod, B. Robustness of a Blind Image Watermarking Scheme. In IEEE Internation Conference on Image Processing, 3: 17–20, 2000.

[75] Pereira, S., Voloshynovskiy, S., & Pun, T. Effective Channel Coding for DCT Watermarks. In IEEE International Conference on Image Processing, 3: 671–673, 2000.

[76] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 125

[77] Tefas, A. & Pitas, I. Multi-bit Image Watermarking Robust to Geometric Distortions. In IEEE International Conference on Image Processing, 3:710–713, 2000.

[78] Chen, B. & Wornell, G. W. An Information-theoretic Approach to the Design of Robust Digital Watermarking Systems. In IEEE Transactions on Acoustics, Speech, and Signal Processing, 4:2061-2064, 1999.

[79] Chou, J., Pradhan S. S., & Ramchandran, K. On the Duality between Distributed Source Coding and Data Hiding. Thirty-third Asilomar Conference on Signals, Systems, and Computers, 2:1503–1507, 1999.

[80] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 127

[81] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 128

[82] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 129

[83] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 130

[84] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 131

[85] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 338

[86] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 339

[87] Kutter, M., Voloshynovskiy, S., & Herrigel, A. *Digital Copyright Technologies*, Stauffacher Strasse 149, 8004 Zurich, Switzerland, CUI-University of Geneva, 1211 Geneva, Switzerland

[88] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 341

[89] Chang, E. C. & Orchard, M. Geometric Properties of Watermarking Schemes. IEEE Int. Conf. on Image Processing, 3:714–717, 2000.

[90] Moulin, P. & O'Sullivan, J. A. Information-theoretic Analysis of Information Hiding. Preprint, available from http://www.ifp.uiuc.edu/moulin/Papers, 1999.

[91] Tanha, M., Torshizi, S. D. S., Abdullah, M. T., Hashim, F. An Overview of Attacks against Digital Watermarking and their Respective Countermeasures, available from http://www.academia.edu/1260639/An_overview_of_attacks_against_digital_watermarking_and_their_respective_countermeasures, 2014.

[92] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 343

[93] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 344

[94] Kutter, M. & Petitcolas, F. A. P. A Fair Benchmark for Image Watermarking Systems. Security and Watermarking of Multimedia Contents, Proc. SPIE-3657:226–239, 1999.

[95] Boeuf, J. & Stern, J. P. An Analysis of One of the SDMI Candidates. In Proceedings of Info Hiding ’01, 2137:395-410, 2001.

[96] Stone, H. S. Analysis of Attacks on Image Watermarks with Randomized Coefficients. Technical Report TR 96-045, NEC Research Institute, 1996.

[97] Boneh, D. & Shaw, J. Collusion-secure Fingerprinting for Digital Data. In Proceedings of Advances in Cryptology—CRYPTO'95, Lecture Notes in Computer Science, 963:452–465, 1995.

[98] Ker, A. Improved Detection of LSB Steganography in Grayscale Images. In J. Fridrich, editor, Preproceedings, Information Hiding, 6th International Workshop, IIH 2004, Toronto, Canada, May 23–25, 2004, LNCS, 3200:97–115. Springer- Verlag, Berlin, 2005.

[99] L'ofvenberg, J. *Random Codes for Digital Fingerprinting*. Ph.D. thesis, Link¨oping University, 1999.

[100] Muratani, H. A Collusion-secure Fingerprinting Code Reduced by Chinese Remaindering and its Random-error Resilience. In Proceedings of International Workshop on Information Hiding '01, 2137:303-315, 2001.

[101] Tardos, G. Optimal Probabilistic Fingerprint Codes. Annual ACM Symposium on Theory of Computing STOC, pp. 116–125, 2003.

[102] Trappe, W., Wu, M., Wang, Z. J. & Liu, K. J. R. Anti-collusion Fingerprinting for Multimedia. IEEE Transactions Signal Processing, 51(4):1069–1087, 2003.

[103] Cayre, F., Fontaine, C., & Furon, T. Watermarking Security: Theory and practice. IEEE Trans. on Signal Processing, 53(10):3976–3987, 2003.

[104] Cayre, F., Fontaine, C., & Furon, T. A Theoretical Study of Watermarking Security. Proceedings International Symposium on Information Theory (ISIT), pp. 1868– 1872, 2005.

[105] Kerckhoff, A. La Cryptographie Militaire. Journal des sciences militaires, (Jan., Feb.), 9:5–38, 9:161–191, 1883.

[106] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 350

[107] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 351

[108] Anderson, R. J. Stretching the Limits of Steganography. In Ross Anderson, editor, Information Hiding: First International Workshop, Springer-Verlag, 174:39–48, 1996.

[109] Craver, S. On Public-key Steganography in the Presence of an Active Warden. In Proceedings 2nd Information Hiding Workshop, 1525:355–368, 1998.

[110] Fridrich, J., Du, R., & Long, M. Steganalysis of LSB Encoding in Color Images. IEEE International Conference on Multimedia and Expo, 3:1279-1282 2000.

[111] Westfeld, A. & Pfitzmann, A. Attacks on Steganographic Systems. In A. Pfitzmann, editor, Information Hiding, 3rd International Workshop, IH'99, Dresden, Germany, September 29–October 1, 1999, Springer-Verlag, New York, 1768: 61–75, 2000.

[112] Schneier, B. *Applied Cryptography*. John Wiley & Sons, 1996.

[113] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, pp. 352-353

[114] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 354

[115] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 355

[116] Cox, I. J., Doerr, G., & Furon, T. Watermarking is not Cryptography. Int. Workshop on Digital Watermarking, 4283:1-15, 2006.

[117] Preuss, R. D., Roukos, S. E., Huggins, A. W. F., Gish, H., Bergamo, M. A., Peterson, P. M., & Derr, A. G. Embedded Signaling. United States Patent, 5,319,735, 1994.

[118] Su, J. K. & Girod, B. On the Imperceptibility and Robustness of Digital Fingerprints. Proc. IEEE International Conference Multimedia Computing and Systems, 2:530–535, 1999.

[119] Eggers, J. J., Su, J. K., & Girod, B. Public Key Watermarking by Eigenvectors of Linear Transforms. In EUSIPCO, Tampere, Finland, 2137:142-156, 2000.

[120] Furon, T. & Duhamel, P. An Asymmetric Public Detection Watermarking Technique. In Proc. of the 3rd International Information Hiding Workshop, Dresden, Germany, 1768:88-100, 1999.

[121] Furon, T. & Duhamel, P. Robustness of an Asymmetric Watermarking Technique. In IEEE International Conference on Image Processing, 3:21–24, 2000.

[122] van Schyndel, R. G., Tirkel, A. Z., & Svalbe, I. D. Key Independent Watermark Detection. In IEEE International Conference on Multimedia Computing and Systems, 1:580–585, 1999.

[123] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 357

[124] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 358

[125] Petitcolas, F. A. P., Anderson, R., & Kuhn, M. G. Information Hiding—a Survey. Proc. IEEE, 87(7):1062–1077, 1999.

[126] Kutter, M. & Petitcolas, F. A. P. A Fair Benchmark for Image Watermarking Systems. Security and Watermarking of Multimedia Contents, Proc. SPIE-3657:226–239, 1999.

[127] Kutter, M., Voloshynovskiy, S., & Herrigel, A. The Watermark Copy Attack. Security and Watermarking of Multimedia Contents II, Proc. SPIE-3971:371–380, 2000.

[128] Holliman, M. & Memon, N. Counterfeiting Attacks on Oblivious Block-wise Independent Invisible Watermarking Schemes. IEEE Trans. on Image Processing, 9(3):432–441, 2000.

[129] Craver, S., Memon, N., Yeo, B.L., & Yeung, M. M. Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, attacks and implications. IEEE Journal of Selected Areas in Communication, 16(4):573–586, 1998.

[130] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 369

[131] Linnartz, J.P.M.G. & van Dijk, M. Analysis of the Sensitivity Attack against Electronic Watermarks in Images. In Workshop on Information Hiding, Portland, OR, 1525:258-272, 1998.

[132] Kalker, T., Linnartz, J.P.M.G., & van Dijk, M. Watermark Estimation through Detector Analysis. In Proceedings. 1998 International Conference on Image Processing, 1:425-429, 1998.

[133] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 373

[134] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 70

[135] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 77

[136] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 234

[137] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 116

[138] Cox, I.J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T.(2007). *Digital Watermarking and Steganography*, Second Edition. Burlington, MA: Morgan Kaufmann Publishers, p. 247