Bachelor Thesis

Hasan Nasir Shaikh

Integration of Fingerprinting Codes for Image Watermarking Application

Fakultät Technik und Informatik
Department Informations- und Elektrotechnik

Faculty of Engineering and Computer Science
Department of Information and Electrical Engineering

Hasan Nasir Shaikh

Integration of Fingerprinting Codes for Image Watermarking Application

Bachelor Thesis based on the examination and study regulations for
the Bachelor of Engineering degree programme
Information Engineering
at the Department of Information and Electrical Engineering
of the Faculty of Engineering and Computer Science
of the University of Applied Sciences Hamburg

Supervising examiner : Prof. Dr.rer.nat Hans-Jürgen Hotop

Second examiner : Prof. Dr.-Ing. Lutz Leutelt

Hasan Nasir Shaikh

Title of the Bachelor Thesis

Integration of Fingerprinting Codes for Image Watermarking Application

Keywords

Copyright Protection, Image Processing, Digital Watermarking, Collusion attacks, Collusion Secure Fingerprinting Codes.

Abstract

Watermarking is an accepted phenomenon to discourage illegal distribution of multimedia. Using a traitor tracing scheme such as Transaction Watermarking one is able to trace a distributed copy back to the responsible.
However, this is no longer true if several customers of the same content collaborate and conduct a so called collusion attack. Collusion attacks aim to manipulate the Transaction Watermark Messages. Common theoretical approaches to counter such attacks are mathematical Fingerprinting Codes which usually require long code length corresponding large embedding payload. Thus, a huge challenge is to apply these schemes for image watermarking. Since images only have limited embedding payload, it is hard to successfully embed a Fingerprinting Code consisting of thousands of bits. The goal of this thesis is to explore which theoretical schemes are appropriate for embedding, and to integrate the most applicable one into a current image watermarking application, maintaining the quality of the image and properties of the watermark.

Hasan Nasir Shaikh

Thema der Bachelorarbeit

Integration von Fingerprinting Codes für Bild Watermarking Applikation

Stichworte

Copyright Protection, Bildverarbeitung, Digitale Watermarking, Absprachen Angriffe, Absprachen Sichere Fingerprinting Codes

Kurzzusammenfassung

Watermarking ist eine anerkannte Methode, um vor einer illegalen Verbreitung von Multimedia Daten abzuschrecken. Mit einem Verräter-Verfolgung System(traitor tracing system)(sounds strange, evtl stay with „traitor tracing“ Schema), z.B. Transaction Watermarking ist man in der Lage eine illegale Kopie zu demjenigen Kunden der diese verbreitete zurück zu verfolgen. Allerdings ist das nicht mehr der Fall, wenn mehrere Kunden zusammenarbeiten und einen sogenannten Koalitionsangriff durchführen. Koalitionsangriffe zielen darauf ab die Transaktion Watermark(Transaction Watermark information or Transaktionswasserzeicheninformation, but consistent) zu manipulieren. Theoretische Ansätze, um diesen Angriffen entgegen zu wirken sind sogenannte mathematische Finger-

printing Codes, welche allerdings für gewöhnlich lange Kodelängen und dementsprechend große Aufnahemekapazität der Multimedia Daten erfordern. Daraus folgt, dass die Anwendung von Fingerprinting Codes eine große Herausforderung für Bildwasserzeichen darstellt. Da Bilder nur eine sehr limitierte Aufnahemekapazität vorweisen, ist es kaum möglich einen Kode der aus tausenden Bit besteht erfolgreich einzubetten.

Das Ziel dieser Arbeit ist es zu untersuchen, welche theorethischen Schemata von Fingerpringing Codes für die Einbettung in der Praxis geeignet sind, um diese dann ohne die Qualität der Bilder zu beeinträchtigen und unter Beibehaltung der Wasserzeicheneigenschaften in die Bildwasserzeichenanwendung zu integrieren.

Contents

List of Figures

Chapter 1

Introduction

In this era of digital world today, where all the information is shaped into multimedia data, one enjoys the ease of access to relevant information at its maximum. Distributors of multimedia data, for example online shops for audio files, images etc use this ease of access to market and sell their products. However, problems arise if their data is illegally distributed over the Internet. This is often the case, as digital data is easy to intercept, copy and re-distribute without any loss in quality. To avoid this misuse, *transaction watermarking* of multimedia data was introduced to discourage illegal distribution of multimedia data.

Transaction watermarking is the process of imperceptibly hiding individual information in digital data. If a distributor encounters with an illegal copy of his property, he is able to detect the invisible message, and hence can trace back to the responsible customer, which is meant to discourage the illegal distribution.

1.1 Motivation

Today, almost all noticeable research organizations in the field of Information technology are working on the security of the multimedia data. Once the data is sold, a distributor has no longer control of what is done with the data. For example it could be shared illegally on the Internet. Hence, the data can be illegally downloaded, shared or distributed without any consequences.

A solution to this problem is to apply digital watermarking for multimedia. Using digital watermarking algorithms one is able to imperceptibly hide information in multimedia. The embedded information is called watermark.

Embedding of the watermark depends on the nature of data. In images, it may correspond to the change of pixel values, whereas in audio it can be the modification in frequencies or the amplitude.

To protect his data, a distributor embeds the watermark on his distributed copies using the embedding algorithm. If any copy is found being illegally distributed over the Internet, the distributor can download it and prove it to be his property by extracting the hidden watermark using the detection algorithm. This is referred to as copyright watermarking.

However, copyright watermarking only leads to the ownership of the distributor. In order to prevent illegal distribution from its customers, or to lead back to the customers involved in illegal distribution, there exist *Transaction Watermarking* techniques. This type of watermarking algorithm embeds individual watermarks, like a serial ID, into each distributed copy. Hence, the distributor uses a detection algorithm on the illegal copy he encountered on the Internet and checks his database to identify the traitor. This scheme is also called traitor tracing, as described in [2] by Noar and Chor.

If the watermark to be embedded is a binary code, i.e. consisting of '0's and '1's, a huge number of copies can be embedded using a small watermark length. For example, with a watermark length of 20, one can embed 2^{20} copies of the data with different watermark combinations. In the context of images, a watermark with a length of 20 bits can be embedded multiple times and therefore is robust against attacks like compression, cropping, rotation or geometric distortions of the images.

However, transaction watermarking is vulnerable against *collusion attacks*. These are attacks for which a group of customers collaborate and by comparing their data find positions where their data differs. As these positions correspond to the individual watermark information, they generate a manipulated copy by for example averaging their data. This manipulated copy thus contains a destroyed or modified watermark message, that can also lead to accuse an innocent customer in the detection process.

To counter these attacks one may apply mathematical codes known as *collusion secure fingerprint codes*. Here, the messages, called fingerprints, provide inter Alia so many undetectable positions, that tracing back is possible even after a collusion attack. These fingerprints are to be created under specific conditions and assumptions that leads to much larger fingerprinting message lengths and thus it is difficult to embed fingerprints into multimedia with limited payload, alike images.

There are many proposed algorithms for these fingerprints, and much work is being done to reduce the fingerprint length ([10],[12],[14],[13],[9]). However, the challenge is that these algorithms produce fingerprints whose code lengths exceed the payload limit to apply them in practice. The important challenge is to provide fingerprints which are highly secure, reliable and, at the same time, provide a sufficiently compact code length according to the typical payload limitations of common watermarking algorithms.

1.2 Goal

This thesis aims at the integration of an existing Fingerprinting Code for the already implemented watermarking algorithm, and evaluating it against different collusion attacks.

A collusion secure fingerprint is a unique mathematically generated code word embedded in each customer's copy. The significance of this message is that it leads the distributor to identify at least one of the responsible customers, in case the data is manipulated under a collusion attack and is being distributed illegally.

A collusion attack is when a group of legal customers of the same content, join hands and make attempt to destroy the fingerprints embedded in their data. These customers are then referred to as *colluders*. They compare their data, and try to manipulate on the positions where they detect differences (also called *detectable positions* in data content). The manipulation attempts to generate a multimedia copy with destroyed or altered watermark. According to Boneh and Shaw [1], manipulating at the detectable positions is their only chance to create a copy of high quality which is not traceable. Additionally, comparing their data is assumed as the only method to find detectable positions, referred to as *Marking Assumption*[1].

However, the code length of these fingerprints is eminent longer than of a comparable transaction watermark message, which makes it impossible to embed them in multimedia data with small payload, such as images or an audio data. For example a fingerprint with code length of about 2000 bits cannot be embedded successfully into an image file of 5000 bits. This is because it is too long to assure all the characteristics of a watermark. Many significant collusion secure fingerprinting algorithms are proposed providing resistance against attacks generated by a group of ten or more colluders, but as the collusion size correlates to the code length, it therefore reaches in thousands. Due to the experience gained in practice, that the majority of the attacks are made by one or two users, there is demand to focus on fingerprinting algorithms resistant against collusion attacks generated by only two colluders.

This work deals with the integration of an already proposed 2-secure fingerprint code [12] into images. The embedding of the fingerprints is done using a watermarking algorithm proposed in [6]. Additionally, this work evaluates the security of the fingerprint against different collusion attacks in the spatial domain as well as in the Fourier domain. The security of the fingerprints embedded is tested against conventional attacks (without the watermark embedding information) as well as advance attacks (attacks using the watermark embedding information). An anlaysis is made on the manipulated fingerprints, and an optimization to the tracing algorithm is proposed so as to catch guilty users.

1.3 Arrangement

- In chapter 2, the watermarking schemes, their applications and the required properties of watermarks are discussed. An emphasis on the need of fingerprinting as well as some existing fingerprinting schemes are described.

- Chapter 3 discusses the 2-secure fingerprint scheme which is used in this work . Additionally, the implemented watermark embedding and detection algorithm is described in detail.

- Chapter 4 covers the conventional collusion attack strategies used for image manipulation in spatial and frequency domain respectively. Attacks generated using the perceptual masks are also discussed, and their performances against the tracing algorithm evaluated.

- Analysis on the reliability of watermark information after collusion attacks is made in Chapter 5. This chapter also proposes an optimized colluder tracing strategy, which is evaluated against the combination of collusion and image processing attacks. This approach provides an improved error rate and a higher tracing accuracy at the crucial attack strategies discussed in the previous chapter.

- Chapter 6 provides the summary of the implemented milestones and thus concludes the thesis work.

Chapter 2

Watermarking and fingerprinting Basics

This Chapter describes the general watermarking scheme, the properties a watermark should possess and the range for application of watermarking. Furthermore, in this chapter the need of fingerprints is discussed, followed by the background of general fingerprinting schemes.

2.1 General watermarking scheme

Digital watermarking is a technique to imperceptibly hide information, also called *watermark message* in multimedia data.

A Watermarking algorithm consists of two stages, the *embedding stage* and the *detection stage*. In the embedding stage, the watermark is integrated into the multimedia data using a secret key. The integration is done by introducing some unnoticeable modifications of the data. Whereas in the detection stage, the embedded message is extracted out of the watermarked copy using the same key as in the embedding stage. This can only be done using the secret key, possessed by the legitimate distributor.

There are two types of watermarking algorithms, namely *blind* and *non-blind*.

In the embedding process of a blind watermarking algorithm (see Figure 2.1) , the multimedia data and the watermark message is fed into the watermark embedder, which by means of a secret key embeds the watermark information on a copy of the original data. The information adds imperceptible modification in original data according to the watermark message. The output of the embedder is a watermarked copy of the original data.

If a distributor wants to prove his ownership on a particular multimedia data, he makes use of the detection process. The distributor inputs the data into the watermark detector, and by means of the same secret key he used in the embedding process, extracts the watermark message.

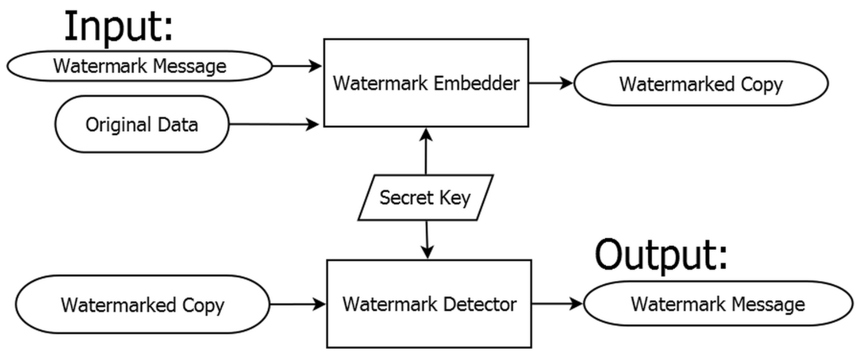


Figure 2.1: Blind watermarking algorithm scheme

The non-blind watermarking algorithm scheme, unlike the blind watermarking algorithm scheme, needs the original data in the detection process, as shown in Figure 2.2. Therefore this scheme is also referred to as *informed watermarking algorithm.* This work focuses on blind algorithms as non-blind algorithms are unmanageable and therefore rarely used.

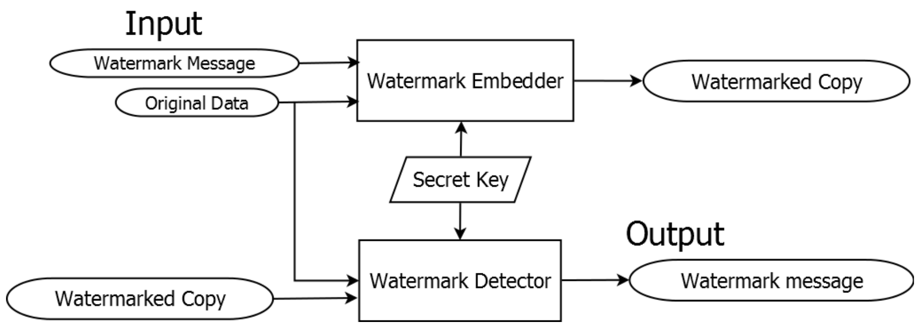


Figure 2.2: Non-blind watermarking algorithm scheme

2.1.1 Watermark properties

To be considered as a good watermark, each type of watermark must meet some requirements as agreed upon by Cox et al. in [3] and Dittmann in [4]. The following being the most important:

- **Robustness:** A watermark message is called robust, if even after any modification on the watermarked data, it can be reliably detected. The modifications being for example compression in images, amplification of audio data, format conversions like mp3, JPEG etc.

- **Imperceptibility:** A watermark is called imperceptible if one is not able to hear or see the difference between the original and the watermarked content. A watermark message hence shall not generate an audible or visual difference that can be noticed by hearing or vision.

- **Security:** A watermark message is called secure, if the watermark message cannot be destroyed or detected, in case the data has undergone some attacks. Even if the watermarking algorithm is known to the attackers and they possess at least one watermarked copy, the watermark message must not be detected as the attackers do not know the secret key. In other words, the security of the watermark message must only depend on secret key possessed by the distributor alike typical cryptographic models.

- **Capacity:** It should be possible to embed a watermark message multiple times in one multimedia file. How much information can be embedded into the original data is referred to as capacity. It is recommended to embed a message several times to increase the reliability of the detection process.

2.1.2 Watermark Applications

Watermarking algorithms are being used worldwide for the security of digital data. Below are listed the most common types of watermarks, they distinguish according to their different applications.

- **Copy control:** Using copy control or broadcast watermarks, one is able to allow for example watching or listening and/or copying of the data or not.

- **Integrity control:** This type of watermarks are used to check if the data is original or has been manipulated.It also checks if there is any additional information embedded. One specific type of these watermarks are content-fragile watermarks, which are very sensitive to changes. This is of interest for distributors to verify that data has not been edited, damaged, or altered since it was marked.

- **Authentication:** Authentication or copyright watermarks are embedded by the distributors to prove the ownership or copyright of the data.

- **Transaction watermarking:** If the data is sold to various customers, and its security has to be maintained, the distributor embeds an individual watermark message in each of the customers' copy. The message can be a serial number or any sort of identification for the customer and is therefore different for every customer. This type of watermarks are called transaction watermarks. With these watermarks, a distributor is able to trace back an illegal copy and thus identify the responsible customer.

2.2 Transaction Watermarking

The focus of this work is on collusion secure fingerprints applied as transaction watermarks, more specifically a collusion secure transaction watermarking on images.

Transaction watermarks are individual watermarks messages embedded for example in multimedia data like images, video and audio files, which cannot be detected by an average human. These watermarks are used to catch customers involved in unauthorized data distribution. Hence every copy of distributed data contains an individual watermark message, which is used to identify the customers.

An example of a transaction watermark can be a binary message representing the customers' number or ID as shown in Figure 2.3.

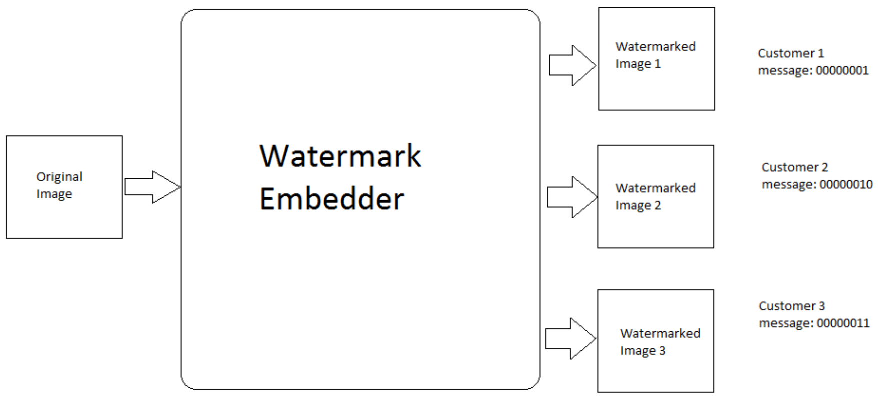


Figure 2.3: The illustration shows the embedding process for three copies of the same content

As seen in figure 2.3, every user gets an individual 8-bit watermark embedded in his multimedia copy, which represents the identity of the customer in distributor's database. In case an illegal distribution of data is encountered, the distributor can trace back to the corresponding customer by the message hidden in this data. Therefore the presence of a watermark and a high probability of reliably tracing back to the responsible customer is for example made public in order to refrain customers to be involved in illegal distribution.

However, their might occur some errors in transaction tracking as well. in case there is no watermark output by the detector, this means the distributor cannot accuse anyone. The corresponding error is called False Negative. The event when the detection algorithm accuses an innocent is denoted as False Positive. The ε_2 rate is the rate the applicant of the fingerprinting codes choses in order for the probability of a False Negative error to be lower than this rate ε_2. Similarly, ε_1 describes the rate the applicant choses during the genereration process in order to keep the probability of a False Positive error lower than this rate. In practice, ε_1 must be very small, otherwise the whole scheme becomes useless.

2.3 Collusion Attacks

A transaction watermark is robust against most of the types of data manipulation by a customer for example, enhancement or compression of a watermarked image, or amplification or filtering of audio data. However, this condition does not hold true in case of a *collusion attack*.

Already two customers of the same individually marked content can collaborate in order to find out positions of the watermark by comparing their copies and looking for the differences. These differences are assumed to be positions of the individual watermarks. Generating a third copy of the content composed from both watermarked copies and only manipulating at the different positions, they can generate a high quality copy containing a destroyed or altered watermark, that with a high probability no longer allows tracing back to the responsible customers. This attack is called **collusion attack**. In case the distributor runs the detection process with an illegal data file generated under a collusion attack, the result can be an unsuccessful attempt to detect the watermark message or even worse, the detected watermark resembles an innocent customer's watermark.

For example let customer 1 and customer 8 be the attackers, also termed as *colluders*. They collaborate to generate a modified copy in such way that the generated copy contains watermark message allocated to customer 9. This example is illustrated in figure 2.4.

In practice, the colluders do not have any information of the embedded message, they just have their data files containing the watermark messages. After comparing their data for example images, they can find differences in pixel intensity values at various positions. At these positions they start the attack, may be modifying the pixel values under some strategy, to generate an illegal image. The watermark message in the manipulated copy, might be destroyed or altered in a way that it could not lead back to the colluders.

According to Boneh and Shaw in [1], creating transaction watermarks as serial numbers do not fulfill the security requirement of a watermark as they fail against the special attacks as collusion attacks. So if the data has

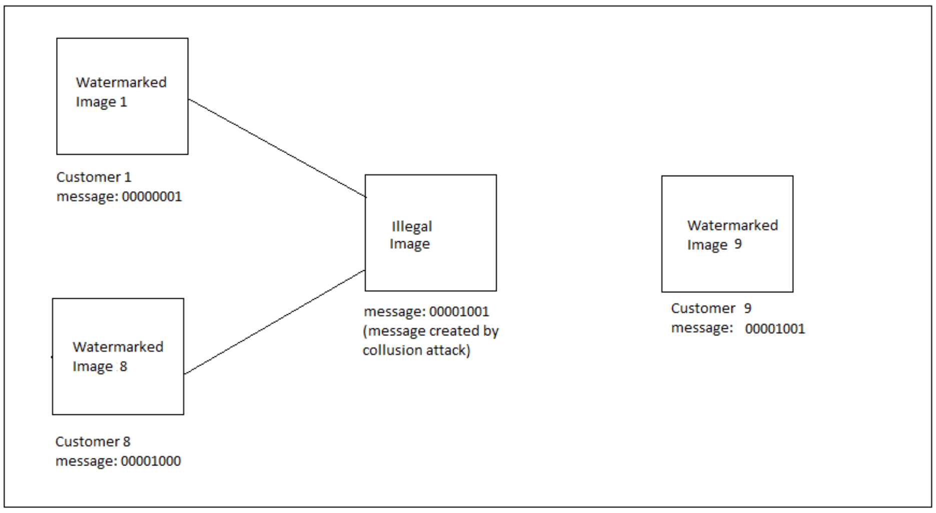


Figure 2.4: Example of a collusion attack

to be protected against attacks generated by a group working together, a distributor must use *collusion secure fingerprint watermarks.*

2.4 Collusion secure fingerprinting

A collusion secure fingerprinting code is a set of collusion secure fingerprints, where each fingerprint is a mathematical code (in this case binary). These fingerprints after undergoing a collusion attack, still carry enough information in the altered message y to trace back to at least one of the colluders. A transaction watermark can be a sequential or a randomly generated watermark message, with an additional block of 12-16 bits which are needed for the cyclic redundancy check (CRC). On the contrary, a fingerprint message is probabilistically generated under some probability density functions, without the CRC or the ECC(error correction code) bits attached. Instead, there are tracing algorithm applying probabilistic techniques for the tracing of fingerprints. However, fingerprints are significantly longer than transaction watermarks. Hence the larger the fingerprint code length for a fixed collusion size c, the higher is the probability to catch the guilty users.

In other words, the larger the number of colluders (guilty users), the longer is the fingerprint to carry sufficient information to accuse at least one colluder.

However, longer fingerprinting codes are not practically useful for multimedia data offering smaller payload. A payload can be defined as the capacity of multimedia data to allow imperceptible modifications during the embedding process. The payload for an image or an audio song file is limited and does not allow to embed thousands of bits of fingerprint. Thus, reducing the code length is the aim of most recent research in fingerprinting.

2.4.1 Existing Fingerprinting Schemes

There are many fingerprinting algorithms, designed to be resistant against large as well as small collusion sizes. Although in this work we focus on fingerprints with smaller code lengths, it is worth mentioning the probabilistic schemes to be resistant against large collusion channels.

For example, Boneh and Shaw [1] presented a binary scheme with the lower bound on the code length m $\leq O(c_o^4 \log |n/\varepsilon_1| * \log |\varepsilon_1^{-1}|)$, where ε_1 is called the False Positive rate i.e. the rate chosen by the applicant to keep the probability of a False Positive error lower than this rate. They also introduced the *Marking Assumption*, stating that the colluders only manipulate the fingerprint at positions where they detect a change when comparing their data. Most of the work on fingerprints is based on this Marking Assumption.

The Tardos code

The Tardos fingerprinting scheme proposed in [14] is based on a probabilistic generation of the fingerprints.

In [14], Tardos introduced a probabilistic fingerprinting generation scheme, generating a fingerprinting matrix, with each row representing one fingerprint. The entries of the matrix are based on probability vector p_i generated from Tardos' probability density function f, which is biased towards the values close to 0 and 1.

In case a distributer finds an unauthorized copy, he extracts the fingerprint message. The detected fingerprint, y, is compared to all the distributed fingerprints. If none of them matches y, the distributor assumes a collusion attack and starts the accusation algorithm. Therefore, the accusation calculation is made according to a pairwise score between each distributed fingerprint and the manipulated one.

The Improved Tardos code

Significant reduction in the code length is proposed by Skoric et al. in [13], where he states that the '0' bits in the fingerprint are equally informative as '1' bits. Tardos only considered bits with value '1' as informative. This improves the code already by a factor of 2 for the code length. Later in his paper he decoupled the false positive rate, ε_1 from the false negative rate ε_2, which further reduced the code length by a factor of 5, however still much higher to be embedded in most of the multimedia application.

Chapter 3

2-secure image Watermarking Algorithm

The goal of this thesis work is to successfully embed Fingerprints as watermarks into images. However, since image files provide very limited payload, it is not possible to embed fingerprints resistant against attacks generated by large group of colluders. However, as most of the attacks encountered are made by two colluders or less, it is preferable to apply fingerprints resistant against collusion attacks generated by two colluders, that therefore come up with a smaller code lengths than not providing any collusion resistance at all. In this thesis work, a 2-secure fingerprinting code proposed by Schfer et al. in [12] is integrated into the watermarking algorithm proposed by *Liu et al.* in [6], and tested against varying collusion attack strategies.

3.1 2-secure fingerprints

A fingerprinting code resistant properly against attacks generated by at most c pirates is called c-secure fingerprinting code. Under the conventional Marking Assumption [1] several c-secure including 2-secure codes have been proposed. In [10] *Nuida et al.* proposes fingerprinting codes resistant against a collusion of two providing outstandingly short code lengths. However, the computational effort is very demanding such that employing in practice might not appeal in real world scenarios. Also, the proposed code maintains to be 2-secure within a (though very low) probability for a false positive error. However, to hold before court this probability is demanded to be zero. Therefore, in [12] Schaefer et al. proposed a collusion-secure code against two colluders based on special conditions on the Hamming Distances between the users fingerprints. Using preconditions for the fingerprint generation and also in the tracing process, the code proposed in [12] needs much less computation effort.

In theory, the false positive error probability is proven to be zero. This makes it secure enough to be used for example in commercial data applications. Along with the zero-false positive error probability, the code length for the fingerprint message is much shorter compared to the Tardos code, being less than 100 bits for approximately a thousand users. For the watermarking algorithm being used, this fingerprint length seems reasonable to embed it in images.

3.1.1 Fingerprint Generation

To get a stronger dependency between each fingerprint, some specific preconditions are met for the fingerprint generation. With these preconditions additional information to the algorithm is included that later supports the accusation scheme in the process of finding the colluders. Therefore, the accusation scheme exposes at least one attacker.

A fingerprint matrix X of size $n \times m$ is to be generated, where n is the number of users and m denotes the fingerprint length.

Each fingerprint is to be created in such a way that its Hamming Distance to every other must fall in a pre-defined interval $[HD_{min}, HD_{max}]$, where $HD_{min} > \frac{1}{2}HD_{max}$. Here, HD_{min} and HD_{max} denote the minimum and maximum Hamming Distance of a fingerprint to every other. In [12], the author propose to create the matrix X by first creating an n' x m matrix X', where n' >> n (empirical experiments yield n' $\approx$ 10n), using a coin-flipping algorithm. To get X, all rows of X' are simply striked out, which do not meet the Hamming Distance condition introduced above.

After the generation, the fingerprints are embedded in the images for different users using an appropriate watermarking algorithm.

3.1.2 Tracing Algorithm

If an unauthorized copy of an image is found, the distributor extracts the fingerprint and compares with all the fingerprints allotted to the users. This manipulated fingerprint is denoted as y. If y does not match with any of the user fingerprints, the distributor assumes a collusion attack. Therefore, to find the colluders, firstly the Hamming Distances between the detected message y and the user fingerprint U_i are calculated. All fingerprints whose Hamming Distance is greater than $\frac{3}{4}HD_{max}$ are considered to belong to innocent users and are striked out. The rest of the fingerprints are distributed into two subsets, one contains the fingerprints with Hamming Distance $< \frac{1}{2}HD_{max}$ to y and the other subset contains the remaining fingerprints with Hamming distances between $\frac{1}{2}HD_{max}$ and $\frac{3}{4}HD_{max}$ to y.

The fingerprints in the first subset are sent to a parent pair search algorithm similar to the one proposed by *Nuida et al.* in [11]. All possible combinations of fingerprint pairs of this subset are generated for the parent pair search. Hence the algorithm analyses for every pair $(U_i; U_j)$ with i , j $\in$ 1,....,n and i $\neq$ j, at every bit position if it is possible that the corresponding two users worked together and could have created the attacked fingerprint. For example, if on any position i in y, there is a bit value $y_i = 1$. Then at least in one of the fingerprints of a pair $(User_A, User_B)$,must have a bit value '1' at position i. If it is not the case, the pair is considered not to be able to construct the message y. The algorithm returns YES if every bit could be created in collaboration of both, otherwise NO. This is denoted as step 1 of the Parent pair search algorithm. The goup of fingerprints which are suspected to be a part of the collusion after step 1 are denoted as U_{s_1}

This is done, because one of the colluders' fingerprint has at least half of its Hamming Distance or more to y compared to the Hamming Distance between the colluders, HD_c.

With the fingerprints of the second subset is proceeded as follows. The fingerprints of U_{s_1} are paired individually with the fingerprints of the second subset. The hamming Distance between the two colluders' fingerprints is denoted as HD_c. As one of the colluder fingerprint has a hamming distance greater than $\frac{1}{2}HD_c$ to y, at least one colluder is always in the process during the first step of the algorithm.

The generated pairs (as mentioned in the paragraph above) are again analyzed in the Parent pair search algorithm. If the algorithm outputs only one suspected pair, it is considered to be the colluder pair. If the algorithm outputs more than one pair, the fingerprint that is present in all the suspected pairs is considered to be one participant of collusion. If there is no common fingerprint in all the suspected pairs, no fingerprint is set guilty of being a member of the collusion and thus, no user gets accused. This thus means a False Negative error occured.

Hence a user can therefore rely on the use of this tracing algorithm as it has a zero-false positive error rate and a very low false negative error. These 2-secure fingerprints are embedded into the images using the watermarking algorithm described in the next section, and their performance is evaluated and discussed in the forthcoming chapters.

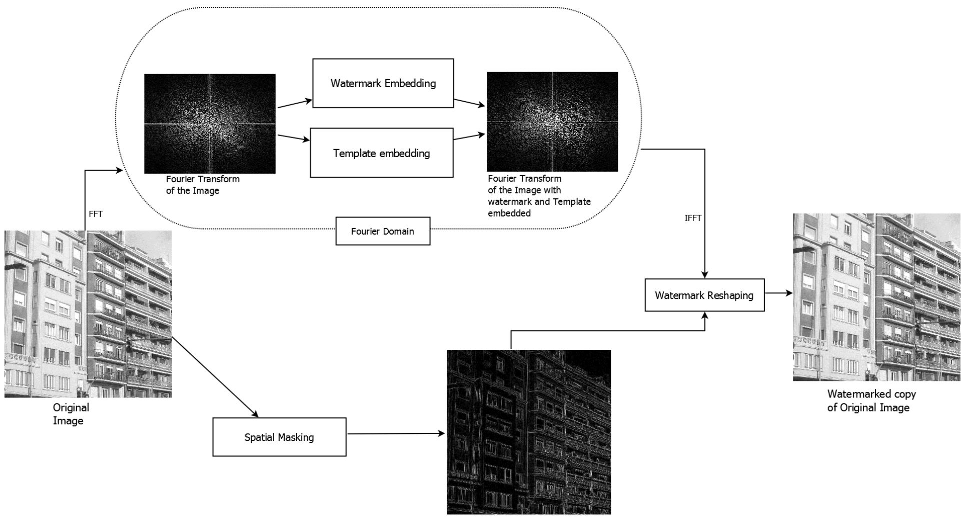
3.2 Watermarking Algorithm

The watermarking algorithm proposed for image watermarking in [6] follows the patchwork approach providing a reasonable watermark payload which meets the requirement of most applications. This watermarking algorithm combines template embedding and patchwork watermarking in Fourier domain. Furthermore, the watermark is reshaped using the spatial perceptual mask after it is re-inverted to the spatial domain. These watermarks are robust against distortions like JPEG compression, cropping, scaling and rotation which are very common processing for digital images.

Geometric distortion is more challenging for watermark robustness. A geometric distortion can be defined as apparent change of shape of objects for example a round object may appear elliptical, or the whole image is shifted by 5 degrees. This distortion can fail the watermark detection by desynchronizing the positions where the watermark was embedded.

To counter this problem, a predefined template signal is embedded to invert the distortions caused by geometric distortions. A template is usually embedded in frequency domain. However, since Fourier transform does not provide a localized frequency analysis, it is difficult in Fourier domain to apply a spatial-localized adaptive embedding. When taking the inverse transform, the template or the energy embedded in the frequency domain is distributed over the whole image, without considering the perceptual model. This can lead to visible artifacts in images, causing image quality degradation.

One solution to counter this problem was to minimize the embedding strength as proposed in [7]. However tests show that even with limited embedding strength the embedded template in Fourier domain can still cause visible artifacts on smooth areas and hence can easily become undetectable after combined distortions. Therefore, to achieve a better quality, it is necessary to combine the robustness achieved by watermarking in Fourier domain with the perceptual masking of the embedded energy in the spatial domain.



Watermark Embedding
Template embedding
Fourier Transform of the Image
Fourier Transform of the Image with watermark and Template embedded
Fourier Domain
FFT
IFFT
Watermark Reshaping
Original Image
Spatial Masking
Watermarked copy of Original Image

Figure 3.1: Illustration of watermarking scheme as described in [6].

3.2.1 Watermark Embedding

Figure 3.1 illustrates the watermarking scheme proposed in [6]. The embedding of the watermark and template is done in the frequency domain, whereas the reshaping of the watermark using a perceptual model is done in the spatial domain. The template embedded is used for re-synchronization of the watermark if the image has undergone geometric distortions like scaling, rotation etc.

In this watermarking algorithm proposed, an informative watermark message is embedded in the magnitude spectrum of the Fourier domain using a patchwork approach. First, the Fourier transform is applied on the original image. If the image is of high resolution i.e. greater than a predefined block size B, it is first divided into blocks of size B $\times$ B before the embedding process starts. Due to the symmetry constraint of the Fourier transform, only the two quadrants in the upper half of the magnitude spectrum are used for watermarking.

For the embedding of a watermark, a middle frequency band is selected. This is because a modification in the lower frequency band will produce noticeable artifacts and coefficients in the high frequency band are not robust to compression. Hence, the frequency band within the interval F_L and F_H are selected for embedding, where F_L and F_H are the lowest and highest frequencies for the middle frequency band.

Denoting the magnitudes of the coefficients in the middle frequency band as F=$\{f_0, f_1, ...f_N\}$, for each watermark bit b_i, a group of coefficients G^i in F are randomly selected

$$G^i = \{f^i_0, f^i_1, ...f^i_n\}, i = (0, 1, 2, ..., L) \tag{3.1}$$

where L is the length of the watermark, n is the number of coefficients in the group. A secret key ensures the security of embedding process by grouping the coefficients randomly.

Now the group G^i is divided into two parts, G_1 and G_2, with E_1 and E_2 being the sum of the coefficients.

$$E_1 = \sum_{i=0}^{n/2} G^i_1 \tag{3.2}$$

$$E_2 = \sum_{i=0}^{n/2} G^i_2 \tag{3.3}$$

If the watermark bit to be embedded i.e. $b_i = 0$, the coefficients in G_1 and G_2 are modified so that the condition of $E_1 < E_2$ is satisfied. If b_i=1, the coefficients are modified so that $E_1 > E_2$.

Modification on every bit coefficient is based on its weighted average share and frequency. For example, if b_i=1,

$$f^i_k = \begin{cases} f^i_k + \alpha^i_k.\Delta, f^i_k \in G_1 \\ f^i_k - \alpha^i_k.\Delta, f^i_k \in G_2 \end{cases} \tag{3.4}$$

where

$$\alpha^i_k = \frac{f^i_k}{\sum\limits_{f \in G^i} f^i_k} \tag{3.5}$$

and

$$\Delta = \begin{cases} D_i + \beta\varepsilon_{f_k^i} . \sum\limits_{f \in G^i} f_k^i, & \text{if } D_i + \beta\varepsilon_{f_k^i} . \sum\limits_{f \in G^i} f_k^i \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

where Δ is the total amount of modifications on all of the coefficients in G_i and α is the weighted average factor for each coefficient. The parameter ε is an adaptive factor based on the corresponding frequency of the coefficient. Thus, the modification of every coefficient is adaptively determined by its magnitude and frequency. The parameter β is a global controlling factor which determines the embedding strength. D_i is the difference between E_1 and E_2 such as

$$D_i = \begin{cases} E_2 - E_1, & \text{if } b_i = 1 \\ E_1 - E_2, & \text{if } b_i = 0 \end{cases} \tag{3.7}$$

As discussed earlier, the embedding of the watermark is done in the upper two quadrants of the frequency domain. After the embedding, the lower two quadrants are also to be modified accordingly so as to fulfill the symmetry constraint of the Fourier transform.

After this embedding process, the inverse Fourier transform is performed to get the watermarked image in spatial domain. However, the watermark has to be masked using a perceptual model to generate the final version of the watermarked image.

3.2.2 Template Embedding

Along with the watermark, a template is also embedded into the images in order to re-synchronize the watermark from geometric distortions. The template consists of two lines, one lying in the first quadrant, and the other in the second quadrant of the magnitude spectrum. Both lines consist of seven points uniformly distributed upon them. It is empirically found that seven points per line can satisfy the low false positive probability during detection [7]. The angles of the two lines θ_1 and θ_2 and the radii of the points vary between F_L and F_H. The angels and radii can be randomly chosen, controlled by a secret key. The template embedding is also done in the middle frequency band, as modification in the low frequency band could cause visible artifacts on the image, whereas high frequency components are not robust against compression.

The embedding strength is determined by local mean and standard deviation of Fourier coefficients. In [7], the embedding strength is proposed to be local mean value plus five times standard deviation, however this embedding strength is strong enough to cause visible artifacts. Embedding with lesser strength, the template can easily become undetectable after compression. However, using the perceptual masking a template with strength of local mean value plus twelve times standard deviation is embedded without causing any visible artifacts. Also, with strong embedding strength, the reliability of the watermark detection is increased.

3.2.3 Watermark reshaping in Spatial domain

Masking Calculation

As watermark and template embedding is done in the Fourier domain, it does not consider perceptual aspects like edges, luminance and the texture of an image. Fourier transform is a global transform, and modification in Fourier domain does not correspond to any localized modification on a particular pixel in the spatial domain. Therefore, after embedding when the Image is re-transformed in spatial domain, the energy is distributed over the entire image without considering the perceptual behavior of the image. This can result in visible artifacts in the smooth regions. Hence, a perceptual mask is calculated in the spatial domain as proposed in [6] so as to reshape the embedded energy before it is imposed on the original pixels.

Three properties of the human visual system (HVS) are taken into account in *Liu et al.* watermarking algorithm ([6]) while calculating the perceptual mask namely luminance, edge approximation and texture.

- The human eye is less sensitive to noise in the bright and dark regions,

- The human eye is less sensitive to noise around edges,

- The human eye is less sensitive to noise in highly textured regions.

Hence, in an image I, for each pixel $I(i,j)$ these properties are calculated.

Let C be a 2×2 block around pixel $I(i,j)$. The normalized average luminance value for this pixel is

$$L_m(i,j) = \frac{1}{256 \times 4} \sum_{(i,j) \in C} I(i,j). \tag{3.8}$$

$$L_w(i,j) = \begin{cases} L_m(i,j), & \text{if } L_m(i,j) > 0.5 \\ 1 - L_m(i,j), & otherwise \end{cases} \tag{3.9}$$

The luminance mask is obtained as

$$L(i,j) = 1 + L_w(i,j) \tag{3.10}$$

The edge proximity is represented by the average gradient energy of pixels in the neighborhood. Edges in the diagonal direction yield higher mask values than those in horizontal and vertical directions, because human eyes are less sensitive to detect the variance in diagonal directions.

$$E(i,j) = \frac{1}{12} \sum_{(i,j),(m,n) \in C} [I(i,j) - I(m,n)]^2 . \begin{cases} \sqrt{2}, & \text{if } |i-m| = |j-n| = 1 \\ 1, & otherwise \end{cases} \tag{3.11}$$

Texture masking is defined as the local variance in C

$$T(i,j) = Var\{I(i,j)\}_{(i,j) \in C} \tag{3.12}$$

Hence, the spatial mask is calculated using the three components as

$$M(i,j) = L(i,j).[E(i,j).T(i,j)]^{\mu} \tag{3.13}$$

where μ is the weighting factor. The normalized value $M(i,j)$ represents the intensity of the modification allowed at pixel $I(i,j)$.

Watermark reshaping

The watermark embedded in the frequency domain is transformed in spatial domain, and is reshaped using the spatial mask M before adding to the original image to generate a watermarked copy.

$$I'(i,j) = I(i,j) + \eta . M_n(i,j).[W(i,j) + T(i,j)] \tag{3.14}$$

where M_n is the normalized masking matrix, W and T are the watermark and template respectively in the spatial domain, I is the original image, and I' is the watermarked image. The parameter η is a scaling factor ensuring that the total embedded energy remains unchanged during watermark reshaping.

The Comparison of the watermarked images generated with and without spatial masking is made by generating their difference images. The resulting images are illustrated in Figure 3.2. The energy embedded without the spatial mask is distributed over the whole image, whereas with spatial masking, energy is allocated to areas with high texture activity, near the edges, light and dark regions.

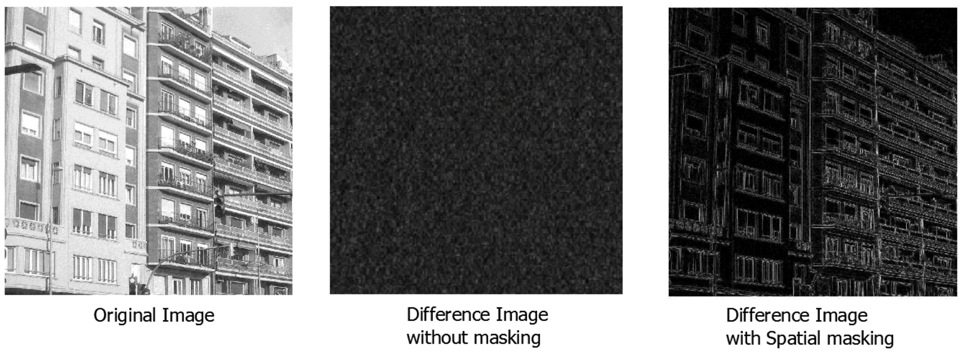

Original Image
Difference Image without masking
Difference Image with Spatial masking

Figure 3.2: Comparison of difference images with and without spatial masking.

Optimization

After the watermark and the template are re-inverted in the spatial domain, the energy to be embedded is to be distributed over the whole image, with each pixel getting its share. In a gray-scale image, the pixel intensity is represented by integer values ranging from 0-255. As many of the pixels may get energy share probably lower than 1, rounding the pixel value after energy addition might lose or enhance the energy added. If the pixel value is floored, the robustness of the watermark is affected, whereas increasing the pixel value might degrade the image quality. Therefore to balance between robustness and quality, the modification is tuned by adaptively rounding the pixel value.

$$I'(i,j) = \begin{cases} I(i,j)+1, & \text{if } 0 < \delta < 1, M_n(i,j) > 0.01 \\ I(i,j)-1, & \text{if } -1 < \delta < 0, M_n(i,j) > 0.01 \end{cases} \tag{3.15}$$

where $\delta = \eta.M_n(i,j).[W(i,j) + T(i,j)]$

3.2.4 Watermark Detection

The detection process tries to detect the watermark bit-by-bit using the secret key. If the bit detection is not reliable i.e. the detection response λ does not meet the required threshold τ, the template detection is performed. According to the template the image is reformed, removing the geometric distortions. After the image reformation the watermark is detected again on the adjusted image.

In the detection process, the image is transformed into the frequency domain and all the coefficient groups are regenerated using the secret key. In every group, E_1 and E_2 are recalculated to extract the watermark bit using

$$\lambda_i = \log \frac{E_{1_i}}{E_{2_i}} \tag{3.16}$$

where λ_i is called the detector response of the bit b_i. The watermark bit b_i is estimated as

$$b_i = \begin{cases} 1, & \text{if } \lambda_i > \tau \\ 0, & \text{if } \lambda_i < -\tau \end{cases} \tag{3.17}$$

where τ is the predefined detection threshold. If λ falls between -τ and τ, the detector response is not considered reliable and the watermark bit can not be estimated.

Chapter 4

Collusion Attacks on Watermarked Images

A group of malicious users, denoted as colluders, can collaborate and compare their legally purchased individually watermarked data of the same content (images in this context) to create a manipulated version. This is called a collusion attack (see section 2.3). In this section, we implement some of the collusion attack strategies colluders can implement in order to attack the watermark. The embedded fingerprints are 2-secure (resistant against attacks implemented by two colluders), therefore the attacks implemented in the spacial as well as in the frequency domain are designed to be applied by two colluders. The implemented attack strategies were designed to ensure the quality of the resulting images, for example the PSNR value of the image is always greater than 40 dB. The attacks we implemented are on gray scale images as well as RGB (colored) images.

In Figure 4.1, a well known image often referred to as 'Lenna' image [5], is used to demonstrate the original image, two generated watermarked copies and a manipulated image created by a collusion attack strategy (i.e. using the image information of watermarked image 1 and 2). As discussed in section 3.1, the watermarking algorithm reshapes the watermark information to avoid visible artifacts on the image. As can be seen in figure 4.1, the watermarked copies of the image do not have artifacts. Also the manipulated version of the image, created by one of the attack strategies discussed later (section4.1), does not display any visible artifacts.

Although some of the implemented attack strategies were used earlier to

Watermarked Image 1
Watermarked Image 2
Original image without watermarking
Manipulated image created by colluders

Figure 4.1: Original, watermarked and manipulated version of a Lena image

test the security of different fingerprinting techniques, they were designed under the assumption that colluders had the knowledge of positions where their fingerprint bits were distinct. Those attacks also assumed that the value of bits at those position were known to the colluders. Hence, the attacks were designed with the assumption of knowing the binary values at different positions of the fingerprint. Moreover, these attacks were implemented directly on the fingerprints, i.e. the digital data embedded with these fingerprints were not used to test the security of fingerprints embedded in them.

However in the real world attacks where the fingerprints are embedded into the images, these strategies are not realizable. This is due to the fact that the watermark is embedded in the frequency domain by *altering* the randomly selected frequency coefficients. As a result, a direct attack on the fingerprint is not possible. Therefore the attacks designed and implemented here are by using the pixel intensity values in the spatial domain and the magnitudes of frequency coefficients in the frequency domain. The forthcoming sections of this chapter outline some information on realization of images in the Spatial and frequency domains, and the methods implemented to use this information in order to design the collusion attacks based on different strategies.

4.1 Attacks in Spatial Domain

A digital gray scale image can be defined as a two-dimensional function, $f(i,j)$, where i and j are the spatial coordinates, and the amplitude of f at any position (i,j) is called the intensity of the image at that point. The intensity here represents the energy possessed by a pixel. A pixel is the smallest unit of an image that can be represented or controlled [5].

In the spatial domain, the image is represented by the intensity values of its pixels. It should be noted that the attackers do not have any prior information regarding the watermark. They follow the Marking Assumption (see section 2.4.1), to identify the positions where watermarks could be present. In order to find these positions, the colluders compare their intensity values pixel by pixel, and note the positions where the intensity value differ. These positions can be termed as *detectable positions* and are considered to be watermarked with different information. For example, on a particular position, attacker 1's image has intensity value of 201, whereas attacker 2's image has the intensity value of 204, it can be assumed that this difference is due to different information embedded. In this thesis, the embedded information is depicted as a binary code, hence one copy can be assumed to be embedded with '0' and the other copy embedded with '1'. However, a single pixel does not represent one bit of watermark information. Hence, to achieve maximum manipulation, colluders try to manipulate on all detectable positions.

Note that as the fingerprints embedded are collusion resistant against attacks generated by two colluders, the attack strategies implemented also focus on attacks generated by two watermarked copies. Some of the collusion attack strategies that are used to manipulate the watermark are as follows:

- **Maximum value Attack:** Two images of the same content but with distinct watermarks embedded are used to generate the attack. The intensity values of both images are compared pixel by pixel. That is the intensity value from the first pixel of image 1 is compare to that of first pixel of image 2, from second pixel of image 1 to that of image 2, and so on. During this comparison, if a difference of values is found, the particular pixel position is assumed to contain watermark information. To manipulate the information, the higher intensity value of the two pixel values is chosen to be placed in the manipulated image.

- **Minimum value Attack:** Another type of systematic collusion attack on the watermarked images, is the so called minimum value attack. Contrary to the maximum value attack where higher intensity value is chosen, here the lower of the two intensity values is taken to be placed in the manipulated image.

  Figure 4.2 shows how the intensity values of the manipulated image are set, after undergoing maximum and minimum value attacks. For simulation purpose, only fifteen intensity values are taken into consideration.

- **Block Segmentation Attack:** Here, the watermarked image of both the colluders are split into blocks of a predefined size, for example a block of size 2 × 2 pixels. A manipulated image copy is generated by alternatively taking blocks from both watermarked copies. Figure 4.3 shows an example of a Block segmentation attack. Blocks highlighted with green are taken from image 1, while highlighted with blue are taken from image 2.

- **Random segmentation Attack:** In this attack strategy, unlike block segmentation attack where a block size is predefined, a segment size is randomly chosen over a predefined length. For example, with a predefined size of 10 pixels, segment of size lesser than 10 × 10 is randomly chosen. Then from any of the two watermarked copies also chosen randomly, this segment is taken to be placed in a manipulated

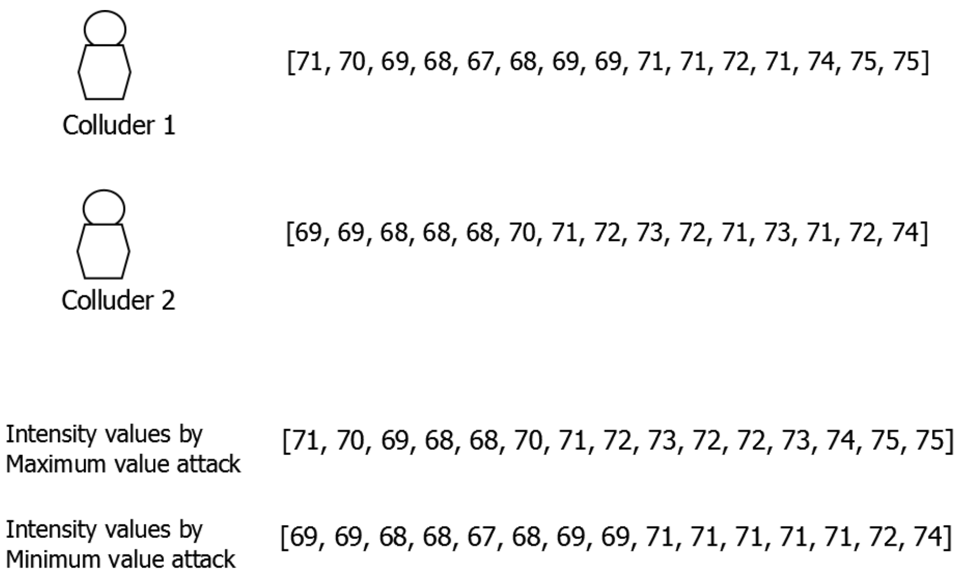


Figure 4.2: Example of a maximum and minimum value attack

version. An illustration of a Random segmentation attack is shown in figure 4.4.

- **Average value Attack:** Here, the colluders calculate bitwise the average of their intensity values. This way colluders try to develop a high quality image that is not tracable.
- **Random value Attack:** As the name suggests, the intensity values from both the images are randomly selected pixel by pixel to create a manipulated image.

These are only some neat examples that can be done easily by 2 colluders.

4.2 Attacks in Frequency Domain

As the embedding of the watermark is done in the frequency domain, it is very important to analyze the behavior of the watermark against attacks generated in frequency domain. It is therefore necessary to understand what happens when an image is transformed into its frequency domain.

The Fast Fourier Transform (FFT) of an image is the sampled Fourier Transform and therefore does not contain all frequencies forming an image, but only a set of samples which is large enough to fully describe the spatial

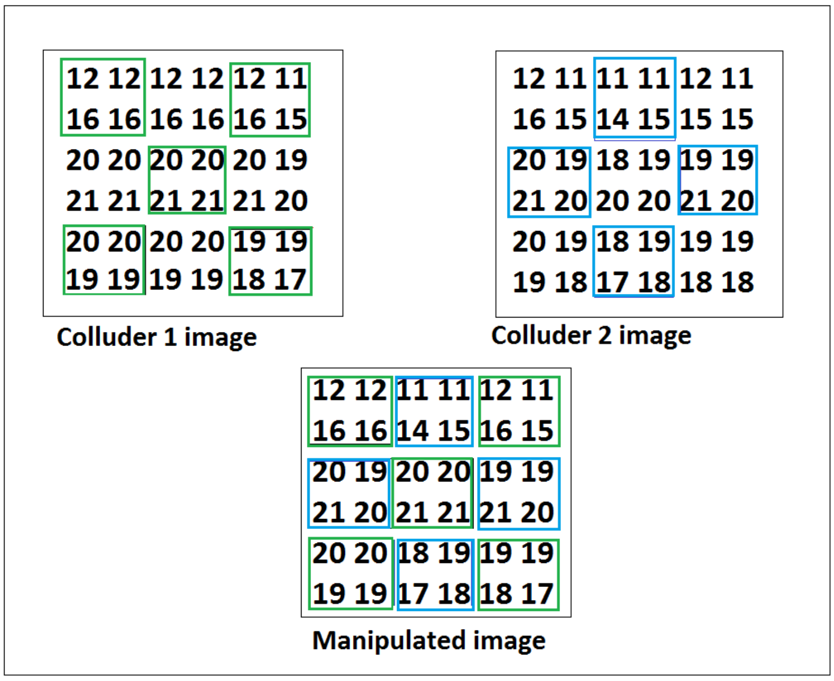


Figure 4.3: Block Segmentation attack by two colluders

domain image. The number of frequencies corresponds to the number of pixels in the spatial domain image, i.e. the image in the spatial and Fourier domain are of the same size [5]. The Fourier transform of an image $f(x, y)$ of size $M \times N$ is given by

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)} \tag{4.1}$$

for u= 0,1,..,M-1 and v = 0,1,2.....,N-1.

The frequency domain is the coordinate system spanned by $F(u, v)$ with u and v being frequency variables. The Spatial domain coordinates are x and y. The frequency rectangular region of the image of size $M \times N$ is of the same size as in the spatial domain.

The inverse Fourier Transform [5] of the image is given by

$$F(x, y) = \frac{1}{M \times N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} f(u, v) e^{j2\pi(ux/M + vy/N)} \tag{4.2}$$

for x= 0,1,..,M-1 and y = 0,1,2.....,N-1.

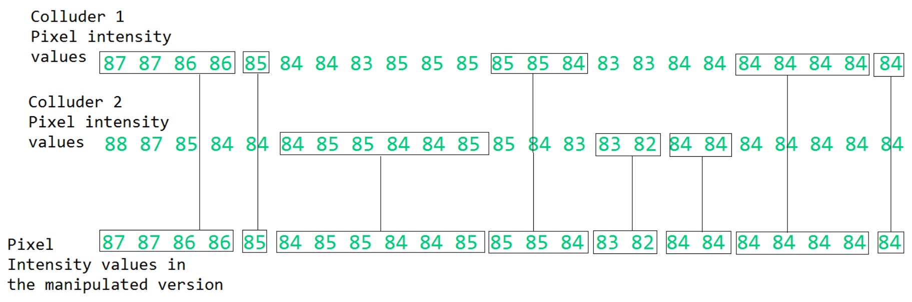


Figure 4.4: Example of a random segmentation attack

Figure 4.5 illustrates the "Lena" image and its transform in the frequency domain. It is a common practice to multiply the Fourier transform of the image by $(-1)^{x+y}$ as it shifts the origin of F(u,v) to frequency coordinates (M/2 +1, N/2 +1), which is the center of the $M \times N$ area occupied by the 2D fft. The frequency components are symmetric about the origin, which is located at coordinates (M/2 +1, N/2 +1). This means the upper right quadrant is symmetric to lower left quadrant, and the upper left quadrant to the lower right. It should also be taken into consideration that the very first row and the very first column do not have symmetric points. This information has to be considered when generating the attack.

As discussed in section 3.2.4, the detection process estimates the embedded bit information by regenerating the groups of the coefficients using the secret key. The logarithmic ratio of the sum of the coefficients selected E_1 and E_2 (section 3.2.4) estimates the bit information. In order to alter a watermark message bit, the difference between E_1 and E_2 has to be minimized, so that the detector response becomes unreliable for this particular position. However, without any knowledge of the secret key, it is impossible to say whether a manipulation implemented over a group of coefficients corresponds to a particular bit. On the other hand, assuming the attackers have knowledge in image processing operations according to human perception, i.e. that the watermark needs to be embedded on a middle frequency band range, attacks can be implemented on any noticeable difference in the

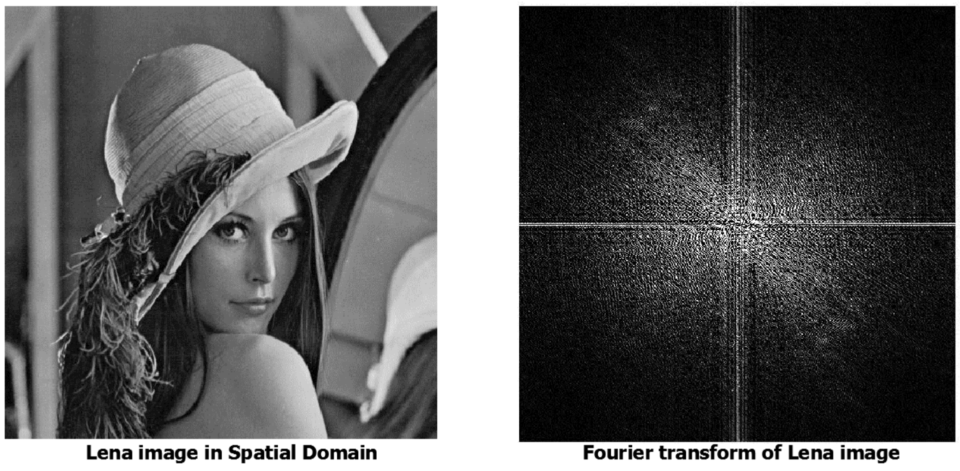


Figure 4.5: Image in spatial and Frequency domain

energy coefficients.

Figure 4.6 illustrates the difference image of the Fourier transform of the original and watermarked images. A difference image with and without the watermarking is generated, on which the bright triangular blocks represents the positions of the embedded watermark. The bright dotted line between the triangles represents the embedded template. Although the colluders do not have the original image, they compare the transforms of their watermarked copy and generate a difference image.

Figure 4.7 shows the difference image generated by comparison of two watermarked copies. The bright triangular region is less dense as it now only shows the positions where different information is embedded. For example, if one watermarked copy is embedded with a '1' on some specific positions, where as in the other copy on same positions a '0' is embedded, it will be displayed in the difference image. Another significant information colluders cannot access is the position of the template. As the template is embedded with equal strength on all watermarked copies, it cannot be detected in the difference image generated from colluders frequency response.

For each attack strategy discussed below, two image copies of the same content with individual watermarks embedded are transformed into the frequency domain. As the watermark is embedded in frequency domain, these attacks directly influence on watermarked positions. However, as the frequency bins on which the watermark information is to be embedded is randomly chosen, the colluders cannot associate any *detectable position* to be

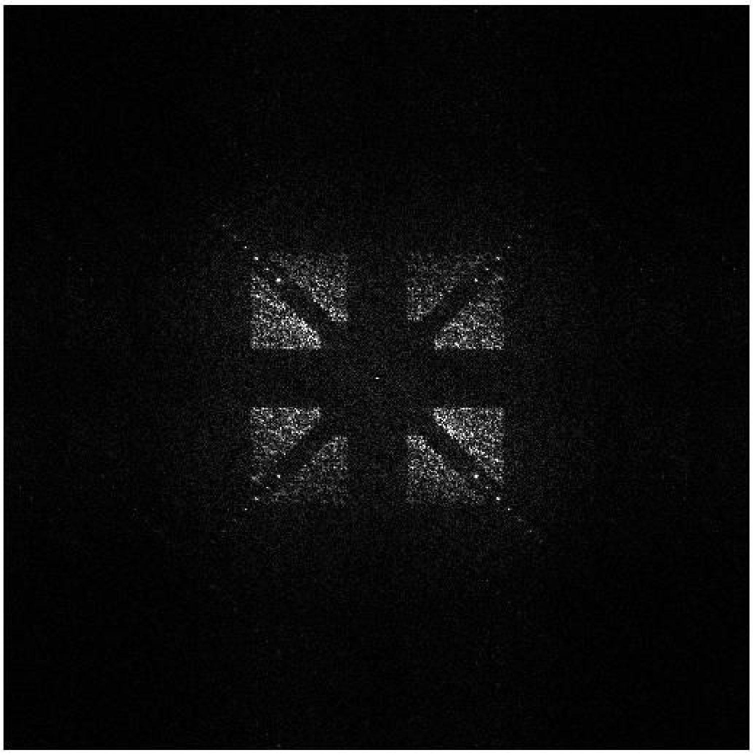


Figure 4.6: Difference image of an Original and Watermarked image frequency domain

an information for a particular watermark bit. Considering this information, we try to manipulate the fingerprint using the following strategies:

- **Maximum value Attack:** To remove the difference between the decision groups E_1 and E_2 (groups of coefficients estimating the watermark bit information, see section 3.2.1), the magnitudes of the coefficients of both the images are compared point by point. If a significant difference is detected, the colluders select the coefficient with higher magnitude to be placed in the manipulated image.
- **Minimum value Attack:** In this attack, coefficients with lower magnitude are selected to create a manipulated image. As discussed in the watermark embedding process (see section 3.2), for embedding a '1' , the coefficients of group E_1 were raised and that of group E_2 were lowered, while for embedding a '0' the coefficients were modified the other

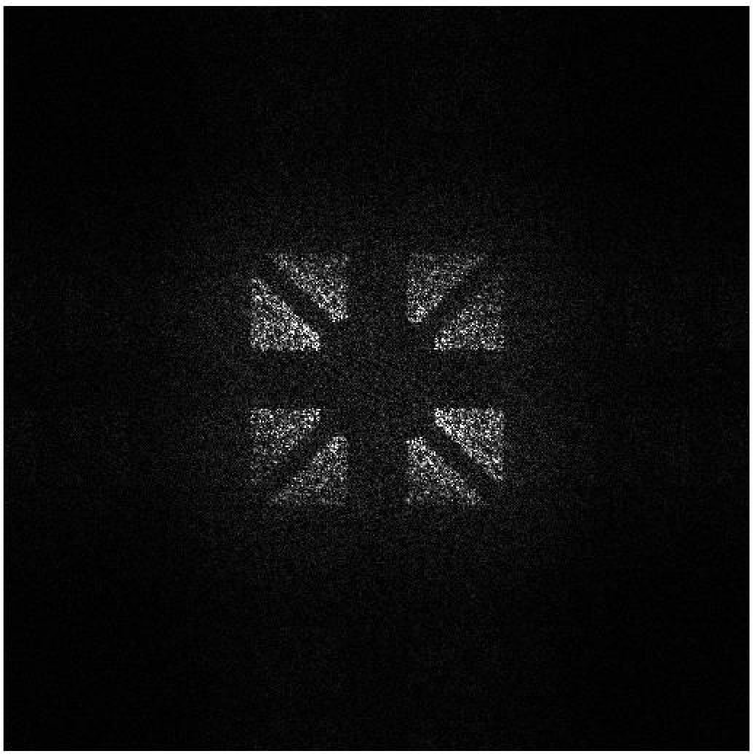


Figure 4.7: Difference image of colluders in frequency domain

way. The maximum and the minimum value attack aim to minimize the difference of coefficient groups E_1 and E_2 on detectable positions so that the estimation of these bits is no longer reliable.

- **Random value Attack:** Here, each coefficient is chosen randomly from any of the two images, in order to create a manipulated image.
- **Local average value Attack:** When comparing the coefficients, if noticeable differences are found, an average magnitude is calculated over the neighboring coefficients from both the images, and this average value is placed in the manipulated image.
- **Average image Attack:** When comparing the coefficients, an average magnitude is calculated over each point in the frequency domain, and this average value is placed in the manipulated image. This results in an average image of two watermarked images. This strategy aims to remove the magnitude difference between E_1 and E_2 on detectable

positions.

- **Image segmentation Attack:**In this attack strategy, alike the random segmentation attack in spatial domain, a segment size is randomly chosen over a predefined length. Then from the frequency rectangle of any of the two watermarked copies also chosen randomly, this segment of coefficients is taken to be placed in a manipulated version.

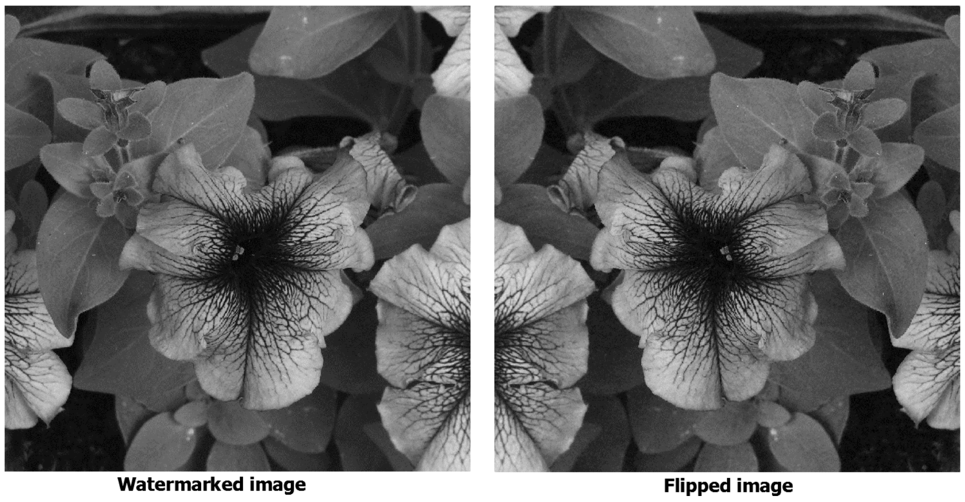


Figure 4.8: illustration of a watermarked image and an image undergone a flip attack

- **Cropped image Attack:** Normally, on the edges of an image lies irrelevant information (for example sky or clouds etc). Two images with different watermarks, containing irrelevant information are cropped from the edges, ensuring no loss of relevant information, but possible loss of watermark information. This strategy aims to the removal of undetectable watermark information i.e. the watermark positions where same information was embedded in both the images. Then, for example, the average of the frequency components of the two cropped images is calculated to generated a manipulated copy. Averaging the two images is considered to be one of the strongest attack, as theoretically it should minimize the difference between groups E_1 and E_2.

- **Flipped image Attack:** The images are mirrored along the y-axis, (i.e. the right half is mirrored to the left and the left half to the right), and for example an average attack is implemented. This attack ensures losing no information from the image as can be seen in figure 4.8, however as the image is mirrored the watermark sequence is also disturbed.

4.3 Attacks using the Perceptual Masking

The attack strategies discussed in sections 4.1 and 4.2 are based on the *marking assumption*[1] i.e. the colluders can only manipulate the fingerprint on positions where a difference is detected. It is obvious that the colluders cannot detect the difference in fingerprint bits only by comparing the two images. As the fingerprint message is imperceptibly embedded, they use marking assumption on the image content. However, using a perceptual model one can estimate the presence of watermark information without following the marking assumption. A perceptual model in imagery defines the range by which a pixel intensity can be modified without causing visible artifacts. Using perceptual model one can estimate the watermark presence even on undetectable positions (positions where the same watermark information is embedded in two images and hence could not be detected under Marking Assumption).

The watermarking algorithm (Chapter 3.2) used in this evaluation reshapes the watermark according to a perceptual model before it is added to the image. Using this perceptual model, colluders can estimate positions where the watermark information is embedded. This section presents attack strategies based on the perceptual model used in the embedding process. Other perceptual models like **watson metric** can also be used to generate the perceptual model.

Liu et al. spatial mask

The perceptual mask discussed in [6] considers three properties of human visual system (HVS) namely background luminance, edge proximity and texture masking. The mask is estimated in spatial domain, so using this mask colluders estimate whether a pixel contains the watermark information or not. Figure 4.9 shows an image and its perceptual mask calculated using the *Liu et al.* algorithm [6], where the bright pixels in the mask correspond to the positions at which watermark information is likely to be present. The mask value of a pixel $M(i, j)$ represents the intensity with which the watermark is embedded. Using this information, colluders try to manipulate on detectable as well as undetectable positions of a watermark.

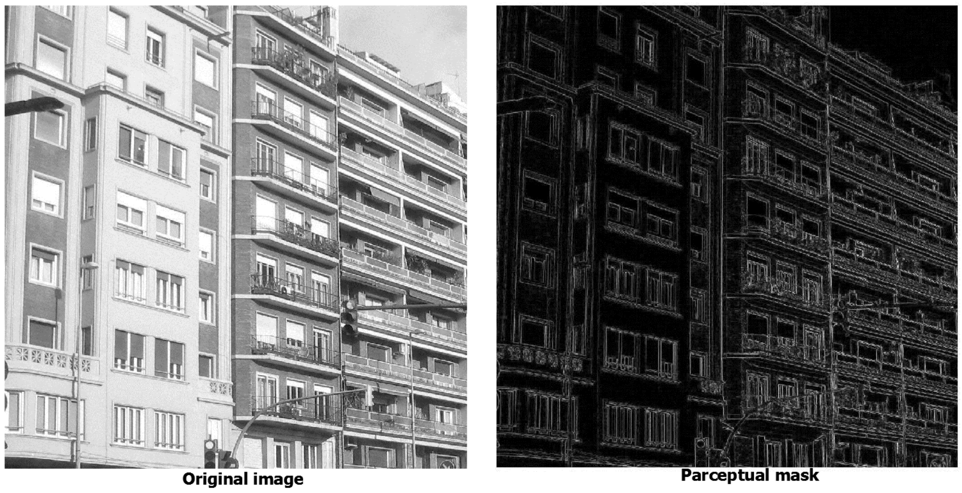


Figure 4.9: Illustration of an image and its Perceptual mask

To manipulate the watermark using the perceptual mask, one can estimate the range with which the pixel values are modified.

Considering the fact that colluders try to manipulate the embedded information as much as possible, we design some attack strategies using the perceptual mask of images. Figure 4.10 illustrates the scheme how colluders can use the perceptual masking to generate a manipulated image. Two colluders may generate a difference image, obtaining the *detectable positions* and also the difference of the detectable pixel intensities. One can generate a perceptual mask from any of the two copies, obtaining the watermark positions and the intensity with which pixels can be modified without noticeable artifacts. To estimate the range of modification of pixels one can relate the difference of pixel values on detectable positions with its masking intensity.

The range of manipulation can be estimated as

$$\Re = \frac{1}{2N}\sum_{i=0}^{n}\sum_{j=0}^{m}\frac{diff(I_1(i,j),I_2(i,j))}{M(i,j)} \qquad if \quad diff(I_1(i,j),I_2(i,j)) \neq 0 \tag{4.3}$$

where I_1 and I_2 are colluder images, n and m being the height and the width of the image respectively, N is the number of positions having differ-

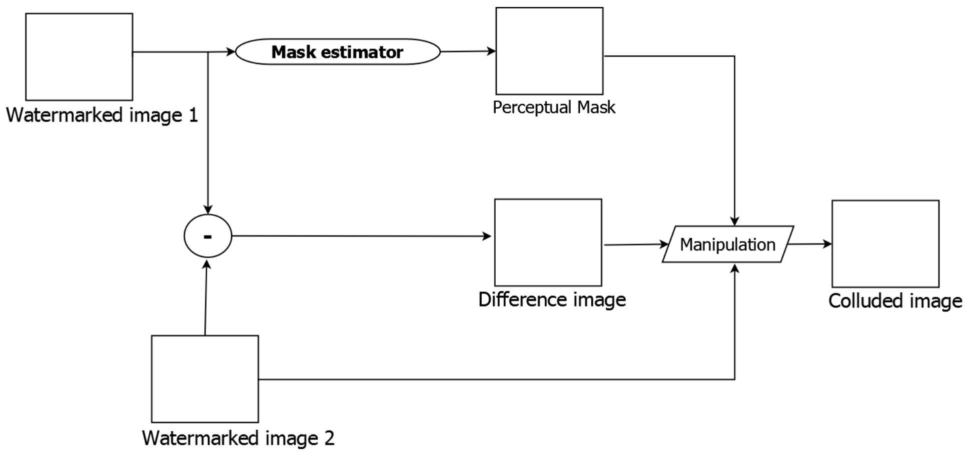


Figure 4.10: Collusion attack scheme using the perceptual mask

ent intensities and M is the masking value.

The range $\Re$ calculated over detectable positions is used to manipulate the watermark information on positions where pixel value of both the images are same, but masking value is non-zero (which means a high probability of the presence of watermark information). Considering the pixel optimization in section 3.2.3, the manipulation algorithm on undetectable position *(i,j)* is defined as

$$\gamma(i,j) = \begin{cases} \Re M_n(i,j), & \text{if } \Re M_n(i,j) > 1 \\ 1 & \text{if } M_n(i,j) > 0.01 \quad and \quad \Re M_n(i,j) < 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

whereas manipulation on detectable positions is done as

$$\gamma(i,j) = \begin{cases} diff(I_1(i,j), I_2(i,j))M_n(i,j), & \text{if } M_n(i,j) > 0.01 \\ \Re diff(I_1(i,j), I_2(i,j))M_n(i,j) & \text{otherwise} \end{cases} \tag{4.5}$$

where γ in the manipulation value on position *(i,j)*.

The adaptive attacks we made using the Manipulation mask are listed as follows

- **Random manipulation attack** In this attack, the manipulation of the image is performed by randomly increasing or decreasing the manipulation masking.

$$Image_c(i,j) = Image_{WM}(i,j) \pm \gamma(i,j) \tag{4.6}$$

- **Incrementation attack** Contrary to the random manipulation, this attack aims to increase the pixel values on undetectable positions to manipulate the watermark

$$Image_c(i,j) = Image_{WM}(i,j) + \gamma(i,j) \tag{4.7}$$

- **Decrementation attack** As the name suggests, pixel values are decreased to obtain a colluded version.

$$Image_c(i,j) = Image_{WM}(i,j) - \gamma(i,j) \tag{4.8}$$

where $Image_{WM}$ is the watermarked image and $Image_c$ is the resulting manipulated or also called *colluded image.*

4.4 Results

This section discusses the results of the attacks implemented on the watermarked images. As the focus is on collusion secure fingerprinting, the strategies discussed in the previous sections are implemented to test the vulnerability of the embedded fingerprints.

4.4.1 Spatial domain Attack Results

To analyze the performance of embedded fingerprints, 100 attacks of each strategies discussed in section 4.1 were implemented on 100 different images. Table 4.1 illustrates the attack results. The most important result is that No innocent user was accused of generating the manipulated version of an image, proving the fingerprint code used to be zero-false positive.

The attacks generated in the spatial domain did not manipulate the watermark to a significant extent as is evident from table 4.1. Almost 90 % of the colluders were successfully traced back from the *colluded* images generated using these attack strategies. The manipulation over the pixel intensity values do not correspond to localize manipulation of frequency domain coefficients, at which the watermark information is embedded. This leads to the low False Negative error (FN error) i.e. the probability of no colluder getting caught.

Attack Strategy	Innocents	Colluders Caught	FN Error
Block segmentation	0	96%	4%
Average value	0	90%	10%
Maximum value	0	88%	12%
Random value	0	92%	8%
Minimum value	0	93%	7%
Random segmentation	0	92%	8%

Table 4.1: Results against attacks in spatial Domain, 100 attacks each

4.4.2 Frequency domain Attack Results

Collusion attacks discussed in section 4.2 were generated in the frequency domain. 200 attacks of each strategy discussed were implemented on 150 images. Table 4.2 illustrates the watermark detection and the colluder tracing result.

Attack Strategy	Innocents	Colluders Caught	FN Error
Minimum value	0	67.5%	32.5%
Maximum value	0	73%	27%
Random value	0	72%	28%
Local average value	0	69%	31%
Average image	0	72%	28%
Segment image	0	65%	35%
Cropped image	0	60%	40%
Flipped image	0	76%	24%

Table 4.2: Results against attacks in Frequency Domain, with τ=0.04

The flipped image attack was implemented to disturb the watermark information sequence, but as the watermark embedding algorithm is symmetric, the secret key is still able to detect the frequencies containing the watermark information.

Although the attacks generated in the frequency domain caught less colluders as compared to that of in the spatial domain, also no innocent user was accused to be a part of collusion. Cropped image attack caught least colluders as during the cropping process, watermark information can be lost. However, the false negative error rate remains below 0.5, which is considered high enough to refrain the users to attempt a collusion attack.

In the frequency domain, colluders try to lower the difference between coefficient groups E_1 and E_2, so as to fail the watermark bit estimation discussed earlier in section 3.2.4. Recalling, a threshold τ was used for the bit estimation, with detector response $> \tau$ estimated as 1 and $< -\tau$ estimated as 0. For the evaluation, we used τ to be 0.04. As expected, many of the attack strategies successfully reduce the detector response, resulting in a failure of the watermark detection. Hence, as the result in table 4.2 indicate, in almost 30-40 % of every attack strategy implemented, no colluder was accused.

4.4.3 Attacks generated using Perceptual masking

Collusion attacks using the perceptual masking information were implemented to test the embedded fingerprints. The focus is on manipulation of as many watermark positions possible, be it detectable or undetectable. However, for each undetectable position holds, that the colluders cannot identify whether the energy has been added or subtracted. Therefore three strategies as discussed in chapter 4.3 were implemented to generate a manipulated copy and trace back the colluders. The results of the tests are illustrated in table 4.3.

Attack Strategy	Innocents	Colluders Caught	FN Error
Random adaptive Attack	0	93.75%	6.25%
Incrementation Attack	0	80.5%	19.5%
Decrementation Attack	0	96%	4%

Table 4.3: Results against attacks using perceptual mask, 200 attacks each

As the results indicate, attacks using the perceptual model were quite unsuccessful to hide colluder's identity. This is because the manipulation was done in spatial domain, which does not manipulate coefficients in frequency domain with the same strength.

Also, as per the perceptual model, watermark is embedded on positions where high texture activity and edges are present. On these positions the pixel intensity value is already quite low. Hence reducing the pixel intensity with a large value will not prove to manipulate the message. For example, on a pixel with intensity value 4, if manipulation algorithm allows a modification of 20, decrement of 20 will not manipulate significant information as the intensity range for gray scale is from 0-255. Therefore the pixel value will be decremented by 4 instead of 20. Whereas in the incrementation attack, the stronger manipulation can manipulate the watermark information.

However, this manipulation on positions with same intensity in both the images have only a 50% chance of disturbing the watermark message. This is because if during the embedding process the pixel intensity was lowered to carry watermark information, any further decrement will enhance the watermarking strength. This theory is also supported in the results posted in table 4.3. The incrementation attack proved better for the colluders as compared to the decrementation attack due to the reasons stated above. However, the attempt of forging a copy with an innocet user's information is again unsuccessful as no innocent user was traced.

As the name suggests, the random adaptive attack randomly adds or subtracts the information on undetectable positions. However, it gives a very low probability of disturbing the watermark as the random manipulation has equal probability of enhancing the watermark strength.

Hence as the result portray, even with the estimation of watermark presence, attacks generated on positions with same watermark information fail to disturb watermark message. No innocent user was accused in the tests made. In significant percentage of the attacks, both the colluders were traced, with incrementation attack being the most successful in colluders' point of view, where the successful tracing of the fingerprints is 80 %, the lowest in relative comparison.

However, the increment in the False Negative error rate (probability of no colluder getting caught) after the attacks generated in frequency domain is of great concern, on which the analysis of improving the tracing strategy is conducted in the next chapter.

Chapter 5

Watermark detection analysis and the Colluder tracing Algorithm optimization

As discussed in section 3.2.4, the estimation of the watermark message bit is based on the logarithmic ratio between the frequency coefficients groups namely E_1 and E_2 respectively (see section 3.17). The bit is estimated to be a '1' if the ratio is greater than τ and a '0' if the ratio is lesser than -τ, where τ is a predefined threshold. The ratio falling between the range of -τ and τ are considered unreliable and hence the bit estimation is not considered reliable. For our convenience and further references, this ratio is called the 'Detector response'.

5.1 Watermark detection analysis

To analyse the performance of a watermark message, image processing attacks and also collusion attacks are implemented to study the detector response. One of the tested watermarked images, its compressed version and and its version generated after an average collusion attack is shown in figure 5.1. Figure 5.2 shows the response of a 104 bit watermark message, without undergoing any attack. The vertical lines represent the positive and the negative detector response threshold $'\tau'$, for the '1' and '0' bit value estima-

Watermarked Image
Watermarked Image after 30% jpeg compression
Watermarked Image after collusion attack

Figure 5.1: The watermarked image, the compressed watermarked image and the colluded watermarked image

tion respectively. The same watermarked image is then tested against some image processing attacks, namely the jpeg compression and the median filter attacks.

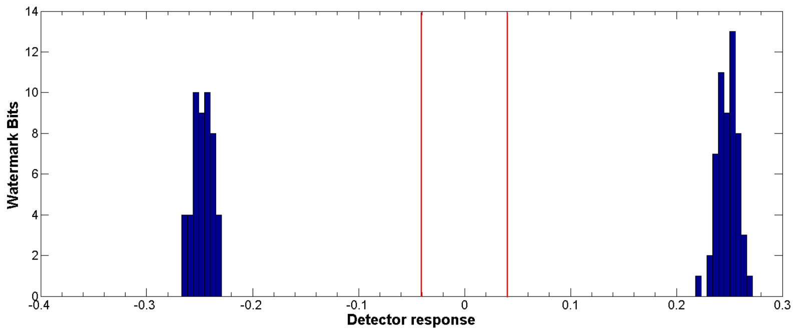


Figure 5.2: Detector response of a watermarked image

The detector response of a watermarked image that is manipulated using the image processing attacks is illustrated in figure 5.3. It can be seen that both the distributions, the one for the positive detector response as well as the one for the negative detector response are affected. However, there is still a large gap between the thresholds and the corresponding nearest bin. Liu et al. mentioned it in [6], that due to the watermark reshaping in the embedding process, the detector response of any watermark message bit shall not fall in the unreliability zone between -τ and tau. This theory stands correct against the image processing attacks, however the collusion attacks proved the contrary.

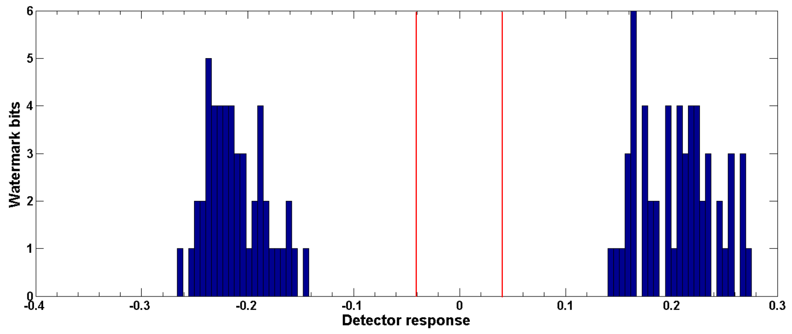


Figure 5.3: Detector response of a watermarked image undergone Jpeg compression amd median filter attacks

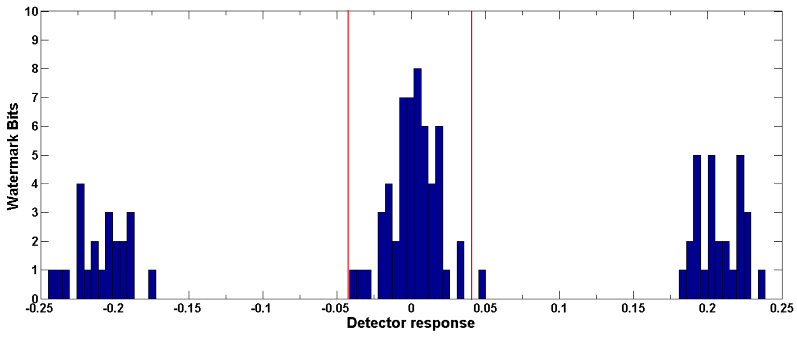


Figure 5.4: Detector response of a watermarked image undergone a collusion attack

The detector response, of a watermarked image that has undergone an average collusion attack is illustrated in figure 5.4. The attack is generated in the frequency domain, which aims at averaging the frequency response of an image. This results in minimizing the logarithmic ratio for many of the watermark message bits. As can be seen in the figure, almost half of the watermark message bits fall between the thresholds τ and -τ, which compromises the reliability of the watermark message i.e. on those positions in the message where the response is unreliable, it is difficult for the watermark detector to estimate a '1' or a '0', and instead would label it as an undefined symbol '?'. Consequently, as the fingerprint tracing algorithm takes the extracted watermark message as the manipulated fingerprint message, the tracing of the colluders is not reliable if the fingerprint message contains undefined symbols. Therefore, another tracing layer over the existing tracing algorithm is proposed in the next section, to counter this problem.

5.2 Colluder tracing algorithm optimisation

The existing 2-secure colluder tracing algorithm proposed in [12] assumes a successful detection of a manpulated fingerprint message. However, with the collusion attacks on images resulting in the generation of undefined symbols '?' in the fingerprint, the tracing of the colluders is not possible and hence increases False-Negative error rate (probability of not being able to trace any colluder). Therefore, this work proposes another tracing strategy to be added to the existing colluder tracing algorithm, so as to reliably trace back the colluders.

Recalling from chapter 4, the colluders compare their images and add modifications on positions where they detect a difference (*Marking Assumption*). These differences correspond to the positions where the watermark embedding information is distinct. Modification on these positions might lower the detector response of some bits. Using this information, the positions in the fingerprint where the detector response for a bit is low, is of significant importance. As the analysis is on the attack generated by two colluders, these positions in the fingerprint correspond to be detectable positions (positions where the embedding bit value for both the colluders is different). For example, if on a specific position i in the watermark, the i^{th} bit has a low detection response, it is a position where the two colluders had distinct bits embedded in their respective fingerprints i.e. one having a '1' and the other having a '0' or vice versa. Using this information, the tracing algorithm is optimised.

Tracing algorithm optimisation

The manipulated fingerpint message extracted from a colluded image is denoted as y. If the existing tracing algorithm (section 3.1.2) is not able to trace back the colluders due to the unreliability of the fingerprint message, all positions in the manipulated fingerprint y showing the undefined symbol are acquired. These positions correspond to the detectable positions, where the two colluders have distinct bit values in their respective fingerprints.

The distributor, makes pairs of all the fingerprints issued for that particular image. For each pair, the distributor checks on each of the detectable positions (extracted from y), if the bit information of both the user fingerprints are distinct. If in *any* detectable position, the bit information in both the fingerprint is same, the pair is considered not to be a collusion-pair, and the scheme continues for the remaining pairs.

If a pair has distinct bit values on all of the detectable positions, the pair is added to the suspect queue. After checking all the pairs, if only one pair is suspected, it is established to be a colluder pair. If the algorithm suspects more than one pair, the common fingerprint that is present in all suspected pairs is considered to be a colluder. However, if two or more pairs are suspected and do not contain a common fingerprint, the algorithm accuses no user to be a part of the manipulation.

The proposed addition to the tracing algorithm is tested against collusion as well as image processing attack strategies, some of which are discussed in the next section.

5.3 New algorithm evaluation

The modification on the tracing algorithm is applied and tested against different attack strategies. In practice, the colluders after generating collusion attack are likely to follow the image processing attacks so as to disturb the watermark. Hence, this algorithm is tested against the combination of collusion attacks and image processing attacks.

5.3.1 Implemented attack strategies

The new tracing algorithm is tested using the image processing application 'Stirmark 3.1' [8], an image processing application used to test the robustness of watermarking algorithms. In the implemented testbench, 100 watermarked images were first undergone an average collusion attack, followed by image processing attacks like jpeg compression, image scaling and rotation, geometric distortions, sharpening and smoothing filter attacks. Prior to the discussion of the results, it is vital to discuss some of the implemented image processing strategies and their effect on the image.

- **JPEG Compression** is a commonly used method of lossy compression for digital photography (image). The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality. JPEG compression can compress with a ratio of 5:1 with little or no visible artifacts. As jpeg comression is performed in the Distrete cosine transform(DCT) domain, it is a lossy compression discarding the high frequency information. This can directly lead to the loss of watermark information, as the watermark is also embedded in the frequency domain.

- **Geometric distortions** in an image can be described as rotating / distorting the image by a small degree along its centre. This changes the pixel positions and hence, affects the frequency transform of the image as well. This can result in watermark manipulation, as the positions where the watermark is embedded, after geometric distortion their lies different information.
- **Image Sharpening filters** are used to enhance the line structures or other details in an image. Line structures and edges can be obtained by applying a sharpening filter (or a high pass filter) on the image. Being a high pass filter, it attenuates the low and middle frequency bins which can disturb the watermark (as the watermark is embedded in the middle frequency range).
- **Gaussian and median filters** are used to smoothen an image and remove noise. These filter represent low-pass characteristics in the frequency domain

The above mentioned image processing attacks were implemented and the manipulated images generated. This is followed by detecting the watermark from the manipulated images. As mentioned before, the new algorithm traces the colluders using the detector response of those bits, whose detector response exceeds the response threshold i.e. in the region between -τ and τ. Therefore, to find an optimum threshold value for τ, the tracing algorithm was tested with different values for τ.

Figure 5.5 represents the test results of various attacks using different threshold values. The false negative error rate (probability of not tracing any colluder) is plotted against different threshold values against the attack strategies discussed above. Analysing both the plots, the threshold value of 0.04 is considered to be an optimum value as the false negative rate is lowest against all the attacks.

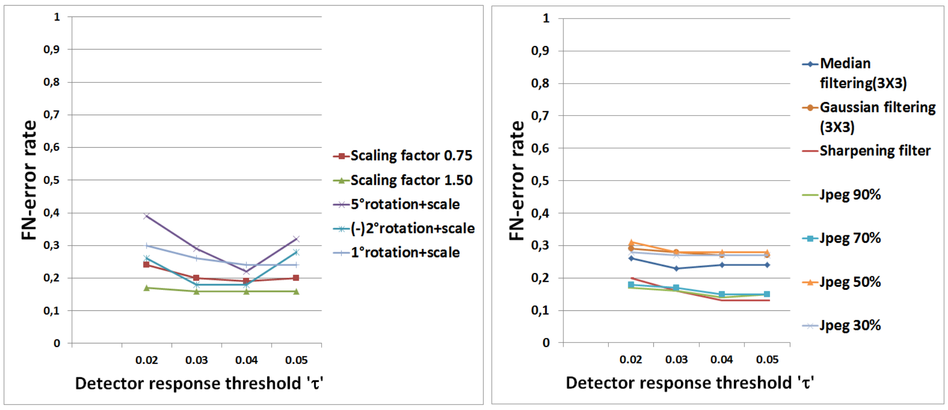


Figure 5.5: FN error rate against collusion attack followed by image processing attacks

Using the optimum detector response threshold, the optimized tracing strategey is added to the tracing algorithm proposed by Schfer et al. [12] as a second security layer, with the motivation of decreasing the false negative rate. Figure 5.6 shows the comparison of the false negative rate between the older tracing strategy [12] and the proposed optimised strategy. As illusterated in this bar chart comparison, the proposed tracing strategy reduces the false-negative error by 3% - 8% against various attack strategies.

Hence, using the optimized tracing colluder strategy, the false-negative rate decreases by an average of 5%, ensuring more colluders getting caught and also retaining its zero-false positve characteristic (probabitlity of accusing an innocent user).

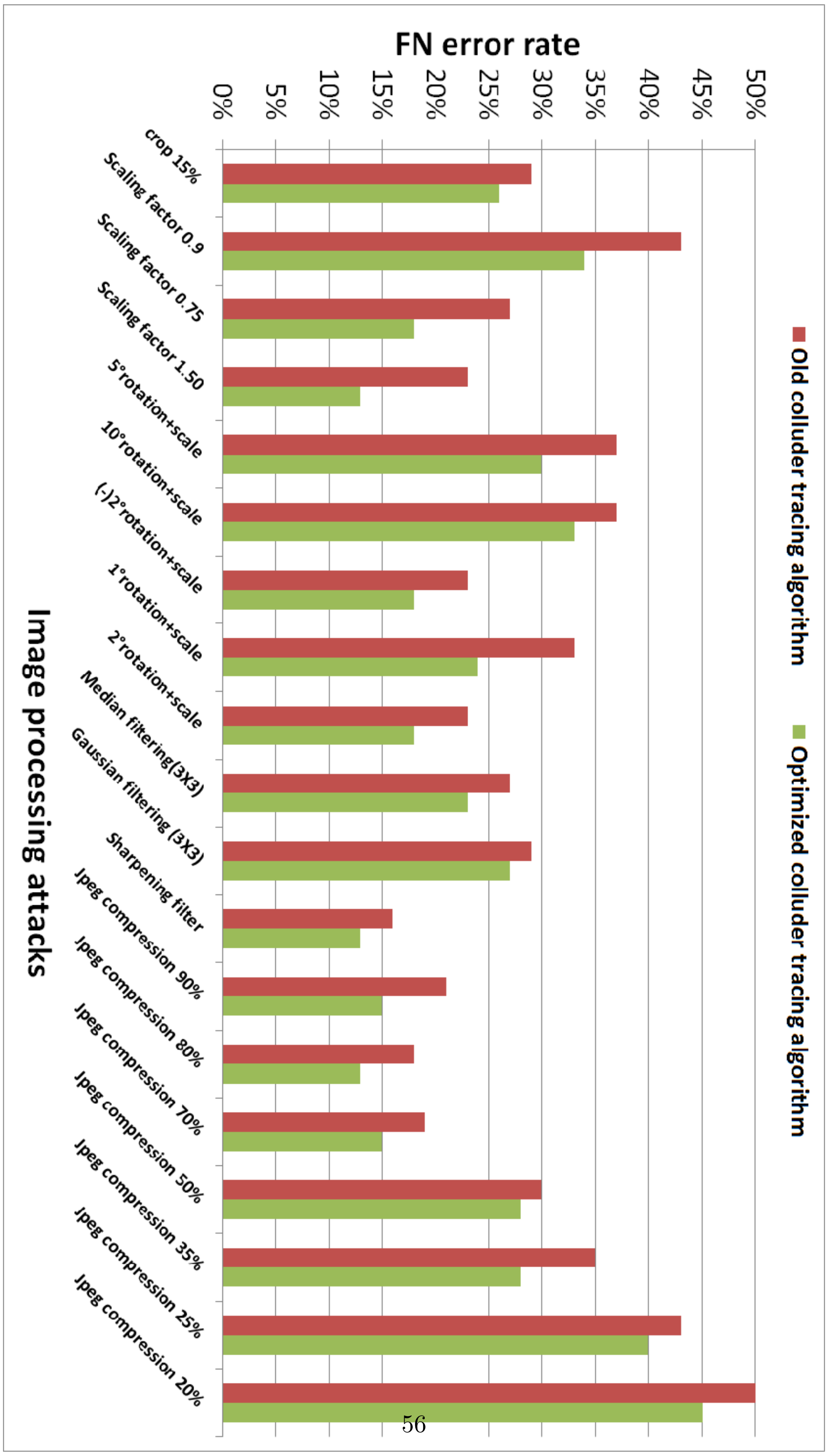


Figure 5.6: FN error rate comparison among the old and optimized colluder tracing strategy

Chapter 6

Conclusion

This thesis work integrates the 2-secure fingerprinting codes into the image watermarking application software licensed by the Fraunhofer SIT. This covers the implementation of 2-secure fingerprint code generation and the tracing algorithm to trace back the colluders from a manipulated fingerprint as proposed in [12]. The watermarking application proposed in [6] is integrated for embedding and detection of these fingerprints. The robustness of the watermark information and the fingerprint performance is evaluated against the implemented collusion attacks strategies in spatial and frequency domain, discussed in chapter 4.

The affects of the collusion attacks on the manipulated fingerprints are discussed in chapter 5. Based on this analysis, this thesis work also proposes the optimisation of the colluder tracing algorithm, and its evaluation in sections 5.2 and 5.3 respectively. The optimisation in the tracing algorithm reduces the false-negative rate by 5%, and proves the fingerprints being resistant against the combination of collusion and image processing attacks.

Chapter 7

Appendix

”This Bachelor Thesis contains an appendix of program listings, hardware descriptions etc. on a CD (disk or supplementary booklet). This Appendix is deposited with Prof. Dr. Hans-Jrgen Hotop.”

Chapter 8

Declaration

I/we declare within the meaning of part 16(5) of the General Examination and Study Degree Programmes at the Faculty of Engineering and Computer Science and the Examinations and Study Regulations of the International Degree course Information Engineering that: this Bachelor Thesis has been completed by myself/ourselves independantly without outside help and only the defined sources and study aids were used. Sections that reflect the thoughts or works of others are made known through the definition of sources.

City, Date Signature

Bibliography

[1] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, 44(5):1897–1905, Sept. 1998.

[2] B. Chor, A. Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Transactions on Information Theory*, 46(3):893–910, May 2000.

[3] I. J. Cox, M. L. Miller, and J. A. Bloom. *Digital Watermarking.* Morgan Kaufmann, 2002.

[4] J. Dittmann and M. Steinebach. Joint watermarking of audio-visual data. In *Proc. IEEE Fourth Workshop on Multimedia Signal Processing*, pages 601–606, 3–5 Oct. 2001.

[5] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001.

[6] M. Steinebach H. Liu. Improved fourier domain patchwork and template embedding using spatial masking. In *SPIE Proceedings of Media Watermarking, Security, and Forensics*, 2012.

[7] Xiangui Kang, Jiwu Huang, Yun Q Shi, and Yan Lin. A dwt-dft composite watermarking scheme robust to both affine transform and jpeg compression. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(8):776 – 786, aug. 2003.

[8] Martin Kutter and Fabien A. P. Petitcolas. A fair benchmark for image watermarking systems. In *To in E. Delp et al. (Eds), in vol. 3657, proceedings of Electronic Imaging '99, Security and Watermarking of Multimedia Contents, San Jose, CA, USA*, January 1999.

[9] Peter Meerwald and Teddy Furon. Towards joint decoding of tardos fingerprinting codes. *CoRR*, abs/1104.5616, 2011.

[10] Koji Nuida. An improvement of short 2-secure fingerprint codes strongly avoiding false-positive. In Stefan Katzenbeisser and Ahmad-Reza Sadeghi, editors, *Information Hiding*, volume 5806 of *Lecture Notes in Computer Science*, pages 161–175. Springer Berlin / Heidelberg, 2009.

[11] Koji Nuida, Satoshi Fujitsu, Manabu Hagiwara, Takashi Kitagawa, Hajime Watanabe, Kazuto Ogawa, and Hideki Imai. An improvement of discrete tardos fingerprinting codes. *Des. Codes Cryptography*, 52:339–362, September 2009.

[12] Marcel Schäfer, Waldemar Berchtold, Sascha Zmudzinski, and Martin Steinebach. Zero false positive 2-secure fingerprinting watermarking based on combining hamming distance conditions and parent pair search. In *Proceeding of The 12th ACM Workshop on Multimedia and Security 2010 (MMSEC 2010)*, Sep 2010.

[13] Boris Skorić, Stefan Katzenbeisser, and Mehmet Celik. Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. *Designs, Codes and Cryptography*, 46:137–166, 2008. 10.1007/s10623-007-9142-x.

[14] G. Tardos. Optimal probabilistic fingerprinting codes. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC), pp. 116-125*, 2003.