1-1-2006

The use of wavelet watermarking and statistical classification techniques for collusion detection and identification in multimedia forensics

Anthony G. Persaud
Iowa State University

The use of wavelet watermarking and statistical classification techniques for collusion detection and identification in multimedia forensics

by

Anthony G. Persaud

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Computer Engineering

Program of Study Committee:
Yong Guan, Major Professor
Jennifer Davidson
Johnny Wong

Iowa State University
Ames, Iowa
2006

Graduate College
Iowa State University

This is to certify that the master's thesis of

Anthony G. Persaud

has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy

DEDICATION

I would like to dedicate this thesis to my parents Tillak Persaud and Heriberta Santiago. I would also like dedicate this to my major professor, Dr. Yong Guan and my committee members, Dr. Jennifer Davidson and Dr. Johnny Wong, for valuable support and interesting discussions.

Also, to all my great friends here at Iowa State University, who I consider my new family, who have helped me and supported my all of my ideas, concerns and frustrations.

Additionally I would like to dedicate this thesis to Dr. Irwin Jacobs and Steve P. Jobs for valuable inspiration and advice.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This research proposes a wavelet-based multimedia fingerprint scheme and statistical clustering algorithm for collusion detection and identification.

The use of digital multimedia has steadily increased using mediums like the Internet. Encryption is generally used to safeguard content while in transmission, but offers no protection against duplication.

Tracing unauthorized content distributors has become an increasing concern for the media industry. Unauthorized duplication, piracy, and illegal redistribution of multimedia content account for several billion dollars in losses every year.

It is important to design reliable investigative techniques against unauthorized duplication and propagation, and provide protection in the form of theft deterrence. Some fingerprint embedding schemes are robust against single-user modification attacks. However, a new breed of attacks, known as collusion attacks, have been used to defeat those underlying schemes. These attacks use the combination of multiple fingerprinted copies to create a new version where the underlying fingerprint is highly attenuated, becoming untraceable to the colluders.

This research adopts the use of wavelet transforms and statistical classification techniques to effectively identify the set of colluders involved in a collusion attack while maintaining low miss rates and false accusation rates. The experimental results show that the solution is effective in identifying large colluder sets without the knowledge of the number of colluders involved in an attack and the collusion attack used.

CHAPTER 1. INTRODUCTION

The availability of content and the economical costs of electronic distribution have helped increase the adoption rate of content stored in electronic format. New digital content, such as music, pictures and movies, is being distributed over the internet at a incredibly fast pace. As an example of the increase adoption rate of this distribution system, it took less than three years for Apple Computer^{TM}to sell its one billionth song on its internet downloading music store, iTunes^{TM}, while it took more than 8 years for McDonald's^{TM}to sell its one billionth burger.(1; 24)

The global nature of the internet and the numerous content distribution applications such as file transfer protocol (FTP), peer-to-peer (P2P) file sharing, and pirate-software download sites such as Warez(63) have made it easier for unauthorized users to receive multimedia content. As more digital content becomes available through the internet for purchase and distribution, the higher the concern for security and unauthorized duplication and proliferation.

Tracing unauthorized content distributors has become an increasing concern in the media industry. According to the Record Industry Association of America, piracy costs the music industry alone around *$4.2 billion* dollars each year.(47) The Motion Picture Association of America esitmates that unauthorized duplication and distribution of movies will cost them an estimated *$15 billion* over the next 4 years.(21). Eight people were charged by authorities in the United States for providing an illegal online release of the last Star Wars movie, "Episode III : Revenge of the Sith".(3) The movie industry states that illegal distribution issues like these are the current cause in the worldwide box office revenue drop of 7.9% in 2005 from 2004.(41)

Although encryption has been used to safeguard the content while in transmission, it does not offer protection against further unauthorized distribution and duplication. As an example, the use of Apple iTunes$^{\mathrm{TM}}$music software lets an authorized user purchase a song. Using its FairPlay$^{\mathrm{TM}}$(62) system, the song is encrypted so that only the user with the specified key is able to perform playback. However, this encryption system has been circumvented, with an application called Jhymn, (22) which mimics an iTunes$^{\mathrm{TM}}$agent. By mimicking the agent, it can download the specified keys from the server and decrypt the song that was purchased by the user. Using its additional features, the application can additionally remove traceable fingerprints such as the *apID* and *cprt* atoms which identify the multimedia artifact to the original purchaser. In this example, this leaves the multimedia audio file available for future distribution.

Digital watermarks were designed to provide a unique method of marking multimedia artifacts to determine the original recipient of the content. Watermarks are designed to be reliable investigative techniques against tracing unauthorized duplication and propagation, and also provide protection in the form of theft deterrence. Some fingerprint embedding schemes were developed to be robust against single-user attacks. However, a new breed of attacks known as *collusion attacks*, have been used to defeat these underlying schemes. These attacks use the combination of multiple fingerprinted copies in order to create a new version where the underlying fingerprint is highly attenuated, becoming untraceable to the colluders. Due to the highly connected digital world, these attacks have been proven to be highly cost-effective for attackers to become untraceable.

Figure 1.1 shows the general process of fingerprinting a specified multimedia artifact. A fingerprint for a user named *Alice*, who wishes to purchase a legal copy of an artifact, will be generated from a specific customer id. This id generates a unique fingerprint sequence that will then be embedded into the original multimedia artifact before being received by Alice. Therefore, Alice will never have in her possession the original non-fingerprinted multimedia artifact.

Given that Alice may know other users who have also purchased the same multimedia artifact, these users can collude together to generate a new copy where the fingerprint has

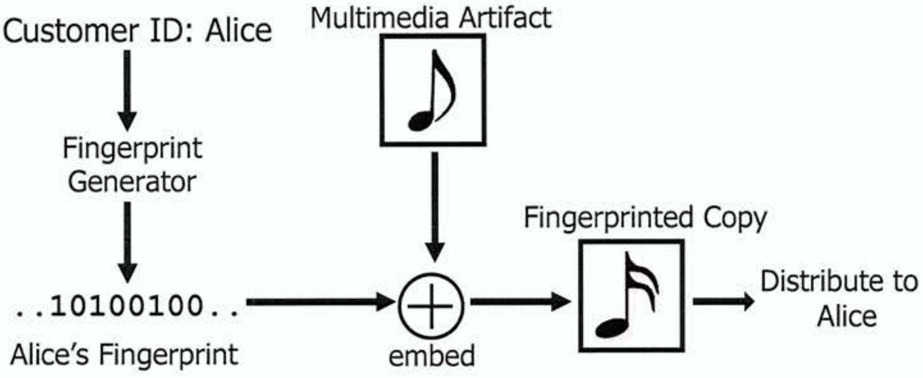

Figure 1.1 Multimedia artifact fingerprinting

been removed or attenuated. This copy can then be distributed to other users, which may have a relationship with Alice, through peer-to-peer software. The framework of the collusion attack described is seen in Figure 1.2.

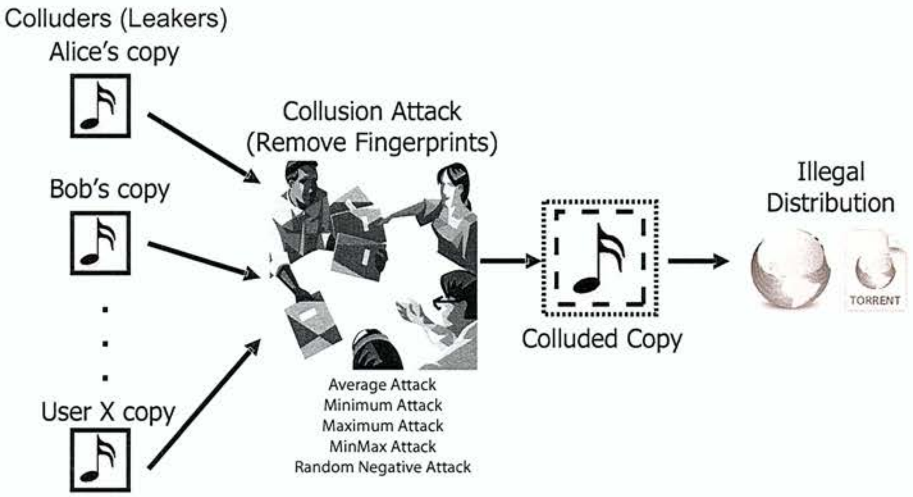

Figure 1.2 Framework of a collusion attack

The general problem faced by a digital forensic investigator in determining the original illegal distributor of a multimedia artifact is shown in Figure 1.3. Given a suspicious or colluded copy, a digital forensic investigator must extract the recoverable fingerprint. Using this recovered fingerprint, which may be a partial recovery, they must determine the original owner of this multimedia copy. If a direct match cannot be determined, one can suspect that it may be a copy generated through collusion. Using the fingerprint information, the digital forensic investigator requires a method for tracing back the conspirators in generating this colluded copy.

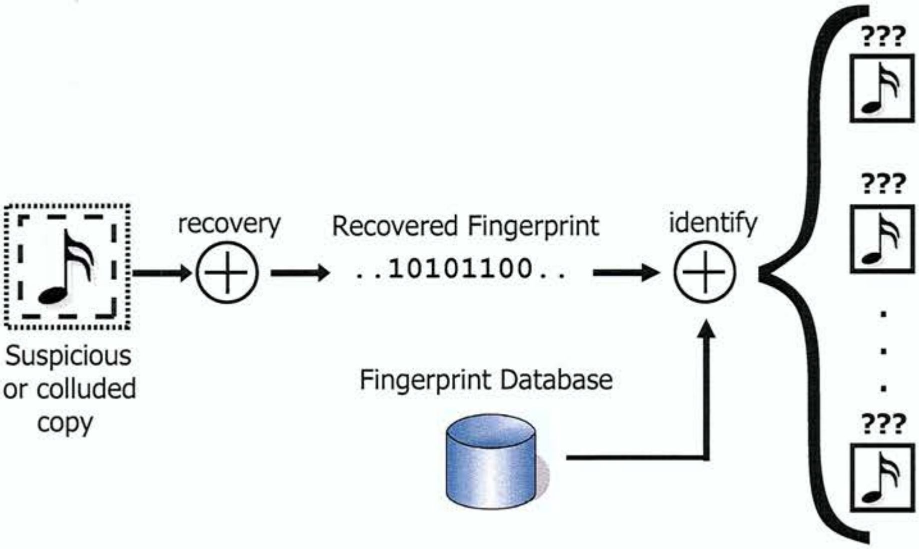

Figure 1.3 Colluder set identification problem

Some colluder detection techniques rely on direct pattern correlation by using orthogonal modulation to generate independent fingerprints.(65) This, however, assumes that the entire fingerprint can be recovered and the entire set of possible leakers is known. In some variations, multiple watermarks are used for every multimedia artifact to increase the probability that an investigator can trace the artifact to the sources used for collusion.

Others methods try to generate all possible combinatorial pairs of possible colluder sets to find a highly correlated fingerprint.(59) This methodology requires a large amount of computational complexity in testing these combinatorial predictions.

Digital forensic investigators require a method that is cost effective in both implementation and time complexity, in order to identify the colluders involved in a collusion attack. In addition, they would like to minimize the number of false accusations made against innocent individuals not involved in an attack. Identifying these colluders would help aid the media industry in preventing future proliferations from those redistributors.

The main objective of this research is to find a solution that, given a colluded copy of a multimedia artifact, identify the possible set of users involved in an attack without the knowledge of the collusion attack used and the total number of colluders involved in that collusion attack. In addition, the research aims to also provide a solution that minimizes the number of misses and false accusations generated.

This research has found that the use of wavelet transforms and unsupervised statistical clustering techniques can be used to effectively identify the set of colluders involved in a collusion attack while minimizing the number of missed colluders and false accusations generated.

The research provides a framework divided into three phases: (I) embedding phase, (II) recovery phase, and (III) identification phase. The embedding phase will describe the process of embedding a digital fingerprint into a multimedia file using the *discrete wavelet transform* (DWT). The recovery phase will specify the process of recovering the fingerprint in a colluded copy. The last phase, will provide details in a new method of identifying colluder sets involved in a multimedia collusion attack. The unique contributions of this research are:

- Full fingerprint recovery is not required.
- The colluder set is built from joint density observations and not from predictions.
- Identifying colluder sets becomes independent of the collusion attack used.
- Independent of the number of colluders involved in the attack.

An evaluation is performed against the developed solution with three different colluder set distributions. Two implementations are proposed in our research: with and without spread spectrum watermarking.

The results demonstrate that colluders can be identified when they employ the use of the minimum, maximum and minimun/maximum attacks. The methodology also minimizes the number of false accusations incurred by the use of these attacks. Although the random negative attack is not completely mitigated, the false accusation rates for the positively and negatively skewed colluder data sets are highly minimized. The average attack seems to be the strongest attack against the proposed work. In general, the evaluation results of this research demonstrate the solution is highly effective at identifying large number of colluders with the collusion attacks tested.

CHAPTER 2. A FRAMEWORK FOR MULTIMEDIA FORENSICS

The proposed fingerprinting and identification scheme can be partitioned into three main phases: (I) embedding phase, (II) recovery phase, and (III) identification phase.

Phase I focuses on using robust embedding methods to embed the watermark information into different multimedia artifacts.

Phase II focuses in performing a recovery of the embedded watermark. In some cases of Phase II, only part of the watermark may be recovered due to various alterations attacks.

Phase III performs the correlation between the embedded watermark and the set of known watermarks that correspond to known users.

Most existing watermarking schemes address issues in Phases I and II. The proposed solution in this research aims at addressing the issues presented in all three phases.

2.1 Phase I : Embedding

There have been various watermark embedding schemes used for multimedia forensics (51): (i) least significant bit (LSB) modification embedding, (ii) correlation-based embedding, (iii) frequency domain embedding and (iv) wavelet watermarking.

The least significant bit (LSB) (57) scheme involves replacing the least significant bits of audio samples or images with the watermark bits. (50) Though this method is easy to implement (25), however, it is easily defeated through any alteration that change the lower significant bits in the image.

The correlation-based technique (CBT) is a spatial embedding technique. It uses a seed key to generate a pseudo-random noise (PN) sequence that will be applied to the image or audio. The strength of the embedded noise is regulated by a gain factor, which increases robustness of the watermark. However, this reduces the quality of the watermarked copy. Although the advantage of this spatial technique is that it can be applied to any image or audio, it lacks the ability of using sub-sequent processing to improve resistance to tampering.

The frequency domain technique scheme uses the *discrete cosine transform* (DCT) to break up the image into a set of frequency bands. (33) The watermark is embedded into the coefficients of the middle frequency bands of an image. This provides the least distortion to the quality of the image.(2) This type of embedding is highly robust against compression and noise attacks.(51)

Wavelets have also been proven to be a more effective and robust scheme for watermarking multimedia.(37) Using wavelet transforms such as the *discrete wavelet transform* (DWT), the image can be decomposed into a set of sub-bands.(68) These sub-bands represent the approximation coefficients of the image, which can be combined with the watermark through additive embedding.(64) One of the main advantages of wavelet embedding is the ability to use higher energy watermarks in regions that are less sensitive to the human visual system (HVS). This allows a higher degree of robustness at little or no impact on quality.(40)

Additionally, any of these embedding schemes can take some advantages of using spread-spectrum sequences (15) or orthogonal codes (60) to generate the fingerprint as a pseudo-noise (PN) like sequence. These sequences make it harder for attackers to alter the embedded fingerprint. However, the issue with using spreading sequences is that they require a larger storage space to hold the larger spreaded fingerprint.

2.2 Phase II : Recovery

The fingerprint extraction process can be either blind, or non-blind. A non-blind approach requires the original artifact to recover the embedded fingerprint, while a blind approach does not. In multimedia forensics, the non-blind fingerprinting scheme is more practical because the unique fingerprint is embedded into the multimedia prior to distribution and it is stored by the content distributor.

If the LSB scheme was used on a fingerprinted copy, to perform recovery, the artifact would be analyzed and the lowest significant bits would be extracted to recreate the numerical sequences.(50; 57)

Using the correlation-based technique, the original seed key is required to generate the specific PN sequence.(51) This sequence is compared to the recovered sequence from the multimedia content. If the correlation value exceeds a certain threshold, the fingerprint is detected and matched. The accuracy of this technique is based on the selected threshold value and the percent of fingerprint recovery.

The frequency domain fingerprint extraction process is similar to its embedding process. The DCT is used to decompose the artifact in 8x8 blocks, to recover the middle frequency bands.(2) The coefficients representing these bands are extracted and the quantization table is analyzed to determine the fingerprint value for that block.(33) After all blocks are processed, the fingerprint has been recreated.

The wavelet watermarking process is similar to the recovery process. The DWT is used to decompose the artifact into its corresponding set of sub-bands.(68) These coefficients are then compared to the original non-fingerprinted coefficients to retrieve the difference in value.(64) This difference in value is the corresponding embedded fingerprint for the sub-band. This is processed for all sub-bands which may have an embedded fingerprint.(40)

If the fingerprint was embedded using spread-spectrum sequences, the extracted signal from the multimedia artifact would be despreaded to recreate the original watermark. Figure 5.4 shows how given a block of five values, the mean can be calculated to recreate the original non-spreaded fingerprint value. This is done for all blocks recovered from the colluded copy.

2.3 Phase III : Identification

Identification becomes feasible by employing fingerprint correlation methods and statistical classification techniques. Correlation can be computed between the recovered colluded fingerprint and the fingerprint of users that had received the original content.

Identification schemes can be categorized into three types: *independent fingerprint correlation*, *detector-oriented model* and *combinatorial (predictive) design*.

The independent fingerprint identification is a classical method which uses orthogonal modulation in order to generate fingerprints. A recovered fingerprint from a colluded copy would then be correlated using a set of matched filters with all possible known fingerprints in a data set. A problem with this methodology is that identification complexity increases substantially with larger number of users.

Judge and Ammar (26) use a hierarchical watermarking system called *WHIM*. Using watermark verification through intermediary nodes, the geographical location of the potential leakers or colluders can be approximated using WHIM.

Using a detector-oriented model requires the use of more than one multimedia watermarking scheme. The system uses a set of watermarks known as *detection keys*(30). If an attacker obtains a watermarked copy and a detection key, and uses the key to remove the embedded watermark, the attacker will actually be inserting a new fingerprint signal into the new copy. Therefore, the corresponding devices can be identified given a colluded copy generated with those detection keys. Segmentation and key compression are some related issues with using this scheme.

Some combinatorial detection schemes rely on direct pattern correlation of the colluded fingerprint to a combination of colluders (65). Some of these assume that the entire fingerprint was recoverable from the colluded copy.(59) In multimedia leaker identifications, Chu, Qiao, and Nahrstedt (10) proposed first generating the entire list of all possible leaker combinations (colluder sets) from the set of members receiving the content. When comparing the fingerprint recovered from the colluded copy, each combination of colluders is compared to the recovered fingerprint. Through the process of elimination, the corresponding colluders are potentially found.

A reference and summary of the notation used in this paper is provided in Table 2.1. Also, the terms watermark and fingerprint will have the same definition and be used interchangeably. Moreover, images will be used as examples for this research work, however, the solution can be applied to other types of multimedia formats.

Table 2.1 Notations used in this paper

Notation	Definition
Ψ	The original image (non-fingerprinted)
Ψ'	A colluded or fingerprinted copy
$\psi_\ell^y(i,j)$	ℓ-level coefficient at component (i,j) of the band-pass image Ψ of sub-band y
$f'(i,j)$	Fingerprint value of the colluded copy F' at component (i,j)
$f(i,j)$	Fingerprint or watermark sequence value at component (i,j)
R	Set of correlation values.
U	Set of users receiving a fingerprinted version of Ψ. $C \cup D = U$
C	Set of colluders involved in a collusion attack. $C \cap D = \emptyset$
D	Set of innocent individuals.
C'	Set of users identified as colluders. This set can contain both real colluders and innocent parties.
A	Set of all real colluders identified. $A \subset C'$, $A \subset C$ and $A \cap D = \emptyset$
B	Set of users identified as potentially innocent. $B = (D - C') \cup (C - C')$
Detection Rate	Also called true positive. It is defined as $\frac{\lvert A\rvert}{\lvert C\rvert}$
False Accusation Rate	Also called false positive rate. Defined as $\frac{\lvert C'-A\rvert}{\lvert D\rvert}$
Miss Rate	Also called false negative rate. Defined as $\frac{\lvert C-A\rvert}{\lvert C\rvert}$

CHAPTER 3. THREAT MODEL

Collusion attacks fall into two main categories: linear and non-linear collusion attacks.(65) A study of the effectiveness of these attacks was studied in (29; 71).

Linear collusion attacks typically synchronize C fingerprinted copies of a multimedia artifact and average out the signal to produce a new copy. This attack is shown as the *average attack* in Table 3.1. In some cases, colluders might use a variant of the average attack by adding a small amount of disturbance or Gaussian noise ε to increase the attenuation of the original fingerprint. Another attack involves colluders cutting out and pasting different portions of their copies to create a new one. This is also known as a *copy-and-paste attack.* The cut-and-paste attack is not studied in this research because it has the same effect as averaging the collusion, therefore, it becomes analogous to the average collusion attack.(65)

Non-linear collusion attacks are more of a statistical approach to attenuating or defeating the underlying fingerprinting scheme. In most cases, the minimum, maximum and median values of each of the C fingerprinted copies are observed and analyzed to create a new less traceable copy. The non-linear attacks in Table 3.1 are the *minimum*, *maximum*, *minmax* and *randomized negative* attacks.

Let $|C|$ out of $|U|$ total users collude so that $C = \{c_1, c_2, ...c_n\}$, where $n = |C|$. Next, let an image be represented in matrix form. Let $\psi'(i, j)$ represent the value of the component of a colluded image Ψ' at pixel location (i, j). Using $|C|$ copies, the component of $\psi'(i, j)$ is generated by combining components of all $c \in C$ using any of the attacks in Table 3.1.

$$c_1 = \begin{bmatrix} 2 & 1 \\ 3 & 8 \end{bmatrix}, c_2 = \begin{bmatrix} 4 & 2 \\ 5 & 4 \end{bmatrix} \tag{3.1}$$

An example of two colluders is presented in 3.1. Let c_1 and c_2 be image information

Table 3.1 Formulations of collusion attacks in this research

Attack	Formulation
Average	$\psi_x^{avg}(i,j) = \varepsilon + \sum_{n=1}^{\lvert K\rvert} \psi_x^{(n)}(i,j)/\lvert K\rvert$
Minimum	$\psi_x^{min}(i,j) = min(\left\{\psi_x^{(k)}(i,j)\right\}_{k\in K})$
Maximum	$\psi_x^{max}(i,j) = max(\left\{\psi_x^{(k)}(i,j)\right\}_{k\in K})$
Minimum/Maximum (MinMax)	$\psi_x^{minmax}(i,j) = \frac{1}{2}\left(\psi_x^{min}(i,j) + \psi_x^{max}(i,j)\right)$
Randomized Negative	$\psi_x^{randneg}(i,j) = \left\{ \begin{array}{c} \psi_x^{min}(i,j) \text{ with prob. } p \\ \\ \psi_x^{max}(i,j) \text{ with prob. } 1-p \end{array} \right\}$

(approximation coefficients) in matrix form of those two colluders. For presentation purposes, let the image sizes be 2-by-2. These two users will combine their watermarked copies to create a colluded copy Ψ'. Using the formulation in Table 3.1, each of these collusion attacks is described.

3.1 Average Attack

This attack takes the corresponding components of every colluder's copy and averages it to produce a new value. As an example, component of $\psi^{avg}(1,1) = \frac{c_1(1,1)+c_2(1,1)}{2} = 3$. Performing this with all the components of c_1 and c_2 a colluded copy Ψ'^{avg} is obtained.

$$\Psi'^{avg} = \begin{bmatrix} 3 & 1 \\ 4 & 6 \end{bmatrix} \tag{3.2}$$

This attack, though simple in implementation, may sometimes yield the best effect against fingerprinted multimedia. The average attack acts as a signal normalizer. As the number of colluded signals being used increases, this weakens every single fingerprint involved in the

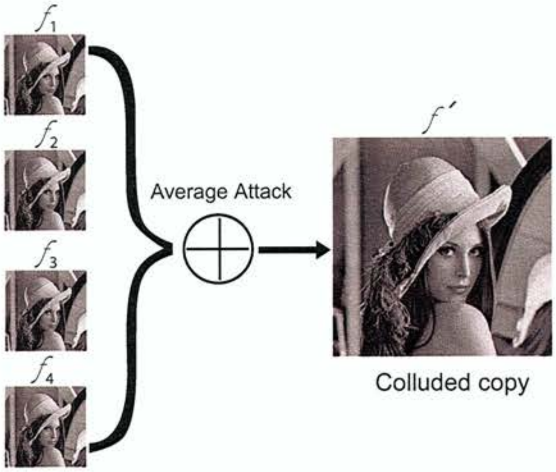

Figure 3.1 Colluded image using average attack

collusion and can in turn normalize the signal of the multimedia artifact to be more similar to the original, non-fingerprinted artifact. Therefore, the average attack can produce a colluded copy that can possibly have better perceptual quality than any fingerprinted signal.

3.2 Minimum Attack

This attack takes the corresponding minimum components of the C fingerprinted copies used in the attack. In the example used, component $\psi^{min}(1,1)$ would be calculated by $min(c_1(1,1), c_2(1,1))$. Performing this for all components of c_1 and c_2, Ψ'^{min} is generated.

$$\Psi'^{min} = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \tag{3.3}$$

3.3 Maximum Attack

This attack takes the corresponding maximum components of the C fingerprinted copies used in the attack. In the example used, component $\psi^{max}(1,1)$ would be calculated by $max(c_1(1,1), c_2(1,1))$. Performing this for all components of c_1 and c_2, Ψ'^{min} is generated.

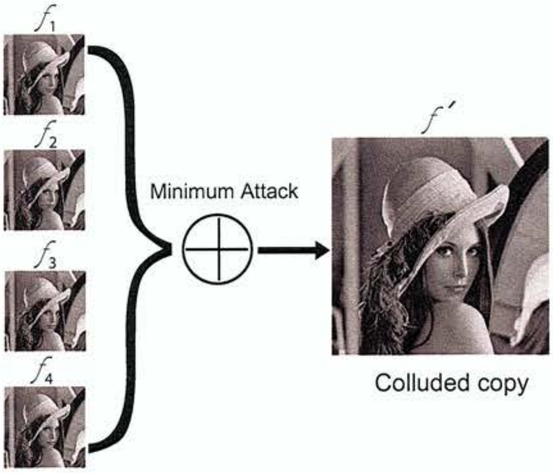

Figure 3.2 Colluded image using minimum attack

$$\Psi'^{max} = \begin{bmatrix} 4 & 2 \\ 5 & 8 \end{bmatrix} \tag{3.4}$$

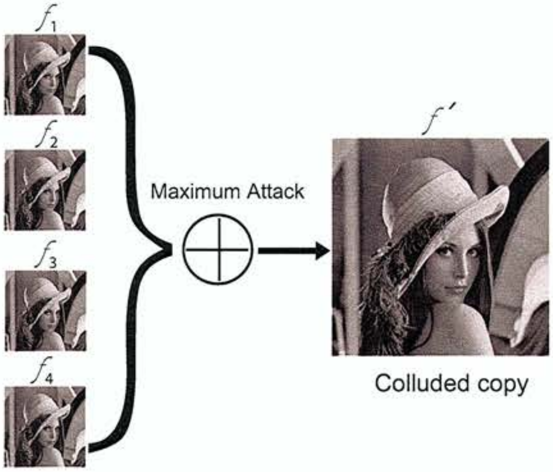

Figure 3.3 Colluded image using maximum attack

This attack is highly similar to the minimum attack previously described. It is a variation where the maximum components are only used for the final colluded copy. Therefore, similar effects are seen on perceptual quality.

3.4 MinMax Attack

In this attack, the average of the minimum and maximum values of the corresponding C copies are used to produce the colluded copy. For this attack, component $\psi^{minmax}(1,1)$ is calculated by the average $\frac{\psi^{min}(1,1)+\psi^{max}(1,1)}{2}$. Performing this for all components of c_1 and c_2, Ψ'^{minmax} is generated.

$$\Psi'^{minmax} = \begin{bmatrix} 3 & 1 \\ 4 & 6 \end{bmatrix} \tag{3.5}$$

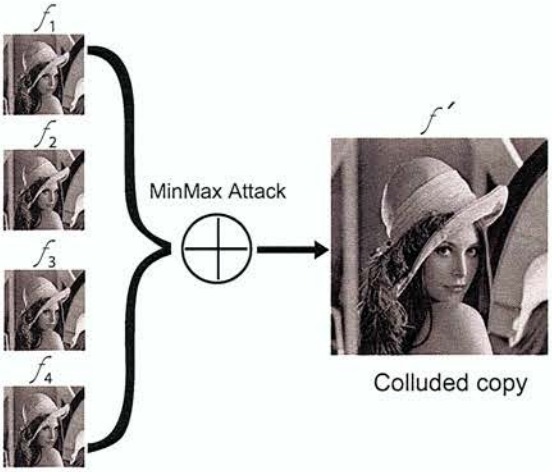

Figure 3.4 Colluded image using minimum/maximum (MinMax) attack

This attack becomes a combination of the minimum or maximum attack paired with the average attack.

3.5 Randomized Negative Attack

The values of each of the components in the colluded copy will take either the minimum or maximum values of the C fingerprinted copies. The value in a component of the colluded copy, such as $\psi^{randneg}(1,1)$ will be set to the minimum value $\psi^{min}(1,1)$ with probability p, otherwise it is set to $\psi^{max}(1,1)$ with probability $(1-p)$. For this study, $p = 0.5$. Assume

ψ^{min} was chosen for $\psi(1,1)$ and $\psi(2,2)$, and ψ^{max} for the other components. The resulting $\Psi'^{randneg}$ shown in the following 3.6 is only 1 of 16 possible colluded results using this attack.

$$\Psi'^{randneg} = \begin{bmatrix} \psi^{min}(1,1) & \psi^{max}(1,2) \\ \psi^{max}(2,1) & \psi^{min}(2,2) \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 5 & 8 \end{bmatrix} \tag{3.6}$$

This attack adds the largest amount of perceptual noise to the final colluded image. The use of four fingerprinted copies in Figure 3.5 are used to generate the colluded copy. The additional noise is seen in the final colluded copy.

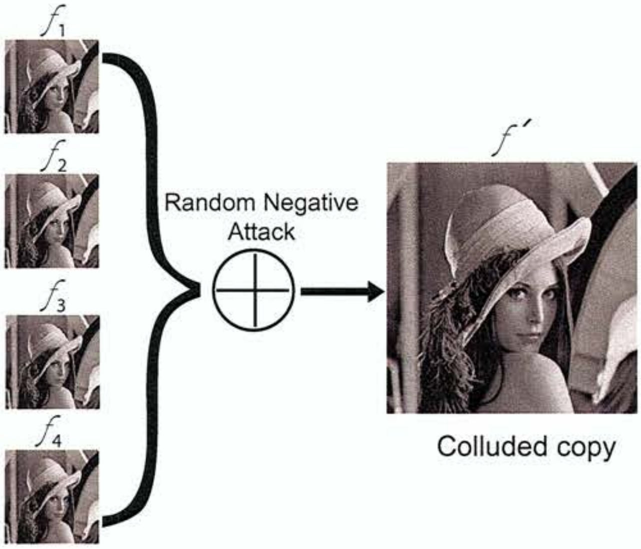

Figure 3.5 Colluded image using the random negative attack

CHAPTER 4. PROBLEM DEFINITION

There is a growing threat by users who leak multimedia artifacts by using collusion attacks. This type of unauthorized proliferation costs the digital entertainment industry billions of dollars every year.(21; 47) The global nature of the internet and the numerous content distribution applications such as file transfer protocol (FTP), peer-to-peer (P2P) file sharing, and pirate-software download sites such as Warez(63) have made it easier for unauthorized users to receive multimedia content. Digital forensic investigators require a efficient means to identify potential leakers of multimedia artifacts to stop future proliferation by those individuals.

Tracing unauthorized content distributors has become an increasing concern in the media industry. According to the Record Industry Association of America, piracy costs the music industry alone around *$4.2 billion* dollars each year.(47) The Motion Picture Association of America esitmates that unauthorized duplication and distribution of movies will cost them an estimated *$15 billion* over the next 4 years.(21).

Although encryption has been used to safeguard the content while transmission, it does not offer protection against further unauthorized distribution and duplication. Fingerprints were developed to determine the possible leakers of the multimedia content. These fingerprints are unique for every copy legally distributed of the same multimedia artifact. Due to the highly connected digital world, these fingerprints not only have to be robust against a single attacker, but new multi-user attacks known as *collusion attacks*. A collusion attack is where more than one adversary combines their copy of the multimedia artifact with other multiple marked copies to produce a new distinctly unmarked version so that the colluders become untraceable.

These attacks come in two categories: linear and non-linear collusion attacks. Linear collusion attacks typically synchronize C fingerprinted copies of a multimedia artifact and average out the signal to produce a new copy. Non-linear collusion attacks are more of a statistical approach to attenuating or defeating the underlying fingerprinting scheme.

The figures in 3.1, 3.2, 3.3, 3.4 and 3.5 seem similar to the human visual system (HVS). However, each one of those colluded images was generated using a different collusion attack.

Some colluder identification schemes try to identify colluders by performing using a correlation threshold between each user and the recovered colluded fingerprint. However, for each case, the correlation threshold needs to be modified for each given attack in order to account for fingerprint attenuation.

Other identification schemes use combinatorial schemes where all possible sets of colluder combinations are created. The fingerprint of these sets are combined for every attack type and correlated against the recovered fingerprint. Although it is effective, it requires a large amount of complexity and bookkeeping when dealing with a large number of colluders.

Forensic investigators require a robust framework for identifying the potential colluders involved in a collusion attack against multimedia content. In addition they require to minimize the probability of missing potential suspects and falsely accusing the innocent. Investigators require the ability to traceback to potential colluders without knowing the number of colluders involved in an attack or the attack used.

The main objective in this research is to identify the set of colluders involved in the generation of a given colluded copy without knowing the total number of colluders involved and the type of collusion attack used.

Let $C \subset U$, where C is the set of users out of all users in U that decide to collude to generate a colluded copy Ψ' from Ψ. Let $C\prime$ be the set of users from U that were identified as probable colluders by any colluder identification scheme. The objective of this research is to propose a fingerprinting and identification scheme using wavelet watermarking and statistical clustering such that $C' \approx C$ while minimizing $\max(|C' - C|, |C - C'|)$.

CHAPTER 5. PROPOSED FINGERPRINTING AND IDENTIFICATION SOLUTION

5.1 Phase I : Embedding Process

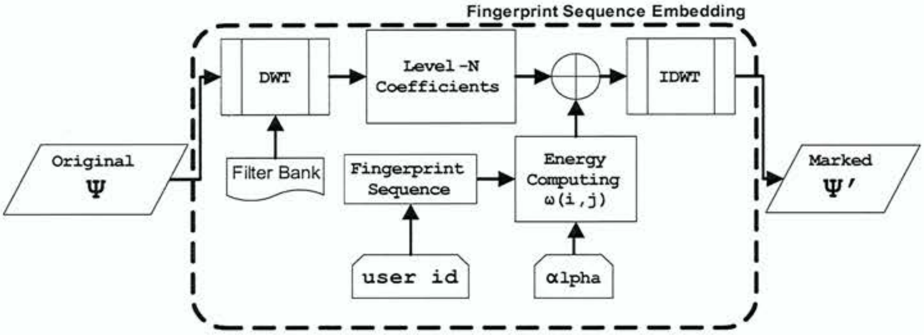

Figure 5.1 Embedding process using DWT

Multimedia artifacts can be represented as discrete signals. As an example, an image can be represented as a matrix where each pixel location $\{i, j\}$ represents a given color value. This property enables the use of the *Discrete Wavelet Transform* (DWT) to easily embed fingerprints in image data.(16) The DWT uses a decomposition process to embed fingerprint coefficients.(13; 32) This is done using band-pass arrays called *filter banks.* A filter bank is a series of high-pass and low-pass filters which partitions the original signal into several components called *sub-bands.* These sub-bands can then be recombined to recreate the original signal. The decomposition process can be repeated to more than one level of decomposition because the wavelet transform is recursive in nature. At each level, the filter bank passes the input through a high-pass filter, $h[\psi]$, which provides the detail coefficients, and low-pass filter, $g[\psi]$, which provide the approximation coefficients.

Figure 5.2 illustrates an ℓ-level decomposition tree using a filter bank. At every level in the tree, the input is decomposed into low and high frequencies using the filters. Figure 5.3 presents the idea behind recursive decomposition of an image using four levels.

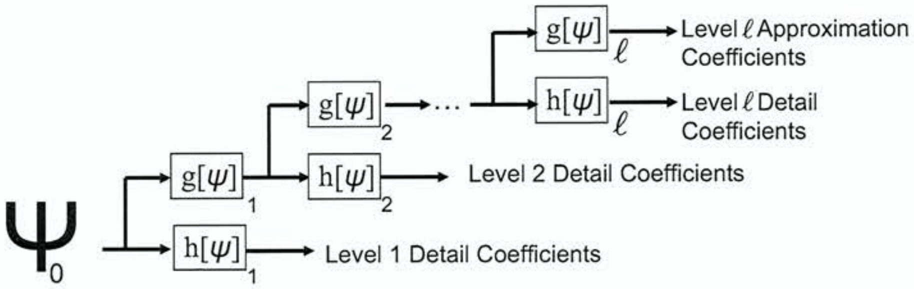
g[ψ]
ℓ
Level ℓ Approximation Coefficients
g[ψ]
2
h[ψ]
ℓ
Level ℓ Detail Coefficients
g[ψ]
1
h[ψ]
2
Level 2 Detail Coefficients
Ψ
0
h[ψ]
1
Level 1 Detail Coefficients

Figure 5.2 ℓ-Level decomposition tree of a filter bank

Even though there are various robust wavelet-based watermarking methods (40), the focus of this research is to use the *constant energy embedding* because it requires the least amount of computation. This technique is usually chosen as a baseline for comparative studies.(64)

$$\psi'_\ell(i,j) = \psi_\ell(i,j) + \alpha \cdot f(i,j) \tag{5.1}$$

Embedding is performed by processing the multimedia artifact with the DWT using ℓ-levels of decomposition. After extracting the corresponding approximation coefficients, an additive embedding of the fingerprint is performed using the constant energy embedding scheme shown in Equation 5.1. After the fingerprint has been embedded, the inverse DWT is performed to recreate the original artifact with the embedded fingerprint.

Let $\psi_\ell(i,j)$ be the component of the original image Ψ at location $\{i,j\}$. Let α be a global energy parameter that determines the fingerprint strength. Let f be the pre-computed fingerprint sequence. Let ℓ specify the decomposition level of the coefficients used to embed the fingerprint. The experimentation results found that $\alpha = 0.1$ and $\ell = 4$ provide enough fingerprinting strength while providing acceptable distortion. A general example is presented using 5.2 and 5.3.

$$\Psi_0 = \begin{bmatrix} 12 & 23 \\ 34 & 45 \end{bmatrix}, f = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \alpha = 2 \tag{5.2}$$

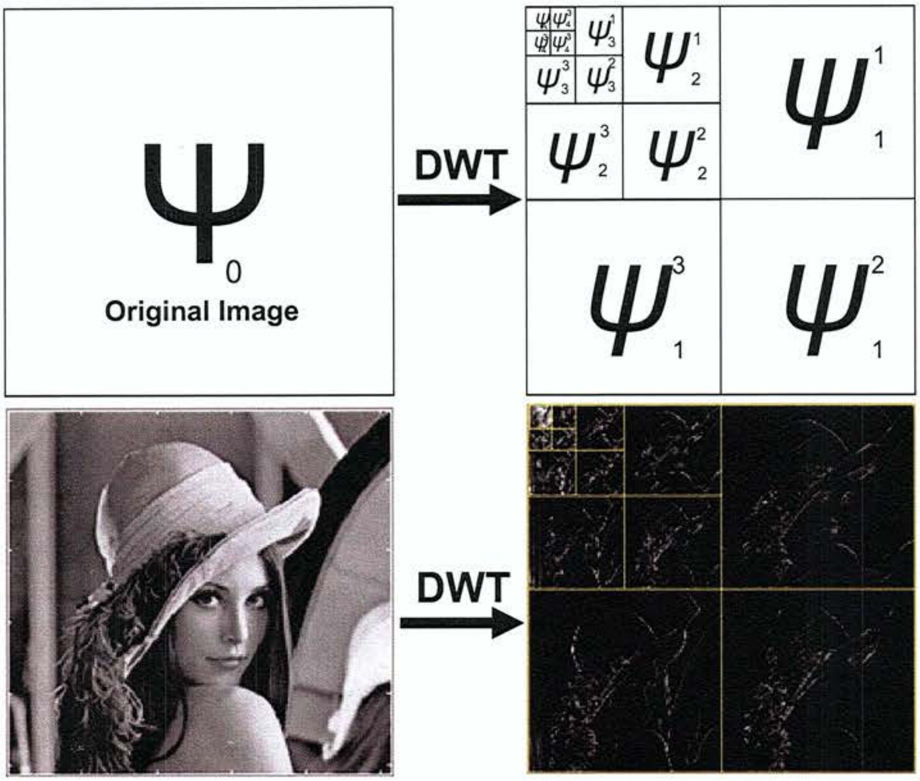

Figure 5.3 Four Level Image Decomposition. Every ψ_l^n is a sub-band

$$\Psi' = \begin{bmatrix} 12 & 23 \\ 34 & 45 \end{bmatrix} + 2 \cdot \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 14 & 23 \\ 36 & 47 \end{bmatrix} \tag{5.3}$$

In addition, spread-spectrum sequences (15) or orthogonal codes (60) can be used to generate the fingerprint f. Figure 5.4 demonstrates how a fingerprint sequence of values can be spread. In the figure, the first value [1] is spread into $[1, 1, 0, 1, 0]$. The second value [0] would be spread as $[0, 1, 0, 0, 1]$. This would be performed for all the values in the fingerprint sequence.

Spreading a fingerprint causes the requirement of additional storage space. In Figure 5.4, every value is spread into a set of five values. In situations where the original multimedia data cannot increase in size, the length of the original fingerprint must be reduced so that the spreaded version is able to fit.

A summary of the embedding process is provided by Algorithm 1.

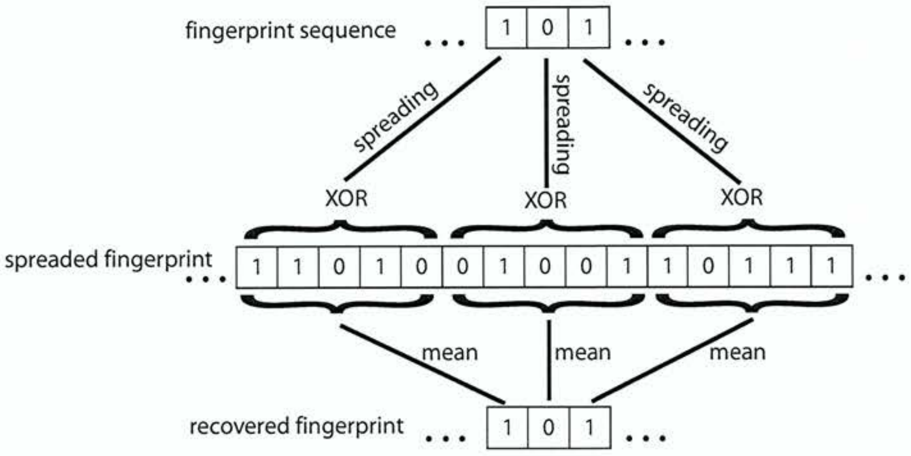

Figure 5.4 Example of spreading a fingerprint sequence

Algorithm 1 $embed(\Psi, f, \alpha, \ell)$

```
$\psi \leftarrow DWT(\Psi, \ell)$
if $size(\psi) \leq size(f)$ then
    return "error: fingerprint too large"
end if
for $i \leftarrow 1$ to $rows[\psi]$ do
    for $j \leftarrow 1$ to $columns[\psi]$ do
        $\psi'_\ell(i,j) = \psi_\ell(i,j) + \alpha \cdot f(i,j)$
    end for
end for
$\Psi' \leftarrow iDWT(\psi', \ell)$
return $\Psi'$
```

5.2 Phase II : Recovery Process

The non-blind extraction process is similar to the embedding process. First, both the original and fingerprinted artifacts are processed with the DWT to extract the approximation coefficients. Next, the difference between these coefficients is calculated using Equation 5.4 where the recovered fingerprint is f'. The fingerprint recovery process is shown in Figure 5.5. The recovery example from the previous embedding process is presented in Equation 5.5. The procedure for recovery is presented in Algorithm 2.

$$f'(i,j) = \frac{1}{\alpha} \cdot (\psi'_\ell(i,j) - \psi_\ell(i,j)) \tag{5.4}$$

$$f' = \frac{1}{2} \cdot \left(\begin{bmatrix} 14 & 23 \\ 36 & 47 \end{bmatrix} - \begin{bmatrix} 12 & 23 \\ 34 & 45 \end{bmatrix} \right) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{5.5}$$

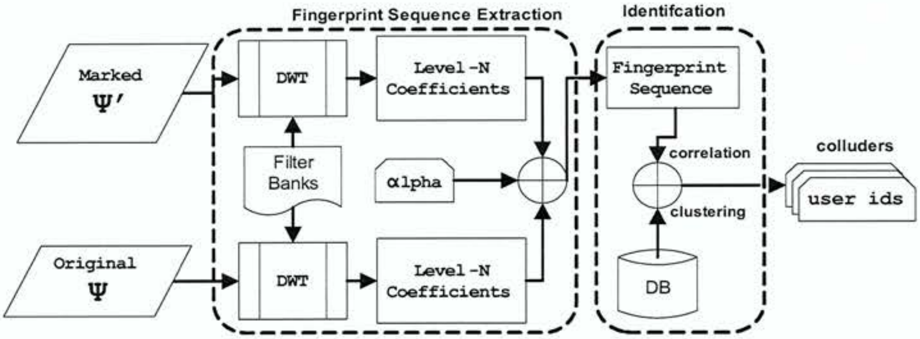

Figure 5.5 Extraction and Correlation process using DWT

Algorithm 2 $recover(\Psi', \Psi, f, \alpha, \ell)$

```
$\psi \leftarrow DWT(\Psi, \ell)$
$\psi' \leftarrow DWT(\Psi', \ell)$
**for** $i \leftarrow 1$ to $rows[\psi']$ **do**
    **for** $j \leftarrow 1$ to $columns[\psi']$ **do**
        $f'(i,j) = \frac{1}{\alpha} \cdot (\psi'_\ell(i,j) - \psi_\ell(i,j))$
    **end for**
**end for**
return $f'$
```

5.3 Phase III : Colluder Identification

After recovering the colluded fingerprint f', the correlation coefficient is calculated between two fingerprints using Equation 5.6, where f is the corresponding fingerprint of a user from a known database. Let $R(i)$ be the correlation value between f' and f_i for user i. The set R contains all of the correlation values between all users and the colluded fingerprint. The correlation value is between -1 and 1.

$$R(i) = corr(f', f_i) = \left| \frac{\sum\sum_{i,j}(f'(i,j) - \overline{f'})(f_i(i,j) - \overline{f_i})}{\sqrt{(\sum\sum_{i,j}(f'(i,j) - \overline{f'})^2)(\sum\sum_{i,j}(f_i(i,j) - \overline{f_i})^2)}} \right| \tag{5.6}$$

After having the set of correlation values R, and their corresponding users, a statistical clustering technique can be used to determine the colluders involved in the attack. The collusion attack and the number of colluders involved is unknown, therefore, the value of a possible *correlation threshold* parameter cannot be determined. The use of a correlation threshold value is not effective.

This research proposes the use of an iterative *2-means* clustering procedure to find possible partitions of colluders. This procedure is adopted from the k-means algorithm.(39) The goal of the algorithm is to classify the correlation values within 2 number of clusters. One cluster will contain the set of detected colluders, and the other, the set of innocent parties. Since it is known that a set of higher correlation values indicates a stronger relationship with the colluded fingerprint, the cluster with the highest mean value will be considered the colluder set C'.

The clusters are partitioned by minimizing the Euclidean distance between every correlation value $R(i)$ and the means of the its cluster called the *centroid.* In the proposed variant, the initial centroids are not selected randomly as other variations, but are calculated based on the *mean* and *standard deviation* of the set R. Two centroids, b and c will be set to $mean(R) \pm stddev(R)$, where $c > b$.

The scheme places two centroids over the entire data set R. At every iteration, a user i is assigned a group (C' or B) based on the shortest distance between $R(i)$ and one of the centroids. After all the users have been assigned to a group, the locations of the centroids are recalculated based on the mean of members of each of the corresponding groups. This entire process is repeated until the locations of the centroids do not change. The final result is the set C' which contains the set of possible colluders involved in a collusion attack to form f'.

The summary of the *2-means* algorithm is presented in Algorithm 3.

A simple working example is shown in Figure 5.6 to present the iterative process. For this example, the set R contains points for users $\{1, 2, 3, 4\}$, so that $R(1)$ is the correlation value

Algorithm 3 $2means(R)$

```
$\overline{c} \leftarrow mean(R) + stddev(R)$
$\overline{b} \leftarrow mean(R) - stddev(R)$
repeat
    $C' \leftarrow B \leftarrow \emptyset$
    for $i \leftarrow 1$ to $|R|$ do
        if $|R(i) - \overline{c}| = min(|R(f) - \overline{c}|, |R(i) - \overline{b}|)$ then
            Assign $R(i)$ to set $C'$
        else
            Assign $R(i)$ to set $B$
        end if
    end for
    $\overline{c}_{last} \leftarrow \overline{c}$
    $\overline{c} \leftarrow mean(C')$
    $\overline{b}_{last} \leftarrow \overline{b}$
    $\overline{b} \leftarrow mean(B)$
until $max(|\overline{c} - \overline{c}_{last}|, |\overline{b} - \overline{b}_{last}|) = 0$
return $C'$
```

Table 5.1 Calculations of first iteration seen in Figure 5.6-(a)

	$R(1) = 0.25$	$R(2) = 0.40$	$R(3) = 0.65$	$R(4) = 0.85$
$\overline{b} = 0.20$	0.05	0.20	0.45	0.65
$\overline{c} = 0.35$	0.10	0.05	0.30	0.50
Group Assignment	B	C'	C'	C'

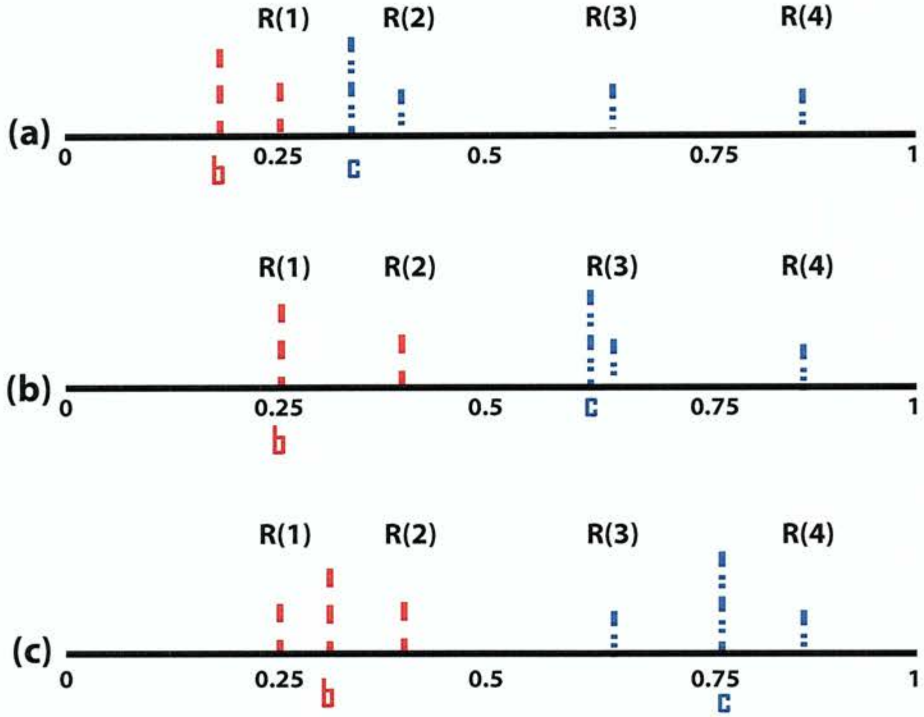

Figure 5.6 2-means algorithm example: (a) Algorithm after first iteration and group assignments based on Table 5.1 (b) Algorithm after second iteration and group assignments based on Table 5.2 (c) Algorithm after third iteration and final group assignments based on Table 5.3

of user 1. The initial values of the centroids $\bar{b}$ and $\bar{c}$, for demonstration purposes, are 0.20 and 0.35 respectively.

Through the first iteration of the algorithm, every point in R is assigned to the group based on the smallest distance to its centroid. Table 5.1, shows the calculations and the group assignments after the first iteration.

After the initial group assignments, the values of the centroids are recalculated to be the mean of the members of the group. Therefore, $\bar{b} = 0.25$ and $\bar{c} = 0.63$. The process is again repeated for the second iteration using the new centroids. The locations of the new centroids are shown in Figure 5.6 part (b). The calculations are presented in Table 5.2. At the end of the second iteration, it is seen that $R(2)$ has moved from being assigned to C' to B. Again, the new centroids are calculated and the process repeated.

Table 5.2 Calculations of second iteration seen in Figure 5.6-(b)

	$R(1) = 0.25$	$R(2) = 0.40$	$R(3) = 0.65$	$R(4) = 0.85$
$\overline{b} = 0.25$	0	0.15	0.40	0.60
$\overline{c} = 0.63$	0.38	0.23	0.02	0.22
Group Assignment	B	B	C'	C'

Table 5.3 Calculations of third iteration seen in Figure 5.6-(c)

	$R(1) = 0.25$	$R(2) = 0.40$	$R(3) = 0.65$	$R(4) = 0.85$
$\overline{b} = 0.325$	0.075	0.075	0.325	0.525
$\overline{c} = 0.75$	0.5	0.35	0.1	0.1
Group Assignment	B	B	C'	C'

At the end of the third iteration, the algorithm notices that the locations of the centroids do not change. Therefore, the algorithm terminates and set C' contains the potential set of colluders because $\overline{c} > \overline{b}$.

Additionally, the final location of centroids b and c identify the joint density of each cluster. As $|c - b| \gg 0$, the more distance between the two cluster exists. The better the cluster separation, the clearer it is to identify the colluders from the user set. A measurement of cluster separation is the *silhouette coefficient.*(27) The silhouette coefficient measures the cohesion and separation between identified clusters in a data set using individual points. The centroids will be used to determine how well the clusters are separated. Let a be the average distance of a centroid to the points in its cluster. Let b be defined as the distance of a centroid to points in another cluster. The silhouette coefficient is given by $SC = 1 - \frac{a}{b}$. The typical value of SC is between 0 and 1, where a value closer to one is preferred.

If the value of SC is closer to 0, there is a possibility of there only being 1 cluster in the data. This cluster would contain either all colluders or all innocent parties. If a forensic investigator knows that the user set used in analysis is the entire possible set of users, then he can assume the entire set is made up of colluders.

The algorithm is successful in determining the colluder set C' because colluders cannot determine the value of the embedded fingerprint in their multimedia artifact. Therefore, they cannot successfully determine which set users to form C such that, $corr(f', f_i) = 1$, for an innocent user u_i.

Furthermore, the algorithm is highly practical because it treats the correlation values in R as random variables, and finds potential relationships based on joint density. Therefore, ***the colluder set is built from observations and not from predictions.*** Furthermore, the identification of sets result in less computation because all possible colluder combinations do not have to be tested.

CHAPTER 6. EVALUATION AND RESULTS

6.1 Experimental Setup

We define a *skew pattern* as the type of distribution colluders exhibit in relation to the entire set of users. The solution will be tested against three data set skew types. In some instances, a digital forensic investigator may not have the entire set of users who had received the original file. It is important to understand the effects of data set skew patterns against the proposed solution.

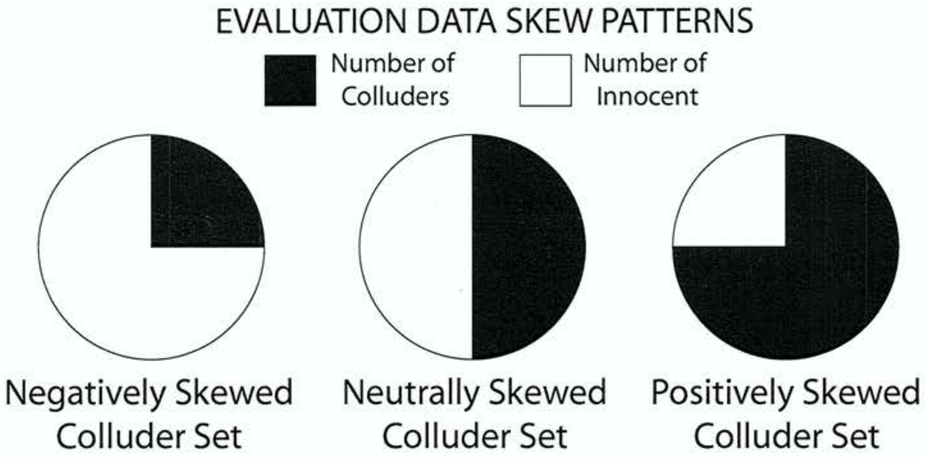

Figure 6.1 Types of data set distributions (skews) of colluders

A *neutrally skewed colluder data set* will be defined as a set where nearly half of the users in the entire data set colluded to generate a colluded copy with a specific collusion attack.

The most common skew type is a *negatively skewed colluder data set.* This will be defined as a set where less than 25% of the users were involved in a collusion attack. A few colluders, in comparison to the entire set of users receiving the multimedia artifact, will collude to generate a new copy. The negatively skewed set will be the closest to a real world scenario where a popular multimedia artifact is purchased by many, but yet a few users decide to conspire.

A *positively skewed colluder data set* will be defined as a set where 75% or more users in a data set were involved in a collusion attack. This is the least likely type of scenario, where most users who obtained the multimedia artifact, conspire to distribute a colluded copy.

Figure 6.1 shows how the set is partitioned for each *skew* type given that the entire user space is contained in the circle.

The evaluation will implemented in two versions: with and without spread spectrum watermarking. The spread spectrum watermarking will be implemented using the pseudo-noise sequence generator provided in MatLab. Both schemes will be tested independently for each skew type. The fingerprints will be a sequence of zero-mean pseudo-randomly generated Gaussian distributed values. Attacks listed in Table 3.1 will be used for evaluation against the scheme.

The solution is evaluated using the *Lena.jpg* image(34) and MatLab 7.0 software. A total set of 400 fingerprinted copies of the image will be created and embedded with the fingerprints using the constant energy embedding technique using the Daubechies-6 filter. A total of 200 colluders will be in set C and a total of 200 innocent individuals will be in set D. A colluded copy is generated when two or more colluders from C join using an attack from Table 3.1. Four levels of decomposition will be used for embedding. The value of α will be set to 0.10.

After generating the set of colluded copies, our 2-means algorithm will be ran against attacks starting from five colluders and increasing the size. These sets will be compared to the true set of colluders to determine our accuracy for each attack type and for each type of skew pattern.

The miss rate and the false accusation rates will be collected from our results for each skew pattern. The identification rate will not be presented in the results. It can be calculated for any given number of colluders in the results by the equation: *1 - Miss Rate.*

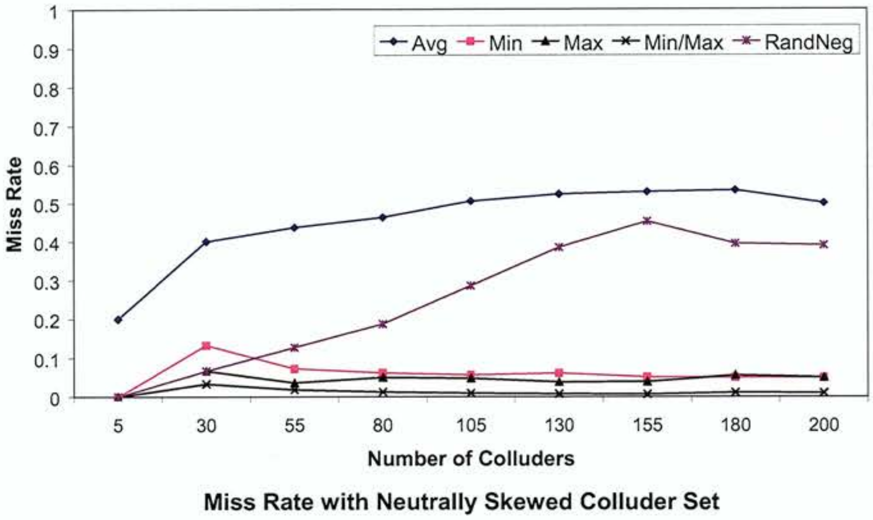

Figure 6.2 Results of the miss rate with a neutrally skewed colluder data set

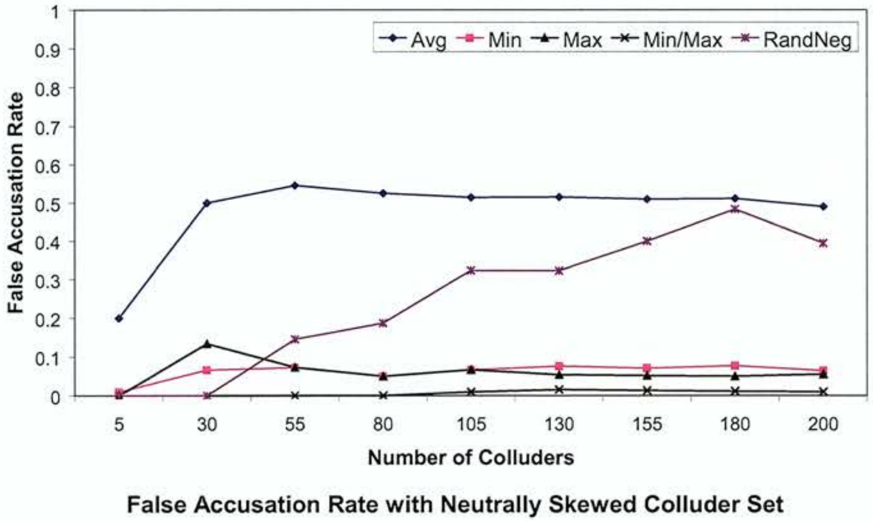

Figure 6.3 Results of false accusations with a neutrally skewed colluder data set

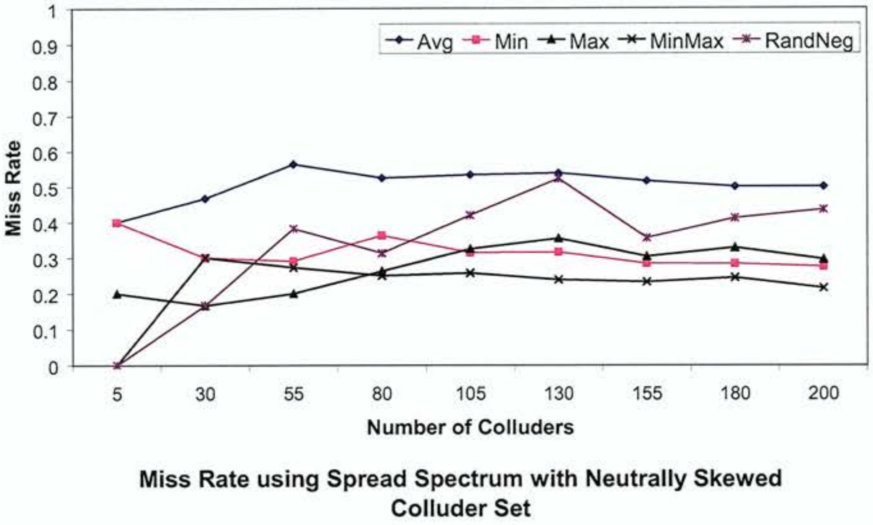

Figure 6.4 Results of miss rate using spread spectrum with a neutrally skewed colluder data set

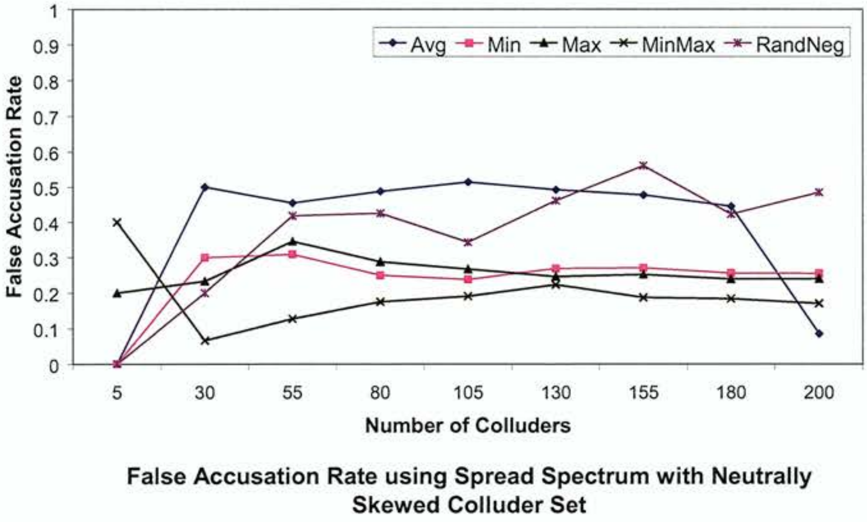

Figure 6.5 Results of false accusations using spread spectrum with a neutrally skewed colluder data set

6.2 Neutrally skewed set evaluation

Figure 6.2 presents the results of the miss rates for a neutrally skewed colluder data set. It is seen from the results that the solution works well in identifying colluders involved in a minimum, maximum and minmax attacks. As the number of colluders increase, less than 10% of colluders are missed by the algorithm on those attacks. The solution works well against the random negative attack where the colluder set size is less or equal to 100 so that we only miss 1/3 of the colluders involved in an attack. This provides a better than 70% probability that a person identified was involved in a random negative attack. When the number of colluders is higher, the rate begins to degrade and flatten out. The average attack is the strongest against our solution.

The performance of false accusation rates is seen in Figure 6.3. Minimum, maximum and minmax attacks are thwarted by the use of the proposed method, regardless of the number of colluders involved. The scheme still provides sufficient protection for innocent parties against the randomized negative attack. However, the random negative and average attack are the strongest in generating false accusations with the solution in this equally weighted data set.

When implementing spread spectrum fingerprinting, our solution degrades its performance. Both the miss and false accusation rates increase in a neutrally colluded set with the use of spread spectrum. This effect is seen in Figures 6.4 and 6.5.

6.3 Negatively skewed set evaluation

The miss rates in a negatively skewed data set, where the number of colluders is much less than the number of individuals in the entire set of users receiving the file, performs similarly to the neutrally skewed data set. The minimum, maximum and minmax attacks are thwarted by identifying most of the colluders (greater 89% of colluders) regardless of the number of colluders involved. This is seen in Figure 6.6.

It is suspect to suggest that due to the larger amounts of innocent individuals in a negatively skewed set, a higher false accusation rate is expected. Interestingly in Figure 6.7, there is a difference in the performance of protecting innocent individuals in a negatively skewed set. The

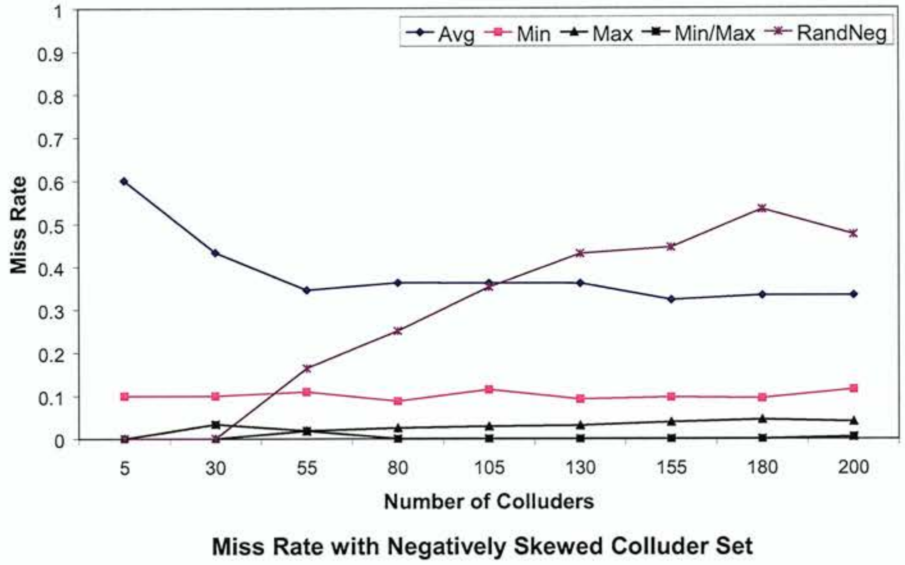

Figure 6.6 Results of the miss rate with a negatively skewed colluder data set

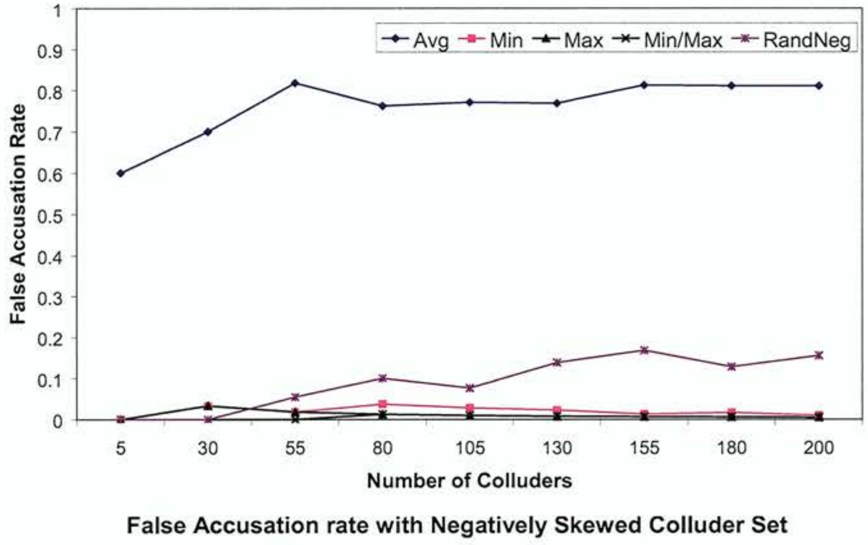

Figure 6.7 Results of false accusations with a negatively skewed colluder data set

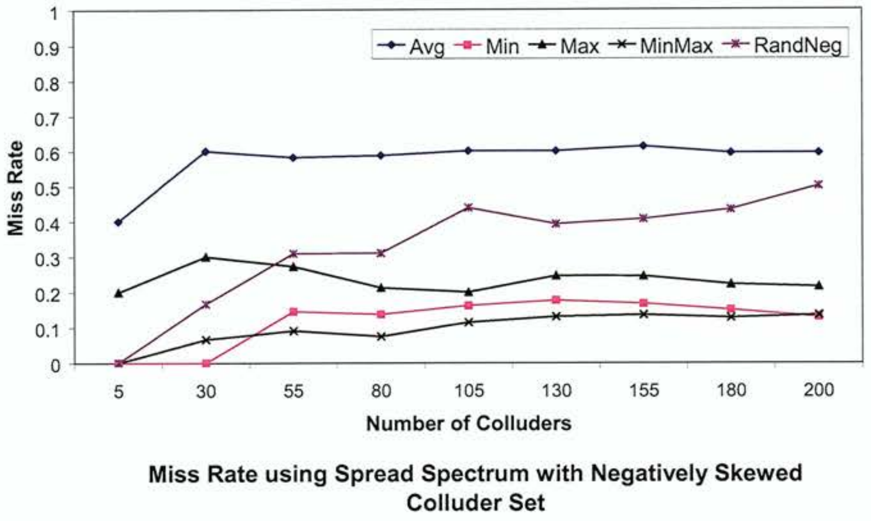

Figure 6.8 Results of miss rate using spread spectrum with a negatively skewed colluder data set

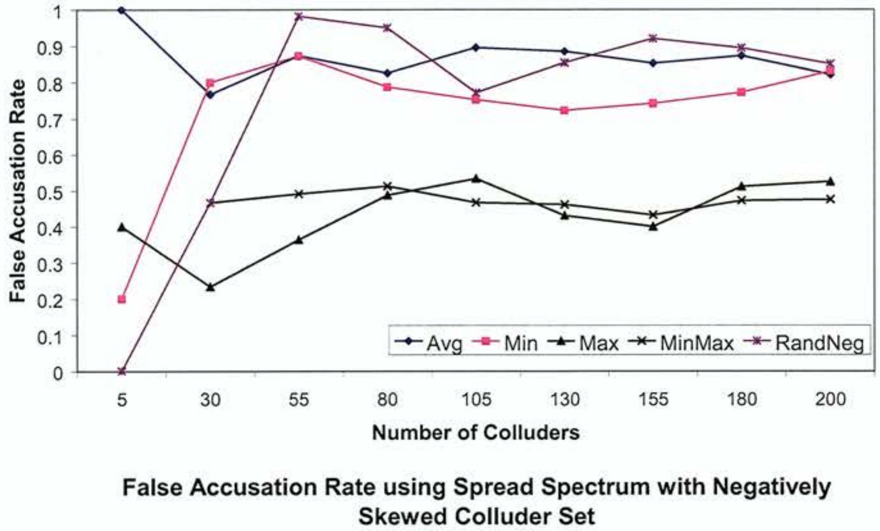

Figure 6.9 Results of false accusations using spread spectrum with a negatively skewed colluder data set

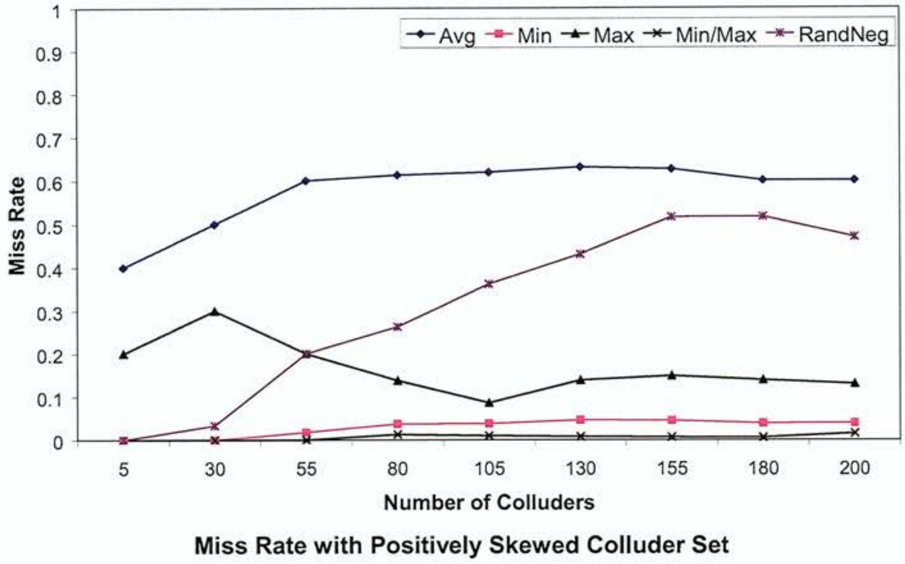

Figure 6.10 Results of the miss rate with a positively skewed colluder data set

solution seems to work effectively against minimum, maximum, minmax and random negative attacks in comparison with the neutrally skewed data set. Again, it is seen that the average attack is the strongest against the solution.

The spread spectrum implementation of our solution undesirably reduces the identification performance, as seen in Figure 6.8. Moreover, as seen in Figure 6.9, spread spectrum substantially increases in accusing almost all of the innocent individuals in the user set. It is clearly seen that spread spectrum has not added any significant value to our proposed solution.

6.4 Positively skewed set evaluation

Figure 6.10 shows the miss rate performance against a positively skewed colluder set distribution. Figure 6.10 shows, that a similar performance is achieved as the neutrally skewed data set, however, there is some adverse effects against the maximum attack. The miss rate for this attack is increased almost 15% compared to the same attack in the other skew patterns.

The solution works effectively well against all the attacks in protecting innocent individuals.

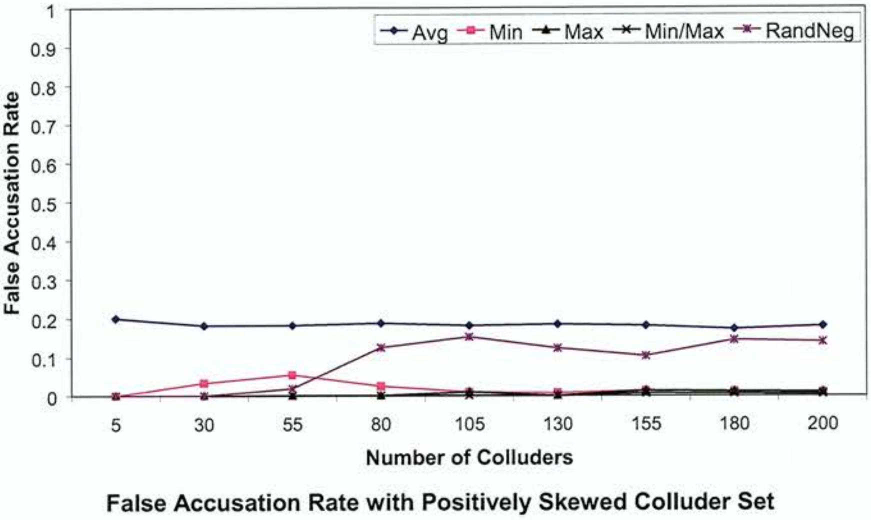

Figure 6.11 Results of false accusations with a positively skewed colluder data set

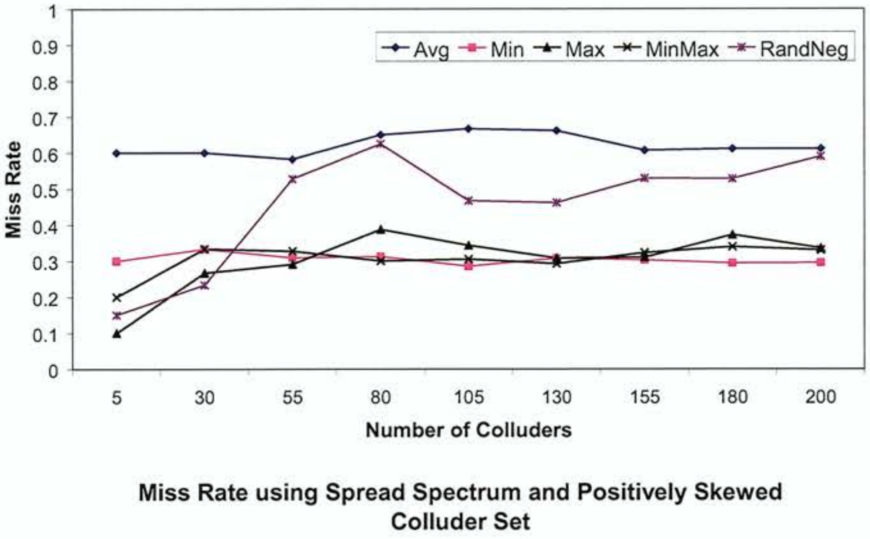

Figure 6.12 Results of miss rate using spread spectrum with a positively skewed colluder data set

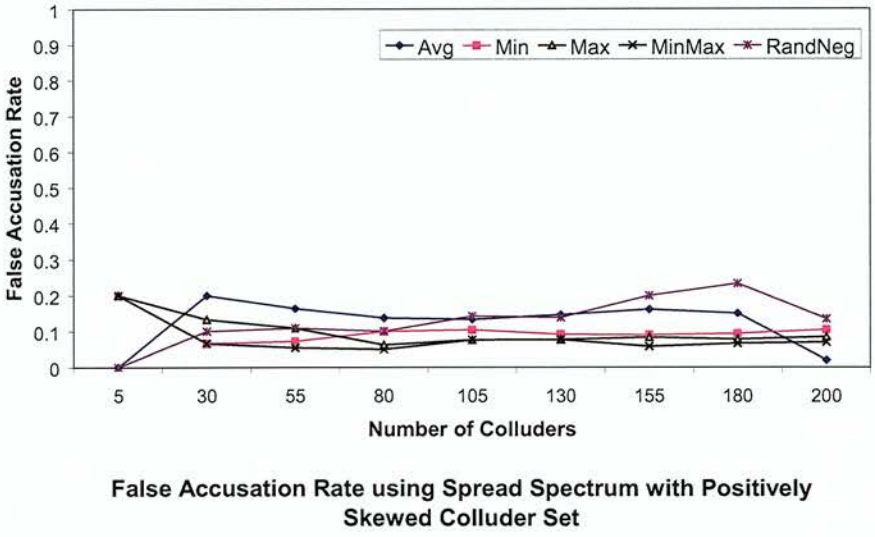

Figure 6.13 Results of false accusations using spread spectrum with a positively skewed colluder data set

Figure 6.11 shows that the false accusation rate with the average attack is attenuated in comparison to the other set distributions. It is also seen that the false accusation rates for the other attacks (minimum, maximum, minmax and random negative) have been mitigated with a large number of colluders.

Including spread spectrum watermarking in our implementation does not seem to add any improvement in detection or reduction in false accusations. In general, it seems that the addition of spread spectrum to the solution causes it to degrade in performance independent of the skew type.

6.5 Continuous Increasing Distribution

The last analysis will use a fixed user set count while increasing the colluders in that set. Let C be the set of real colluders, and D the set of innocent individuals, out of a total user set U. Given a colluder size $|C|$, a distribution set U is generated where $|U| = |C| + |D|$, where $D = U - C$. The experiment starts with a small C and increased sequentially while updating

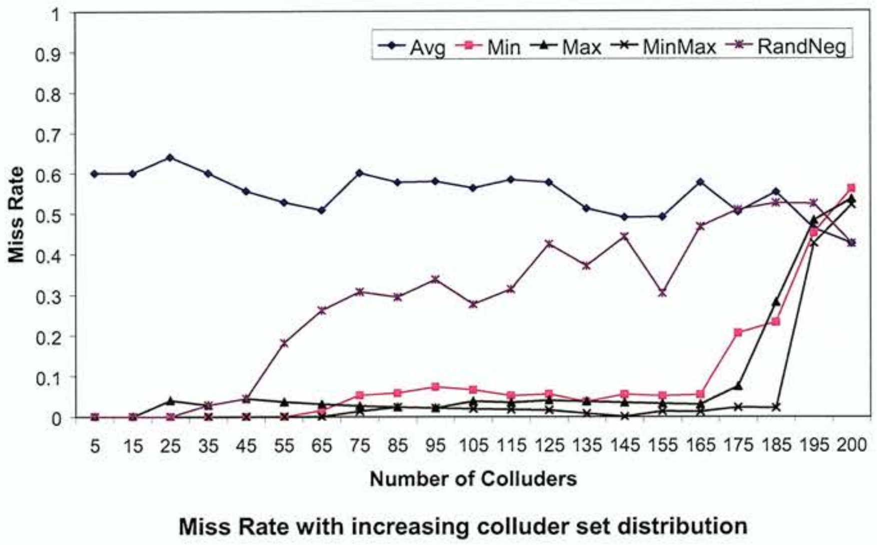

Figure 6.14 Results of miss rate with increasing colluder size while maintaining fixed user set

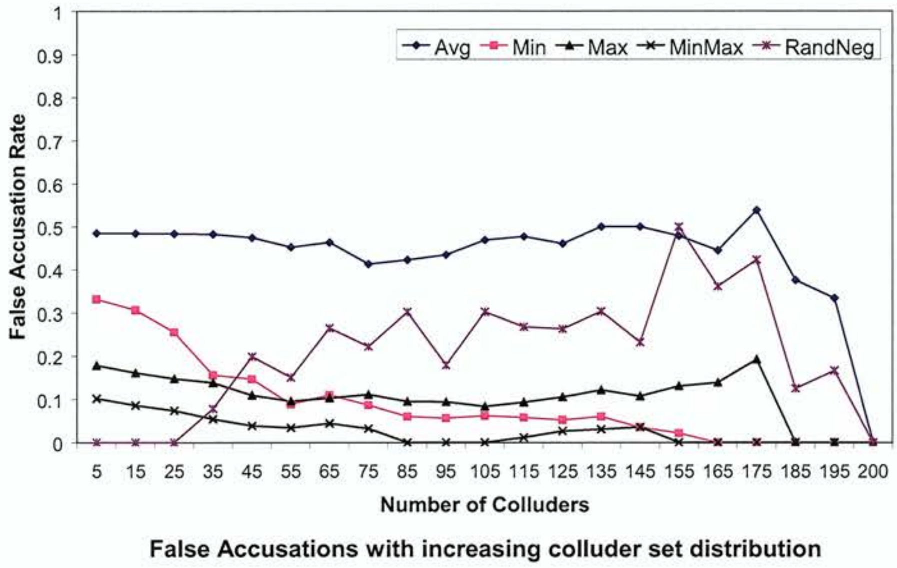

Figure 6.15 Results of false accusations with increasing colluder size while maintaining fixed user set

$D = U - C$. As an example, we start with 5 colluders (C) and 195 innocent individuals (D). As the number of colluders is increased, the number of innocent individuals in the entire set is decreased so that $|U|$ remains fixed. The results are shown in Figures 6.14 and 6.15.

It can be seen that as the total number of colluders increases in relation to the total fixed set of all users, the colluders gain the advantage of having a larger sample size of fingerprints that enable them to thwart leaving any type of statistical signature.

The figures demonstrate that as the number of colluders increase to the full set size, the algorithm starts to degrade in separating the two clusters sets. This is caused due to the final distance between the two centroids being reduced. As the number of colluders increase in relation to the decrease in innocent users, the joint density of the colluder cluster overwhelms the joint density of the innocent cluster. This causes our algorithm to only detect one cluster.

Using the SC, one can identify the possible set of attacks used on a colluded copy. If SC is closer to 1, there is a high probability that the minimum, maximum or minmax attacks were used to generate the colluded copy. Therefore, a user identified by the algorithm has a greater than 90% probability of being correctly identified. Otherwise, a SC closer to 0 indicates that the average or random negative attacks were used and it is hard to determine the certainty of that individual.

CHAPTER 7. CONCLUSION

There is an increasing need for digital forensic investigators to identify multimedia content distributors involved in a collusion attack to prevent future proliferation by those individuals.

In this research, the use of wavelet-based watermarking and statistical clustering techniques is used to detect and identify colluders involved in a collusion attack. The wavelet-based watermarking technique provides a framework of fingerprint embedding that provides a high recovery rate. This high recovery rate enables the use of a *2-means* statistical clustering algorithm to identify colluder sets involved in a collusion attack while minimizing miss rates and false accusation rates. The algorithm is effective at identifying the colluder sets in an attack without the knowledge of the number of colluders involved and the collusion attack used.

The solution was implemented with and without spread spectrum watermarking. An evaluation was performed against the developed solution with three different colluder set distributions: neutrally, negatively and positively skewed colluder data sets.

The average attack was found to do the most damage to the proposed solution because it acts as a signal normalizer. As the number of colluded signals increases, this weakens every single fingerprint involved in the collusion. This, in turn, normalizes the signal of the multimedia artifact to be more similar to the original artifact.

Therefore, the average attack can produce a colluded copy that can possibly have better perceptual quality than any fingerprinted signal. The results show that solutions that implement independent fingerprints that are Gaussian distributed values, the average attack would basically generate a copy that would have fingerprints of zero value. This is because the fingerprint generated values are from a Gaussian distribution. Given enough fingerprinted copies, colluders can use the average attack to reduce the colluded fingerprint equal to the mean of the distribution. However, the farther away the colluded fingerprint is from the mean of the distribution, the better the performance of our solution in identifying colluders in an average attack.

The proposed solution works best for the minimum, maximum and minimum/maximum attacks. The is due to the process on how these copies are generated. These attacks take the total minimum or maximum values for all components, in an artifact, and for all colluders. It becomes highly probable, that every colluder has a set of minimum or maximum component values that contribute directly to the final colluded copy. Since the fingerprints are orthogonal, there will be a set of component values that directly match a contributing colluder. In general terms, these attacks cause the colluders to leave their own signature in the final colluded copy. Leaving this signature increases the joint density of the relationship between those colluders and colluded copy, so that our 2-means algorithm becomes highly effective.

Another reason for this effect is the fact that Gaussian distributed values with zero-mean and variance were chosen. Since minimum and maximum values create a generated colluded copy whose fingerprint is further away than the mean of the distributed values, it becomes easier to detect a fingerprint that was not generated through the original process. This is because the colluded fingerprint created does not retain the properties of a Gaussian distributed fingerprint.

These two properties, colluders leaving their signature values and the colluded copy not exhibiting Gaussian properties, are the reason why our *2-means* algorithm obtains a high rate of effectiveness against the minimum, maximum and minmax attacks.

Our results show that spread spectrum watermarking causes degradation in identifying colluders and increases false accusation rates. Spread spectrum sequences generate a new type of noise-like sequence from the original watermark. This new embedded watermark is larger than the original. This property, coupled with any of the collusion attacks, causes a higher change in the watermark values in the final generated colluded copy. The main disadvantage of using the spread spectrum watermarking is that because the signal from a colluded copy is despreaded after extraction, causing it to be reduced in size, the fingerprint becomes part of a smaller fingerprint value space. This causes an increase in highly correlated fingerprints between innocent and colluders. Therefore, spread spectrum watermarking is not recommended as part of a solution that uses unsupervised statistical classification.

Our algorithm can be improved with an additional change. The algorithm should additionally check the ending distance of the two centroids. Our algorithm relies on finding statistical clusters within the given data set. However, if the final location of the two centroids is similar, one may conclude that there are no two clusters in the data set. Therefore, the entire set is either composed of mostly all innocent individuals or mostly all colluders. If this is the case, a threshold methodology can be used to further analyze and identify the type of scenario.

The main advantage of our algorithm is that it builds the colluder set from joint density observations and not set predictions. This research uses statistical classification techniques to find possible unknown relationships or signatures between the colluded copy and the fingerprinted copies of suspicious users. The use of statistical classification techniques coupled with strong fingerprinting schemes, such as wavelets, enable a new type of colluder identification method for digital forensic investigators. This methodology aids in finding colluders even if they use a new breed of collusion attack.

The results demonstrate that colluders can be identified when they employ the use of the minimum, maximum and minimun/maximum attacks. The methodology also minimizes the number of false accusations incurred by the use of these attacks. Although the random negative attack is not completely mitigated, the false accusation rates for the positively and negatively skewed colluder data sets are highly minimized. The average attack seems to be the strongest attack against the proposed work. In general, the evaluation results of this research demonstrate the solution is highly effective at identifying large colluder sets against the collusion attacks tested and can easily be implemented in current fingerprinting frameworks.

CHAPTER 8. ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Yong Guan for his guidance, patience and support throughout this research and the writing of this thesis. His insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank Dr. Jennifer Davidson for valuable insight in the area of digital data hiding, and providing me the initial interests in this area. I would also like to thank Dr. Johnny Wong for supporting me as a committee member and providing valuable discussions.

In addition, without the financial support from GEM, Iowa State University and Qualcomm Incorporated, my academics would not have been possible.

BIBLIOGRAPHY

[1] Apple iTunes. Apple Computer 2006. Available: http://www.apple.com/itunes/1billion/ .

[2] M. Barni, F. Bartolini, V. Capellini, and A. Piva, A DCT-domain system for robust image watermarking, in IEEE Signal Processing, vol. 6 No. 3, 1998, pp. 357372.

[3] BBC NEWS. "Eight charged over Star Wars leak", September 28th, 2005. Available: http://news.bbc.co.uk/1/hi/entertainment/film/4289030.stm .

[4] I. Biehl and B. Meyer, "Protocols for collusion-secure assymetric fingerprinting", Proceedings of STACS, 1997.

[5] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1897-1905, Sept. 1998.

[6] J. Brassil, S. Low, N. Maxemchuk and L. O'Gorman, *Electronic marking and identification techniques to discourage document copying* Proceedings of Infocom '94, pp. 1278-1287, June 1994.

[7] G. Caronni, *Assuring ownership rights for digital images*, H.H. Brueggemann and W. Gerhardt-Haeckl (Ed.) Proceedings of 'reliable IT systems' (verlaessliche IT-Systeme) VIS '95 Vieweg Publishing Company, Germany, 1995.

[8] B. Chen and G.W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," IEEE Trans. Information Theory, vol. 47, pp.1423-1443, May 2001.

[9] B. Chor, A. Fiat and M. Naor, *Tracing traitors*, Proceedings of Crypto '94, pp. 257-270.

[10] Hao-hua Chu, L. Qiao, K. Nahrstedt, H. Wang, and R. Jain, A secure multicast protocol with copyright protection, SIGCOMM Comput. Commun. Rev., vol. 32, no. 2, pp. 4260, 2002.

[11] C. J. Colbourn and J.H. Dinitz, *The CRC Handbook of Combinatorial Designs.* Boca Raton, FL: CRC Press, 1996.

[12] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms.* New York: McGraw Hill, 1989.

[13] I. Cox, J. Bloom, and M. Miller, Digital Watermarking: Principles and Practice. San Mateo, CA: Morgan Kaufman, 2001.

[14] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, *A secure robust watermark for multimedia*, Information Hiding, LNCS 1174, Springer-Verlag, 1996, 185-206.

[15] I. Cox; J. Kilian; F.T. Leighton; and T. Shamoon, Resistance of digital watermarks to collusive attacks, in Proceedings IEEE International Symposium Information Theory, Aug. 1998.

[16] I. Daubechies, Ten lectures on wavelets. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992.

[17] J.H. Dinitz and D.R. Stinson, *Contemporary Design Theory: A Collection of Surveys.* New York: Wiley, 1992.

[18] J. Dittmann, P. Schmitt, E. Saar, J. Schwenk, and J. Ueberberg, "Combining digital watermarks and collusion secure fingerprints for digital images," *SPIE J. Electron. Imaging*, vol. 9, no. 4, pp. 459-467, 2000.

[19] D.Z. Du and H. Park, "On competitive group testing," *SIAM J. Comput.*, vol. 23, pp. 1019-1025, Oct. 1994.

[20] F. Ergun, J. Kilian, and R. Kumar, "A note on the limits of collusion-resistant watermarks," in *Proc. Eurocrypt'99*, pp. 140-149, 1999.

[21] Film Journal International. "MPAA: Piracy could cost biz 15 billion over next 4 years" October 22, 2004 Available: http://www.hollywoodreporter.com/thr/film/brief_display.jsp?vnu_content_id=1000682391 .

[22] FutureProof, hymn - decrypt itunes and ipod music / unprotect aac files. [Online]. Available: http://hymn-project.org/jhymndoc/ .

[23] A. Herrigel, J. Oruanaidh, H. Petersen, S. Pereira, and T. Pun, "Secure copyright protection techniques for digital images," in *Second Information Hiding WOrkshop (IHW)* (Lecture Notes in Computer Science, vol. 1525). New York: Springer-Verlag, 1998.

[24] Jobs, Steve. CEO of Apple Computer. Apple Special Event, February 28th, 2006. Available: http://www.combo-blog.com/lvevent/Live Event/Live Event.html.

[25] N.F. Johnson and S.C. Katezenbeisser, A survey of steganographic techniques, Information Techniques for Steganography and Digital Watermarking, pp. 4375, Dec. 1999.

[26] Paul Judge and Mostafa Ammar, WHIM: Watermarking Multicast Video with a Hierarchy of Intermediaries, Comput. Networks, vol. 39, no. 6, pp. 699712, 2002.

[27] L. Kaufman, and P. J. Rousseeuw, *Finding groups in data. an introduction to cluster analysis.* Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York: Wiley, 1990.

[28] S. Kay, *Fundamentals of Statistical Signal Processing, Volume II : Detection Theory.* Englewood Cliffs, NJ: Prentice-Hall, New Jersey, 1998.

[29] J. Kilian; T. Leighton, L. Matheson, T. Shamoon, R. Tarjan, and F. Zane, Secure spread spectrum watermarking for multimedia, Proceedings of the IEEE 20 International Conference on Image Processing, ICIP 97, vol. 6, pp. 16731687, Oct. 1997.

[30] D. Kirovski, H.S. Malvar, and Y. Yacobi, "Multimedia content screening using a dual watermarking and fingerprinting system," in Proc. ACM Multimedia, 2002, pp. 372-381.

[31] D. Kirovski and F.A. Petitcolas, "Blind pattern matching attack on watermarking systems," *IEEE Trans. Signal Processing*, vol. 51, pp. 1045-1053, Apr. 2003.

[32] Martin Kutter, Frdric Jordan, and Frank Bossen, Digital watermarking using multiresolution wavelet decomposition, in In Proceedings of IEEE ICASSP 1998, Seattle, WA, May 1998, pp. Volume 5, pages 2969 2972.

[33] G. Langelaar, I. Setyawan, and R.L. Lagendijk, Watermarking digital image and video data, IEEE Signal Processing Magazine, vol. 17, pp. 2043, Sept. 2000.

[34] Lena.jpg Available: http://www.cs.nyu.edu/ nitin/minip/lena.jpg .

[35] C-Y. Lin, M. Wu, Y-M. Lui, J.A. Bloom, M.L. Miller, and I.J. Cox, "Rotation, scale, and translation resilient public watermarking for images," *IEEE Trans. Image Processing*, vol. 10, pp. 767-782, May 2001.

[36] J. Lubin, J. Bloom, and H. Cheng, "Robust, content-dependent, high-fidelity watermark for trackingin digital cinema," *Security and Watermarking of Multimedia Contents V, Proc. SPIE*, vol. 5020, pp. 536-545, Jan. 2003.

[37] A. Lumini and D. Maio, A wavelet-based image watermarking scheme, pp. 122 127, Mar.

[38] C.M. Luong, *Introduction to Computer Vision and Image Processing*, Department of Pattern Recognition and KNowledge Engineering, Institute of Information Technology, Hanoi, Vietnam, 2004.

[39] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, in Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, University of California Press, 1967.

[40] Peter Meerwald and Andreas Uhl, A survey of wavelet-domain watermarking algorithms, in Proceedings of SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents III, vol. 4314. San Jose, CA, USA: SPIE, January 2001.

[41] Motion Picture Association of America. "2005 Theatrical Market Statistics" Available: http://www.mpaa.org/researchStatistics.asp .

[42] P. Moulin and J.A. O'Sullivan. (2001, Dec.) Information-theoretic analysis of information hiding. Available: http://www.ifp.uiuc.edu/ moulin/paper.html .

[43] A. Persaud and Y. Guan, "Collusion Detection and Identification for Multimedia Forensics" Second Annual IFIP WG 11.9 International Conference on Digital Forensics, National Center for Forensic Science, Orlando, Florida, January 2006.

[44] B. Pfitzmann and M. Waidner, *Assymetric fingerprinting for large collusions*, Proceedings of the 4th ACM Conference on Computer and Communication Security, 1997.

[45] C. Podilchuk and W. Zeng, "Image adaptive watermarking using visual models," *IEEE J. Selected Areas Commun.*, vol. 16, pp. 525-538, May 1998.

[46] J.G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2000.

[47] Record Industry Association of America (RIAA). "Anti-Piracy". (2006) Available: http://www.riaa.com/issues/piracy/default.asp .

[48] J. Sachs, *Digital Image Basics*, Digital LIght and Color, Cambridge, Massachusetts, 1999.

[49] B. Schneier, *Applied cryptography*, Wiley, 1994.

[50] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, A digital watermark, in In Proceedings of the International Conference on Image Processing. Vol 2, 1994, pp. pages 86 90.

[51] Chris Shoemaker, Hidden bits: A survey of techniques for digital watermarking, 2002. [Online]. Available: http://www.vu.union.edu/shoemakc/watermarking/watermarking.html .

[52] H.S. Stone, "Analysis of attacks on image watermarks with randomized coefficients," NEC Research Inst., Princeton, NJ, Tech. Rep. 96-045, 1996.

[53] K. Su, D. Kundur, and D. Hatzinakos, "A content-dependent spatially localized video watermarked for resistance to collusion and interpolations attacks," in *Proc. IEEE INt. Conf. Image Processing*, Oct. 2001, pp. 818-821.

[54] J.K. Su, J.J. Eggers, and B. Girod, "Capacity of digital watermarks subjected to an optimal collusion attack," in *European Signal Processing Conf. (EUSIPCO 2000)* 2000.

[55] M.D. Swanson, B. Zhu, and A.T. Tewfik, "Multiresolution scene-based video watermarking using perceptual models," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 540-550, May 1998.

[56] K. Tanaka, Y. Nakamura and K. Matsui. *Embedding secret information into a dithered multi-level image*, Proceedings of the 1990 IEEE Military Communications Conference, pp. 216-220, September 1990.

[57] L. F. Turner, Digital data security system. patent IPN WO 89/08915, 1989.

[58] N. Wagner, *Fingerprinting*, Proceedings of the 1983 IEEE Symposium on Security and Privacy, April, 1983, pp. 18-22.

[59] Z.J. Wang, M. Wu, W. Trappe, and K.J.R. Liu, "Group-oriented fingerprinting for multimedia forensics," to be published.

[60] Z.J. Wang, Min Wu, H.V. Zhao, W. Trappe, and K.J.R. Liu, Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation, IEEE Transactions on Image Processing, vol. 14, pp. 804821, June 2005.

[61] Z.J. Wang, M. Wu, H. Zhao, W. Trappe, and K.J.R. Liu, "Resistance of orthogonal Gaussian fingerprints to collusion attacks," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP'03)*, Hong Kong, Apr. 2003, pp. 724-727.

[62] Wikipedia, Fairplay - from wikipedia, the free encyclopedia. 2006. [Online]. Available: http://en.wikipedia.org/wiki/FairPlay .

[63] Wikipedia, Warez - from wikipedia, the free encyclopedia. 2006. [Online]. Available: http://en.wikipedia.org/wiki/Warez .

[64] C-S Woo, J. Du, and B. Pham, Performance factors analysis of wavelet-based watermarking method, Newcastle, Austrailia, 2005.

[65] M. Wu, W. Trappe, and Z. Wang; and K.J.R. Liu, Collusion resistant fingerprinting for multimedia, IEEE Signal Processing Magazine: Special Issue on Digital Rights Management, pp. 1527, Mar. 2004.

[66] M. Wu and B. Liu, *Multimedia Data Hiding*, New York: Springer-Verlag, Oct. 2002.

[67] M. Wu and B. Liu, "Data hiding in image and video: Part-I - Fundamental issues and solution," *IEEE Trans. Image Processing*, vol. 12, pp. 685-695, Jun 2003.

[68] X.-G., Xia and C. G. Boncelet and G. R. Arce, Wavelet transform based watermark for digital images, Optics Express 3, p. p. 497, Dec. 1998.

[69] Y. Yacobi, "Improved Boneh-Shaw content fingerprinting," in *Proc. CT-RSA 2001*, 2001, pp. 378-91.

[70] W. Zeng and B. Liu, "A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images," *IEEE Trans. Image Processing*, vol. 8, pp.1534-1548, Nov. 1999.

[71] H. Zhao; M. Wu; Z.J. Liu, and K.J.R., Nonlinear collusion attacks on independent fingerprints for multimedia, in ICME 2003 Proc. International Conference on Multimedia Expo., July 2003, pp. Vol 1. 613616.

APPENDIX A. MATLAB SOURCE CODE

The following is the source code that was used in the evaluation of the proposed scheme.

A.1 setup.m

```
X = imread('lena.jpg');
X = double(X); %perform operations without warnings
total = input('Set Total Number of Copies: '); step = input('Set
Collusion step size: '); dwtmode('per');
disp(sprintf('Creating fingerprinted copies... %d',total));
C=generate(X,total); %colluders images/watermarks
I=generate(X,total); %innocent images/watermarks
```

A.2 generate.m

```
function [aW]=genereate(X,k);
 for n = 1:k
     disp(sprintf('Generating copy (%d)...',n));
     [aW(n).im, aW(n).wm] = wavemark(X);
 end
 return;
```

A.3 generate.m

```
function [cent_b,cent_c]=calc_centroids(Cr,Ir);
```

```
%we divide by 100.0, (to offset MatLab rounding problem)
cent_c = mean([Cr Ir]) + (std([Cr Ir])/100.0);

cent_b = mean([Cr Ir]) - (std([Cr Ir])/100.0); return;
```

A.4 wavemark.m

```
function [W,w]=wavemark(X);

[M,N] = size(X); alpha = .1;

%=================start of encoder====================

% set dwtmode to periodization so that size(cA)=size(cH)=size(cV)=size(cD)=ceil(sx/2)
dwtmode('per');

% perform 4-level Daubechies-6 wavelet decomposition
[C,S] = wavedec2(double(X),4,'db6');

% watermark creation
[cH4,cV4,cD4] = detcoef2('all',C,S,4);
% calc number of lowpass approximation coeffs

randn('state',sum(100*clock));
w = randn(1,M*N - prod(size(cH4))); % 1x65280 watermark for r=4 resolution

% watermark insertion
highpass = C(prod(size(cH4))+1:end); lowpass =
C(1:prod(size(cH4)));
```

```
mod_highpass = highpass + (alpha*w); % additive formula


mod_C = [lowpass mod_highpass];

% catenate modified highpass with untouched lowpass coeff


% Inverse 4-level Daubechies-6 wavelet decomposition

W = waverec2(mod_C,S,’db6’);

%W = waverec2(C,S,’db6’);


return;
```

A.5 wavemark_ext.m

```
function u=wavemark_ext(O,W);


[M,N] = size(O); [wM,wN] = size(W); alpha = .1;


% set dwtmode to periodization so that

%size(cA)=size(cH)=size(cV)=size(cD)=ceil(sx/2)

dwtmode(’per’);


% perform 4-level Daubechies-6 wavelet decomposition

[C,S] = wavedec2(double(O),4,’db6’); [C_est,S_est] =

wavedec2(double(W),4,’db6’);


% watermark creation

[cH4,cV4,cD4] = detcoef2(’all’,C,S,4);

% calc number of lowpass approximation coeffs
```

```
highpass_est = C_est(prod(size(cH4))+1:end); highpass =
C(prod(size(cH4))+1:end);

u = (highpass_est - highpass) ./ alpha; return;
% inverse formula
```

A.6 avgattk.m

```
function [tImg]=avgattk(k,X);
%Takes as input a series of k copies
%of X structures that will be combined linearly
[m,n] = size(X); [img_m,img_n] = size(X(1).im); tImg =
zeros(img_m,img_n);

for t = 1:k
    tImg = tImg + X(t).im;
end

tImg = tImg / n; return;
```

A.7 minattk.m

```
function [tImg]=maxattk(k,X); [m,n] = size(X); [img_m,img_n] =
size(X(1).im); tImg = X(1).im;
for i = 1:n
   tImg = min(tImg,X(i).im);
end
return;
```

A.8 maxattk.m

```
function [tImg]=maxattk(k,X);

[m,n] = size(X);


[img_m,img_n] = size(X(1).im); tImg = X(1).im;


for i = 1:n

   tImg = max(tImg,X(i).im);

end

return;
```

A.9 minmaxattk.m

```
function [tImg]=minmaxattk(k,X);


%Takes as input a series of k copies of X structures that will be combined linearly


[m,n] = size(X); [img_m,img_n] = size(X(1).im); tImg =

zeros(img_m,img_n); tImg_1 = zeros(img_m,img_n); tImg_2 =

zeros(img_m,img_n);


tImg_1 = minattk(k,X); tImg_2 = maxattk(k,X);


tImg = (tImg_1 + tImg_2) /2;


return;
```

A.10 randneg.m

```
function [tImg]=randneg(k,X);

%Takes as input a series of k copies of structures X that will be combined linearly

[m,n] = size(X); [img_m,img_n] = size(X(1).im); tImg =

zeros(img_m,img_n);


randn('state',sum(100*clock)); tImg = X(1).im;


i=1; j=1;


for i = 1:img_n

    for j = 1:img_m


    p = randn();

    if( p > 0)

            for t = 1:k

                F = X(t).im;

                tImg(i,j) = max(tImg(i,j), F(i,j));

            end;

    else

            for t = 1:k

                F = X(t).im;

                tImg(i,j) = min(tImg(i,j), F(i,j));

            end

    end;


        %tImg = max(tImg , X(t).im);

    end
```

```
end

return;
```

A.11 test_avgattk_kmeans.m

```
%now I have 2 K fingerprinted copies of Lena
%Trying Geometric linear attack.
disp(sprintf('Running attacks'));

% total - 5 (cause we start at 5, but we want to include (5), so
% -5 + 1 = -4
TP = zeros(1,(total/step)+1); FP = zeros(1,(total/step)+1); FN =
zeros(1,(total/step)+1); cCorr = zeros(1,(total/step)+1); iCorr =
zeros(1,(total/step)+1); dCorr = zeros(1,(total/step)+1); allCorr
= zeros(1,(total/step)+1); index = [(5:step:total) total]; counter
= 1; for n = [index]

Cr = zeros(1,n); Ir = zeros(1,n);
disp(sprintf('Generating collusion copy with n = %d',n));

  K = avgattk(n,C);

  extK  = wavemark_ext(X,K); %extract watermark

  for colluder = 1:n
        r = corr2(extK,C(colluder).wm); %corr between fprint and colluder
        %disp(sprintf('Colluder %d of %d: corr %f',colluder,n,r));
        Cr(colluder) = r;
  end

  for innocent = 1:n
        r = corr2(extK,I(innocent).wm); %corr between fprint and colluder
        %disp(sprintf('Innocent %d of %d: corr %f',innocent,n,r));
        Ir(innocent) = r;
  end

rT = [Cr' ; Ir']; %put colluder group first, then innocent
%so, 1 -> n is colluders, and n+1->end is innocent always
[cent_b,cent_c] = calc_centroids(Cr,Ir); [IDX,centroids] =
kmeans(rT,2,'start',[cent_b ; cent_c]);

if(centroids(1) > centroids(2))
    %colluder group is 1
       ident = COUNT(IDX(1:n), '==1'); %true positives - identifications
       falseacc = COUNT(IDX(n+1:end), '==1');%false positives - false accusations
       misses = COUNT(IDX(1:n), '==2'); %true positives - identifications
else
    %colluder group is 2
       ident = COUNT(IDX(1:n), '==2'); %true positives - identifications
       falseacc = COUNT(IDX(n+1:end), '==2');%false positives - false accusations
       misses = COUNT(IDX(1:n), '==1'); %true positives - identifications
end

TP(counter) = ident/n;
FN(counter) = misses/n; %misses
```

```
FP(counter) = falseacc/n; %wrongly accused

%allCorr(counter) = [Cr',Ir'];
cCorr(counter) = mean(Cr); iCorr(counter) = mean(Ir);
dCorr(counter) = abs(mean(Cr)) - abs(mean(Ir));
disp(sprintf('TruePositives : %d',TP(counter)));
disp(sprintf('FalseNegatives: %d',FN(counter)));
disp(sprintf('FalsePositives: %d',FP(counter)));
counter = counter + 1;

end
```

A.12 test_minattk_kmeans.m

```
%now I have 2 K fingerprinted copies of Lena
%Trying Geometric linear attack.
disp(sprintf('Running attacks'));

% total - 5 (cause we start at 5, but we want to include (5), so
% -5 + 1 = -4
TP = zeros(1,(total/step)+1); FP = zeros(1,(total/step)+1); FN =
zeros(1,(total/step)+1); cCorr = zeros(1,(total/step)+1); iCorr =
zeros(1,(total/step)+1); dCorr = zeros(1,(total/step)+1); allCorr
= zeros(1,(total/step)+1); index = [(5:step:total) total]; counter
= 1; for n = [index]

Cr = zeros(1,n); Ir = zeros(1,n);
disp(sprintf('Generating collusion copy with n = %d',n));
```

```
  K = minattk(n,C);

  extK  = wavemark_ext(X,K); %extract watermark

  for colluder = 1:n
        r = corr2(extK,C(colluder).wm); %corr between fprint and colluder
        %disp(sprintf('Colluder %d of %d: corr %f',colluder,n,r));
        Cr(colluder) = r;
  end

  for innocent = 1:n
        r = corr2(extK,I(innocent).wm); %corr between fprint and colluder
        %disp(sprintf('Innocent %d of %d: corr %f',innocent,n,r));
        Ir(innocent) = r;
  end

rT = [Cr' ; Ir']; %put colluder group first, then innocent
%so, 1 -> n is colluders, and n+1->end is innocent always
[cent_b,cent_c] = calc_centroids(Cr,Ir); [IDX,centroids] =
kmeans(rT,2,'start',[cent_b ; cent_c]);

if(centroids(1) > centroids(2))
    %colluder group is 1
       ident = COUNT(IDX(1:n), '==1'); %true positives - identifications
       falseacc = COUNT(IDX(n+1:end), '==1');%false positives - false accusations
       misses = COUNT(IDX(1:n), '==2'); %true positives - identifications
else
```

```
    %colluder group is 2
        ident = COUNT(IDX(1:n), '==2'); %true positives - identifications
        falseacc = COUNT(IDX(n+1:end), '==2');%false positives - false accusations
        misses = COUNT(IDX(1:n), '==1'); %true positives - identifications
end

TP(counter) = ident/n;
FN(counter) = misses/n; %misses
FP(counter) = falseacc/n; %wrongly accused

%allCorr(counter) = [Cr',Ir'];
cCorr(counter) = mean(Cr); iCorr(counter) = mean(Ir);
dCorr(counter) = abs(mean(Cr)) - abs(mean(Ir));
disp(sprintf('TruePositives : %d',TP(counter)));
disp(sprintf('FalseNegatives: %d',FN(counter)));
disp(sprintf('FalsePositives: %d',FP(counter)));
counter = counter + 1;

end
```

A.13 test_maxattk_kmeans.m

```
%now I have 2 K fingerprinted copies of Lena
%Trying Geometric linear attack.
disp(sprintf('Running attacks'));

% total - 5 (cause we start at 5, but we want to include (5), so
% -5 + 1 = -4
```

```
TP = zeros(1,(total/step)+1); FP = zeros(1,(total/step)+1); FN =
zeros(1,(total/step)+1); cCorr = zeros(1,(total/step)+1); iCorr =
zeros(1,(total/step)+1); dCorr = zeros(1,(total/step)+1); allCorr
= zeros(1,(total/step)+1); index = [(5:step:total) total]; counter
= 1; for n = [index]

Cr = zeros(1,n); Ir = zeros(1,n);
disp(sprintf('Generating collusion copy with n = %d',n));

  K = maxattk(n,C);

  extK  = wavemark_ext(X,K); %extract watermark

  for colluder = 1:n
        r = corr2(extK,C(colluder).wm); %corr between fprint and colluder
        %disp(sprintf('Colluder %d of %d: corr %f',colluder,n,r));
        Cr(colluder) = r;
  end

  for innocent = 1:n
        r = corr2(extK,I(innocent).wm); %corr between fprint and colluder
        %disp(sprintf('Innocent %d of %d: corr %f',innocent,n,r));
        Ir(innocent) = r;
  end

rT = [Cr' ; Ir']; %put colluder group first, then innocent
%so, 1 -> n is colluders, and n+1->end is innocent always
[cent_b,cent_c] = calc_centroids(Cr,Ir); [IDX,centroids] =
kmeans(rT,2,'start',[cent_b ; cent_c]);

if(centroids(1) > centroids(2))
    %colluder group is 1
       ident = COUNT(IDX(1:n), '==1'); %true positives - identifications
       falseacc = COUNT(IDX(n+1:end), '==1');%false positives - false accusations
       misses = COUNT(IDX(1:n), '==2'); %true positives - identifications
else
    %colluder group is 2
       ident = COUNT(IDX(1:n), '==2'); %true positives - identifications
       falseacc = COUNT(IDX(n+1:end), '==2');%false positives - false accusations
       misses = COUNT(IDX(1:n), '==1'); %true positives - identifications
end

TP(counter) = ident/n;
FN(counter) = misses/n; %misses
FP(counter) = falseacc/n; %wrongly accused

%allCorr(counter) = [Cr',Ir'];
cCorr(counter) = mean(Cr); iCorr(counter) = mean(Ir);
dCorr(counter) = abs(mean(Cr)) - abs(mean(Ir));
disp(sprintf('TruePositives : %d',TP(counter)));
disp(sprintf('FalseNegatives: %d',FN(counter)));
disp(sprintf('FalsePositives: %d',FP(counter)));
counter = counter + 1;

end
```

A.14 test_minmaxattk_kmeans.m

```
%now I have 2 K fingerprinted copies of Lena

%Trying Geometric linear attack.

disp(sprintf('Running attacks'));


% total - 5 (cause we start at 5, but we want to include (5), so

% -5 + 1 = -4

TP = zeros(1,(total/step)+1); FP = zeros(1,(total/step)+1); FN =

zeros(1,(total/step)+1); cCorr = zeros(1,(total/step)+1); iCorr =

zeros(1,(total/step)+1); dCorr = zeros(1,(total/step)+1); allCorr

= zeros(1,(total/step)+1); index = [(5:step:total) total]; counter

= 1; for n = [index]


Cr = zeros(1,n); Ir = zeros(1,n);

disp(sprintf('Generating collusion copy with n = %d',n));


  K = minmaxattk(n,C);


  extK  = wavemark_ext(X,K); %extract watermark


  for colluder = 1:n

        r = corr2(extK,C(colluder).wm); %corr between fprint and colluder

        %disp(sprintf('Colluder %d of %d: corr %f',colluder,n,r));

        Cr(colluder) = r;

  end


  for innocent = 1:n
```

```
        r = corr2(extK,I(innocent).wm); %corr between fprint and colluder
        %disp(sprintf('Innocent %d of %d: corr %f',innocent,n,r));
        Ir(innocent) = r;
  end

rT = [Cr' ; Ir']; %put colluder group first, then innocent
%so, 1 -> n is colluders, and n+1->end is innocent always
[cent_b,cent_c] = calc_centroids(Cr,Ir); [IDX,centroids] =
kmeans(rT,2,'start',[cent_b ; cent_c]);

if(centroids(1) > centroids(2))
    %colluder group is 1
       ident = COUNT(IDX(1:n), '==1'); %true positives - identifications
       falseacc = COUNT(IDX(n+1:end), '==1');%false positives - false accusations
       misses = COUNT(IDX(1:n), '==2'); %true positives - identifications
else
    %colluder group is 2
       ident = COUNT(IDX(1:n), '==2'); %true positives - identifications
       falseacc = COUNT(IDX(n+1:end), '==2');%false positives - false accusations
       misses = COUNT(IDX(1:n), '==1'); %true positives - identifications
end

TP(counter) = ident/n;
FN(counter) = misses/n; %misses
FP(counter) = falseacc/n; %wrongly accused

%allCorr(counter) = [Cr',Ir'];
cCorr(counter) = mean(Cr); iCorr(counter) = mean(Ir);
```

```
dCorr(counter) = abs(mean(Cr)) - abs(mean(Ir));
disp(sprintf('TruePositives : %d',TP(counter)));
disp(sprintf('FalseNegatives: %d',FN(counter)));
disp(sprintf('FalsePositives: %d',FP(counter)));
counter = counter + 1;

end
```

A.15 test_randnegattk_kmeans.m

```
%now I have 2 K fingerprinted copies of Lena
%Trying Geometric linear attack.
disp(sprintf('Running attacks'));

% total - 5 (cause we start at 5, but we want to include (5), so
% -5 + 1 = -4
TP = zeros(1,(total/step)+1); FP = zeros(1,(total/step)+1); FN =
zeros(1,(total/step)+1); cCorr = zeros(1,(total/step)+1); iCorr =
zeros(1,(total/step)+1); dCorr = zeros(1,(total/step)+1); allCorr
= zeros(1,(total/step)+1); index = [(5:step:total) total]; counter
= 1; for n = [index]

Cr = zeros(1,n); Ir = zeros(1,n);
disp(sprintf('Generating collusion copy with n = %d',n));

  K = randnegattk(n,C);

  extK  = wavemark_ext(X,K); %extract watermark
  for colluder = 1:n
        r = corr2(extK,C(colluder).wm); %corr between fprint and colluder
        %disp(sprintf('Colluder %d of %d: corr %f',colluder,n,r));
        Cr(colluder) = r;
  end

  for innocent = 1:n
        r = corr2(extK,I(innocent).wm); %corr between fprint and colluder
        %disp(sprintf('Innocent %d of %d: corr %f',innocent,n,r));
        Ir(innocent) = r;
  end

rT = [Cr' ; Ir']; %put colluder group first, then innocent
%so, 1 -> n is colluders, and n+1->end is innocent always
[cent_b,cent_c] = calc_centroids(Cr,Ir); [IDX,centroids] =
kmeans(rT,2,'start',[cent_b ; cent_c]);

if(centroids(1) > centroids(2))
    %colluder group is 1
       ident = COUNT(IDX(1:n), '==1'); %true positives - identifications
       falseacc = COUNT(IDX(n+1:end), '==1');%false positives - false accusations
       misses = COUNT(IDX(1:n), '==2'); %true positives - identifications
else
    %colluder group is 2
       ident = COUNT(IDX(1:n), '==2'); %true positives - identifications
       falseacc = COUNT(IDX(n+1:end), '==2');%false positives - false accusations
       misses = COUNT(IDX(1:n), '==1'); %true positives - identifications
```

```
end

TP(counter) = ident/n;
FN(counter) = misses/n; %misses
FP(counter) = falseacc/n; %wrongly accused

%allCorr(counter) = [Cr',Ir'];
cCorr(counter) = mean(Cr); iCorr(counter) = mean(Ir);
dCorr(counter) = abs(mean(Cr)) - abs(mean(Ir));
disp(sprintf('TruePositives : %d',TP(counter)));
disp(sprintf('FalseNegatives: %d',FN(counter)));
disp(sprintf('FalsePositives: %d',FP(counter)));
counter = counter + 1;

end
```

A.16 batchrun.m

```
setup

disp(' '); disp('Average Attack');
test_avgattk_kmeans

avg.TP = TP';
avg.FP = FP';
avg.FN = FN';
avg.cCorr = cCorr';
avg.iCorr = iCorr';
avg.dCorr = dCorr';
```

```
disp(' '); disp('Min Attack');
test_minattk_kmeans
min.TP = TP';
min.FP = FP';
min.FN = FN';
min.cCorr = cCorr';

min.iCorr = iCorr';
min.dCorr = dCorr';

disp(' '); disp('Max Attack');
test_maxattk_kmeans

max.TP = TP';
max.FP = FP';
max.FN = FN';
max.cCorr = cCorr';
max.iCorr = iCorr';
max.dCorr = dCorr';

disp(' '); disp('MinMax Attack');
test_minmaxattk_kmeans

minmax.TP = TP';
minmax.FP = FP';
minmax.FN = FN';
minmax.cCorr = cCorr’;
minmax.iCorr = iCorr’;
minmax.dCorr=dCorr’;



disp(’ ’); disp(’RandNeg Attack’);
test_randnegattk_kmeans

randneg.TP = TP’;
randneg.FP = FP’;
randneg.FN = FN’;
randneg.cCorr = cCorr’;
randneg.iCorr = iCorr’;
randneg.dCorr=dCorr’;

index_v = [(5:step:total) total]’;

ALL.TP = [index_v, avg.TP, min.TP, max.TP, minmax.TP, randneg.TP];
ALL.FP = [index_v, avg.FP, min.FP, max.FP, minmax.FP, randneg.FP];
ALL.FN = [index_v, avg.FN, min.FN, max.FN, minmax.FN, randneg.FN];

ALL.cCorr = [index_v,
             avg.cCorr,
             min.cCorr,
             max.cCorr,
             minmax.cCorr,
             randneg.cCorr];
```

```
ALL.iCorr = [index_v,
             avg.iCorr,
             min.iCorr,
             max.iCorr,
             minmax.iCorr,
             randneg.iCorr];

ALL.dCorr = [index_v,
             avg.dCorr,
             min.dCorr,
             max.dCorr,
             minmax.dCorr,
             randneg.dCorr];

disp('Exporting to Excel file wave_results.xls');
xlswrite('wave_results.xls', ALL.TP, 'True Positive','A2');
xlswrite('wave_results.xls', ALL.FP, 'False Positive','A2');
xlswrite('wave_results.xls', ALL.FN, 'False Negative','A2');
```