Chapter 9

COLLUSION DETECTION USING MULTIMEDIA FINGERPRINTS

Anthony Persaud and Yong Guan

Abstract The large-scale distribution of digital multimedia over the Internet has seen steep increases in the numbers of criminal cases involving the unauthorized sharing and duplication of copyrighted multimedia content. Consequently, it is important to design reliable investigative techniques to combat unauthorized duplication and propagation, and to provide protection in the form of theft deterrence. Several fingerprint embedding schemes have been developed to combat single-user attacks. However, a new breed of attacks known as "collusion attacks" can defeat these schemes. Collusion attacks use the combination of multiple fingerprinted copies to create a new version of the multimedia artifact in which the underlying fingerprint is attenuated to render the colluders untraceable.

This paper proposes a wavelet-based fingerprinting scheme and a clustering algorithm for collusion attack detection and colluder identification. Experimental results show that the scheme can identify colluders while maintaining low miss rates and false accusation rates.

Keywords: Multimedia content, collusion attacks, multimedia fingerprinting

1. Introduction

Digital watermarks are often used to uniquely mark multimedia artifacts to help identify the original recipients of the artifacts. Watermarks are useful for investigating the unauthorized duplication and propagation of multimedia content. Also, they provide protection in the form of theft deterrence. Several fingerprint embedding schemes have been developed to combat single-user attacks, i.e., the duplication and dissemination of content by individuals. However, a new breed of attacks known as "collusion attacks" can defeat these schemes. Collusion attacks combine multiple fingerprinted copies of multimedia content, creating a new ver-

sion of the artifact in which the fingerprint is attenuated to render the colluders untraceable. In the highly interconnected digital world, collusion attacks have become a popular technique for defeating multimedia fingerprinting embedding schemes.

Most colluder detection techniques [14] rely on direct pattern correlation of the colluded fingerprint to the set of colluders. This assumes that the entire fingerprint can be recovered and that the entire set of possible colluders is known. However, such assumptions are not realistic in many real-world scenarios.

This paper proposes a wavelet-based multimedia fingerprinting technique and a clustering algorithm for collusion attack detection and colluder identification. The scheme engages wavelet transforms and statistical clustering techniques to detect and identify the colluders involved in a collusion attack. Experimental results show that it can identify colluders while maintaining low miss rates and false accusation rates.

Our approach has four main benefits. First, full fingerprint recovery is not required. Second, the colluder set is built from joint density observations, not from predictions. Third, the approach requires less computational overhead for colluder identification. Finally, the identification of colluder sets is independent of how multiple marked copies are combined in a collusion attack and the number of colluders involved.

The next section describes the framework used for multimedia forensics. Next, a threat model involving linear and non-linear collusion attacks is presented. The following sections describe the wavelet-based fingerprinting scheme and clustering algorithm. The final two sections present the experimental results and conclusions.

2. Multimedia Forensics Framework

The overall problem of multimedia fingerprinting and colluder identification has three components: (i) fingerprint embedding, (ii) fingerprint recovery and (iii) collusion attack detection and identification. Fingerprint embedding focuses on using robust methods to embed watermark information in different multimedia artifacts. Fingerprint recovery deals with the recovery of embedded watermarks. In some cases, only part of the watermark is recoverable due to various alteration attacks. Collusion attack detection and colluder identification involve analyzing correlations between the embedded watermark and the set of known watermarks that correspond to known users. Most watermarking schemes address fingerprint embedding and/or recovery. Our scheme is unique in that it addresses all three components of the overall problem.

Wavelets have proven to be the most effective and robust scheme for watermarking multimedia artifacts [8]. Fingerprint embedding typically uses wavelet transforms, e.g., Discrete Wavelet Transform (DWT), to decompose an image into sub-bands [15]. These sub-bands represent the image approximation coefficients, which can be combined with the watermark via additive embedding [13]. 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. This provides a higher degree of robustness with little or no impact on quality [10].

Fingerprint recovery is similar to the fingerprint embedding process. DWT is used to decompose the artifact into its corresponding set of sub-band coefficients [15]. These coefficients are compared with the original non-watermarked coefficients to retrieve the differences in values [13]. The difference in values is the corresponding embedded watermark for each sub-band. The recovery process is performed for all sub-bands that may have an embedded watermark [10].

Collusion attack detection and colluder identification involve the application of watermark correlation methods. Correlations are computed between the recovered colluded fingerprint and the fingerprints of users who received the content.

To identify multimedia colluders, Chu, *et al.* [1] have proposed that the list of all possible colluder combinations be generated from the set of individuals who received the multimedia content. The fingerprint for each combination of possible colluders is compared to the retrieved watermark, and the colluders are identified by the process of elimination.

Judge and Ammar [5] have developed a hierarchical watermarking system called *WHIM*. The location of the potential colluders can be approximated using watermark verification through intermediary nodes.

Other detection techniques rely on direct pattern correlation of a colluded fingerprint with a combination of colluders [14]. Some of these techniques assume that the entire watermark is recoverable from the colluded copy.

3. Threat Model

Collusion attacks fall into two main categories: linear and non-linear attacks [14]. Interested readers are referred to [3, 16] for details about these attacks.

Collusion attacks typically synchronize multiple fingerprinted copies of a multimedia artifact and average the signal to produce a new copy. In some cases, colluders might use a variant of the average attack by adding a small amount of Gaussian noise ε to increase the attenuation

Table 1. Formulations of collusion attacks used in this study.

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	$\psi_x^{max}(i,j) = max(\left\{\psi_x^{(k)}(i,j)\right\}_{k\in K})$
MinMax	$\psi_x^{minmax}(i,j) = \frac{1}{2}\left(\psi_x^{min}(i,j) + \psi_x^{max}(i,j)\right)$
Randomized Negative	$\psi_x^{randneg}(i,j) = \left\{ \begin{array}{c} \psi_x^{min}(i,j) \text{ with prob. } p \\ \psi_x^{max}(i,j) \text{ with prob. } 1-p \end{array} \right\}$

of the original fingerprint. Other attacks use statistical approaches to attenuate fingerprints. In most cases, the minimum, maximum and median values of each of the fingerprinted copies are analyzed to create a new less traceable copy. The collusion attacks considered in this study are shown in Table 1.

Let $|C|$ out of $|U|$ total users collude so that $C = \{c_1, c_2, ...c_n\}$, where $n = |C|$. Let $\psi'(i,j)$ represent the component of a colluded copy Ψ' at location (i,j). Using $|C|$ copies, the component of $\psi'(i,j)$ is generated by combining the components of all $c \in C$ using any of the attacks formulated in Table 1.

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

$$\Psi'^{avg} = \begin{bmatrix} 3 & 1 \\ 4 & 6 \end{bmatrix}, \Psi'^{min} = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \tag{2}$$

$$\Psi'^{max} = \begin{bmatrix} 4 & 2 \\ 5 & 8 \end{bmatrix}, \Psi'^{minmax} = \begin{bmatrix} 3 & 1 \\ 4 & 6 \end{bmatrix} \tag{3}$$

$$\Psi'^{randneg} = \begin{bmatrix} \psi^{min}(1,1) & \psi^{max}(1,2) \\ \psi^{max}(2,1) & \psi^{min}(2,2) \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 5 & 8 \end{bmatrix} \tag{4}$$

An example involving two colluders is presented in Equation 1. The terms c_1 and c_2 denote image information in matrix form possessed by the two colluders. The two colluders combine their watermarked copies to create a colluded copy Ψ'. In the following, we discuss each of the collusion attacks presented in Table 1 and specified by Equations 2–4.

3.1 Average Attack

This attack averages the corresponding components of each colluder's copy to produce a new value. In the example, component $\psi^{avg}(1,1)$ is

calculated as $\frac{c_1(1,1)+c_2(1,1)}{2} = 3$. The colluded copy Ψ'^{avg} is obtained by performing this computation for all the components of c_1 and c_2.

3.2 Minimum Attack

This attack takes the corresponding minimum components of the $|C|$ fingerprinted copies of the colluders. In the example, component $\psi^{min}(1,1)$ is calculated as $min(c_1(1,1), c_2(1,1))$. The colluded copy Ψ'^{min} is generated by performing this computation for all the components of c_1 and c_2.

3.3 Maximum Attack

This attack takes the corresponding maximum components of the $|C|$ fingerprinted copies used in the attack. Component $\psi^{max}(1,1)$ is calculated as $max(c_1(1,1), c_2(1,1))$. Performing this computation for all the components of c_1 and c_2 yields Ψ'^{min}.

3.4 MinMax Attack

In this attack, the averages of the minimum and maximum values of the corresponding components of the $|C|$ fingerprinted copies are used to produce the colluded copy. Component $\psi^{minmax}(1,1)$ is computed as the average $\frac{\psi^{min}(1,1)+\psi^{max}(1,1)}{2}$. Performing this computation for all the components of c_1 and c_2 yields Ψ'^{minmax}.

3.5 Randomized Negative Attack

In this attack, the values of each of the components in the colluded copy take either the minimum or maximum values of the corresponding components of the $|C|$ fingerprinted copies. The value of a component of the colluded copy, $\psi^{randneg}(1,1)$, is set to the minimum value $\psi^{min}(1,1)$ with probability p and is set to $\psi^{max}(1,1)$ with probability $(1-p)$. Assume that $p = 0.5$, and suppose that ψ^{min} is chosen for $\psi(1,1)$ and $\psi(2,2)$, and ψ^{max} for the other components. The resulting $\Psi'^{randneg}$, which is shown in Equation 4, is just one of the sixteen possible colluded results.

4. Fingerprinting and Colluder Identification

This section describes our collusion attack detection and colluder identification scheme. Wavelet watermarking is used to maximize fingerprint recovery. Statistical clustering techniques are used to accurately identify

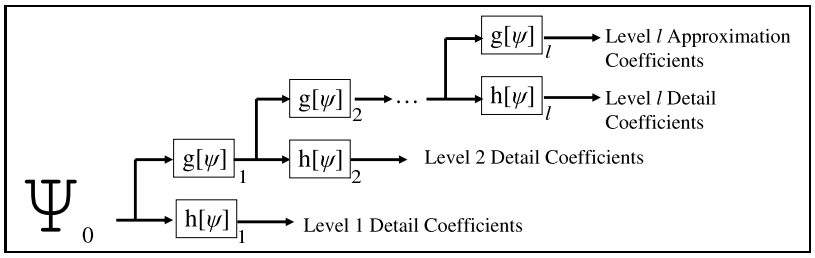

Figure 1. Level ℓ decomposition tree of a filter bank.

large colluder sets while minimizing $max(|C' - C|, |C - C'|)$, where C is the set of real colluders and C' is the set of identified colluders.

4.1 Fingerprint Embedding

Multimedia artifacts can be represented as discrete signals, e.g., an image is represented as a matrix where each pixel location $\{i, j\}$ represents a color value. This property enables the use of the Discrete Wavelet Transform (DWT) [4]. DWT uses a decomposition process to embed fingerprint coefficients [2, 7]. This is done using band-pass arrays called "filter banks." A filter bank is a series of high-pass and low-pass filters that partition the original signal into several components called "sub-bands." The sub-bands can be recombined to recreate the original signal. The decomposition process can be applied 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 a low-pass filter, $g[\psi]$, which provides the approximation coefficients.

Figure 1 illustrates an ℓ-level decomposition tree using a filter bank. Note that the filters decompose the input into low and high frequencies at every level. Figure 2 shows a four-level recursive decomposition of the well-known *Lena* image.

Several robust wavelet-based watermarking methods have been developed [10]. This work employs the constant energy embedding technique because it requires the least amount of computation. In fact, the constant energy embedding technique is used as a baseline in most comparative studies [13]:

$$\psi'_\ell(i, j) = \psi_\ell(i, j) + \alpha \cdot f(i, j) \tag{5}$$

Embedding is performed by processing the multimedia artifact using the DWT with ℓ-levels of decomposition. After extracting the corresponding approximation coefficients, an additive embedding of the wa-

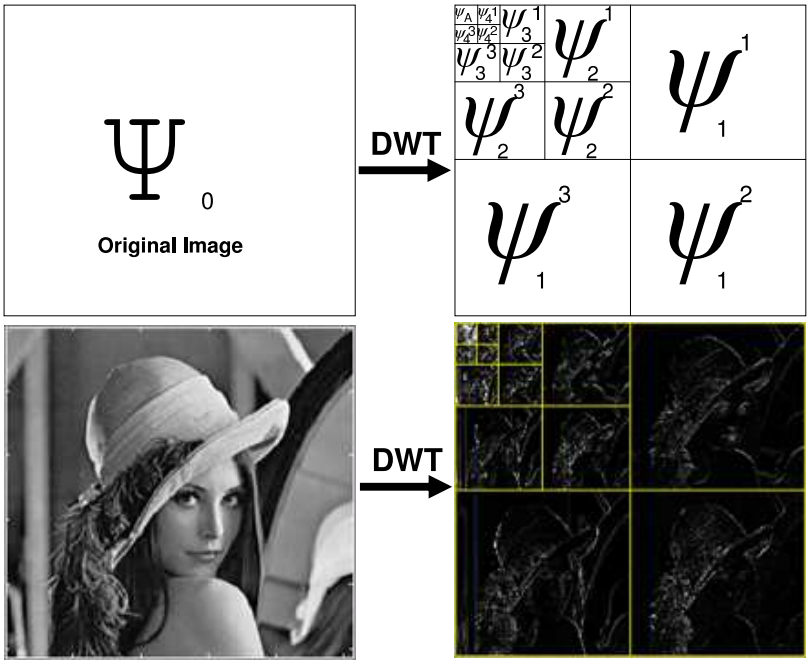

Figure 2. Four-level image decomposition (every ψ_l^n is a sub-band).

termark is performed using the constant energy embedding technique in Equation 5. After the fingerprint has been embedded, the inverse DWT is performed to recreate the original artifact with the embedded watermark.

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, and ℓ specify the decomposition level of the coefficients used to embed the fingerprint. Spread-spectrum sequences [6] or orthogonal codes [12] can be used to generate a fingerprint f. In this study, a set of zero-mean Gaussian distributed random values is used to generate f. Our experimental results indicate that $\alpha = 0.1$ and $\ell = 4$ provide adequate fingerprinting strength with acceptable distortion.

$$\Psi_0 = \begin{bmatrix} 12 & 23 \\ 34 & 45 \end{bmatrix}, f = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \alpha = 2 \tag{6}$$

$$\Psi' = \begin{bmatrix} 12 & 23 \\ 34 & 45 \end{bmatrix} + 2 \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 14 & 23 \\ 36 & 47 \end{bmatrix} \tag{7}$$

Equations 6 and 7 provide an example of the computations involved in fingerprint embedding. Algorithm 1 formalizes the fingerprint embedding process.

Algorithm 1 : Fingerprint Embedding

```
1: ψ ← DWT(Ψ, ℓ)
2: if size(ψ) ≤ size(f) then
3:    "error: fingerprint too large"
4: end if
5: for i ← 1 to rows[ψ] do
6:    for j ← 1 to columns[ψ] do
7:       ψ'_ℓ(i, j) = ψ_ℓ(i, j) + α · f(i, j)
8:    end for
9: end for
10: Ψ' ← iDWT(ψ', ℓ)
11: return Ψ'
```

4.2 Fingerprint Extraction

The non-blind fingerprint extraction process is similar to the fingerprint embedding process. First, the original and fingerprinted artifacts are processed using DWT to extract the approximation coefficients. Next, the difference between these coefficients is calculated using Equation 8. The recovered fingerprint is denoted by f'.

$$f'(i,j) = \frac{1}{\alpha} \cdot \left(\psi'_\ell(i,j) - \psi_\ell(i,j)\right) \tag{8}$$

The fingerprint extraction process is formalized in Algorithm 2.

Algorithm 2 : Fingerprint Extraction

```
1: ψ ← DWT(Ψ, ℓ)
2: ψ' ← DWT(Ψ', ℓ)
3: for i ← 1 to rows[ψ'] do
4:    for j ← 1 to columns[ψ'] do
5:       f'(i, j) = 1/α · (ψ'_ℓ(i, j) − ψ_ℓ(i, j))
6:    end for
7: end for
8: return f'
```

4.3 Colluder Identification

After recovering the colluded fingerprint, f', the correlation coefficient is calculated between two fingerprints, where f is the corresponding fingerprint of a user from a known database. Let $R(f)$ be the correlation

value between f' and f. The set R contains the correlation values between the colluded fingerprint and the fingerprint of each user.

Having determined the set of correlation values R and their corresponding users U, a statistical clustering technique can be applied to identify the colluders involved in the attack. Our scheme uses an iterative *2-means* clustering algorithm to obtain possible partitions in the set of colluders. This algorithm is a specialization of the well-known k-means algorithm [9]. The algorithm classifies the correlation values into two clusters, one is the set of detected colluders, and the other is the set of innocent individuals. Since higher correlation values indicate stronger relationships with the colluded fingerprint, the cluster with the highest mean value is considered to be the colluder set C'.

The clusters are partitioned by minimizing the Euclidean distance between every correlation value $R(f)$. The mean of a cluster is called its "centroid." In our variant of the algorithm, initial centroids are not selected randomly as in other algorithms, but are calculated based on the mean and standard deviation of the set R.

The algorithm computes two centroids for the entire data set R. During each iteration, a user f is assigned a group, C' or B, based on the shortest distance between $R(f)$ 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 members of each group. The 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 that yields f'.

The *2-means* algorithm is summarized below (Algorithm 3).

Algorithm 3 : *2-means* Clustering

1: $\overline{c} \leftarrow mean(R) + stddev(R)$
2: $\overline{b} \leftarrow mean(R) - stddev(R)$
3: **repeat**
4: $C' \leftarrow B \leftarrow \emptyset$
5: **for** $f \leftarrow 1$ to $|R|$ **do**
6: **if** $|R(f) - \overline{c}| = min(|R(f) - \overline{c}|, |R(f) - \overline{b}|)$ **then**
7: Assign $R(f)$ to set C'
8: **else**
9: Assign $R(f)$ to set B
10: **end if**
11: **end for**
12: $\overline{c}_{last} \leftarrow \overline{c}$; $\overline{c} \leftarrow mean(C')$
13: $\overline{b}_{last} \leftarrow \overline{b}$; $\overline{b} \leftarrow mean(B)$
14: **until** $min(|\overline{c} - \overline{c}_{last}|, |\overline{b} - \overline{b}_{last}|) = 0$
15: return C'

Table 2. Three iterations of the *2-means* algorithm.

Iteration 1	$R(1) = 0.25$	$R(2) = 0.40$	$R(3) = 0.65$	$R(4) = 0.85$
$\overline{b} = 0.20$	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'
Iteration 2				
$\overline{b} = 0.25$	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'
Iteration 3				
$\overline{b} = 0.325$	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'

We present a simple example to illustrate the clustering algorithm. In the example, the set R contains users $\{1, 2, 3, 4\}$, and $R(i)$ is the correlation value of user i. The initial values of the centroids, $\overline{b}$ and $\overline{c}$, are 0.20 and 0.35, respectively. During the first iteration of the algorithm, every point in R is assigned to a group based on the least distance to the corresponding centroid. Table 2 shows the group assignments after the first iteration.

After the initial group assignments, the values of the centroids are recalculated as the means of the members of each of the two groups. Therefore, $\overline{b} = 0.25$ and $\overline{c} = 0.63$. The process is repeated for the second iteration using the new centroids, and the new group assignments are presented in Table 2. At the end of Iteration 2, note that $R(2)$ has moved from C' to B. Again, the new centroids are calculated and the process is repeated.

The locations of the centroids are unchanged at the end of Iteration 3. Therefore, the algorithm terminates and C' contains the potential set of colluders because $\overline{c} > \overline{b}$.

This scheme is successful at determining the colluder set C' because colluders cannot obtain the value of the embedded fingerprint in their multimedia artifact. Therefore, they cannot determine which users from set C satisfy the condition $corr(f', f_i) = 1$ for an innocent user u_i. This becomes more difficult for colluders when orthogonal codes [12] and spread-spectrum watermarking [6] are used.

The algorithm is very practical because it treats the correlation values in R as random variables, and finds potential relationships based on joint densities. Therefore, the colluder set is built from observations not predictions. Furthermore, less computational overhead is involved because all possible colluder combinations do not have to be tested.

5. Experimental Results

Our collusion attack detection and colluder identification scheme was evaluated for collusion attacks on the *Lena* image (Figure 2). The fingerprints used were a sequence of pseudo-randomly generated Gaussian distributed values. MatLab 7.0 software was used for the computations.

A set of 400 fingerprinted copies of the *Lena* image were created. The fingerprints were embedded using the constant energy embedding technique and the Daubechies-6 filter. Set C contained a maximum of 200 colluders and set D contained a maximum of 200 innocent individuals. Colluded copies were generated for attacks from Table 1 involving two or more colluders from C. Four levels of decomposition were used for embedding, and the value of α was set to 0.10.

The experimental results show that the colluder identification scheme is highly effective against minimum, maximum and minmax attacks. The scheme works well for the randomized negative attack, although it degrades a little as the number of colluders increases. The scheme does not work quite as well against the average attack when the colluded fingerprint is closer to the mean of the distribution used to generate the fingerprint. The further the colluded fingerprint is from the mean, the better the scheme performs at identifying colluders who have used average attacks.

Figure 3 shows the miss rate of colluders, i.e., the number of colluders that are not detected, as the number of colluders increase.

Figure 4 shows the performance of the scheme with respect to false accusation rates as the number of colluders increase.

In summary, the colluder identification scheme is effective against minimum, maximum and minmax attacks. Also, it provides satisfactory protection for innocent parties in the event of randomized negative attacks.

6. Conclusions

The wavelet-based watermarking scheme and the statistical clustering technique proposed in this paper are useful for detecting and identifying individuals involved in collusion attacks. The wavelet-based watermarking technique provides a high fingerprint recovery rate. The *2-*

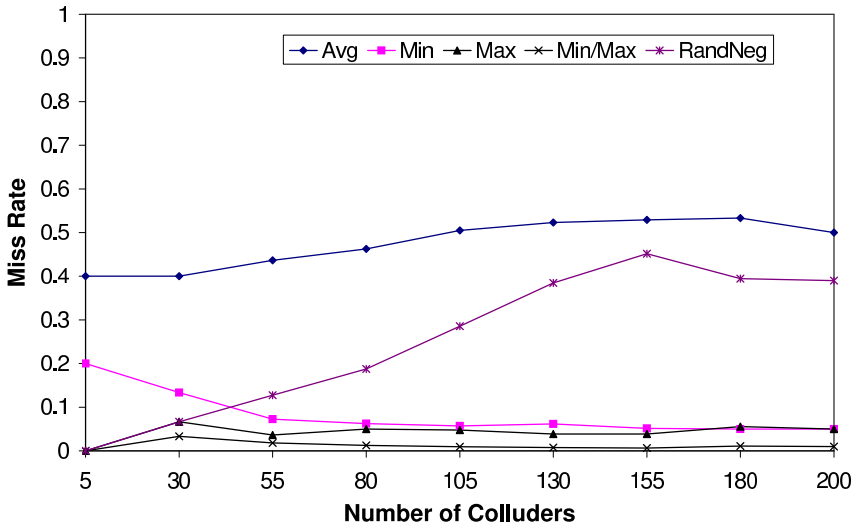

Figure 3. Miss rates with increasing numbers of colluders.

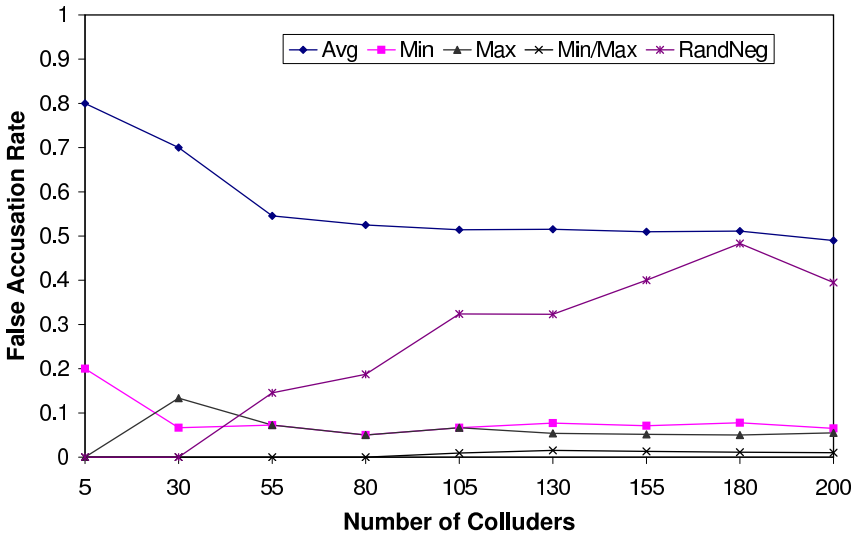

Figure 4. False accusation rates with increasing numbers of colluders.

means clustering algorithm is effective against collusion attacks because it builds the colluder set from value observations, not predictions. The experimental results show that the scheme is effective at thwarting common collusion attacks and determining colluder sets for a large number of colluders.

Acknowledgements

This research was partially supported by NSF Grant DUE-0313837, ARDA Contract NBCHC030107 and the GEM Fellowship Program. The authors also wish to thank Dr. Jennifer Davidson and anonymous reviewers for their advice and comments on earlier versions of this paper.

References

[1] H. Chu, L. Qiao, K. Nahrstedt, H. Wang and R. Jain, A secure multicast protocol with copyright protection, *ACM Computer Communication Review*, vol. 32(2), pp. 42-60, 2002.

[2] I. Cox, J. Bloom and M. Miller, *Digital Watermarking: Principles and Practice*, Morgan Kaufmann, San Mateo, California, 2001.

[3] I. Cox, J. Kilian, T. Leighton and T. Shamoon, Secure spread spectrum watermarking for multimedia, *IEEE Transactions on Image Processing*, vol. 6(12), pp. 1673-1687, 1997.

[4] I. Daubechies, *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992.

[5] P. Judge and M. Ammar, WHIM: Watermarking multicast video with a hierarchy of intermediaries, *Computer Networks*, vol. 39(6), pp. 699-712, 2002.

[6] J. Kilian, T. Leighton, L. Matheson, T. Shamoon, R. Tarjan and F. Zane, Resistance of Digital Watermarks to Collusive Attacks, Technical Report TR-585-98, Department of Computer Science, Princeton University, Princeton, New Jersey, 1998.

[7] M. Kutter, F. Jordan and F. Bossen, Digital watermarking using multiresolution wavelet decomposition, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, pp. 2969-2972, 1998.

[8] A. Lumini and D. Maio, A wavelet-based image watermarking scheme, *Proceedings of the International Symposium on Information Technonogy*, pp. 122-127, 2000.

[9] J. MacQueen, Some methods for classification and analysis of multivariate observations, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281-297, 1967.

[10] P. Meerwald and A. Uhl, A survey of wavelet-domain watermarking algorithms, *Proceedings of SPIE: Electronic Imaging, Security and Watermarking of Multimedia Contents III*, vol. 4314, 2001.

[11] C. Shoemaker, Hidden bits: A survey of techniques for digital watermarking (www.vu.union.edu/∼shoemakc/watermarking/watermarking.html), 2002.

[12] Z. Wang, M. Wu, H. Zhao, W. Trappe and K. Liu, Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation, *IEEE Transactions on Image Processing*, vol. 14, pp. 804-821, 2005.

[13] C. Woo, J. Du and B. Pham, Performance factors analysis of a wavelet-based watermarking method, *Proceedings of the Third Australasian Information Security Workshop*, pp. 89-98, 2005.

[14] M. Wu, W. Trappe, Z. Wang and K. Liu, Collusion resistant fingerprinting for multimedia, *IEEE Signal Processing Magazine*, pp. 15-27, March 2004.

[15] X. Xia, C. Boncelet and G. Arce, Wavelet transform based watermark for digital images, *Optics Express*, vol. 3(12), pp. 497-511, 1998.

[16] H. Zhao, M. Wu, Z. Wang and K. Liu, Nonlinear collusion attacks on independent fingerprints for multimedia, *Proceedings of the International Conference on Multimedia and Expo*, vol. 1, pp. 613-616, 2003.