JOINT CODING AND EMBEDDING FOR COLLUSION-RESISTANT FINGERPRINTING

Wade Trappe, Min Wu, and K. J. Ray Liu

Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742

ABSTRACT

An effective attack against fingerprints for multimedia is collusion, where several differently marked copies of the same content are averaged or combined to disrupt the underlying fingerprint. In this paper, we investigate the problem of designing fingerprints that can withstand collusion and allow for the identification of colluders. We begin by introducing the collusion problem for additive embedding, and introduce an efficient detection algorithm for orthogonal modulation that identifies the fingerprints associated with K colluders and requires $\mathcal{O}(K \log(n/K))$ correlations for a group of n users. We present a construction of collusion-resistant fingerprints based upon anti-collusion codes (ACC) and binary code modulation. Using ACC, we build fingerprints that identify groups of K or less colluders. We present a construction of binary-valued ACC under the logical AND operation using the theory of combinatorial designs. Our code construction requires only $\mathcal{O}(\sqrt{n})$ orthogonal signals to accommodate n users. We demonstrate the performance of our ACC for fingerprinting multimedia and identifying colluders through experiments using Gaussian signals.

1. INTRODUCTION

The rapid advancement of communication networks and multimedia technologies has created a need for mechanisms that ensure that content is used for its intended purpose, and by legitimate users with appropriate distribution rights. Digital fingerprinting is an effective tool used to control the redistribution of content. These fingerprints can be embedded in multimedia content through a variety of robust watermarking techniques[1, 2]. However, a cost-efficient attack against watermarking can be waged by a coalition of users with the same content that contains different marks. One of the simplest approaches to performing such a *collusion* attack is to average multiple copies of the content together[3]. Other collusion attacks might involve forming a new content by selecting different pixels or blocks from the different colluders' content. By gathering a large enough coalition of colluders, it is possible to sufficiently attenuate each of the colluders' identifying fingerprints and produce a new version of the content with no detectable fingerprints. It is therefore important to design fingerprints that are not only able to resist collusion, but also identify the colluders.

In this paper, we investigate the problem of making fingerprints for multimedia content that are resistant to averaging-based collusion attacks. Our investigation differs from collusion-resistant schemes for generic data sources by incorporating the special properties of multimedia, such as the embedding method and appropriate choice of detection statistics. In Section 2 we describe multimedia fingerprinting, and introduce the problem of user collusion for a class of additive watermark schemes. We then review orthogonal modulation in Section 3, describe the effect that collusion has upon the constellation points of the modulation scheme, and present a detection algorithm with reduced complexity. In Section 4, we present our design of anti-collusion codes (ACC), which are used in conjunction with binary code modulation to construct fingerprints that are resistant to collusion and able to identify members of a colluder set. Our approach is suitable for both averaging-based collusion attacks, and for interleaving collusion attacks. Finally, we present conclusions in Section 5.

2. FINGERPRINTING AND COLLUSION

We first review additive embedding. Suppose that the host signal is a vector denoted as $\mathbf{x}$ and that we have a family of watermarks $\{\mathbf{w}_j\}$ that are fingerprints associated with the different users who purchase the rights to access $\mathbf{x}$. Before the watermarks are added to the host signal, every component of each $\mathbf{w}_j$ is scaled by an appropriate factor that corresponds to an amplification, i.e. $\mathbf{s}_j(k) = \alpha(k)\mathbf{w}_j(k)$, where we refer the the kth component of a vector $\mathbf{w}_j$ by $\mathbf{w}_j(k)$. Corresponding to each user is a marked version of the content $\mathbf{t}_j = \mathbf{x} + \mathbf{s}_j$, which typically experiences additional distortion $\mathbf{z}_j$ that is due to such factors as compression and attacks made to remove the embedded fingerprints. We will denote the combination of the noise and the interference of the original signal by $\mathbf{d}_j = \mathbf{x} + \mathbf{z}_j$. We can thus assume that each user will be given a marked content $\mathbf{y}_j = \mathbf{s}_j + \mathbf{d}_j$. Typically, the watermarks $\{\mathbf{w}_j\}$ are chosen to correspond to orthogonal noise-like signals [1], or are constructed using code modulation and represented using a basis of orthogonal noise-like signals $\mathbf{u}_i$ via $\mathbf{w}_j = \sum_{i=1}^{v} b_{ij}\mathbf{u}_i$, where $b_{ij} \in \{0, 1\}$ or $b_{ij} \in \{\pm 1\}$.

We can identify a user who is redistributing marked content $\mathbf{y}_j$ by detecting the watermark associated with the user to whom $\mathbf{y}_j$ was sold. The detection of additive watermarks $\mathbf{w}_j$ or the corresponding codes $\{b_{ij}\}$ can be formulated as a hypothesis testing problem, where the embedded data is considered as the signal that is to be detected in the presence of noise. If the distribution of the components of $\mathbf{d}_j$ is modelled as independent Gaussian, the optimal detector is a set of correlators of $\mathbf{y}$ and $\mathbf{u}_i$ with proper normalization. The detector can be further refined with more realistic statistical model for $\mathbf{d}_j$.

When two parties who have the same image but fingerprinted differently come together, they can perform a collusion attack to generate a new image from the two fingerprinted images so that the traces of either fingerprint in the new image is attenuated. For fingerprinting through additive embedding, this can be done by averaging the two fingerprinted images $\mathbf{y}_c = \lambda_1\mathbf{y}_1 + \lambda_2\mathbf{y}_2$ where $\lambda_1 + \lambda_2 = 1$, so that the energy of each of the fingerprints is reduced to ${\lambda_i}^2$ of the corresponding original and the detection statistics with respect

The authors may be contacted at wxt, minwu, and kjrliu @eng.umd.edu.

to the i-th fingerprint is scaled by a factor of λ_i. In a K-colluder averaging-collusion the watermarked content signals $\mathbf{y}_j$ are combined according to $\sum_{j=1}^{K} \lambda_j \mathbf{y}_j$. The objective of each colluder is to avoid being detected, yet remain fair to his fellow colluders and retain good image quality. We have shown in [4], that under realistic assumptions about the detection statistics for each user, choosing $\lambda_j = 1/K$ for all j is the most fair choice for each colluder to avoid detection.

3. ORTHOGONAL MODULATION AND COLLUSION

In orthogonal modulation, there are v orthogonal signals $\mathbf{s}_j$ that are used to convey a $B = \log_2 v$ bit ID by inserting one of the v signals into the host signal. The effect of collusion on orthogonal modulation is studied by calculating the distance between the constellation points and averages of the constellation points, as well as the distance between the averages of the constellation points and the origin. Suppose each watermark is embedded using $\mathcal{E}$ energy. If we average K watermarks, then the distance from the colluded mark to any of the watermarks used in forming it is $\sqrt{\mathcal{E}(K-1)/K}$. The distance from the colluded mark to any of the other watermarks not used in the collusion is $\sqrt{\mathcal{E}(K+1)/K}$. Further, the distance of the colluded mark from the origin is $\sqrt{\mathcal{E}/K}$. Thus, as K increases, the watermarks in the colluded mark will become harder to detect.

The classical method for estimating which signal was embedded in the host signal is done via $v = 2^B$ correlators. The fact that detection complexity is linear in the amount of signals was considered a major drawback of the method of orthogonal modulation[1].

For a set $A = \{\mathbf{w}_j\}_{j \in J}$ where J is an indexing set, we define the sum of A by $SUM(A) = \sum_{j \in J} \mathbf{w}_j$. We present a recursive algorithm for efficiently detecting the identity of K colluders as follows: Let us denote by $S = \{\mathbf{w}_j\}$ the set of orthogonal watermark signals, and suppose the test signal is $\mathbf{y}$. At each stage we divide S into two non-overlapped sets S_0 and S_1, and perform a correlation of $\mathbf{y}$ against $SUM(S_0)$ and $SUM(S_1)$, respectively. If a set passes a threshold test, we further decompose it and test the correlations. We repeat until we are no longer able to decompose further, at which point we output the element in the corresponding set. There are many possible choices for dividing S into S_0 and $S_1 = S \backslash S_0$ in such an algorithm. For example, if we choose S_0 such that $|S_0| = 2^{\lceil \log_2 |S| \rceil - 1}$, then the number of correlations, denoted as $C(n, K)$, that must be performed to identify K signals in a test signal $\mathbf{y}$ satisfies $C(n, K) \leq 2\left(-1 + K\left(\log_2(2^{\lceil \log_2 n \rceil}/K) + 1\right)\right)$. This is an improvement over the traditional linear computational complexity and is demonstrated by our experiments described in detail in [4].

4. CODE MODULATION EMBEDDING AND ACC

A drawback for using orthogonal modulation in data embedding is the large number of orthogonal signals needed to convey B bits. In this section we use code modulation to convey more bits of information for a given amount of basis vectors than orthogonal modulation. We use this modulation technique, in conjunction with appropriately designed codewords, known as anti-collusion codes, to construct a family of fingerprints that have the ability to identify members of the colluding set of users.

In code modulation, there are v orthogonal basis signals $\{\mathbf{u}_j\}$, and information is encoded into a watermark signal $\mathbf{w}_j$ via $\mathbf{w}_j = \sum_{j=1}^{v} b_{ij} \mathbf{u}_i$, where $b_{ij} \in \{0, 1\}$ or $b_{ij} \in \{\pm 1\}$. The first of the two possibilities for choosing the values of b_{ij} corresponds to on-off keying (OOK) while the second choice of $\{\pm 1\}$ corresponds to an antipodal form. The determination of each b_{ij} is done by correlating with the $\mathbf{u}_i$, and comparing against a decision threshold.

We assign a different bit sequence $\{b_{ij}\}$ for each user u_j. We may view the assignment of the bits b_{ij} for different watermarks in a matrix $\mathbf{B}$, which we call the *derived* code matrix, where each column of $\mathbf{B}$ contains a *derived* codevector for a different user. In the following section, we shall design a code matrix $\mathbf{C}$ whose elements are either 0 or 1. By applying a suitable mapping depending on whether the OOK or antipodal form of code modulation is used, the code matrix $\mathbf{C}$ is used to derive the matrix $\mathbf{B}$,

4.1. Anti-Collusion Codes

In this section we design a family of codevectors $\{\mathbf{c}_j\}$ whose overlap with each other can identify groups of colluding users. A similar idea was proposed in [5], where projective geometry was used to construct such code sequences. As we will explain in this section, our proposed code construction makes more efficient usage of the basis vectors than the codes described in [5].

We assume, when a sequence of watermarks is averaged and detection is performed, that the detected binary sequence is the logical AND of the codevectors $\mathbf{c}_j$ used in constructing the watermarks. For example, when the watermarks corresponding to the codevectors (1110) and (1101) are averaged, the output of the detector is (1100). This assumption might not necessarily hold since the average of many 1's and a few 0's may produce a decision statistic large enough to pass through the detector as a 1.

We want codes that can identify up to K colluders. We prefer shorter codes since longer codes would distribute the fingerprint energy over more basis vectors, which would lead to a higher error rate in the detection process. To identify colluders, we first require that there is some non-zero component remaining in the code when the codes for these K colluders are combined. Secondly, we require that there are no repetitions in the different combinations of K or fewer codevectors. We call codes that satisfy these properties anti-collusion codes.

Definition 1. *A binary code $\mathcal{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_n\}$ such that the logical AND of any subset of k or fewer codevectors is non-zero and distinct from the logical AND of any other subset of k or fewer codevectors is a k-resilient AND anti-collusion code.*

We now present a construction of a K-resilient AND-ACC that requires only $\mathcal{O}(\sqrt{n})$ basis vectors for n users. Our construction uses balanced incomplete block designs (BIBD)[6]. A (v, k, λ)-BIBD has $n = \lambda(v^2 - v)/(k^2 - k)$ blocks. Corresponding to a block design is the $v \times n$ incidence matrix $\mathbf{M} = (m_{ij})$, where m_{ij} is 1 if the ith element belongs to the jth block, and 0 otherwise. If we define the codematrix $\mathbf{C}$ as the bit-complement of $\mathbf{M}$, and assign the codevectors $\mathbf{c}_j$ as the columns of $\mathbf{C}$, then we have a $(k-1)$-resilient AND-ACC[4]. Our codevectors are therefore v-dimensional, and we are able to accommodate $n = \lambda(v^2 - v)/(k^2 - k)$ users. Assuming that a BIBD exists, for n users we therefore need $v \approx \mathcal{O}(\sqrt{n})$ basis vectors. In general, (v, k, λ)-BIBDs do not necessarily exist for an arbitrary choice of v and k. The existence of different BIBDs, and techniques for constructing BIBDs can be found in [6].

A useful metric for evaluating the efficiency of an AND-ACC for a given resiliency is its rate $R = v/n$, which describes the amount of basis vectors needed per user. AND-ACCs with lower rates are better. For (v, k, λ)-BIBD AND-ACC, their rate is $R =$

```
Algorithm: $SuspectAlg(\mathbf{\Gamma})$
$\mathbf{\Phi} = \mathbf{1}$;
Define $J$ to be the set of indices where $\Gamma_j = 1$ ;
for $t = 1$ to $|J|$ do
    $j = J(t)$ ;
    Define $\mathbf{e}_j$ to be the $j$th row of $\mathbf{C}$;
    $\mathbf{\Phi} = \mathbf{\Phi} \cdot \mathbf{e}_j$;
end
```

Algorithm 1: Algorithm $SuspectAlg(\mathbf{\Gamma})$, which determines the vector $\mathbf{\Phi}$ that describes the suspect set.

$(k^2 - k)/(\lambda(v-1))$. By Fisher's Inequality[6], we also know that $n \geq v$ for a (v, k, λ)-BIBD, and thus $R \leq 1$ using the BIBD construction. In contrast, the k-resilient construction in [5] has rate much larger than 1, and thus requires more spreading sequences (or marking locations) to accommodate the same amount of users as our scheme. It is possible to use the collusion-secure code constructions of [7] in conjunction with code modulation for embedding. However, the construction described in [7] has codelength $\mathcal{O}(\log^4 n \log^2(1/\epsilon))$, where $\epsilon < 1/n$ is the decision error probability. This codelength is considerably large for small error probabilities and practical n values. Additionally, for the same amount of users, the use of code modulation watermarking with an AND-ACC constructed using a $(v, k, 1)$-BIBD requires v orthogonal sequences for $n = (v^2 - v)/(k^2 - k)$ users, while orthogonal modulation discussed in Section 3 would require n sequences.

Given that the output of the detector is a vector $\mathbf{\Gamma} = (\Gamma_1, \Gamma_2, \cdots, \Gamma_n)$, we would like to narrow down the entire user set to a subset of suspect users by using $\mathbf{\Gamma}$ to determine a *suspicious* set from the entire user set. In Algorithm 1, we determine a vector $\mathbf{\Phi} = (\Phi_1, \Phi_2, \cdots, \Phi_n) \in \{0,1\}^n$ that describes the suspicious set via the location of components whose value are 1. Thus, if $\Phi_j = 1$, then the jth user is suspected of colluding. In the algorithm, we denote the jth row vector of $\mathbf{C}$ by $\mathbf{e}_j$, and use the fact that the element-wise multiplication "$\cdot$" of the binary vectors corresponds to the logical AND operation. The algorithm starts with $\mathbf{\Gamma}$ and $\mathbf{\Phi} = \mathbf{1}$, where $\mathbf{1}$ is the n dimensional vector consisting of all ones. The algorithm then uses the indices where $\mathbf{\Gamma}$ is equal to 1, and narrows down the suspicious set through updates to $\mathbf{\Phi}$ by performing the AND of $\mathbf{\Phi}$ with the rows of the code matrix $\mathbf{C}$ corresponding to indices where $\mathbf{\Gamma}$ is 1.

We now focus on the detector involved in detecting collusion for binary code modulation. Suppose that a codevector $\mathbf{c}_j$ has weight $\omega = wt(\mathbf{c}_j)$. In the OOK case the remaining $v - \omega$ positions would be zeros, while in the antipodal case the remaining $v - \omega$ positions would be -1. If we allocate $\mathcal{E}$ energy to this codevector, then the OOK case would use $\mathcal{E}/\omega$ energy to represent each 1, while the antipodal case would use $\mathcal{E}/v$ energy to represent each ± 1. The amplitude separation between the constellation points for the 0 and 1 in OOK is $\sqrt{\mathcal{E}/\omega}$, while the separation between -1 and 1 in antipodal is $2\sqrt{\mathcal{E}/v}$. Since it is desirable to have the separation between the constellation points as large as possible, we should choose OOK only when $\omega < v/4$. In the AND-ACCs presented in Section 4.1, the weight of each codevector is $\omega = v - k$. OOK is advantageous when $(3/4)v < k$, and antipodal modulation is preferable otherwise. Typically, in BIBDs with $\lambda = 1$, $k << v$ and therefore the antipodal form is preferred.

If K colluders come together and average their marked content, then they produce an averaged test signal $\mathbf{y}$ whose contribution in the $\mathbf{u}$ component is the average of the b_{ij} values for that basis vector. For example, in the antipodal case, the b_{ij} are either -1 or 1 and therefore the values $-1, -(K-2)/K, -(K-4)/K, \cdots, (K-4)/K, (K-2)/K, 1$ are possible for the average $\bar{b}$ of the b_{ij} values for a basis vector $\mathbf{u}$. From these possibilities, it is clear that larger values of K are undesirable from a detection point-of-view. In the antipodal case, the separation between the $\bar{b} = (K-2)/K$ and $\bar{b} = 1$ hypotheses is critical to the validity of using AND as the binary operation in designing an ACC. In order to strengthen the validity of the AND assumption for a K-resilient AND-ACC, the separation between the $\bar{b} = (K-2)/K$ and $\bar{b} = 1$ hypotheses can be increased by devoting more energy $\mathcal{E}$ to the watermark, or by increasing the coding gain though employing longer orthogonal basis vectors $\{\mathbf{u}_j\}$.

4.2. ACC Simulations with Gaussian Signals

In this section we study the behavior of our AND-ACC when used with code modulation in an abstract model, where $\mathbf{y}_j = \mathbf{x} + \mathbf{s}_j = \mathbf{x} + \alpha \sum_{i=1}^{v} b_{ij} \mathbf{u}_i$. The host signal $\mathbf{x}$ and the orthogonal basis signals $\mathbf{u}_i$ are assumed to be independent and each of them are vectors of i.i.d. Gaussian samples. In the simulations that follow, we used a $(16, 4, 1)$ BIBD to construct our AND-ACC code[6] and used the antipodal form of code modulation. The $(v, 4, 1)$ codes exist if and only if $v \equiv 1$ or $4 \pmod{12}$ and can uniquely identify up to $K = 3$ colluders. Our host signal $\mathbf{x}$ was a $N = 10000$ point vector whose components were Gaussian $\mathcal{N}(0,1)$. The scaling factor α was applied equally to each component of the watermark, and was determined from the desired WNR $= 10 \log_{10} \|\mathbf{s}\|^2 / \|\mathbf{x}\|^2$dB.

We first studied the behavior of the detector and the legitimacy of the AND logic for the detector under the collusion scenario. We randomly selected 3 users as colluders and averaged their marked content signals to produce $\mathbf{y}_c$. The colluded content signal was correlated using T_N.

For three colluders, there are 4 possible values for $\bar{b}$, namely $-1, -1/3, 1/3$, and 1. We refer to the cases $-1, -1/3$ and $1/3$ as the non-1 hypothesis. We calculated $p(1|1)$ and $p(1|\text{non-1})$ as a function of WNR for different thresholds. The thresholds used were $\tau_1 = 0.9E(T_N)$, $\tau_2 = 0.7E(T_N)$, and $\tau_3 = 0.5E(T_N)$. To calculate $E(T_N)$, we assumed that the detector knows the the WNR and hence the power of the distortion. The plot of $p(1|1)$ for different threshold strategies is presented in Figure 1(a), and the plot of $p(1|\text{non-1})$ is presented in Figure 1(b). We observe that for the smaller threshold of $0.5E(T_N)$ the probability $p(1|1)$ is higher, but at an expense of a higher probability of false classification $p(1|\text{non-1})$. Increasing the threshold allows us to decrease the probability of falsely classifying a bit as a 1, but at an expense of decreasing the probability of correctly classifying a bit as a 1.

We calculated the fraction of colluders that were captured as well as the fraction of the total group that were falsely placed under suspicion for different WNRs and different thresholds. We assumed that there were always 3 colluders, which were randomly selected from the entire user set. We used Algorithm 1 to determine the set of suspicious users. The fraction of the colluders that belong to the suspicious set, and the fraction of the total user set that are innocents falsely placed under suspicion were calculated and averaged over 2000 realizations for each WNR. The results are presented in Figure 2. Compared to lower thresholds, for all WNRs, the higher threshold is able to capture more of the colluders, but also places more innocent users falsely under suspicion. As WNR increases, the

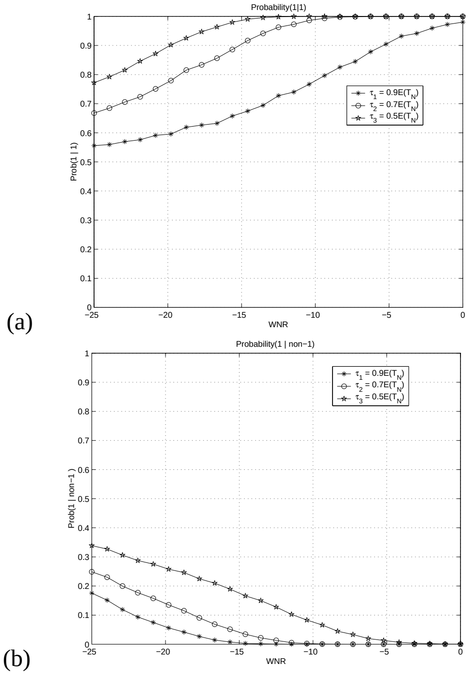

Fig. 1. (a) The probability of detection $p(1|1)$ and (b) the probability of false alarm $p(1|\text{non-1})$.

detector has lower $p(1|\text{non-1})$, and therefore does not incorrectly eliminate colluders from suspicion. Similarly, at higher WNR, the detector has a higher $p(1|1)$, thereby correctly identifying more 1's, which allows for us to eliminate more innocents from suspicion. Therefore, at higher WNR we can capture more colluders as well as place less innocent users under suspicion. We note, however, that in Figure 2(b), at low WNR between -25dB and -15dB, the fraction of innocents under suspicion using threshold τ_1 is lower than at slightly higher WNR. This behavior can be explained by examining Figure 1(a) and Figure 1(b). We observe that at low WNR, the $p(1|\text{non-1})$ is higher than slightly higher WNR, particularly for the threshold τ_1. However, for τ_1 the $p(1|1)$ at these WNR is relatively flat. These two observations combined indicate that at lower WNR we falsely decide 1 more often than at slightly higher WNR, while we do not experience much difference in the amount of correctly identified 1's. As more 1's pass through the detector we remove more users from suspicion. Since the amount of correctly detected 1's is roughly constant for WNRs between -25dB and -15dB, the additional 1's from false detections at lower WNR eliminates more innocent users (as well as colluders) from suspicion.

5. CONCLUSION

In this paper, we investigated the problem of making fingerprints for multimedia content that are resistant to collusion attacks. We developed an efficient detection scheme for orthogonal modulation that is able to identify K colluders in an amount of correlations that is logarithmic in the number of orthogonal signals. Further, we developed a fingerprinting scheme based upon code modulation that requires only $\mathcal{O}(\sqrt{n})$ orthogonal signals to accommodate n users. We proposed anti-collusion codes (ACC) that are used in conjunction with modulation to fingerprint multimedia sources. We constructed binary-valued ACC under the logical AND operation using combinatorial designs, and is suitable for both the OOK and antipodal form of binary code modulation. We performed experiments to evaluate the proposed ACC-based fingerprints. We used a Gaussian signal model to examine the ability of the ACC to identify the colluders, as well as reveal the amount of innocent users that would be falsely placed under suspicion. We observed that decreasing WNR increases the amount of false 1s that pass through the detector, which leads to a small amount of colluders captured at low WNR. By raising the threshold, we improve the ability to capture colluders at all WNR, but also increase the amount of innocents who are falsely placed under suspicion. The ACC fingerprinting proposed in this paper has also been applied to natural images and shown effective in tracing traitors and colluders [4].

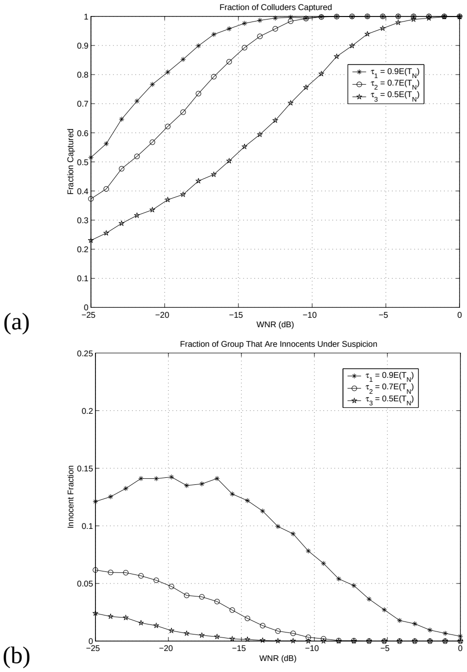

Fig. 2. (a) The fraction of colluders placed under suspicion, and (b) the fraction of the total group falsely placed under suspicion.

6. REFERENCES

[1] I. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Tran. on Image Proc.*, vol. 6(12), pp. 1673–1687, December 1997.

[2] C. Podilchuk and W. Zeng, "Image adaptive watermarking using visual models," *IEEE Journal on Selected Areas in Communications*, vol. 16(4), pp. 525–540, May 1998.

[3] H. S. Stone, "Analysis of attacks on image watermarks with randomized coefficients," *NEC Technical Report 96-045*, 1996.

[4] W. Trappe, M. Wu, and K.J.R. Liu, "Anti-collusion fingerprinting for multimedia," *Submitted to IEEE Trans. on Signal Processing*, 2001, (Preprint available at www.eng.umd.edu/˜wxt/papers/accfingerprint_sp_v5.pdf).

[5] J. Dittmann, P. Schmitt, E. Saar, J. Schwenk, and J. Ueberberg, "Combining digital watermarks and collusion secure fingerprints for digital images," *SPIE Journal of Electronic Imaging*, vol. 9, pp. 456–467, 2000.

[6] C. J. Colbourn and J. H. Dinitz, *The CRC Handbook of Combinatorial Designs*, CRC Press, 1996.

[7] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Tran. on Information Theory*, vol. 44, pp. 1897–1905, September 1998.