ANTI-COLLUSION CODES: MULTI-USER AND MULTIMEDIA PERSPECTIVES

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

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

ABSTRACT

Digital fingerprinting is an effective method to identify users who might try to redistribute multimedia content, such as images and video. These fingerprints are typically embedded into the content using watermarking techniques that are designed to be robust to a variety of attacks. A cheap and effective attack against such digital fingerprints is collusion, where several differently marked copies of the same content are averaged or combined to disrupt the underlying fingerprint. We present a construction of collusion-resistant fingerprints based upon anti-collusion codes (ACC) and binary code modulation. ACC have the property that the composition of any subset of K or fewer codevectors is unique. Using this property, we build fingerprints that allow for the identification of 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 by identifying colluders through experiments using real images.

1. INTRODUCTION

The development of ubiquitous broadband communication networks and multimedia technologies will lead to the creation of a digital marketplace where a broad range of multimedia content, such as images, video, audio and speech, will be available. However, before viable businesses can be established to market content on these networks, mechanisms must be in place to ensure that content is used for its intended purpose, and by legitimate users who have purchased appropriate distribution rights.

In order to control the redistribution of content, digital fingerprinting is used to trace the consumers who use their content for unintended purposes[1]. These fingerprints can be embedded in multimedia content through a variety of watermarking techniques[2, 3]. *Collusion* attacks provide a cost-effective approach to removing an identifying watermark. One of the simplest approaches to performing a collusion attack is to average multiple copies of the content together[4]. 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 resist collusion and identify the colluders, thereby discouraging attempts at collusion by the users.

The problem of designing fingerprints that are resistant to collusion has been considered for generic digital data in [1]. Such generic schemes, however, do not consider the actual marking process associated with specific applications and media types. Indeed, the design of collusion-resistant fingerprinting should consider application-specific issues such as the inherent, special properties of multimedia data since the fingerprinting process for multimedia involves a chain of events including the selection of the embedding method and appropriate choice of detection statistics.

In this paper, we investigate the problem of making fingerprints for multimedia content, such as images and video, that are resistant to collusion attacks by averaging. In Section 2 we describe multimedia fingerprinting, and introduce the problem of user collusion for a class of additive watermark schemes. In Section 3, 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. The proposed ACC concept is applicable to all multimedia data types. For the convenience of discussion, however, we will use images in our experiments, since the extension to audio or video is straightforward. Finally, we present conclusions in Section 4.

2. FINGERPRINTING AND COLLUSION

In this section, we will review additive embedding, where a watermark signal is added to a host signal. 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 noiselike signals [2], or are constructed using code modulation and represented using a basis of orthogonal noiselike signals $\mathbf{u}_i$ via

$$\mathbf{w}_j = \sum_{i=1}^{v} b_{ij}\mathbf{u}_i, \qquad (1)$$

where $b_{ij} \in \{0, 1\}$ or $b_{ij} \in \{\pm 1\}$[5]. The first case corresponds to the on-off keying form of code modulation, and makes less effi-

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

cient usage of energy than the second case, which corresponds to the antipodal form of code modulation. Therefore, we shall only consider $b_{ij} \in \{\pm 1\}$ for the remainder of the paper.

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 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. For the popular spread spectrum embedding [2, 3], the detection performance can be studied via the following simplified antipodal model:

$$\begin{cases} H_0 : y_i = -s_i + d_i \quad (i = 1, ..., N) \qquad \text{if } b = -1 \\ H_1 : y_i = +s_i + d_i \quad (i = 1, ..., N) \qquad \text{if } b = +1 \end{cases} \tag{2}$$

where $\{s_i\}$ is a deterministic spreading sequence (often called the *watermark*), b is the one bit to be embedded and is used to antipodally modulate s_i, d_i is the total noise, and N is the number of samples/coefficients to carry the hidden information. In non-blind detection, where the original source is available, d_i comes from the processing and/or attacks; in blind detection, d_i consists of the host media as well as distortion from processing and attacks. If d_i is modelled as i.i.d. Gaussian $\mathcal{N}(0, \sigma_d^2)$, the optimal detector is a (normalized) correlator with a detection statistics T_N given by

$$T_N \quad = \quad \mathbf{y}^T\mathbf{s}/\sqrt{\sigma_d^2 \cdot \|\mathbf{s}\|^2} \tag{3}$$

where $\mathbf{y} = [y_1, ..., y_N]^T$, $\mathbf{s} = [s_1, ..., s_N]^T$ and $\|\mathbf{s}\|$ is the Euclidean norm of $\|\mathbf{s}\|$. Under the i.i.d. Gaussian assumption for d_i, T_N is Gaussian distributed with unit variance and a mean value $E(T_N) = b \cdot \sqrt{\|\mathbf{s}\|^2/\sigma_d^2}$.

The i.i.d. Gaussian noise assumption is critical for the optimality of a correlator-type detector, but it may not reflect the statistical characteristics of the actual noise and interference. For example, the noise and interference in different frequency bands are different. In such a scenario, we should first normalize the observations $\{y_i\}$ by the corresponding noise standard deviation to make the noise distribution i.i.d. before taking the correlation. That is,

$$T'_N \quad = \quad \sum_{i=1}^{N} \frac{y_i \cdot s_i}{\sigma_{d_i}^2} / \sqrt{\sum_{i=1}^{N} \frac{{s_i}^2}{\sigma_{d_i}^2}}. \tag{4}$$

In this paper, we use the correlator with normalized noise variance as described in (4).

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 to the i-th fingerprint is scaled by a factor of λ_i. Collusion can be extended to more than two parties. 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 [6], that under realistic assumptions about the detection statistics for each user, choosing $\lambda_j = 1/K$ for all j is that most fair choice for each colluder to avoid detection.

3. CODE MODULATION EMBEDDING AND ANTI-COLLUSION CODES

In this section we construct fingerprints using code modulation and appropriately designed codewords to construct a family of fingerprints that have the ability to identify the members of a 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 (1) where $b_{ij} \in \{\pm 1\}$. At the detector side, 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 the ith user. 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 $\mathbf{C}$ with values from $\{0, 1\}$ and map $\mathbf{C} \rightarrow \mathbf{B}$ with values $\{\pm 1\}$ prior to use in code modulation.

In binary code modulation, if we average two watermarks, $\mathbf{w}_1$ and $\mathbf{w}_2$ corresponding to bit sequences b_{j1} and b_{j2}, then when $b_{j1} \neq b_{j2}$ the contributions cancel. However, when $b_{j1} = b_{j2}$ the contributions do not attenuate.

3.1. 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 [7], 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 [7].

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 to design codes such that when K or fewer users collude, we can identify the 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 $\mathcal{C} = \{c^1, \cdots, c^n\}$ 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, or an AND-ACC code.*

We present a $(n-1)$-resilient AND-ACC. Let $\mathcal{C}$ consist of all n-bit binary vectors that have only a single 0 bit. For example, when $n = 4$, $\mathcal{C} = \{1110, 1101, 1011, 0111\}$. It is easy to see when $K \leq n-1$ of these vectors are combined under AND, that this combination is unique. This code has cardinality n, and can produce at most n differently watermarked media. It is desirable to shorten the codelength to squeeze more users into fewer basis

```
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.

vectors. We now present a construction of a K-resilient AND-ACC that requires $\mathcal{O}(\sqrt{n})$ basis vectors for n users.

Our construction uses balanced incomplete block designs (BIBD)[8]. 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 $\mathbf{M} = (m_{ij})$ defined by

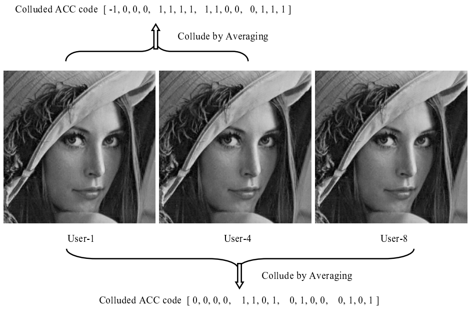
$$m_{ij} = \begin{cases} 1 & \text{if the } i\text{th element belongs to the } j\text{th block,} \\ 0 & \text{otherwise.} \end{cases}$$

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[6]. Our codevectors are therefore v-dimensional, and we are able to accommodate $n = \lambda(v^2 - v)/(k^2 - k)$ users with these v basis vectors. 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 [8].

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 $R = (k^2 - k)/(\lambda(v-1))$. By Fisher's Inequality[8], 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 [7] 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 [1] in conjunction with code modulation for embedding. However, the construction described in [1] 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 signaling would require n sequences.

3.2. Detector Design and Performance

In this section we focus on the detector involved in detecting collusion for binary code modulation. In order for the detector to determine whether $\mathbf{u}_i$, $-\mathbf{u}_i$, or neither exists in the test signal $\mathbf{y}$, the detector correlates $\mathbf{y}$ with $\mathbf{u}_i$. If K colluders come together and average their marked content, then they produce an averaged test signal $\mathbf{y}$ whose contribution in the $\mathbf{u}_i$ component is the average of the b_{ij} values for that basis vector. The values of $-1, -(K-2)/K, -(K-4)/K, \cdots, (K-4)/K, (K-2)/K, 1$ are possible for the average $\bar{b}$ of the b_{ij} values for a particular basis vector $\mathbf{u}_i$. From these possibilities, it is clear that larger values of K are undesirable from a detection point-of-view. 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 $\{\mathbf{u}_j\}$.


Colluded ACC code [-1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]
Collude by Averaging
User-1
User-4
User-8
Collude by Averaging
Colluded ACC code [0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1]

User 1:	-1,-1,-1,-1,1,1, 1,1, 1,1, 1,1 ,1,1,1,1
User 4:	-1, 1, 1, 1,1,1, 1,1, 1,1,-1,-1,-1,1,1,1
User 8:	1,-1, 1, 1,1, 1,-1,1,-1,1,1,1,1,1,-1,1
User(1,4) Average:	-1, 0, 0, 0,1,1, 1,1, 1,1, 0,0,0,1,1,1
User(1,4,8) Average:	-$\frac{1}{3}$,-$\frac{1}{3}$,$\frac{1}{3}$,$\frac{1}{3}$,1,1,$\frac{1}{3}$,1 ,$\frac{1}{3}$,1,$\frac{1}{3}$,$\frac{1}{3}$,$\frac{1}{3}$,1,$\frac{1}{3}$,1
After thresholding:	0, 0, 0, 0,1,1, 0,1, 0,1,0,0,0,1,0,1

Fig. 1. Illustration of collusion by averaging two and three images fingerprinted with ACC codes, respectively. Also presented are the derived codevectors from a $(16, 4, 1)$ AND-ACC for user 1, 4, and 8 as well as example vectors from two collusion scenarios.

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.

3.3. ACC Experiments with Images

In order to demonstrate the performance of our AND-ACC with code modulation fingerprinting on real images for fingerprinting users and detecting colluders, we used an additive spread spectrum watermarking scheme similar to that in [3], where the perceptually weighted watermark was added to 8×8 block DCT coefficients. The detection of the watermark is performed without the knowledge of the host image via the detection statistics as shown in (4). In the simulations, we used a $(16, 4, 1)$ BIBD [8] to construct our AND-ACC code. The 512×512 Lenna and Baboon images were

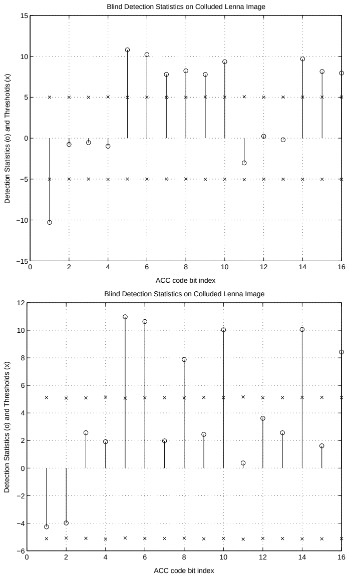


Fig. 2. Example blind detection statistics values for 2 users' and 3 users' collusion with a $(16, 4, 1)$-BIBD AND-ACC fingerprint. (top) User 1 and 4 perform averaging, resulting in the output of the detector as $(-1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1)$. (bottom) User 1, 4, & 8 perform averaging, resulting in the output of the detector as $(0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1)$.

used as the host signals for the fingerprints. The fingerprinted images have no visible artifacts with an average PSNR of 41.2dB for Lenna, and 33.2dB for Baboon.

Two collusion examples, as well as the three derived code vectors that were assigned to user 1, 4, and 8, and the colluded versions are presented in Figure 1. The detection statistics of the two examples are shown in Figure 2. The colluded images are further compressed using JPEG with quality factor (QF) 50%. Also shown in Figure 2 are the thresholds determined from the estimated mean of the detection statistics $E(T_N)$. We then estimated the fingerprint codes by thresholding the detection statistics, and the estimated fingerprint codes are identical to the expected ones. In Figure 3, we present histograms of the T_N statistics from several collusion cases with different distortions applied to the colluded Lenna images. We see that there is a clear distinction between the three decision regions, facilitating the accurate determination of the AND-ACC codes from colluded images.

4. CONCLUSION

In this paper, we investigated the problem of making fingerprints for multimedia content that are resistant to collusion attacks. We proposed anti-collusion codes (ACC) that are used in conjunction with modulation to fingerprint multimedia sources. Our anti-collusion codes have the property that the composition of any subset of K or fewer codevectors is unique, which allows for the identification of subgroups of K or fewer colluders. We constructed binary-valued ACC under the logical AND operation using combinatorial designs. Our codes are efficient in that they require only $\mathcal{O}(\sqrt{n})$ orthogonal signals to accommodate n users. For practical values of n this is an improvement over prior work on fingerprinting generic digital data. We evaluated our fingerprints on

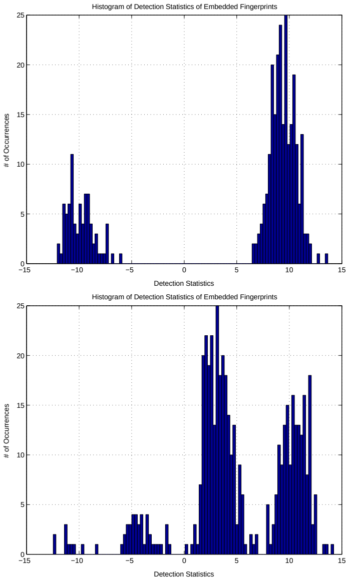


Fig. 3. Histograms of blind detection statistics of embedded fingerprints: (top) single fingerprint case, (bottom) 3-colluder case.

real images, and observed that the values of the detection statistics were well-separated. This behavior allows the detector to accurately determine the colluders by correctly extracting a fingerprint codevector that corresponds to the colluder set.

5. REFERENCES

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

[2] 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.

[3] 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.

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

[5] M. Wu and B. Liu, "Modulation and multiplexing techniques for multimedia data hiding," in *Proc. of SPIE ITcom'01, SPIE vol 4518*, Aug. 2001.

[6] 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).

[7] 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.

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