On Anti-Collusion Codes and Detection Algorithms for Multimedia Fingerprinting

著者	Cheng Minquan, Miao Ying
journal or publication title	IEEE transactions on information theory
volume	57
number	7
page range	4843-4851
year	2011-07
権利	(C) 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works
URL	http://hdl.handle.net/2241/113559

doi: 10.1109/TIT.2011.2146130

On Anti-Collusion Codes and Detection Algorithms for Multimedia Fingerprinting

Minquan Cheng and Ying Miao*

***Abstract*—Multimedia fingerprinting is an effective technique to trace the sources of pirate copies of copyrighted multimedia information. AND anti-collusion codes can be used to construct fingerprints resistant to collusion attacks on multimedia contents. In this paper, we first investigate AND anti-collusion codes and related detection algorithms from a combinatorial viewpoint, and then introduce a new concept of logical anti-collusion code to improve the traceability of multimedia fingerprinting. It reveals that frameproof codes have traceability for multimedia contents. Relationships among anti-collusion codes and other structures related to fingerprinting are discussed, and constructions for both AND anti-collusion codes and logical anti-collusion codes are provided.**

***Index Terms*—AND anti-collusion code (AND-ACC), detection algorithm, disjunct matrix, frameproof code, logical anti-collusion code (LACC), multimedia fingerprinting, NAGTA, separable code, separable matrix.**

I. Introduction

THE advancement of multimedia technologies, coupled with the development of an infrastructure of ubiquitous broadband communication networks, has led to a tremendous use of multimedia contents in digital marketplace. However, such an advantage also poses the challenge of insuring that multimedia contents are appropriately used, especially in view of the ease of copying and manipulating multimedia data. Devising techniques for copyright protection of multimedia contents has been an urgent problem to be solved.

In order to hinder the unauthorized redistribution of digital data, digital fingerprinting was introduced to trace the authorized customers who redistribute their contents for unintended purposes [1]. Fingerprints for multimedia data can be embedded through a variety of watermarking techniques prior to their authorized distribution [6], [14]. Nowadays attacks mounted by individuals are no longer a main security issue in digital rights management. The global nature of the Internet makes authorized customers with differently marked versions of the same content easy to work together and collectively mount attacks against the fingerprints. Multiuser collusion attacks provide a cost-effective approach to remove the embedded fingerprints. One of the most feasible approaches to perform a collusion attack is to average multiple copies of the content together [18]. Other collusion attacks might involve creating a new content by selecting different pixels or blocks from different colluders' marked contents. By gathering a large enough coalition of colluders, in an improperly designed embedding and identification scheme, it is possible to produce a colluded version of the content where the colluders' fingerprints are sufficiently attenuated so that tracing and identifying the colluders becomes impossible. It is desirable, therefore, to design fingerprints that can resist collusion and identify the colluders, thereby discouraging attempts at collusion by the authorized customers.

Especially, the averaging attack is a serious problem in multimedia fingerprinting. The averaging attack is an attempt to remove the embedded fingerprints by averaging all the fingerprinted signals with an equal weight for each colluder, so that no colluder would take more of a risk than any other colluders. This attack reduces the power of each contributing fingerprint and makes the colluded signal have better perceptual quality. When the sizes of potential coalitions are small, a usual watermarking method that embeds orthogonal signals as watermarks, called orthogonal fingerprinting, can overcome the averaging attack. However, as the size of coalition increases, the limitation of orthogonal fingerprinting is clear, and a more sophisticated fingerprint design for multimedia data is eagerly expected.

Trappe et al. [20], [21] introduced the notion of an AND anti-collusion code (AND-ACC) against the averaging attack, and proposed a construction for AND-ACCs by using the bit complement of the incidence matrix of a balanced incomplete block design. A similar idea was proposed in [8], where projective geometries were used to construct such anti-collusion codes. Constructions via other mathematical structures such as cover-free families can be found in [10]. Li and Trappe [11] investigated collusion-resistant fingerprints from sequence sets satisfying the Welch bound equality.

In this paper, we investigate AND-ACCs from the standpoint of combinatorial group testing. We first provide a combinatorial characterization of an AND-ACC in terms of a separable matrix, a mathematical structure used in combinatorial group testing and satellite communications [9], [7]. As an immediate consequence of this equivalence, all constructions for separable matrices can be adopted to AND-ACCs. In fact, all the known constructions for AND-ACCs are special cases of those for separable matrices. We introduce the notion of a separable code, show that separable codes are closely related with separable matrices, and then discuss the relationships between AND-ACCs and other codes related to digital fingerprinting. We also investigate the problem of detecting colluders when AND-ACCs are used with code modulation to construct

This work was supported by JSPS Grant-in-Aid for Scientific Research (C) under Grant No. 21540108.

Minquan Cheng and Ying Miao are with Department of Social Systems and Management, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan. E-mail: {mqcheng, miao}@sk.tsukuba.ac.jp.

Manuscript received ?????; revised ?????.

multimedia fingerprints. We point out some flaws of the hard detection algorithm based on AND-ACCs proposed in [19], and describe a revised detection algorithm based on special AND-ACCs. The highlight of this paper is the introduction of a new anti-collusion code called logical anti-collusion code (LACC), where not only the logical AND operation but also the logical OR operation is exploited to identify colluders. We find an equivalence between a binary LACC and a binary separable code. We then describe an efficient identification algorithm based on LACCs constructed from frameproof codes. Frameproof codes were widely considered as having no traceability for generic digital data (see for example [17]). However, our result shows that frameproof codes actually have traceability for multimedia contents. This greatly strengthens the importance of frameproof codes in fingerprinting. Finally we provide a few constructions for separable codes and frameproof codes.

The paper is organized as follows. In Section 2, we review the basic concepts of fingerprinting, collusion and detection. In Sections 3 and 4, we discuss the properties of AND-ACCs and detection algorithms based on AND-ACCs. In Section 4, we investigate LACCs and detection algorithms based on LACCs. Conclusion is drawn in Section 5.

II. Fingerprinting, Collusion, and Detection

In this section, for the convenience of readers, we recapitulate some basic concepts of fingerprinting, collusion, and detection. The interested reader is referred to [12] for more detailed information.

In general, collusion-resistant fingerprinting requires the design of fingerprints that can survive collusion attacks to trace and identify colluders, as well as robust embedding of fingerprints into multimedia host signals. Spread-spectrum additive embedding is a widely employed robust embedding technique [6], [14], which is nearly capacity optimal when the host signal is available in detection [4], [13]. Its capability of putting multiple marks in overlapped regions also limits the effective attack strategies mountable by colluders [23]. In spread-spectrum embedding, a watermark signal, often represented by noise-like orthonormal basis signals, is added to the host signal. As usual, all signals are regarded as vectors in some signal spaces. Now let $\mathbf{x}$ be the host multimedia signal, and $\{\mathbf{w}_j = (\mathbf{w}_j(1), \mathbf{w}_j(2), \ldots, \mathbf{w}_j(n)) \mid 1 \le j \le M\}$ be a family of watermarks that are fingerprints associated with different users U_j, $1 \le j \le M$, who have purchased the rights to access $\mathbf{x}$. In practical watermarking, before $\mathbf{w}_j$ is added to $\mathbf{x}$, every of its coordinates is usually scaled by an appropriate factor to achieve the imperceptibility as well as to control the energy of the embedded watermark, where the factor can be chosen according to the just-noticeable-difference from human visual model [14]. Each authorized user U_j, $1 \le j \le M$, is then assigned with a watermarked version of the content $\mathbf{y}_j = \mathbf{x} + \mathbf{w}_j$. The fingerprints $\mathbf{w}_j$, $1 \le j \le M$, are often chosen to be noise-like orthonormal signals [6], or are built by a linear modulation scheme employing an orthonormal basis $\{\mathbf{u}_i \mid 1 \le i \le n\}$ of noise-like signals via $\mathbf{w}_j = \sum_{i=1}^{n} b_{ij}\mathbf{u}_i$, where $b_{ij} \in \{0, 1\}$, which corresponds to the on-off keying form of code modulation, or $b_{ij} \in \{-1, 1\}$, which corresponds to the antipodal form of code modulation [22], [21]. Since signals represented by a linear combination of noise-like orthonormal $\mathbf{u}_i$, $1 \le i \le n$, can distinguish different users' fingerprints $\mathbf{w}_j$ to the maximum extent [12], and usually antipodal form of code modulation makes more efficient usage of energy, in this paper, we only consider the case $\mathbf{w}_j = \sum_{i=1}^{n} b_{ij}\mathbf{u}_i$ where $b_{ij} \in \{-1, 1\}$.

The fingerprint $\mathbf{w}_j$ assigned to the authorized customer U_j can be represented uniquely by a vector $\mathbf{b}_j = (b_{1j}, b_{2j}, \ldots, b_{nj})^T \in \{-1, 1\}^n$ because of the linear independence of the basis $\{\mathbf{u}_i \mid 1 \le i \le n\}$. The $n \times M$ matrix $B = (b_{ij})$, with column j corresponding to the fingerprint $\mathbf{w}_j$ for user U_j, $1 \le j \le M$, is the derived code matrix of the fingerprints $\{\mathbf{w}_j \mid 1 \le j \le M\}$. Since distinct code matrices correspond to distinct fingerprinting strategies, we would like to strategically design a code matrix to accurately identify the contributing fingerprints involved in collusion attacks. In the remainder of this paper, before code modulation, we will first design an $n \times M$ code matrix $C = (c_{ij})$ with entries from $\{0, 1\}$, then transform C to B by $c_{ij} \longmapsto b_{ij} = 2c_{ij} - 1$. In code modulation phase, information is encoded into a watermark signal $\mathbf{w}_j$ via $\mathbf{w}_j = \sum_{i=1}^{n} b_{ij}\mathbf{u}_i$ with $b_{ij} \in \{-1, 1\}$.

When t authorized users, say $U_{j_1}, U_{j_2}, \ldots, U_{j_t}$, who have the same host content but distinct fingerprints come together, we assume that they have no way of manipulating the individual orthonormal signals, that is, the underlying codeword needs to be taken and proceeded as a single entity, but they can carry on a linear collusion attack to generate a pirate copy from their t fingerprinted contents, so that the venture traced by the pirate copy can be attenuated. For fingerprinting through additive embedding, this is done by linearly combining the t fingerprinted contents $\sum_{l=1}^{t} \lambda_{j_l}\mathbf{y}_{j_l}$, where the weights $\{\lambda_{j_l} \mid 1 \le l \le t\}$ satisfy the condition $\sum_{l=1}^{t} \lambda_{j_l} = 1$ to maintain the average intensity of the original multimedia signal. In such a collusion attack, the energy of each of the watermarks $\mathbf{w}_{j_l}$ is reduced by a factor of $\lambda_{j_l}^2$, therefore the trace of U_{j_l}'s fingerprint becomes weaker and thus U_{j_l} is less likely to be caught by the detector. In fact, since normally no colluder is willing to take more of a risk than any other colluder, the fingerprinted signals are typically averaged with an equal weight for each user. Averaging attack choosing $\lambda_{j_l} = 1/t$, $1 \le l \le t$, is the most fair choice for each colluder to avoid detection, as claimed in [18], [21]. This attack also makes the pirate copy have better perceptional quality.

Any circulated copy of the host multimedia content may experience an additional distortion $\mathbf{z}$ before it is tested for the existence of a fingerprint. This additional noise $\mathbf{z}$ could be due to the effect of compression or from an attack mounted by colluders in an attempt to hinder the detection of the fingerprint. Based on the averaging attack model, the observed content $\mathbf{y}$ after collusion is

$$\mathbf{y} = \frac{1}{t}\sum_{l=1}^{t} \mathbf{y}_{j_l} + \mathbf{z} = \frac{1}{t}\sum_{l=1}^{t} \mathbf{w}_{j_l} + \mathbf{x} + \mathbf{z},$$

where $\mathbf{z}$ is assumed to follow an i.i.d. Gaussian distribution with zero mean and variance $\sigma_{\mathbf{z}}^2$ in this paper. For simplicity

of notation, we can combine $\mathbf{x}$ and the possible distortion $\mathbf{z}$ into a single term denoted by $\mathbf{d}$. Therefore,

$$\mathbf{y} = \frac{1}{t}\sum_{l=1}^{t}\mathbf{w}_{j_l} + \mathbf{x} + \mathbf{z} = \frac{1}{t}\sum_{l=1}^{t}\mathbf{w}_{j_l} + \mathbf{d} = \sum_{l=1}^{t}\sum_{i=1}^{n}\frac{b_{ij_l}}{t}\mathbf{u}_i + \mathbf{d}.$$

The problem of detecting colluders can be posed in a hypothesis testing framework where fingerprints are signals to be detected in the presence of noise. Due to the orthogonality of the orthonormal basis $\{\mathbf{u}_i \mid 1 \le i \le n\}$, in colluder detection phase, we only need to consider the correlation vector $\mathbf{T} = (\mathbf{T}(1), \mathbf{T}(2), \ldots, \mathbf{T}(n))$, where $\mathbf{T}(i) = \frac{1}{\sigma_{\mathbf{z}}}\langle \mathbf{y}, \mathbf{u}_i\rangle$, $1 \le i \le n$, and $\langle \mathbf{y}, \mathbf{u}_i\rangle$ is the inner product of $\mathbf{y}$ and $\mathbf{u}_i$. It is straightforward to check that

$$\mathbf{T} = \frac{1}{t\sigma_{\mathbf{z}}}(B\mathbf{\Phi})^T + \frac{1}{\sigma_{\mathbf{z}}}(\langle \mathbf{d}, \mathbf{u}_1\rangle, \ldots, \langle \mathbf{d}, \mathbf{u}_n\rangle),$$

where the vector $\mathbf{\Phi} \in \{0,1\}^M$ indicates colluders via the location of the coordinates whose value is 1. Three detection strategies, namely, hard detection, adaptive sorting approach, and sequential algorithm, were provided in [12] to efficiently estimate the colluder vector $\mathbf{\Phi}$.

III. AND ACCs and Related Detection Algorithms

In [20], [21], the notion of an AND anti-collusion code (AND-ACC) was introduced for protecting multimedia contents, which, with code modulation, can be used to construct a family of fingerprints with the ability to survive collusion and trace colluders. Let n, M and q be positive integers, and Q an alphabet with $|Q| = q$. A set $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_M\} \subseteq Q^n$ is called an (n, M, q) code and each $\mathbf{c}_i$ is called a codeword. Without loss of generality, we may assume $Q = \{0, 1, \ldots, q-1\}$. When $Q = \{0, 1\}$, we also use the word "binary". Given an (n, M, q) code, the incidence matrix $M(\mathcal{C})$ is the $n \times M$ matrix on $Q = \{0, 1, \ldots, q-1\}$ in which the columns are the M codewords in $\mathcal{C}$. Often, we make no difference between an (n, M, q) code and its incidence matrix unless otherwise stated.

Intuitively, we say that an (n, M, q) code $\mathcal{C}$ is an anti-collusion code, briefly ACC(n, M, q), if the coordinates shared between codewords uniquely identify sets of codewords. Below we give a formal definition of AND anti-collusion codes.

Definition 3.1: An $(n, M, 2)$ code $\mathcal{C}$ is a t-resilient AND-ACC, briefly t-AND-ACC$(n, M, 2)$, if the bitwise AND of any subset of t or fewer codewords of $\mathcal{C}$ is distinct from the bitwise AND of any other subset of t or fewer codewords of $\mathcal{C}$.

AND-ACCs are closely related to separable matrices.

Definition 3.2: A binary matrix S is a $\bar{t}$-separable matrix if the bitwise OR of any subset of t or fewer column vectors of S is distinct from the bitwise OR of any other subset of t or fewer column vectors of S. If for any column, the number of 1s is a constant k, then we say S is a k-uniform $\bar{t}$-separable matrix.

For any $(n, M, 2)$ code $\mathcal{C}$, its complementary code $\overline{\mathcal{C}}$ is defined to be

$$\overline{\mathcal{C}} = \{\overline{\mathbf{c}} = (\overline{\mathbf{c}(1)}, \ldots, \overline{\mathbf{c}(n)})^T \mid \mathbf{c} = (\mathbf{c}(1), \ldots, \mathbf{c}(n))^T \in \mathcal{C}\},$$

where, as usual,

$$\overline{\mathbf{c}(j)} = \begin{cases} 1, & \text{if } \mathbf{c}(j) = 0, \\ 0, & \text{if } \mathbf{c}(j) = 1. \end{cases}$$

Theorem 3.3: An $(n, M, 2)$ code $\mathcal{C}$ is a t-AND-ACC$(n, M, 2)$ if and only if the incidence matrix of its complementary code $\overline{\mathcal{C}}$ is an $n \times M$ $\bar{t}$-separable matrix.

Proof: The result follows from the facts that $\overline{\bigwedge_{\mathbf{c}\in\mathcal{C}_0}\mathbf{c}} = \bigvee_{\mathbf{c}\in\mathcal{C}_0}\overline{\mathbf{c}}$ and $\overline{\bigvee_{\mathbf{c}\in\mathcal{C}_0}\mathbf{c}} = \bigwedge_{\mathbf{c}\in\mathcal{C}_0}\overline{\mathbf{c}}$ for any $\mathcal{C}_0 \subseteq \mathcal{C}$, where $\bigwedge$ and $\bigvee$ are logical AND and OR, respectively. ∎

From Theorem 3.3, we immediately know that to construct t-AND-ACCs, we only need to construct their equivalent $\bar{t}$-separable matrices. There are systematic methods for constructing infinite families of $\bar{t}$-separable matrices [7], which thus provide a vast supply of t-AND-ACCs. In fact, almost all t-AND-ACCs constructed so far (see, for example, [12] and references therein) can be found in [7] under the disguise of $\bar{t}$-separable matrices.

As was pointed out in [12], constructing fingerprints is only half of the battle in battling illicit content manipulation and redistribution. It is also essential to devise instruments that will allow content distributors to effectively identify those authorized users involved in creating pirate copies. In the remaining of this section, we discuss the problem of identifying colluders when AND-ACCs are used with code modulation to construct multimedia fingerprints.

An algorithm to identify colluders was described in [9], [7] by means of a $\bar{t}$-separable matrix used as the complement of the incidence matrix of a t-AND-ACC. For any pirate copy created by averaging attack from t or fewer users, that algorithm can identify the set of colluders in time $\Theta(nM^t)$, where M is the total number of authorized users and n is the length of the code. For large values of t, that algorithm is clearly not sufficiently efficient. A more efficient algorithm is definitely desired.

Trappe et al. [19] developed the following hard detection algorithm based on a t-AND-ACC$(n, M, 2)$ to find a suspicious set of colluders in time $O(nM)$. Here, the outcome vector $\mathbf{y} = (\mathbf{y}(1), \ldots, \mathbf{y}(n))^T$ is obtained by applying hard thresholding to the detection statistics $\mathbf{T}(i)$ such that $\mathbf{y}(i) = 1$ if $\mathbf{T}(i) > \tau_{\mathbf{a}}$ and $\mathbf{y}(i) = 0$ otherwise, where $\tau_{\mathbf{a}}$ is the threshold appropriately determined by the detector. In this algorithm, the operation $\cdot$ denotes the bitwise AND of binary vectors.

Algorithm 1: `HardDetAlg`

Define J to be the set of indices where $\mathbf{y}(j) = 1$ and $\mathbf{J} = (\mathbf{J}(1), \ldots, \mathbf{J}(|J|))$ to be the vector representing $\mathbf{y}$'s non-zero coordinates
$\mathbf{\Phi} = \mathbf{1}^T$;
for $t = 1$ **to** $|J|$ **do**
  $j = \mathbf{J}(t)$;
  Define $\mathbf{e}_j$ to be the jth row of C;
  $\mathbf{\Phi} = \mathbf{\Phi} \cdot \mathbf{e}_j^T$;
end
return $\mathbf{\Phi}$

We should note that `HardDetAlg` cannot identify the set

of colluders if it is based only on a conventional AND-ACC. In fact, it is even not an algorithm which can effectively narrow the suspicious set of colluders if the number of colluders is greater than t. The following shows such examples.

Example 3.4: We first consider a $(7,7,2)$-code $\mathcal{C}_1 = \{\mathbf{c}_{11}, \ldots, \mathbf{c}_{17}\}$, where $\mathbf{c}_{11} = (0,0,1,0,1,1,1)^T$, $\mathbf{c}_{12} = (1,0,0,1,0,1,1)^T$, $\mathbf{c}_{13} = (1,1,0,0,1,0,1)^T$, $\mathbf{c}_{14} = (1,1,1,0,0,1,0)^T$, $\mathbf{c}_{15} = (0,1,1,1,0,0,1)^T$, $\mathbf{c}_{16} = (1,0,1,1,1,0,0)^T$, and $\mathbf{c}_{17} = (0,1,0,1,1,0,0)^T$. It is easily checked that $\mathcal{C}_1$ is a 2-AND-ACC$(7,7,2)$, so we can identify up to 2 colluders by means of $\mathcal{C}_1$. For example, suppose that user U_1, who is assigned with $\mathbf{c}_{11}$, and user U_7, who is assigned with $\mathbf{c}_{17}$, come together to carry out a collusion attack. Then the outcome vector detected is $\mathbf{y} = (0,0,0,0,1,0,0)^T$, which is different from the logical AND of any subset of up to 2 codewords, so U_1 and U_7 are identified precisely as the colluders. However, algorithm `HardDetAlg` cannot identify U_1 and U_7 correctly. It can only output $\mathbf{\Phi} = (1,0,1,0,0,1,1)^T$, that is, it can only show that U_1, U_3, U_6 and U_7 are suspicious colluders.

Next we consider another $(7,7,2)$-code $\mathcal{C}_2 = \{\mathbf{c}_{21}, \ldots, \mathbf{c}_{27}\}$, where $\mathbf{c}_{21} = (0,0,1,0,1,1,1)^T$, $\mathbf{c}_{22} = (1,0,0,1,0,1,1)^T$, $\mathbf{c}_{23} = (1,1,0,0,1,0,1)^T$, $\mathbf{c}_{24} = (1,1,1,0,0,1,0)^T$, $\mathbf{c}_{25} = (0,1,1,1,0,0,1)^T$, $\mathbf{c}_{26} = (1,0,1,1,1,0,0)^T$, and $\mathbf{c}_{27} = (0,1,0,1,1,1,0)^T$. It is again straightforward to check that $\mathcal{C}_2$ is a 2-AND-ACC$(7,7,2)$, and we can identify up to 2 colluders. However, if users U_1, U_5 and U_7, assigned with $\mathbf{c}_{21}$, $\mathbf{c}_{25}$ and $\mathbf{c}_{27}$, respectively, come together to carry out a collusion attack, then the outcome vector detected is $\mathbf{y}' = (0,0,0,0,0,0,0)^T$, and `HardDetAlg` can only output $\mathbf{\Phi} = (1,1,1,1,1,1,1)^T$, that is, all users are suspicious colluders.

To make `HardDetAlg` complete, we need the notion of a t-disjunct matrix. A binary vector $\mathbf{x}$ is said to cover another binary vector $\mathbf{y}$ of the same length if whenever $\mathbf{y}$ has a 1 in the ith coordinate, so does $\mathbf{x}$.

Definition 3.5: A binary matrix D is called a t-disjunct matrix if the bitwise OR of any subset of t column vectors of D does not cover any other column vector of D. If for any column of D, the number of 1s is a constant k, we say D is a k-uniform t-disjunct matrix.

Disjunct matrices are closely related to separable matrices, as the following Lemma 3.6 shows. The first half of Lemma 3.6 is obvious, while the second half was proved in [9].

Lemma 3.6: Any t-disjunct matrix is a $\bar{t}$-separable matrix. Conversely, any $\bar{t}$-separable matrix is a $(t-1)$-disjunct matrix.

The interested reader is referred to [7] for more relationships between separable matrices and disjunct matrices.

Separable matrices and disjunct matrices can be described in terms of set systems. A set system is a pair $(X, \mathcal{B})$, where $X = \{x_1, \ldots, x_v\}$ is a set of elements called points, and $\mathcal{B} = \{B_1, \ldots, B_b\}$ is a set of subsets of X called blocks. The incidence matrix of $(X, \mathcal{B})$ is the $v \times b$ binary matrix $A = (a_{ij})$ defined by

$$a_{ij} = \begin{cases} 1, & \text{if } x_i \in B_j, \\ 0, & \text{if } x_i \notin B_j. \end{cases}$$

Conversely, given an incidence matrix, we can define an associated set system in an obvious way.

The following descriptions can be easily seen by regarding separable matrices (or disjunct matrices, respectively) in the same light as incidence matrices.

Lemma 3.7: Let A be a binary $v \times b$ matrix. Then A is a $\bar{t}$-separable matrix if and only if A is the incidence matrix of a set system $(X, \mathcal{B})$ where $|X| = v$, $|\mathcal{B}| = b$, and for any two distinct subsets $\mathcal{B}_1$, $\mathcal{B}_2$ of $\mathcal{B}$ with $|\mathcal{B}_1| \le t$, $|\mathcal{B}_2| \le t$, it holds that $\bigcup_{B \in \mathcal{B}_1} B \neq \bigcup_{B \in \mathcal{B}_2} B$.

Lemma 3.8: Let A be a binary $v \times b$ matrix. Then A is a t-disjunct matrix if and only if A is the incidence matrix of a set system $(X, \mathcal{B})$ where $|X| = v$, $|\mathcal{B}| = b$, and for any subset $\mathcal{B}_0 \subseteq \mathcal{B}$ with $|\mathcal{B}_0| \le t$ and for any $A \in \mathcal{B} \setminus \mathcal{B}_0$, it holds that $A \not\subseteq \bigcup_{B \in \mathcal{B}_0} B$.

The dual of a set system $(X, \mathcal{B})$ is the set system $(\mathcal{B}, X)$ where $B \in \mathcal{B}$ is contained in $x \in X$ if and only if $x \in X$ is contained in $B \in \mathcal{B}$.

Definition 3.9: A set system $(X, \mathcal{B})$ is called a non-adaptive group testing algorithm, or briefly t-NAGTA(v, b), if $|X| = v$, $|\mathcal{B}| = b$, and for any two distinct subsets X_1, X_2 of X with $|X_1| \le t$, $|X_2| \le t$, it holds in the dual of $(X, \mathcal{B})$ that $\bigcup_{x \in X_1} x \neq \bigcup_{x \in X_2} x$.

Clearly, an $n \times M$ $\bar{t}$-separable matrix corresponds to the transpose of the incidence matrix of a t-NAGTA(M, n).

The following well-known detection algorithm `IdenAlg` (see [16] for example) based on a t-NAGTA(M, n), or equivalently a t-AND-ACC$(n, M, 2)$, with computational complexity $O(nM)$, is an improved version of `HardDetAlg`. Given the outcome vector $\mathbf{y} = (\mathbf{y}(1), \ldots, \mathbf{y}(n))^T$, which is obtained by applying hard thresholding to the detection statistics $\mathbf{T}(i)$, algorithm `IdenAlg` will output a suspicious set of colluders U if $|U| \le t$ and report that $|U| > t$ otherwise, where C is the transpose of the complement of the incidence matrix of a t-NAGTA(M, n).

Algorithm 2: `IdenAlg`

Define J to be the set of indices where $\mathbf{y}(j) = 1$ and $\mathbf{J} = (\mathbf{J}(1), \ldots, \mathbf{J}(|J|))$ to be the vector representing $\mathbf{y}$'s non-zero coordinates
$\mathbf{\Phi} = \mathbf{1}^T$;
$U = \emptyset$;
for $t = 1$ **to** $|J|$ **do**
  $j = \mathbf{J}(t)$;
  Define $\mathbf{e}_j$ to be the jth row of C;
  $\mathbf{\Phi} = \mathbf{\Phi} \cdot \mathbf{e}_j^T$;
end
for $i = 1$ **to** M **do**
  if $\Phi(i) = 1$ then
   $U = \{i\} \cup U$;
  end
end
if $|U| \le t$, then
  output U;
else
  output "the set of colluders has size at least $t + 1$".
end

The case where algorithm `IdenAlg` outputs the exact set

of colluders U when $U \leq t$ is of more interest.

Definition 3.10: A t-NAGTA(v, b), $(X, \mathcal{B})$, is said to be strong if for any subset $X' \subseteq X$ with $|X'| \leq t$, it holds that $\texttt{IdenAlg}(\mathbf{y}(X')) = X'$, where $\mathbf{y}(X')$ is the outcome vector of X' in the dual of $(X, \mathcal{B})$. $(X, \mathcal{B})$ is said to be k-uniform if for any $x \in X$, it holds in the dual of $(X, \mathcal{B})$ that $|x| = k$. $(X, \mathcal{B})$ is proper if for any $B \in \mathcal{B}$, it holds that $|B| \geq 2$.

Example 3.11: We consider the set system $(X_2, \mathcal{B}_2)$ corresponding to the 2-AND-ACC$(7, 7, 2)$, $\mathcal{C}_2$, in Example 3.4, where $X_2 = \{1, 2, \ldots, 7\}$ and $\mathcal{B}_2 = \{B_1, \ldots, B_7\}$ with $B_1 = \{1, 5, 7\}$, $B_2 = \{1, 2, 6\}$, $B_3 = \{2, 3, 7\}$, $B_4 = \{1, 3, 4\}$, $B_5 = \{2, 4, 5\}$, $B_6 = \{3, 5, 6\}$, $B_7 = \{4, 6, 7\}$. It is easy to check that $(X_2, \mathcal{B}_2)$ is a proper and strong 3-uniform 2-NAGTA$(7, 7)$.

Theorem 3.12: The transpose of an $n \times M$ (k-uniform) t-disjunct matrix is equivalent to the incidence matrix of a strong (k-uniform) t-NAGTA(M, n).

Proof: Let $\mathcal{D}$ be an arbitrary subset of column vectors of an $n \times M$ matrix D with $|\mathcal{D}| \leq t$. The result follows from the fact that a column vector $\mathbf{d}'$ of D is included in $\mathcal{D}$ if and only if $\overline{\mathbf{d}'}$ covers $\bigwedge_{\mathbf{d} \in \mathcal{D}} \overline{\mathbf{d}}$, that is, $\bigvee_{\mathbf{d} \in \mathcal{D}} \mathbf{d}$ covers $\mathbf{d}'$, where the equivalence of the uniformity is obvious. ■

Theorem 3.12 shows that if we use an $n \times M$ t-disjunct matrix instead of an $n \times M$ $\bar{t}$-separable matrix, then we can identify the exact set $\mathcal{C}_0$ of colluders in time $O(nM)$ provided that $|\mathcal{C}_0| \leq t$.

We are more interested in strong NAGTAs. We notice that the strong 2-NAGTA$(7, 7)$, $(X_2, \mathcal{B}_2)$, in Example 3.11 is in fact a balanced incomplete block design $(7, 3, 1)$-BIBD.

Definition 3.13: A balanced incomplete block design, or briefly (v, k, λ)-BIBD, is a set system $(X, \mathcal{A})$ with $|X| = v$ such that each block in $\mathcal{A}$ contains exactly $k \geq 2$ points of X, and every parir of distinct points in X is contained in exactly λ blocks of $\mathcal{A}$.

It is easy to see ([5]) that in a (v, k, λ)-BIBD, there are exactly $b = \frac{\lambda v(v-1)}{k(k-1)}$ blocks, and each point appears in exactly $r = \frac{\lambda(v-1)}{k-1}$ blocks.

The following theorem shows that a $(v, k, 1)$-BIBD can be used to construct a strong $(k-1)$-NAGTA(b, v).

Theorem 3.14: ([16]) If there exists a $(v, k, 1)$-BIBD, then there exists a proper and strong k-uniform $(k-1)$-NAGTA(b, v).

The example described in [19] was constructed from a $(16, 4, 1)$-BIBD. Theorem 3.14 provides a theoretic explanation for the fact that `HardDetAlg` worked properly for up to $k - 1$ colluders.

IV. Relationships between AND-ACCs and Other Codes

In Section III, we showed an equivalence between a binary AND-ACC and a separable matrix. In fact, AND-ACCs also have close relationships with other structures related to digital fingerprinting. In this section, we discuss relationships between AND-ACCs and several other codes and hash families.

For any code $\mathcal{C} \subseteq Q^n$, we define the set of ith coordinates of $\mathcal{C}$ as

$$\mathcal{C}(i) = \{\mathbf{c}(i) \in Q \mid \mathbf{c} = (\mathbf{c}(1), \ldots, \mathbf{c}(n))^T \in \mathcal{C}\}$$

for any $1 \leq i \leq n$.

For any subset of codewords $\mathcal{C}_0 \subseteq \mathcal{C}$, we define the descendant code of $\mathcal{C}_0$ by

$$\mathsf{desc}(\mathcal{C}_0) = \{(\mathbf{x}(1), \ldots, \mathbf{x}(n))^T \in Q^n \mid \mathbf{x}(i) \in \mathcal{C}_0(i), 1 \leq i \leq n\}.$$

The set $\mathsf{desc}(\mathcal{C}_0)$ consists of the n-tuples that could be produced by a coalition holding the codewords in $\mathcal{C}_0$.

Using the notions of descendant codes and sets of ith coordinates of codes, we can define the following two different types of codes.

Definition 4.1: Suppose $\mathcal{C}$ is an (n, M, q) code and $t \geq 2$ is an integer.

(1) $\mathcal{C}$ is a $\bar{t}$-separable code, or $\bar{t}$-SC(n, M, q), if for any $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}$ such that $|\mathcal{C}_1| \leq t$, $|\mathcal{C}_2| \leq t$ and $\mathcal{C}_1 \neq \mathcal{C}_2$, we have $\mathsf{desc}(\mathcal{C}_1) \neq \mathsf{desc}(\mathcal{C}_2)$, that is, there is at least one coordinate i, $1 \leq i \leq n$, such that $(\mathsf{desc}(\mathcal{C}_1))(i) \neq (\mathsf{desc}(\mathcal{C}_2))(i)$.

(2) $\mathcal{C}$ is a t-frameproof code, or t-FPC(n, M, q), if for any $\mathcal{C}' \subseteq \mathcal{C}$ such that $|\mathcal{C}'| \leq t$, we have $\mathsf{desc}(\mathcal{C}') \bigcap \mathcal{C} = \mathcal{C}'$, that is, for any $\mathbf{c} = (\mathbf{c}(1), \ldots, \mathbf{c}(n))^T \in \mathcal{C} \setminus \mathcal{C}'$, there is at least one coordinate i, $1 \leq i \leq n$, such that $\mathbf{c}(i) \notin (\mathsf{desc}(\mathcal{C}'))(i)$.

Define $X = \{1, 2, \ldots, n\} \times Q$, and for each codeword $\mathbf{c} = (\mathbf{c}(1), \ldots, \mathbf{c}(n))^T \in \mathcal{C}$, define an n-subset of X as follows:

$$B_{\mathbf{c}} = \{(i, \mathbf{c}(i)) \mid i = 1, 2, \ldots, n\}.$$

Also define $\mathcal{B} = \{B_{\mathbf{c}} \mid \mathbf{c} \in \mathcal{C}\}$. Then we obtain the following relationship between a $\bar{t}$-separable code and a $\bar{t}$-separable matrix.

Theorem 4.2: Let $\mathcal{C}$ be an (n, M, q) code on an alphabet Q. Then $\mathcal{C}$ is a $\bar{t}$-separable code if and only if the incidence matrix of the set system $(X, \mathcal{B})$ is an n-uniform $nq \times M$ $\bar{t}$-separable matrix.

Proof: The result follows from the fact that for any subset of codewords $\mathcal{C}_0 \subseteq \mathcal{C}$ and for any index $i \in \{1, 2, \ldots, n\}$, we have $\bigcup_{\mathbf{c} \in \mathcal{C}_0} B_{\mathbf{c}} \bigcap (\{i\} \times Q) = \{i\} \times (\mathsf{desc}(\mathcal{C}_0))(i)$. ■

A similar result on the relationship between a t-frameproof code and a t-disjunct matrix was proved in [15] in terms of a cover-free family, which is equivalent to a disjunct matrix [17].

Theorem 4.3: ([15]) Let $\mathcal{C}$ be an (n, M, q) code on an alphabet Q. Then $\mathcal{C}$ is a t-frameproof code if and only if the incidence matrix of the set system $(X, \mathcal{B})$ is an n-uniform $nq \times M$ t-disjunct matrix.

Now we consider relationships between a separable code and a frameproof code. The following is a consequence of Theorem 4.3 and Lemma 3.6.

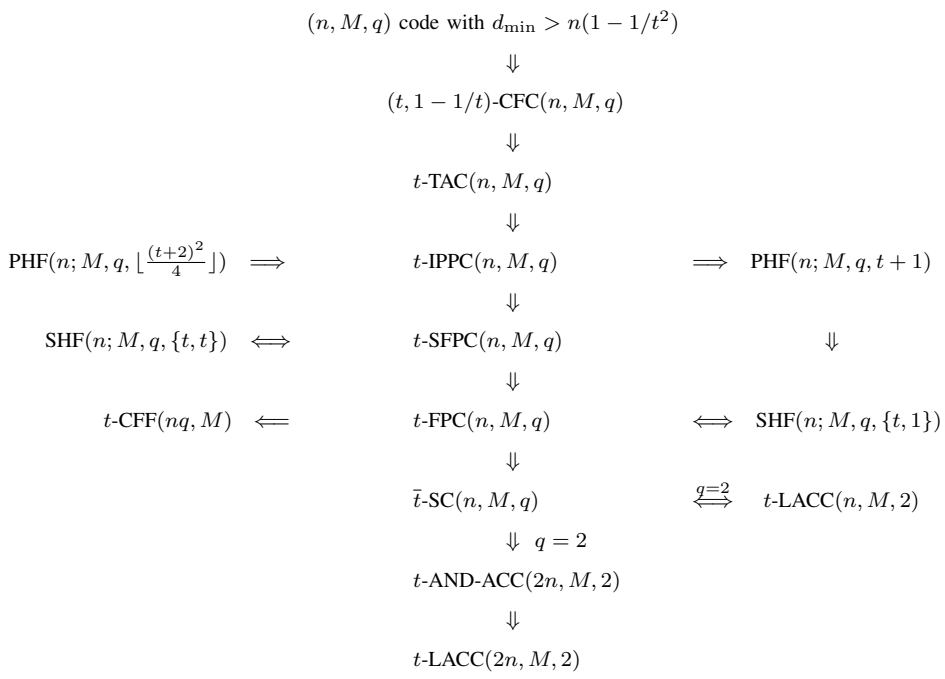
Corollary 4.4: Let $\mathcal{C}$ be an (n, M, q) code on an alphabet Q. If $\mathcal{C}$ is a t-frameproof code, then the incidence matrix of the set system $(X, \mathcal{B})$ defined in Theorem 4.2 is an n-uniform $nq \times M$ $\bar{t}$-separable matrix.

As an immediate consequence of Corollary 4.4 and Theorem 4.2, we have the following result.

Lemma 4.5: If $\mathcal{C}$ is a t-FPC(n, M, q), then $\mathcal{C}$ is also a $\bar{t}$-SC(n, M, q).

We note that the converse of Lemma 4.5 is not true. For example, consider the following $(3, 3, 2)$ code $\mathcal{C} = \{(0, 0, 1)^T,$

Figure 1: Relationships among different types of codes and hash families



Key

$d_{\min}$	minimum distance of the code
CFC	cover-free code
TAC	traceability code
IPPC	identifiable parent property code
SFPC	secure frameproof code
FPC	frameproof code
SC	separable code
AND-ACC	AND anti-collusion code
LACC	logical anti-collusion code
PHF	perfect hash family
SHF	separating hash family
CFF	cover-free family

$(1, 0, 1)^T, (1, 1, 0)^T\}$. According to the definition, $\mathsf{desc}(\{(0, 0, 1)^T\}) = \{0\} \times \{0\} \times \{1\}$, $\mathsf{desc}(\{(1, 0, 1)^T\}) = \{1\} \times \{0\} \times \{1\}$, $\mathsf{desc}(\{(1, 1, 0)^T\}) = \{1\} \times \{1\} \times \{0\}$, $\mathsf{desc}(\{(0, 0, 1)^T, (1, 0, 1)^T\}) = \{0, 1\} \times \{0\} \times \{1\}$, $\mathsf{desc}(\{(0, 0, 1)^T, (1, 1, 0)^T\}) = \{0, 1\} \times \{0, 1\} \times \{0, 1\}$, $\mathsf{desc}(\{(1, 0, 1)^T, (1, 1, 0)^T\}) = \{1\} \times \{0, 1\} \times \{0, 1\}$. These six distinct subsets are the all subsets of $\mathcal{C}$ with cardinality less than or equal to 2, so we know that $\mathcal{C}$ is a $\overline{2}$-separable code. But $(1, 0, 1)^T \in \mathsf{desc}(\{(0, 0, 1)^T, (1, 1, 0)^T\})$, which means that $\mathcal{C}$ is not a 2-frameproof code. However, by Theorem 4.2, Lemma 3.6 and Theorem 4.3, the following assertion holds.

Lemma 4.6: If $\mathcal{C}$ is a $\bar{t}$-SC(n, M, q), then $\mathcal{C}$ is also a $(t-1)$-FPC(n, M, q).

In [15], Stinson et al. summarized the relationships among different types of codes and hash families in their Figure 1. Now their Figure 1 can be extended to include the newly introduced separable code. Here we only provide the definition of a separating hash family. For those undefined terms in Figure 1, the reader is refereed to [15] for the details. Note that the term "LACC" (logical anti-collusion code) will be defined in Section V.

Definition 4.7: An $(n, m, \{w_1, w_2\})$-separating hash family is a set of functions $\mathcal{F}$ such that $|X| = n$, $|Y| = m$, $f : X \longrightarrow Y$ for each $f \in \mathcal{F}$, and for any $C_1, C_2 \subseteq X$ with $|C_1| = w_1$, $|C_2| = w_2$ and $C_1 \bigcap C_2 = \emptyset$, there exists at least one function $f \in \mathcal{F}$ such that $\{f(x) \mid x \in C_1\} \bigcap \{f(x) \mid x \in C_2\} = \emptyset$. The notation SHF$(N; n, m, \{w_1, w_2\})$ is used to denote an (n, m, w_1, w_2)-separating hash family with $|\mathcal{F}| = N$.

V. Logical Anti-Collusion Codes

In the beginning of this paper, we assumed that user U_j is assigned with a fingerprint $\mathbf{w}_j = \sum_{i=1}^{n} b_{ij} \mathbf{u}_i$, and we derived an $n \times M$ matrix $C = (c_{ij})$ with $c_{ij} = (1 + b_{ij})/2$. If we examine the matrix C in more details, we may notice that an AND-ACC can only reflect partial of the properties C holds. Apart from the bitwise AND of the codewords, we can also exploit the bitwise OR of the codewords for fingerprinting. The

corresponding threshold $\tau_{\mathbf{o}}$ for bitwise OR of the codewords can be easily taken as $\tau_{\mathbf{o}} = 1 - \tau_{\mathbf{a}}$, since the complement of OR (respectively, AND) of bits is AND (respectively, OR) of complements of those bits. In this section, we will introduce the notion of a logical anti-collusion code, provide some constructions for this type of codes, and describe a detection algorithm based on such an anti-collusion code.

Definition 5.1: Suppose that $\mathcal{C}$ is an $(n, M, 2)$-code. $\mathcal{C}$ is said to be a t-resilient logical anti-collusion code if for any $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}$ such that $|\mathcal{C}_1| \leq t$, $|\mathcal{C}_2| \leq t$ and $\mathcal{C}_1 \neq \mathcal{C}_2$, at least one of the following boolean inequalities holds:

$$\bigvee_{\mathbf{c}\in\mathcal{C}_1} \mathbf{c} \neq \bigvee_{\mathbf{c}\in\mathcal{C}_2} \mathbf{c}, \qquad \bigwedge_{\mathbf{c}\in\mathcal{C}_1} \mathbf{c} \neq \bigwedge_{\mathbf{c}\in\mathcal{C}_2} \mathbf{c}.$$

We will say that $\mathcal{C}$ is a t-LACC$(n, M, 2)$ for short.

For a given number of users, to design a good fingerprinting code to trace the potential colluders means to make the codeword length as short as possible while maintaining its capability to trace the colluders. From the definitions of a t-AND-ACC$(n, M, 2)$ and a t-LACC$(n, M, 2)$, we immediately know that a t-AND-ACC$(n, M, 2)$ is also a t-LACC$(n, M, 2)$, and consequently, a t-LACC$(n, M, 2)$ is usually better than a t-AND-ACC$(n, M, 2)$.

Example 5.2: Consider a $(3, 4, 2)$ code $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_4\}$, where $\mathbf{c}_1 = (1, 1, 0)^T$, $\mathbf{c}_2 = (1, 0, 1)^T$, $\mathbf{c}_3 = (0, 1, 1)^T$ and $\mathbf{c}_4 = (0, 0, 0)^T$. Then
$\mathbf{c}_1 \vee \mathbf{c}_2 = (1, 1, 1)^T$, $\mathbf{c}_1 \vee \mathbf{c}_3 = (1, 1, 1)^T$, $\mathbf{c}_1 \vee \mathbf{c}_4 = (1, 1, 0)^T$,
$\mathbf{c}_2 \vee \mathbf{c}_3 = (1, 1, 1)^T$, $\mathbf{c}_2 \vee \mathbf{c}_4 = (1, 0, 1)^T$, $\mathbf{c}_3 \vee \mathbf{c}_4 = (0, 1, 1)^T$.
However,
$\mathbf{c}_1 \wedge \mathbf{c}_2 = (1, 0, 0)^T$, $\mathbf{c}_1 \wedge \mathbf{c}_3 = (0, 1, 0)^T$, $\mathbf{c}_2 \wedge \mathbf{c}_3 = (0, 0, 1)^T$; and
$\mathbf{c}_1 \wedge \mathbf{c}_4 = (0, 0, 0)^T$; $\mathbf{c}_2 \wedge \mathbf{c}_4 = (0, 0, 0)^T$; $\mathbf{c}_3 \wedge \mathbf{c}_4 = (0, 0, 0)^T$.
Therefore, by performing these twelve logical operations, we can know that $\mathcal{C}$ is a 2-LACC$(3, 4, 2)$, although $\mathcal{C}$ is not a 2-AND-ACC$(3, 4, 2)$.

We can also show this fact by first checking $\bigwedge$ inequalities and then $\bigvee$ inequalities.
$\mathbf{c}_1 \wedge \mathbf{c}_2 = (1, 0, 0)^T$, $\mathbf{c}_1 \wedge \mathbf{c}_3 = (0, 1, 0)^T$, $\mathbf{c}_1 \wedge \mathbf{c}_4 = (0, 0, 0)^T$,
$\mathbf{c}_2 \wedge \mathbf{c}_3 = (0, 0, 1)^T$, $\mathbf{c}_2 \wedge \mathbf{c}_4 = (0, 0, 0)^T$, $\mathbf{c}_3 \wedge \mathbf{c}_4 = (0, 0, 0)^T$.
However,
$\mathbf{c}_1 \vee \mathbf{c}_4 = (1, 1, 0)^T$, $\mathbf{c}_2 \vee \mathbf{c}_4 = (1, 0, 1)^T$, $\mathbf{c}_3 \vee \mathbf{c}_4 = (0, 1, 1)^T$.
These nine logical operations are also sufficient to show that $\mathcal{C}$ is a 2-LACC$(3, 4, 2)$.

As showed in Example 5.2, although conceptually there is no difference between the boolean inequality checks beginning with $\bigvee$ and those begining with $\bigwedge$, computationally, the two check procedures may have different computational complexities. It is an interesting problem to find an optimal check procedure so that the number of boolean inequalities to be checked is the smallest. We conjecture that the computational complexity of the optimal one is impossible to be $O(nM)$, where M is the number of codewords and n is the length of the code.

It is interesting that LACCs are closely related with separable codes.

Theorem 5.3: Let $\mathcal{C}$ be an $(n, M, 2)$ code. Then $\mathcal{C}$ is a t-LACC$(n, M, 2)$ if and only if it is a $\bar{t}$-SC$(n, M, 2)$.

Proof: For any subset of codewords $\mathcal{C}_0 \subseteq \mathcal{C}$ and any index i, $1 \leq i \leq n$, the pair of AND and OR, respectively, of ith coordinates of codewords in $\mathcal{C}_0$ possesses the whole information on $(\mathsf{desc}(\mathcal{C}_0))(i)$; namely, we have (AND, OR)$= (1, 1)$ (respectively $(0, 0)$, or $(0, 1)$) if and only if $(\mathsf{desc}(\mathcal{C}_0))(i) = \{1\}$ (respectively $\{0\}$, or $\{0, 1\}$).

So, for any two subsets of codewords $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}$ with $|\mathcal{C}_1| \leq t$, $|\mathcal{C}_2| \leq t$ and $\mathcal{C}_1 \neq \mathcal{C}_2$, there exists a coordinate i, $1 \leq i \leq n$, such that the inequality $(\mathsf{desc}(\mathcal{C}_1))(i) \neq (\mathsf{desc}(\mathcal{C}_2))(i)$ holds if and only if the pairs of (AND, OR) of $\mathcal{C}_1$ and $\mathcal{C}_2$ are distinct. ■

In order to construct a t-LACC$(n, M, 2)$, according to Theorem 5.3, we only need to construct its corresponding $\bar{t}$-SC$(n, M, 2)$. The following is a recursive construction for $\bar{t}$-separable codes.

Lemma 5.4: If there exist a $\bar{t}$-SC(n_1, M, q) and a $\bar{t}$-SC(n_2, q, q'), then there exists a $\bar{t}$-SC$(n_1 n_2, M, q')$.

Proof: Let $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_M\}$ be a $\bar{t}$-SC(n_1, M, q) and $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_q\}$ be a $\bar{t}$-SC(n_2, q, q'), respectively. Let $f : \{0, 1, \ldots, q-1\} \longrightarrow \mathcal{C}$ be a bijective mapping such that $f(i) = \mathbf{c}_{i+1}$. For any $\mathbf{b} = (\mathbf{b}(1), \ldots, \mathbf{b}(n_1))^T \in \mathcal{B}$, we define $f(\mathbf{b}) = (f(\mathbf{b}(1)), \ldots, f(\mathbf{b}(n_1)))^T$. Obviously, $f(\mathbf{b})$ is a q'-ary vector of length $n_1 n_2$. We define a new $(n_1 n_2, M, q')$ code $\mathcal{F} = \{f(\mathbf{b}_1), \ldots, f(\mathbf{b}_M)\}$. We are going to show that $\mathcal{F}$ is in fact a $\bar{t}$-separable code.

Consider any two codeword sets $\mathcal{F}_1, \mathcal{F}_2 \subseteq \mathcal{F}$ with $|\mathcal{F}_1| \leq t$, $|\mathcal{F}_2| \leq t$ and $\mathcal{F}_1 \neq \mathcal{F}_2$. They correspond to two codeword sets $\mathcal{B}_1, \mathcal{B}_2 \subseteq \mathcal{B}$, respectively, such that $|\mathcal{B}_1| \leq t$, $|\mathcal{B}_2| \leq t$ and $\mathcal{B}_1 \neq \mathcal{B}_2$, where $\mathcal{F}_1 = \{f(\mathbf{b}) \mid \mathbf{b} \in \mathcal{B}_1\}$ and $\mathcal{F}_2 = \{f(\mathbf{b}) \mid \mathbf{b} \in \mathcal{B}_2\}$. Then there must be a coordinate i_0, $1 \leq i_0 \leq n_1$, such that $\mathcal{B}_1(i_0) \neq \mathcal{B}_2(i_0)$, because $\mathcal{B}$ is a $\bar{t}$-separable code. This implies that $\{f(\mathbf{b}(i_0)) \mid \mathbf{b} \in \mathcal{B}_1\} \neq \{f(\mathbf{b}(i_0)) \mid \mathbf{b} \in \mathcal{B}_2\}$. Clearly, $\{f(\mathbf{b}(i_0)) \mid \mathbf{b} \in \mathcal{B}_1\} \subseteq \mathcal{C}$, $\{f(\mathbf{b}(i_0)) \mid \mathbf{b} \in \mathcal{B}_2\} \subseteq \mathcal{C}$, and $|\{f(\mathbf{b}(i_0)) \mid \mathbf{b} \in \mathcal{B}_1\}| \leq t$, $|\{f(\mathbf{b}(i_0)) \mid \mathbf{b} \in \mathcal{B}_2\}| \leq t$. But $\mathcal{C}$ is also a $\bar{t}$-separable code, so we have $\mathsf{desc}(\{f(\mathbf{b}(i_0)) \mid \mathbf{b} \in \mathcal{B}_1\}) \neq \mathsf{desc}(\{f(\mathbf{b}(i_0)) \mid \mathbf{b} \in \mathcal{B}_2\})$. Then $\mathsf{desc}(\{(f(\mathbf{b}(1)), \ldots, f(\mathbf{b}(n_1)))^T \mid \mathbf{b} \in \mathcal{B}_1\}) \neq \mathsf{desc}(\{(f(\mathbf{b}(1)), \ldots, f(\mathbf{b}(n_1)))^T \mid \mathbf{b} \in \mathcal{B}_2\})$, that is, $\mathsf{desc}(\mathcal{F}_1) \neq \mathsf{desc}(\mathcal{F}_2)$, which means that $\mathcal{F}$ is a $\bar{t}$-separable code. ■

As was shown in Theorem 5.3, we can use any binary $\bar{t}$-separable code as a t-LACC to resist averaging attack and trace up to t colluders. In this case, we should investigate all subsets of users with cardinality up to t, and thus the computational complexity to trace colluders is $O(nM^t)$, where M is the total number of users to whom multimedia content is distributed and n is the length of the code. Clearly, it is desirable that we find a structure which could be used as a t-LACC in an algorithm with a lower computational complexity to trace colluders.

Theorem 5.5: Any t-FPC$(n, M, 2)$ is a t-LACC$(n, M, 2)$, and can be used to identify the colluders with computational complexity $O(nM)$ by using the algorithm `LACCIdenAlg` (Algorithm 3 below).

Proof: Suppose that $\mathcal{C}$ is a t-FPC$(n, M, 2)$. By Lemma 4.5 and Theorem 5.3, $\mathcal{C}$ is also a t-LACC$(n, M, 2)$.

Let $\mathbf{a} = (\mathbf{a}(1), \ldots, \mathbf{a}(n))^T$ and $\mathbf{o} = (\mathbf{o}(1), \ldots, \mathbf{o}(n))^T$ be the outcome binary AND and OR vectors, respectively, which are obtained by applying hard thresholding twice to the

detection statistics $\mathbf{T}(i)$ such that $\mathbf{a}(i) = 1$ if $\mathbf{T}(i) > \tau_a$ and $\mathbf{a}(i) = 0$ otherwise, and $\mathbf{o}(i) = 0$ if $\mathbf{T}(i) < \tau_o$ and $\mathbf{o}(i) = 1$ otherwise, where τ_a and τ_o are the two thresholds satisfying $\tau_o + \tau_a = 1$, $0 \leq \tau_o < \tau_a \leq 1$ appropriately determined by the detector. It is shown [19], [21] that an appropriate threshold τ_a would make $\mathbf{a}$ to be the logical AND of column vectors in C corresponding to the contributing fingerprints involved in the collusion attack, where C is the incidence matrix $M(\mathcal{C})$ of the code $\mathcal{C}$. Similarly, since the complement of OR (respectively AND) of bits is AND (respectively, OR) of complements of those bits, the appropriate threshold τ_o would also make $\mathbf{o}$ to be the logical OR of column vectors in C corresponding to the contributing fingerprints involved in the collusion attack. By deleting all columns $\{\mathbf{c} \mid \mathbf{c} \in \mathcal{C}$ such that $\exists\ 1 \leq i \leq n,\ \mathbf{a}(i) = 1, \mathbf{c}(i) = 0,$ or $\mathbf{o}(i) = 0, \mathbf{c}(i) = 1\}$, we obtain a sub-matrix C' of C with at most t columns. The following detection algorithm shows the procedure described above.

Algorithm 3: `LACCIdenAlg`

Define $J_{\mathbf{a}}$, $J_{\mathbf{o}}$ to be the sets of indices where $\mathbf{a}(j) = 1$, $\mathbf{o}(j) = 0$ respectively, and $\mathbf{J_a} = (\mathbf{J_a}(1), \ldots, \mathbf{J_a}(|J_{\mathbf{a}}|))$, $\mathbf{J_o} = (\mathbf{J_o}(1), \ldots, \mathbf{J_o}(|J_{\mathbf{o}}|))$ to be the vector representing $\mathbf{a}$'s non-zero and $\mathbf{o}$'s zero coordinates.
$\Phi = \mathbf{1}^T$;
$U_1 = \emptyset$;
for $t = 1$ **to** $|J_{\mathbf{a}}|$ **do**
  $j = \mathbf{J_a}(t)$;
  Define $\mathbf{e}_j$ to be the jth row of C;
  $\Phi = \Phi \cdot \mathbf{e}_j^T$;
end
for $i = 1$ **to** M **do**
  if $\Phi(i) = 1$ then
    $U_1 = \{i\} \bigcup U_1$;
  end
end
$\Phi = \mathbf{1}^T$;
$U_2 = \emptyset$;
for $t = 1$ **to** $|J_{\mathbf{o}}|$ **do**
  $j = \mathbf{J_o}(t)$;
  $\Phi = \Phi \cdot \overline{\mathbf{e}_j}^T$;
end
for $i = 1$ **to** M **do**
  if $\Phi(i) = 1$ then
    $U_2 = \{i\} \bigcup U_2$;
  end
end
$U = U_1 \bigcap U_2$;
if $|U| \leq t$, then
  output U;
else
  output "the set of colluders has size at least $t + 1$".
end

We claim that the set of columns in C' corresponds exactly to the set of colluders. Suppose that $\mathcal{C}_0 = \{\mathbf{c}'_1, \ldots, \mathbf{c}'_r\}$, $1 \leq r \leq t$, is the set of colluders. It is clear that any column corresponding to a colluder cannot be deleted by this algorithm. We then consider a column $\mathbf{c}$ in C, and suppose that $\mathbf{c}$ is not a codeword assigned to any colluder. Since $\mathcal{C}$ is a t-FPC$(n, M, 2)$, and $|\mathcal{C}_0| = r \leq t$, $\mathbf{c} \in \mathcal{C} \setminus \mathcal{C}_0$, we know that there must be a coordinate i, $1 \leq i \leq n$, such that $\mathbf{c}(i) \notin \mathcal{C}_0(i)$. If $\mathbf{c}(i) = 1$, then $\mathbf{c}'_1(i) = \mathbf{c}'_2(i) = \ldots = \mathbf{c}'_r(i) = 0$, and the column $\mathbf{c}$ should be deleted since $\mathbf{o}(i) = 0$. If $\mathbf{c}(i) = 0$, then $\mathbf{c}'_1(i) = \mathbf{c}'_2(i) = \ldots = \mathbf{c}'_r(i) = 1$, and the column $\mathbf{c}$ should be deleted since $\mathbf{a}(i) = 1$. This means that every column in C' is corresponding to a colluder. Therefore we can identify all the colluders exactly by this algorithm and its computational complexity is $O(nM)$. ∎

We would like to make an important remark here. Frameproof codes were widely considered as having no traceability for generic digital contents (see for example [17]). Surprisingly enough, Theorem 5.5 shows that frameproof codes have traceability for multimedia contents. This phenomenon is in fact due to the special embedding method for fingerprinting multimedia contents.

We would also like to point out the following fact. By definitions, an $n \times M$ t-disjunct matrix implies a t-FPC$(n, M, 2)$; however, the converse is not necessarily true. It can be easily checked that the 2-LACC$(3, 4, 2)$ in Example 5.2

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

is a 2-FPC$(3, 4, 2)$, but $(0, 1, 1)^T$ is covered by the bitwise OR of $(1, 1, 0)^T$ and $(1, 0, 1)^T$, which means that the above matrix is not a 2-disjunct matrix.

In a similar fashion to the proof of Lemma 5.4, we can prove the following result.

Lemma 5.6: If there exist a t-FPC(n_1, M, q) and a t-FPC(n_2, q, q'), then there exists a t-FPC$(n_1 n_2, M, q')$.

As was shown in Figure 1, a t-FPC(n, M, q) is equivalent to an SHF$(n; M, q, \{t, 1\})$. Separating hash families have been extensively investigated by numerous researchers, and many interesting results on separating hash families have been obtained. We can apply Lemma 5.6 with known separating hash families to produce infinite series of frameproof codes. For example, the following result [2] can be used to produce an infinite series of frameproof codes.

Theorem 5.7: ([2]) If q is a prime power, then there exists an SHF$(2d + 1; q^{d+1}, q, \{2, 1\})$ with $2d \leq q$.

Corollary 5.8: For any non-negative integer s, there exists an SHF$(\prod_{0 \leq i \leq s}(2^{i+1} + 1); 2^{\prod_{0 \leq i \leq s}(2^i+1)}, 2, \{2, 1\})$, or equivalently, a 2-FPC$(\prod_{0 \leq i \leq s}(2^{i+1} + 1), 2^{\prod_{0 \leq i \leq s}(2^i+1)}, 2)$.

Proof: We already knew that the 2-LACC$(3, 4, 2)$ in Example 5.2 is a 2-FPC$(3, 4, 2)$, that is, an SHF$(2^1+1; 2^{2^0+1}, 2^1, \{2, 1\})$. Applying Theorem 5.7 with $d = 2^1$, we obtain an SHF$(2^2 + 1; 2^{(2^0+1)(2^1+1)}, 2^{2^0+1}, \{2, 1\})$. By Lemma 5.6, we obtain an SHF$((2^1 + 1)(2^2 + 1); 2^{(2^0+1)(2^1+1)}, 2, \{2, 1\})$. Iterating this procedure with $d = 2^i$ in step i until $i = s$, we obtain the desired result. ∎

Similar to Corollary 5.8, by applying Lemma 5.4 with Theorem 5.7, we can get the following $\overline{2}$-separable codes.

Corollary 5.9: For any positive integer s, there exists an $\overline{2}$-SC$(\prod_{1 \leq i \leq s}(2^i + 1), 5^{(2^s+1)^{-1}\prod_{1 \leq i \leq s}(2^i+1)}, 2)$.

Proof: It is clear that an SHF$(n; M, 2, \{2, 1\})$ is also a $\overline{2}$-SC$(n, M, 2)$. The starting $\overline{2}$-SC$(2^1+1, 5, 2)$ is listed below: $\mathcal{C}_1 = \{(1,1,0)^T, (1,0,1)^T, (0,1,1)^T, (0,0,0)^T, (1,1,1)^T\}$. ■

We also know the following result.

Theorem 5.10: ([2]) For any prime power q and any integer t with $t \leq q-1$, there exists an SHF$(t+1; q^2, q, \{t, 1\})$.

In a similar fashion, we have the following consequence.

Theorem 5.11: For any prime power q, any positive integer $t \leq q-1$, and any non-negative integer s, there exists a t-FPC$(q(t+1)^s, q^{2^s}, 2)$.

Proof: For any prime power q and any positive integer $t \leq q-1$, the $(q, q, 2)$ code

$$\mathcal{C} = \{(1,0,\ldots,0)^T, (0,1,\ldots,0)^T, \ldots, (0,0,\ldots,1)^T\}$$

is clearly a t-FPC$(q, q, 2)$. We also have a t-FPC$(t+1, q^2, q)$ by Theorem 5.10. Applying Lemma 5.6, we obtain a t-FPC$(q(t+1), q^2, 2)$. There is a t-FPC$(t+1, q^{2^2}, q^2)$ by Theorem 5.10. Applying Lemma 5.6, we obtain a t-FPC$(q(t+1)^2, q^{2^2}, 2)$. Iterating this procedure, we obtain the desired result. ■

Other constructions for frameproof codes can be found in, for example, [3].

VI. Conclusion

In this paper, we investigated anti-collusion codes for multimedia fingerprinting. We showed an equivalence between a t-AND-ACC and a $\overline{t}$-separable matrix. We pointed out some flaws of algorithm `HardDetAlg`, improved it, and furthermore, described an efficient detection algorithm `IdenAlg` based on t-AND-ACCs constructed from t-disjunct matrices. We investigated relationships between AND-ACCs and other structures related to fingerprinting. We also introduced LACCs for multimedia fingerprinting, which can be used to identify more colluders than AND-ACCs. In fact, we proposed an efficient identification algorithm based on LACCs constructed from frameproof codes. This showed an important fact that frameproof codes have traceability for multimedia contents. We also provided a few constructions for separable codes and frameproof codes.

It would be of interest if we could find more properties and constructions of separable codes and frameproof codes. It would be also interesting if we could find an optimal check procedure for LACC so that the number of boolean inequalities to be checked is the smallest.

Acknowledgment

The authors express their sincere thanks to the two anonymous reviewers for their valuable comments and suggestions which greatly improved this paper, and to Professor Rei Safavi-Naini, the Associate Editor for Complexity and Cryptography, for her excellent editorial job.

References

[1] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inform. Theory*, vol. 44, no. 5, pp. 1897–1905, 1998.

[2] M. Bazrafshan and Tran van Trung, "On optimal bounds for separating hash families," abstracted in *Proc. Germany Africa Workshop on Inform. Commun. Tech.*, Essen, Germany, 2008.

[3] S. R. Blackburn, "Frameproof codes", *SIAM J. Discrete Math.*, vol. 16, no. 3, pp. 499–510, 2003.

[4] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provable good methods for digital watermarking and information embedding," *IEEE Trans. Inform. Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.

[5] C. J. Colbourn and J. H. Dinitz (eds.), *The CRC Handbook of Combinatorial Designs*, Second Edition, Boca Raton, FL: Chapman & Hall/CRC, 2007.

[6] I. J. Cox, J. Kilian, F. T. Leighton, and T. G. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, 1997.

[7] D. Du and F. K. Hwang, *Combinatorial Group Testing and Its Applications*, Second Edition, Singapore: World Scientific, 2000.

[8] J. Dittmann, P. Schmitt, E. Saar, J. Schwenk, and J. Ueberberg, "Combining digital watermarks and collusion secure fingerprints for digital images," *SPIE J. Electron. Imag.*, vol. 9, no. 4, pp. 456–467, 2000.

[9] W. H. Kautz and R. R. Singleton, "Nonrandom binary superimposed codes," *IEEE Trans. Inform. Theory,* vol. 10, pp. 363–377, 1964.

[10] Q. Li, X. Wang, Y. Li, Y. Pan, and P. Fan, "Construction of anti-collusion codes based on cover-free families," in *2009 Sixth Int. Conf. Inform. Tech.: New Generations*, pp. 362–365, Las Vegas, NV, 2009.

[11] Z. Li and W. Trappe, "Collusion-resistant fingerprints from WBE sequence sets," In *Proc. IEEE Int. Conf. Commun.*, vol. 2, pp. 1336–1340, Seoul, Korea, 2005.

[12] K. J. R. Liu, W. Trappe, Z. J. Wang, M. Wu, and H. Zhao, *Multimedia Fingerprinting Forensics for Traitor Tracing*, NY: Hindawi, 2005.

[13] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Trans. Inform. Theory*, vol. 49, no. 3, pp. 563–593, 2003.

[14] C. I. Podilchuk and W. Zeng, "Image-adaptive watermarking using visual models," *IEEE J. Select. Areas Commun.*, vol. 16, no. 4, pp. 525–539, 1998.

[15] J. N. Staddon, D. R. Stinson, and R. Wei, "Combinatorial properties of frameproof and traceability codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 3, pp. 1042–1049, 2001.

[16] D. R. Stinson, *Combinatorial Designs: Constructions and Analysis*, NY: Springer, 2004.

[17] D. R. Stinson, Tran van Trung, and R. Wei, "Secure frameproof codes, key distribution patterns, group testing algorithms and related structures," *J. Statist. Plann. Inference,* vol. 86, no. 2, pp. 595–617, 2000.

[18] H. S. Stone, "Analysis of attacks on image watermarks with randomized coefficients," *Tech. Rep.* 96–045, NEC Research Institute, Princeton, NJ, 1996.

[19] W. Trappe, M. Wu, and K. J. R. Liu, "Anti-collusion codes: multi-user and multimedia perspectives," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, pp. 981–984, Rochester, NY, 2002.

[20] W. Trappe, M. Wu, and K. J. R. Liu, "Collusion-resistant fingerprinting for multimedia," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, pp. 3309–3321, Orlando, FL, 2002.

[21] W. Trappe, M. Wu, Z. J. Wang, and K. J. R. Liu, "Anti-collusion fingerprinting for multimedia," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 1069–1087, 2003.

[22] M. Wu and B. Liu, "Modulation and multiplexing techniques for multimedia data hiding," in *Proc. SPIE, Multimedia Systems and Applications IV*, vol. 4518, pp. 228–238, Denver, CO, 2001.

[23] M. Wu, W. Trappe, Z. J. Wang, and K. J. R. Liu, "Collusion resistant fingerprinting for multimedia," *IEEE Signal Process. Magazine, Special Issue on Digital Rights Management*, pp. 15–27, 2004.

Minquan Cheng received the M.S. degree in applied mathematics from Guangxi Normal University, Guilin, Guangxi, P. R. China, in 2008. He is now a Ph.D. student at Department of Social Systems and Engineering, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki, Japan. His research interests include digital right management, combinatorial design theory, and coding theory.

Ying Miao received the D.Sci. degree in mathematics from Hiroshima University, Hiroshima, Japan, in 1997.

From 1989 to 1993, he worked for Suzhou Institute of Silk Textile Technology, Suzhou, Jiangsu, P. R. China. From 1995 to 1997, he was a Research Fellow of the Japan Society for the Promotion of Science. During 1997–1998, he was a Postdoctoral Fellow at the Department of Computer Science, Concordia University, Montreal, QC, Canada. In 1998, he joined the University of Tsukuba, Tsukuba, Ibaraki, Japan, where he is currently an Associate Professor at Department of Social Systems and Management, Graduate School of Systems and Information Engineering. His research interests include combinatorics, coding theory, cryptography, bioinformatics, and their interactions.

Dr. Miao is on the Editorial Boards of both *Graphs and Combinatorics* and *Journal of Combinatorial Designs*. He received the 2001 Kirkman Medal from the Institute of Combinatorics and its Applications.