IEICE TRANS. INF. & SYST., VOL.E92–D, NO.5 MAY 2009

PAPER Special Section on Information and Communication System Security

Fingerprinting Codes for Internet-Based Live Pay-TV System Using Balanced Incomplete Block Designs

Shuhui HOU[†], ***Nonmember***, **Tetsutaro UEHARA**[††], **Takashi SATOH**[†††], **Yoshitaka MORIMURA**[†], ***and*** **Michihiko MINOH**[††], ***Members***

SUMMARY In recent years, with the rapid growth of the Internet as well as the increasing demand for broadband services, live pay-television broadcasting via the Internet has become a promising business. To get this implemented, it is necessary to protect distributed contents from illegal copying and redistributing after they are accessed. Fingerprinting system is a useful tool for it. This paper shows that the anti-collusion code has advantages over other existing fingerprinting codes in terms of efficiency and effectivity for live pay-television broadcasting. Next, this paper presents how to achieve efficient and effective anti-collusion codes based on unital and affine plane, which are two known examples of balanced incomplete block design (BIBD). Meanwhile, performance evaluations of anti-collusion codes generated from unital and affine plane are conducted. Their practical explicit constructions are given last.

key words: *fingerprinting code, anti-collusion code, BIBD, unital, affine plane*

1. Introduction

In the last few years, businesses offering digital contents (such as music and video) via the Internet have come to be established with the development of broadband networks. But it is not realized to serve live pay-television broadcasts via the Internet to a number of people simultaneously at the moment. The authors are developing a system to realize such internet-based pay-TV system under the assumption that the same datagram could be transmitted to a number of receivers simultaneously via IP-multicasting.

Broadcast encryption schemes are techniques that allow a center to broadcast encrypted contents to an arbitrary subset of privileged receivers out of a large set of receivers so that collusions of receivers not in the privileged set cannot decrypt the encrypted contents. Such schemes are useful to limit the access and decryption of encrypted contents but can not limit the illegal copying and redistributing of decrypted contents. Without quality distortion, digital contents are easy to be manipulated by copying and redistributing. To protect digital contents from illegal copying and redistributing after they are decrypted, fingerprinting schemes play an important role.

Fingerprinting schemes embed unique user information (e.g., ID or fingerprints) into each user's copy as a watermark and if an illegal copy appears, user information can be extracted to help trace or identify illegal users. The goal of digital fingerprinting is to deter or discourage people from illegally redistributing the digital data that they have legally purchased. The major challenge to fingerprinting is collusion attacks from illegal users. A collusion attack is a cost-effective attack, where colluders (illegal users) combine several copies with the same content but different fingerprints to try to remove the original fingerprints or frame innocent users.

The research on digital fingerprinting can be broadly divided into two main directions: designing fingerprinting codes and jointly considering the fingerprint coding and embedding. The former direction mainly focuses on the coding theories, in which different emphasis is placed on criteria such as attack assumptions, collusion size, code size, code length, ability to trace one or all pirates, etc [1]–[10]. In contrast, the latter direction mainly focuses on fingerprint embedding, as well as advanced detection [12]–[18]. In addition, there also exists non-coded fingerprinting [11], where mutually orthogonal spreading sequences are assigned to users as fingerprints. While the non-coded fingerprinting is easy to implement, the required number of spreading sequences and the computational complexity of detection would increase linearly with the number of users due to no step of coding. Hence code designing is a crucial step to a successful fingerprinting scheme. This paper mainly focuses on designing desirable fingerprinting codes for the internet-based live pay-TV system rather than how to embed fingerprinting codes.

The authors are aiming to develop live pay-TV system based on the Internet infrastructure, in which the same datagram could be transmitted to $10^4 \sim 10^5$ receivers simultaneously via IP-multicast. The major reason for choosing the range $10^4 \sim 10^5$ consists in: 10^5 or less receivers are so few that general broadcasting infrastructure (e.g., television broadcasting) cannot profit from its broadcasting, while the receivers exceeding 10^5 are so many that it is unworthy to transmit datagram via IP-multicasting with respect to the Internet traffic. Accordingly, fingerprinting codes for internet-based pay-TV system need to manage $10^4 \sim 10^5$ users from a business and economic point of view. To discourage users from illegal copying, high resistance against collusion at-

Manuscript received August 4, 2008.
Manuscript revised December 24, 2008.

[†]The authors are with the Dept. of Intelligence Science and Technology, Graduate School of Informatices, Kyoto University, Kyoto-shi, 606–8501 Japan.
[††]The authors are with the Academic Center for Computing and Media Studies, Kyoto University, Kyoto-shi, 606–8501 Japan.
[†††]The author is with the Dept. of Information and Media Engineering, Faculty of Environmental Engineering, The University of Kitakyushu, Kitakyushu-shi, 808–0135 Japan.

DOI: 10.1587/transinf.E92.D.876

tacks is required. The collusion resistance is generally measured by collusion size (strictly, the maximum tolerated collusion size) for a given number of users. It is obvious that the larger the collusion size is, the securer the pay-TV system is. Notice that the Internet traffic would extremely increase with the collusion size increasing. It is desirable to shorten the code length (to squeeze more users into fewer bits) for easily implementing code. For better scalability, the code whose length increases linearly with the number of users is not suitable for our system. As three important factors of a fingerprinting code, the number of users (equal to code size), collusion resistance and code length should be considered synthetically. So far, there have been almost no evaluations for what fingerprinting codes adapt to internet-based pay-TV broadcasting.

In this paper, efficiency and effectivity are introduced for evaluating fingerprinting codes. In terms of efficiency and effectivity, anti-collusion code (ACC, [13]) is shown to be superior to other existing fingerprinting codes. By making a survey on fingerprinting codes and an analysis from their underlying restrictions and ability to trace or identify colluders, the authors consider ACC is applicable for live pay-television broadcasting.

This paper also presents how to achieve efficient and effective ACC by using unital and affine plane, which are two known examples of balanced incomplete block design (BIBD). Meanwhile, performances of the ACCs derived from unital and affine plane are evaluated from the above-mentioned factors. Last, their practical explicit constructions are described.

The rest of this paper is organized as follows. In Sect. 2, two useful metrics for fingerprinting codes are introduced. Section 3 investigates major existing fingerprinting codes and shows ACC is more efficient and more effective than other fingerprinting codes for live pay-television broadcasting. In Sect. 4, after investigating balanced incomplete block design (BIBD) which is one method to generate ACC, the authors present how to generate efficient and effective ACC from unital and affine plane. Meantime, their performances are evaluated through concrete examples. Section 5 addresses practical constructions of ACC derived from unital and affine plane. Section 6 compares our proposed ACC with recent related work and the conclusions are drawn in Sect. 7.

2. Efficiency and Effectivity of Fingerprinting Codes

Fingerprinting codes generated with different methods are different at the structure and the principle how to trace colluders. In order to compare the property of fingerprinting codes and evaluate what fingerprinting codes are appropriate for pay-TV broadcasting, we need some evaluation metrics which can provide a simple way for comparing the different types of fingerprinting codes.

For a given fingerprinting code, let its length be l, its maximum tolerated collusion size (to put it shortly, collusion size) be c and its size (i.e., the number of users that it can support) be n. In [13] and [15], code efficiency, as a metric, is defined as follows.

Definition 1: The *efficiency* on a fingerprinting code for a given collusion size is referred to as $\beta = n/l$, which describes the number of users that can be supported by the code length.

Within a given collusion size, the higher β indicates the fingerprinting code with fewer bits that support more users. In addition, we need another metric to compare the collusion resistance of fingerprinting codes, so we give the following definition.

Definition 2: The *effectivity* on a fingerprinting code is defined by $\gamma = c/n$, which describes the resistance against collusion attacks.

Among the codes with the same code length, a fingerprinting code with higher γ is resistant to a larger size collusion. That is, a fingerprinting code with higher γ is secure against collusion attack than one with lower γ under the same number of users.

Regarding the requirements from the internet-based pay-TV system (large number of users ($10^4 \sim 10^5$), high collusion resistance and easily implementing), a kind of high efficient and high effective fingerprinting code is preferable for such a system. Until now, lots of fingerprinting codes are presented and studied, but not all of them are proposed with tracing capability and collusion resistance. So we examine them in terms of *efficiency* and *effectivity* in next section.

3. ACC Superior to Other Fingerprinting Codes for Pay-TV System

3.1 Attacking Assumptions: Marking Assuption and *Envelope*

Under an unrestricted, arbitrary collusion attack, it is difficult to analyze the code and the tracing capability, so restrictive assumptions need to be made concerning what collusion can be allowed and what collusion cannot be allowed, given a set of codewords. The most widely used assumption is called marking assumption [2].

1. Marking Assumption
   A fingerprint, as a codeword, is regarded as a sequence of marks. q marks such as $\{0, 1, \cdots, q-1\}$ are used to indicate q different states of a position in an object. Users do not know the positions of marks in the object and also do not know which mark represents which state. Marking assumption states that:
   - In a collusion attack, the positions of the marks in an object can be either undetectable or detectable. An undetectable position is a position where the mark is the same for all the colluders so the position of the mark is unknown. The detectable position is known to colluders since their marks differ in this position.
   - Colluding users cannot change the marks in the

undetectable positions without rendering the fingerprinted object useless, but it is considered possible for colluding users to change the marks in the detectable positions into any state.

A mark without being included in $\{0, 1, \cdots, q-1\}$ is referred to as an unreadable mark. According to whether a mark in the detectable position can be allowed to change into an unreadable state or not, and the range within which pirated codes can fall, there are four cases described by *envelope* [5].

2. *Envelope* under Marking Assumption
"Under Marking Assumption" means that colluders can only alter those marks in the detectable positions.
Let an alphabet $Q = \{0, 1, \cdots, q-1\}$, then $|Q| = q$ where q is the number of elements. Let Q^l be the set of all l-tuples of elements of Q. Considering a code C over Q, $C \subseteq Q^l$ and is called an (l, n, q)-code when $|C| = n$. l is called the length and n is called the size of q-ary code C, respectively. The elements of C are called codewords and each codeword x^j has the form $(x_1^j, x_2^j, \cdots, x_l^j)$, where $x_i^j \in Q, 1 \le i \le l, j = 1, \cdots, n$. Codeword $x^j (j = 1, \cdots, n)$ is assigned uniquely to j-th user, so code size n is the maximum number of users that can be supported by this code. Shortly speaking, code size n is also the number of users.
Let $U = \{u_1, u_2, \cdots, u_t\}$ be a collusion of t users that collude to generate a pirated code $y = (y_1, y_2, \cdots, y_l)$. Without loss of generality, assume that $X = \{x^1, x^2, \cdots, x^t\}$ are the fingerprints (or codewords) assigned to the members of U. Let $Z(X)$ denote the set of undetectable positions for X and symbol $*$ denotes the unreadable state. The range within which the pirated code out of X can fall is defined as *envelope* as follows:

- Narrow-sense envelope $e(X)$ is the set

$$e(X) = \{y \in Q^l | \\ y_i \in \{x_i^1, \cdots, x_i^t\}, 1 \le i \le l\}$$

  Under the narrow-sense case, the generation of a pirated code is restricted to allowing, in any given (detectable) position i, only the marks appearing in position i of any codeword in X to be chosen.
- Expanded narrow-sense envelope $e^*(X)$ is the set

$$e^*(X) = \{y \in \{Q \cup \{*\}\}^l | \\ y_i \in \{x_i^1, \cdots, x_i^t, *\}, 1 \le i \le l\}$$

  Under the expanded narrow-sense case, the restriction is relaxed. The marks in the detectable positions are allowed to change into an unreadable state.
- Wide-sense envelope $E(X)$ is the set

$$E(X) = \{y \in Q^l | \\ y|_{Z(X)} = x^1|_{Z(X)} = \cdots = x^t|_{Z(X)}\}$$

  We refer to $y|_{Z(X)}$ as the restriction of y to the positions in $Z(X)$, where $x^1|_{Z(X)}, x^2|_{Z(X)}, \cdots, x^t|_{Z(X)}$ are with the same meaning. The wide-sense case gives a larger range for colluders to generate a pirated code. Where, marks in the detectable positions can be chosen from alphabet Q.
- Expanded wide-sense envelope $E^*(X)$ is the set

$$E^*(X) = \{y \in \{Q \cup \{*\}\}^l | \\ y|_{Z(X)} = x^1|_{Z(X)} = \cdots = x^t|_{Z(X)}\}$$

  Based on $E(X)$, the expanded wide-sense case allows the marks in the detectable positions to be changed into an unreadable state.

It is clear that $e(X) \subseteq E(X) \subseteq E^*(X)$ and $e(X) \subseteq e^*(X) \subseteq E^*(X)$. In addition, few fingerprinting codes are defined on the expanded narrow-sense envelope $e^*(X)$.

3.2 Major Existing Fingerprinting Codes

Based on the above-mentioned attack assumptions, major existing fingerprinting codes are introduced without giving their strict definition here. Please refer to the corresponding reference for details.

Let us consider (l, n, q)-code C. As described above, a pirated code is an l-tuple that is produced by the colluders whose fingerprints are codewords from C. The major existing codes are introduced below.

1. c-frameproof code [1]–[3]
   In c-frameproof code, no collusion of at most c users can frame a user who is not a member of the collusion.
2. c-secure frameproof code [1], [3]
   A c-secure frameproof code is a stronger form of c-frameproof code. A code is c-secure frameproof code if it is impossible for a collusion C_1 of size maximum c to frame a disjoint collusion C_2 of size maximum c by generating a pirated code that could have been generated by C_2.
   c-frameproof code and c-secure frameproof code do not have traceability, namely, the identification of guilty users cannot be guaranteed.
3. c-identifiable parent property code [1], [12]
   In c-identifiable parent property code, no collusion of size maximum c can generate an l-tuple that cannot be traced back to at least one member of the collusion.
4. c-traceability code [1], [4], [12]
   A c-traceability code is also a c-identifiable parent property code, but a c-traceability code has an advantage that it allows an efficient (i.e., linear-time) algorithm to determine one member of the collusion.
5. separating code [5]
   Separating code is equivalent to secure frameproof code, which is described by a separating system.
6. c-secure code with ϵ-error [2], [6]–[10]
   In a c-secure code with ϵ-error, a member of collusion of maximum size c can be traced back with probability at least $1 - \epsilon$.

Table 1 Taxonomy-1: Research on designing fingerprinting codes.

1.code designing	narrow-sense	wide-sense	expanded wide-sense
under marking assumption	In [1], q-ary FP, SFP, IPP and TA code are defined, and their relationship is investigated, where TA $\Rightarrow$ IPP $\Rightarrow$ SFP $\Rightarrow$ FP; q-ary SPC is also studied in [5], which is equivalent to SFP.	In [3], binary FP and SFP code are defined, and their relationship is SFP $\Rightarrow$ FP.	In [2], binary FP and SEC are investigated. A secure code with ϵ-error restricted in narrow-sense is an IPP code if $\epsilon = 0$.
under weak marking assumption	In [6], weak marking assumption allows the mark in the undetectable position to be erasable but assumes that the colluders can only erase a percentage of the marks, where q-ary c-secure code with ϵ-error, c-traceablity code tolerating e-erasures are defined.		
under relaxed marking assumption	The relaxed version of the marking assumption in [7] is: At any position where the codeword of all pirates agree, the pirates still have a δ probability of being able to output a different symbol. The binary SEC [2] is generalized in [7].		
no marking assumption	Tardos [8] shows that his improvements on the scheme [2] prevent framing innocent users even when the marking assumption does not hold; [9] proposes a variant of Tardos code which is practical for various applications against a small number of pirates.		

Table 2 Taxonomy-2: Research on joint coding-embedding fingerprinting.

2. joint coding-embedding	
under marking assumption	In [12], q-ary TA and IPP are defined under narrow-sense case and that TA codes generally offer better collusion resistance than IPP codes is demonstrated.
	In [13], binary ACC is defined, but no explicit construction of efficient and effective ACC.
	In [14], a new constructive algorithm for binary ACC is presented based on GD-PBIBD (group-divisible partially balanced incomplete block design) theory, which is more efficient in designing fingerprinting code. Further, [15] also describes a method to embed and detect fingerprinting code derived from GD-PBIBDs.
	Several fingerprinting schemes (e.g., [16], [17] and [18]) have been proposed to improve the BIBD based ACC.
Remark: [11] presents non-coded fingerprinting system, which is easy to implement but the required number of spreading sequences and the computational complexity of detection would increase linearly with the number of users.	

7. k-anti-collusion code (ACC) [13], [14]
   k-anti-collusion code (ACC) has the property that the composition of any subset of k or fewer codewords is unique and therefore can identify groups of k or fewer colluders.

   Remark: The variations of above codes are also proposed in the related works, some of which performed better due to the modifications of their parameters. For example, there exist several generalized versions of c-secure code in [7]–[10] such as Tardos code, where the marking assumption is relaxed.

3.3 Analyses of Fingerprinting Codes

For the brevity of description, the abbreviations used in this paper are listed below.

FP	frameproof code
SFP	secure frameproof code
IPP	identifiable parent property code
TA	traceability code
SPC	separating code
SEC	secure code with ϵ-error
ACC	anti-collusion code

Several surveys on existing fingerprinting codes are conducted. Table 1 and Table 2 classify them according to the attack assumptions mentioned previously. Where, "TA $\Rightarrow$ IPP $\Rightarrow$ SFP $\Rightarrow$ FP" means that a c-TA code is a c-IPP code, a c-IPP code is a c-SFP code and a c-SFP code is a c-FP code. Note that generally the converse is not true.

In the case of weak/relaxed/no marking assumption, the probability of accusing an innocent user or the probability of identifying a colluder incorrectly is not zero (see Table 1). This will weaken the force for confirming the illegal user, so those cases under weak/relaxed/no marking assumption are not considered. SEC is also not considered for the same reason. According to [20], it is possible to design a perfect fingerprinting code if using the marking assumption as a foundation. In the meantime, whether marking assumption holds or not depends on the embedding approach of a fingerprinting code, we focus on how to design codes and do not discuss how to meet the marking assumption in this paper.

The authors assume that marking assumption holds and only take account of the fingerprinting codes (except SEC) under marking assumption, which include FP, SFP, IPP, TA and ACC codes. TA code is stronger than FP, SFP, and IPP codes in terms of tracing ability, i.e., TA code is more effective (more secure against collusion attacks) than FP, SFP, and IPP codes. To implement the live pay-television broadcasting system, ACC is more appropriate than TA code in terms of efficiency and effectivity. The main reason lies in three aspects:

1. ACC is adaptable for multimedia data while TA is adaptable for text data.
   - ACC traces or identifies colluders by utilizing the fact that the common marks in the undetectable positions among any group of colluders are unique. The marks in the detectable positions

can be allowed to change into any state including unreadable state, which provides the resistance to linear attacks or nonlinear attacks for multimedia data.
For multimedia data, components of the fingerprinting sequence are spread over the whole object rather than linked to a small area in the object. Different bits of fingerprinting code that are additively embedded in multimedia may not be easily identifiable and arbitrarily manipulated. Thus, from colluders' perspective, collusion attacks like averaging, interleaving, "cut and paste"and so on, are easy to operate instead of combining their fingerprinting bits.

- TA is defined under narrow-sense case and the marks in the detectable positions can only be changed by combinatorial methods, where, the pirated code stems from combination of colluders' fingerprinting bits. TA traces or identifies one colluder based on the fact that the hamming distance between the pirated code and the colluders' codewords is smaller than others not in collusion. Therefore, TA only adapts to combinatorial attacks, which is far from the requirement for protecting multimedia data.

2. ACC has higher efficiency than TA
One sufficient condition for the existence of the TA codes is given in [1]: there exist an (l, n, q) c-TA code, where $q = 2c^2$ and $l = 4c^2 \log n$. The code length is $4c^2 \log n$ with the maximum tolerated collusion size c for n users. If an ACC code derived from (l, c, λ)-BIBD (see **Definition 4**) exists, $n = \lambda(l^2 - l)/(c^2 - c)$ users are supported, with the maximum tolerated collusion size $c - 1$. The code length is l, which is approximately $O(c\sqrt{n})$. It is clear that the length of TA is longer than ACC with the same n and c. According to the **Definition 1**, ACC has higher efficiency than TA. Note that there exist q-ary and binary TA codes while there only exist binary ACC codes.
3. ACC has higher effectivity than TA
Within the maximum tolerated collusion size, ACC identifies all colluders while TA only traces one colluder.

ACC is adaptable for multimedia data and more efficient and effective than TA, while TA is more effective than FP, SFP and IPP codes. So the authors regard ACC as the most desirable fingerprinting code for the internet-based pay-TV system. However, the practical construction of ACC has been still a big problem. Consequently, the authors devote to answering how to generate and construct efficient and effective ACC.

4. How to Generate Efficient and Effective ACC

4.1 Status Quo of ACC

In [13], an anti-collusion code (ACC) is defined as follows:
Definition 3: Let $G = \{0, 1\}$. A code $C = \{c_1, \cdots, c_n\}$ of vectors belonging to G^v is called a K-resilient AND anti-collusion code (AND-ACC) when any subset of K or fewer codevectors combined element-wise under AND is distinct from the element-wise AND of any other subset of K or fewer codevectors.

v denotes the dimension of vector space G, i.e., the length of codevectors $c_1, \ldots, c_n$ are v. A generation method of ACC is using BIBD.
Definition 4: A (v, k, λ) balanced incomplete block design (BIBD) is a pair $(\mathcal{X}, \mathcal{A})$, where $\mathcal{A}$ is a collection of k-element subsets (blocks) of a v-element set $\mathcal{X}$, such that each pair of elements of $\mathcal{X}$ occur together in exactly λ blocks.

A (v, k, λ)-BIBD has a total of $n = \lambda(v^2 - v)/(k^2 - k)$ blocks. Corresponding to it, there is the $v \times n$ incidence matrix $M = (m_{ij})$ defined by

$$m_{ij} = \begin{cases} 1, & \textit{if the ith element belongs to the jth block,} \\ 0, & \textit{otherwise.} \end{cases}$$

If the code matrix C is defined as the bit complement of M and the codevectors c_j are assigned as the columns of C, then a $(k - 1)$-resilient AND-ACC is obtained, with code length v for n users. [13] gave out the following theorem.

Theorem 1: Let $(\mathcal{X}, \mathcal{A})$ be a $(v, k, 1)$-BIBD and M the corresponding incidence matrix. If the codevectors are assigned as the bit complement of the columns of M, then the resulting scheme is a $(k - 1)$-resilient AND-ACC.

According to [19], a $(v, k, 1)$-BIBD is unavailable when the number of users is more than 1641. Hence, how to generate and how to construct ACC still remain problems due to most conditions on parameters v, k and λ are necessary condition but not sufficient condition for the existence of a BIBD. PBIBD (partially balanced incomplete block designs) as a natural extension of BIBD is developed to deal with this drawback in combinatorial design theory.

In [14], InKoo Kang et al. aimed to support several thousands or more users and overcome the limitation of BIBD existence. They presented a recursive construction algorithm for ACC using GD (group-divisible) PBIBD and the derived ACC can support more users than those derived from the table of BIBDs [19]. Differently from BIBD theory in which there is only one group and any two elements occur together only λ times, GD-PBIBD can flexibly control elements replication numbers in block such that the resulting fingerprinting code has high efficiency than BIBD. However, ACCs derived from GD-PBIBD show a worse collusion resistance. For example, A $(243, 6561, 3)$-code from the GD-PBIBD can support 6561 users but only resist

Table 3 Existing examples of BIBD.

Block Design	(v, k, λ)
affine plane	$(m^2, m, 1)$
Fano plane	$(7, 3, 1)$
Hadamard design	$(4m + 3, 2m + 1, m)$
projective plane	$(m^2 + m + 1, m + 1, 1)$
Steiner triple system	$(v, 3, 1)$
unital	$(p^3 + 1, p + 1, 1)$

Table 4 Evaluations of unital and affine plane.

Block Design $(v, k, 1)$ $n = v(v-1)/k(k-1)$	*Efficiency* $\beta = n/v$	*Effectivity* $\gamma = (k-1)/n$
affine plane $(m^2, m, 1)$	$1 + 1/m$	$(m-1)/m(m+1)$
unital $(p^3 + 1, p + 1, 1)$	$p - 1 + 1/(p+1)$	$1/p(p^2 - p + 1)$

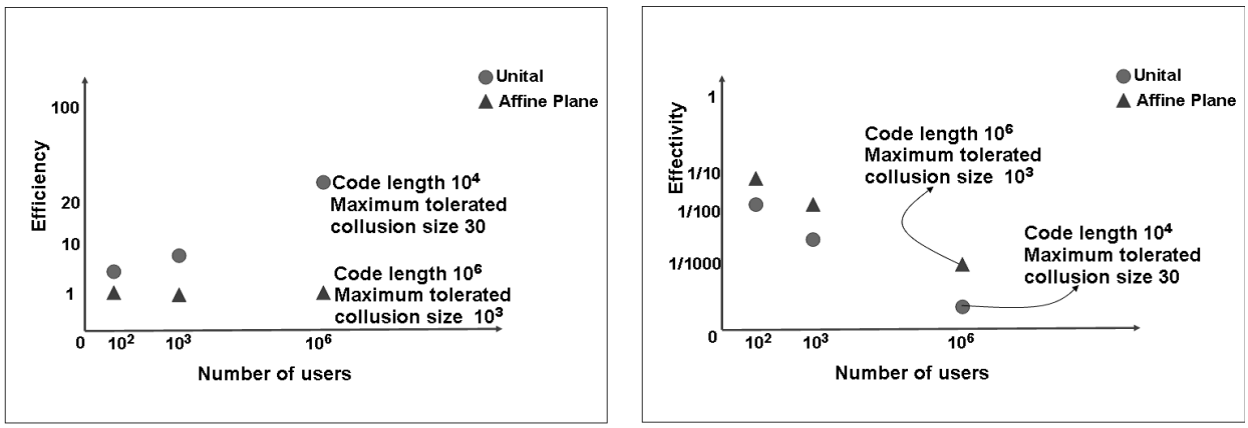

Fig. 1 Performance comparisons of unital and affine plane under the same number of users.

against 2 colluders, which is not practical. To meet the requirements of the internet-based live pay-TV system, ACCs which can not only support more users but also resist against more colluders (namely, high *efficiency* and high *effectivity*) are required. [13] pointed out that another approach to construct BIBDs is to use d-dimensional projective and affine geometry, which include some known examples of BIBD. Attributing to their definite existences, the authors attempt to explore the known examples of BIBD for generating desirable ACCs.

4.2 Existing BIBD Examples

Low-density parity-check codes (LDPC, [21]) are constructed based on unital design, which also motivates the authors to generate ACC from known examples of BIBD, which is listed in Table 3. Hadamard design is not considered because ACC can be achieved only when $\lambda = 1$ (**Theorem 1**). Neither Fano plane nor Steiner triple system is under consideration since their parameters are either constant or small. Projective plane is excluded due to its symmetry ($v = n$), whatever the parameter m is, the *efficiency* $\beta = 1$ since the code length is equal to code size. Unital and affine plane show better *efficiency* and *effectivity* than the above-mentioned, their performances on live pay-TV broadcasting are investigated below.

4.3 Unital and Affine Plane

Given a $(k-1)$-resilient ACC (derived from $(v, k, 1)$-BIBD), let the number of users that can be supported be n. Then, the *efficiency* on such an ACC is $\beta = n/v$ and the *effectivity* on such an ACC is $\gamma = (k-1)/n$.

Unital and affine plane are investigated in terms of *efficiency* and *effectivity* and their details are listed in Table 4. To make more clearly, we compared their efficiency and effectivity under the same condition. Assume that unital and affine plane could support the same number of users in Fig. 1, it is obvious that unital has higher *efficiency* but lower *effectivity*, and affine plane has higher *effectivity* but lower *efficiency*. In other words, unital supports more users but exhibits weaker resistance, while affine plane exhibits stronger resistance but supports fewer users. Assume that they could resist against colluders with the same maximum tolerated collusion size in Fig. 2 or be at the same code length in Fig. 3, similar conclusions will be drawn. It should be pointed out that Fig. 1, Fig. 2 and Fig. 3 are merely schematic illustrations, which assist us with examining the

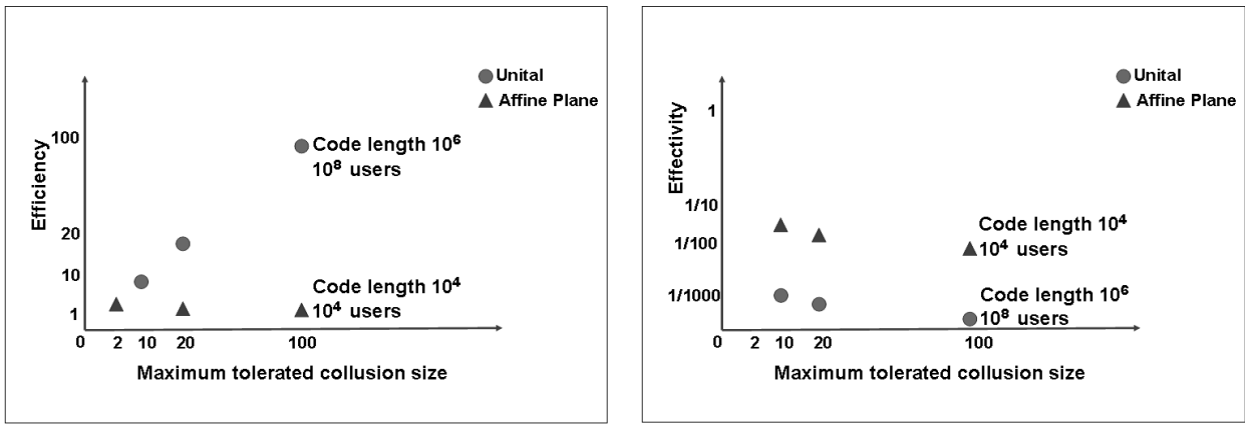

Fig. 2 Performance comparisons of unital and affine plane under the same maximum tolerated collusion size.

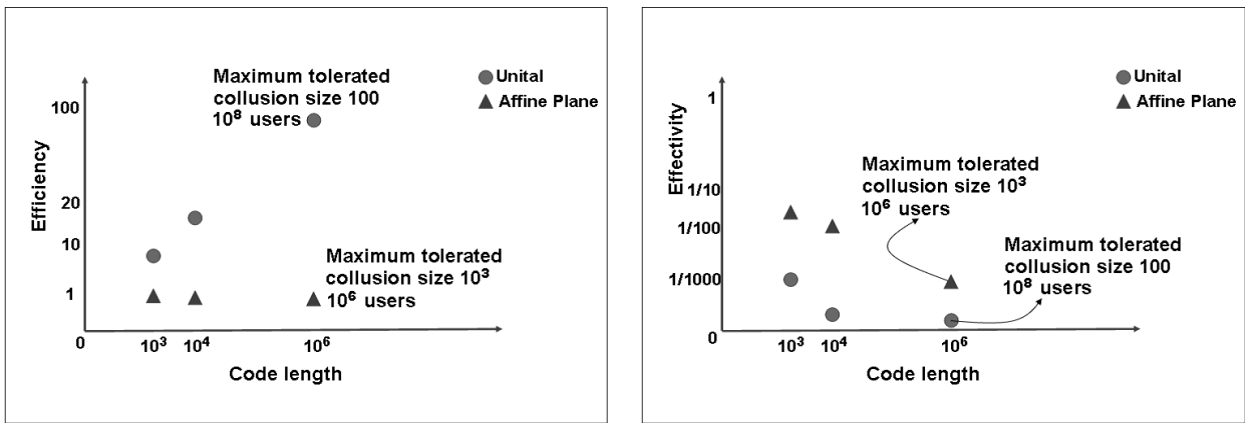

Fig. 3 Performance comparisons of unital and affine plane under the same code length.

Table 5 Performance evaluations of unital code and affine plane code.

Case	Block Design	Number of Users n	Code Length l	Collusion Size $(k-1)$
(a) under the same number of users	affine plane $(m^2, m, 1)$	$n = m(m+1)$ $n: 10^4$	$l = m^2$ $l: \sim 10^4$	$k = m$ $k: \sim 100$
	unital $(p^3+1, p+1, 1)$	$n = p^2(p^2-p+1)$ $n: 10^4$	$l = p^3+1$ $l: \sim 10^3$	$k = p+1$ $k: \sim 10$
(b) under the same collusion size	affine plane $(m^2, m, 1)$	$n = m(m+1)$ $n: \sim 10^4$	$l = m^2$ $l: \sim 10^4$	$k = m$ $k: 101$
	unital $(p^3+1, p+1, 1)$	$n = p^2(p^2-p+1)$ $n: \sim 10^8$	$l = p^3+1$ $l: \sim 10^6$	$k = p+1$ $k: 101$
(c) under the same code length	affine plane $(m^2, m, 1)$	$n = m(m+1)$ $n: \sim 10^4$	$l = m^2$ $l: 10^4$	$k = m$ $k: \sim 100$
	unital $(p^3+1, p+1, 1)$	$n = p^2(p^2-p+1)$ $n: \sim 2\times 10^5$	$l = p^3+1$ $l: 10^4$	$k = p+1$ $k: \sim 20$

performances of unital and affine plane.

Performances of unital and affine plane are further evaluated through the following concrete examples.

Assume that unital code (ACC derived from unital) and affine plane code (ACC derived from affine plane) support the same amount of users, for example, 10^4 users, their code length, collusion size are shown in Table 5 (a). Affine plane code exhibits very high resistance where its maximum tolerated collusion size is about 100. It may be considered difficult to bring together 100 illegal users simultaneously.

Assuming the collusion sizes of unital code and affine plane code as 101 (Table 5 (b)), unital code can support about 10^8 users. Our goal is to support $10^4 \sim 10^5$ users, so these codes are supposed to serve well our multicasting system.

Similarly, assume that unital code and affine plane code

have the same implementing facility, for example, their code length are fixed to 10^4 bits (Table 5 (c)). Affine plane code performs better with respect to collusion resistance and unital code performs better with respect to the amount of users.

For live pay-television broadcasting system, unital code is preferable when higher *efficiency* is required and affine plane is preferable when higher *effectivity* is required.

5. Constructions of Unital Code and Affine Plane Code

We construct unital and affine plane code based on finite geometry and implement them on Magma

http://magma.maths.usyd.edu.au/magma/

5.1 Constructing Unital Code

The results about the existence of unital are: Unital is constructed from finite projective plane ($(m^2+m+1, m+1, 1)$-BIBD) and the only known projective planes have orders that are prime powers (order $m = p^d$).

This paper adapts hermitian unital to construct ACC because it has deterministic construction.

Definition 5: A hermitian unital in a projective plane of order p^2 is the set of p^3+1 points that meets every line $p+1$ points.

The constructing procedure is as follows:

1. Constructing finite projective plane $PG(2, p^2)$
   a. Define the points as follows:
      - Consider triples $X = (x, y, z)$ of elements from the finite field $GF(p^2)$, where (x, y, z) are not all zero. There exist $(p^2)^3 - 1$ triples.
      - Identify triples X and Y if $Y = \eta X$ for some non-zero $\eta \in GF(p^2)$, and say that X and Y are equivalent. Denote the equivalence class of X by $[X]$. Each equivalence class has p^2-1 members, corresponding to the $p^2 - 1$ possible non-zero values of η, and so there are $[(p^2)^3 - 1]/(p^2 - 1) = (p^2)^2 + p^2 + 1$ different classes $[X]$, which is taken as the point of $PG(2, p^2)$.

   b. Define the lines (or blocks) as follows:
      If $A = (a_0, a_1, a_2)$ is a triple of elements of $GF(p^2)$, not all zero, define the line $[\alpha]$ to be the set of all points $[X]$ such that $a_0x + a_1y + a_2z = 0$. There are $(p^2)^2 + p^2 + 1$ lines by an argument similar to the case for points in (a). On each line, there are $p^2 + 1$ points.

2. Constructing hermitian unital
   The hermitian unital is the set of points $(x, y, z) \in PG(2, p^2)$ satisfying $x^{p+1} + y^{p+1} + z^{p+1} = 0$.
   There are p^3+1 points in hermitian unital, which meets every line (described in (b)) in $p + 1$ points. Hermitian unital can be denoted by pair $(\mathcal{H}, \mathcal{B})$, where $\mathcal{H}$ is the set of $p^3 + 1$ points and $\mathcal{B}$ is the collection of subsets consisting of $p+1$ intersection points (referring to **Definition 4**).
3. Constructing unital code
   The hermitian unital $(\mathcal{H}, \mathcal{B})$ is a $(p^3+1, p+1, 1)$-BIBD. Compute its corresponding incidence matrix M, take the bit complement of the columns of M as codewords and assign them to users. Thus, a p-resilient AND-ACC is achieved according to **Theorem 1**.

The above constructing procedure of hermitian unital is implemented in Magma. Magma is a computer algebra system and designed to provide a software environment for computing with the structures which arise in areas such as algebra, number theory, algebraic geometry and (algebraic) combinatorics. Its online calculator exists on

http://magma.maths.usyd.edu.au/calc/

Using the Magma codes in [22], we can generate an incidence matrix M of $(p^3 + 1, p + 1, 1)$-BIBD in practical computation time, where $p < 13$. For example, when $p = 11$, the execution time is 3.629 seconds and memory usage is 44.53 MB (Pentium M 1.20 GHz, 504 MB RAM). Taking the bit complement of the columns of M as codewords, a unital code with code length 1332, code size 13431 (i.e., the number of users is 13431), and the maximum tolerated collusion size 11 can be easily achieved in short time.

5.2 Constructing Affine Plane Code

There exists an affine plane $(m^2, m, 1)$-BIBD of order m if and only if there exists a projective plane $(m^2 + m + 1, m + 1, 1)$-BIBD of order m. A finite projective plane $(m^2 + m + 1, m + 1, 1)$-BIBD exists when the order m is a prime power (i.e., $m = p^d$). Affine plane code is constructed as below:

1. Construct a finite projective plane $PG(2, p^2)$ with Magma [22].
2. Construct a finite affine plane $AG(2, p^2)$ by deleting a single line $[\iota]$ of $PG(2, p^2)$ and all of the points of it, and altering the other lines to delete the point which is intersection of this line and line $[\iota]$.
3. Compute the corresponding incidence matrix M of $AG(2, p^2)$, then the bit complement of the columns of M are corresponding to users' codewords.

Affine plane codes can be obtained when $p < 13$. Especially, when $p = 11$, an affine plane code with code length 14641, code size 14762 (i.e., be able to support 14762 users), and the maximum tolerated collusion size 120 can be achieved.

With Magma calculator, affine plane $AG(2, p^2)$ and its incidence matrix can be directly achieved when $p \leq 7$. For example, when $p = 7$, the source code is:

```
p:=7;
m:=p^2;
FAP:=FiniteAffinePlane(m);
IM:=IncidenceMatrix(FAP);
```

As a result, the execution time is 0.660 seconds and the

memory usage is 30.89 MB.

Remark: In the case of constructing unital code and affine plane code, the restrictions on parameter p (for instance, $p < 13$ or $p \leq 7$) are derived from using Magma. The problem such as "Computation used more memory than allowed"will occur on Magma when $p \geq 13$ or $p > 7$.

On the other hand, the main issue of ACC is a practical construction problem. In the tables [19], a $(1641, 41, 1)$-BIBD which can support 1641 users exists, but there is not definite results if a (v, k, λ)-BIBD exists when the number of users exceeds 1641. Thus, an ACC for more than 1641 users can not be achieved from general BIBDs. Here, our proposed unital code (adapt to 13,431 users) and affine plane code (adapt to 14,762 users) are not subject to this situation. For the goal of supporting $10^4 \sim 10^5$ users simultaneously, our construction methods provide potential use for more than 14,762 users since the restriction $p < 13$ comes from Magma software rather than from the finite geometry theory. Through our method, the code corresponding with the concrete requirements can be constructed flexibly by compromising its three factors, number of users, collusion size and code length. This is desirable from content distributors' point of view. In addition, the practical computation time indicates that our construction methods for unital and affine plane codes are feasible to support large number of users.

6. Comparison between Our Proposed ACCs and Recent Related Works

6.1 Comparing Unital and Affine Plane Codes with GD-PBIBD Codes

According to [14], the bit complement of a $(v, b, r, k, \lambda_1, \lambda_2)$-GD-PBIBD with $\lambda_1 = 0$ and $\lambda_2 = 1$ is an AND-ACC for $b = n$ users with code length v and collusion size $k - 1$. [14] proposed a fingerprinting code set for $n = p^{2s-2}$ users, $p - 1$ colluders and a code length of p^s using a $(p^s, p^{2s-2}, p^{s-1}, p, 0, 1)$-GD-PBIBD for p, a prime number, and s, a positive integer ($s \geq 2$). p and s are user-defined numbers. A $(243, 6561, 3)$-code (for $n = 3^{2*5-2} = 6561$ users with code length $l = 3^5 = 243$ and collusion size $k = 3 - 1 = 2$) from the GD-PBIBD is obtained when $p = 3$ and $s = 5$. The code length p^s will become remarkably long when increasing the collusion size $p - 1$. For instance, the code length is 10^{10} when the collusion size is 101 like Table 5 (b), while our unital code length is 10^6 and affine plane code length is 10^4.

The *efficiency* on the ACC derived from the above-described GD-PBIBD is $\beta = n/v = p^{2s-2}/p^s = p^{s-2}$ and *effectivity* is $\gamma = (k-1)/n = (p-1)/p^{2s-2}$. In terms of *efficiency* and *effectivity*, we compared GD-PBIBD code with unital and affine plane code under the same maximum tolerated collusion size (see schematic illustration Fig. 4). The following relation holds when there are more than 2 colluders.

$$min(\beta_{GD-PBIBD}) > \beta_{unital} > \beta_{affine\ plane}$$

$$max(\gamma_{GD-PBIBD}) < \gamma_{unital} < \gamma_{affine\ plane}$$

$\beta_{GD-PBIBD}$ and $\gamma_{GD-PBIBD}$ denote the *efficiency* and *effectivity* on GD-PBIBD codes, respectively. In GD-PBIBD, s is a user-defined parameter. We assume $s \geq 3$, then $min(\beta_{GD-PBIBD})$ and $max(\gamma_{GD-PBIBD})$ are the *efficiency* and *effectivity* at $s = 3$.

Although the efficiency on unital and affine plane codes is not as high as GD-PBIBD, they can also support more users than those general BIBDs in [19], which is comparable to GD-PBIBD. Besides, our proposed fingerprinting codes provide more practical collusion resistance than GD-PBIBD.

6.2 Comparing Unital and Affine Plane Codes with Tardos Codes

It is needless to say that code length is a key factor in code implementation. Because c-secure codes are too long to use in practical, Tardos codes have been paid more attentions in recent related works [8]–[10]. One reason is that Tardos code has length of theoretically minimal order among all

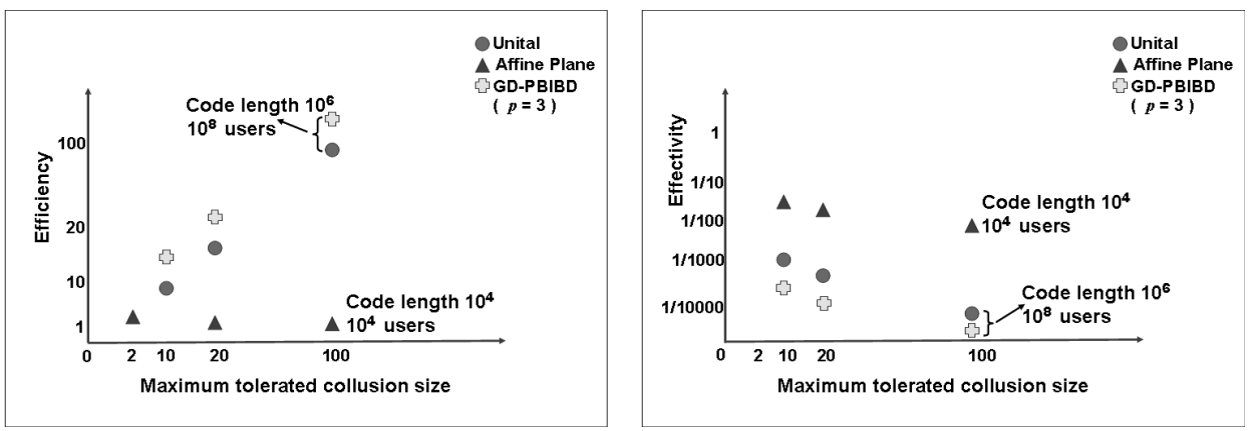

Fig. 4 Performance comparisons of unital, affine plane and GD-PBIBD under the same maximum tolerated collusion size.

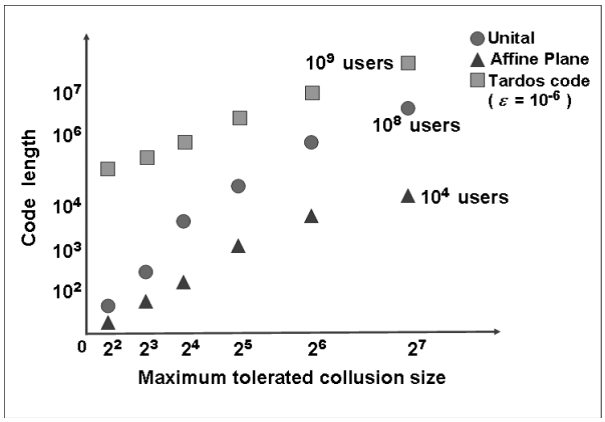

Fig. 5 Comparison of code length between our proposed codes and Tardos codes.

possible c-secure codes with respect to the maximum tolerated collusion size c. Another reason is that when the probability of accusing an innocent user and the probability of not accusing any guilty one are very small (e.g., 10^{-6}), Tardos code can be considered secure against collusion.

We stress that ACCs are more secure in respect of collusion resistance than Tardos codes. Detailly, within the maximum tolerated collusion size,

- ACCs identify all the colluders, so the error of accusing an innocent user and the error of not accusing any guilty one are zeros;
- Tardos codes attempt to restrict the error probability of accusing an innocent user and the error probability of not accusing any guilty one to a reasonable value.

It should be pointed out that ACCs are intrinsically different from Tardos codes. More elaborately, the three factors of ACC, number of users, collusion size and code length, are not independent. Whichever is fixed, and then the other two are fixed correspondingly. In contrast, the code length of Tardos code mainly depends on its collusion size. We give a schematic illustration (Fig. 5) to show the difference between our proposed codes and Tardos codes with respect to code length under the same collusion size. The related data about Tardos code come from [10], where, the number of users is 10^9 and the error probability ϵ is 10^{-6}.

As shown in Fig. 5, our proposed unital and affine plane codes have shorter code length than Tardos codes with respect to the same collusion size. So, unital and affine plane codes are more feasible to implement.

6.3 Comparing Unital and Affine Plane Codes with Enhanced ACCs

In the last few years, several fingerprinting schemes (e.g., [16], [17] and [18]) have been proposed to enhance $(v, k, 1)$-BIBD based ACC. As mentioned previously, one drawback of $(v, k, 1)$-BIBD based ACC is the inability to support large number of users. To support more users, [16] attempted to make large number of fingerprints from small $(v, k, 1)$-BIBD. In [16], there are v orthogonal basis signals s_i and the binary codevector c_j for the j-th user is encoded into a fingerprint signal $w_j = (s_1, s_2, \ldots, s_v) \cdot c_j$, where $\cdot$ indicates the scalar product of vectors. Then, the fingerprint signal w_j is extended by adding a Gaussian distributed random variable μ with a repetition constant N. For example, if $N = 4$, all the fingerprint signals are arranged as follows.

$$
\begin{array}{cccc}
w_1 & \mu & \mu & \mu \\
w_2 & \mu & \mu & \mu \\
\vdots & & & \vdots \\
w_n & \mu & \mu & \mu \\
\mu & w_1 & \mu & \mu \\
 & \vdots & \vdots & \\
\mu & w_n & \mu & \mu \\
 & & \vdots & \\
\mu & \mu & \mu & w_1 \\
 & & \vdots & \vdots \\
\mu & \mu & \mu & w_n
\end{array}
$$

Therefore, the extended fingerprint signals can support up to $N \times n$ users, where n is the number of users which is supported by ACC before extension. It is obvious that supporting more users will result in longer fingerprint signal. We can not directly calculate the efficiency of the enhanced ACC since there is no clear code structure. However, we can estimate that the effectivity is decreased to $1/N$ of the value before extension since collusion size remains unchangeable while the number of users is increased from n to $N \times n$. Further, the extension will lead to more overhead with the number of users increasing in contrast to our proposed unital and affine plane codes which merely adjust their parameters (p and m) to support large number of users.

Another drawback of the $(v, k, 1)$-BIBD based ACC is the failure of detection when the number of colluders exceeds $(k - 1)$. To cope with this, [17] modified the incidence matrix of BIBD to form fingerprinting codes that are collusion resistant even if all the users collude. However, the collusion resistance against all users results in a dramatic increase of code length. The efficiency is decreased to $(k - 1)/(k^2 - k + 1)$ from 1, where v is considered to be proportional to k^2 when $n = v$ and the number of users n is large. Regardless of code length, our proposed unital and affine plane codes can also be resistant to collusion from all users if setting the collusion size as the number of users.

In addition to the above drawbacks, error extraction of every single bit of ACC may lead to the misjudgment of a user. In order to enhance the resistance of ACC and improve the reliability of ACC when transmitting, [18] took advantage of the good error correcting capability of Turbo code and combined Turbo code with ACC. Similarly, our proposed unital and affine plane codes can be improved through combining with the error correcting code.

7. Conclusions and Future Work

We aim to achieve fingerprinting codes for internet-based pay-TV system. By conducting analyses on fingerprinting codes, this paper has shown that ACC has advantages over other existing fingerprinting codes in terms of efficiency and effectivity. Next, it is presented how to generate efficient and effective ACC by using unital and affine plane. The generated unital and affine plane codes exhibit higher collusion resistance than GD-PBIBD codes and have shorter code length than Tardos codes. Then, their practical construction methods are given out based on finite geometry. Last, unital and affine plane codes which can support more than 10^4 users are derived by the implementation on Magma. The practical computation time indicates that our construction methods for unital and affine plane codes are feasible to support large number of users. We consider these codes are applicable to internet-based pay-TV system.

Our work is a theoretical foundation of fingerprinting code for internet-based pay-TV broadcasting system. Whether the proposed ACC code can identify each individual user and survive collusion attacks closely depends on the process of fingerprint embedding/detection, and the performance of the proposed ACC code may vary with the implementation environment. We discussed these issues and gave out the simulations on fingerprinting real image/video in [25]. As a future work, we intend to implement the achieved ACCs on internet-based pay-TV broadcasting and to examine their validity further.

References

[1] J.N. Staddon, D.R. Stinson, and R. Wei, "Combinatorial properties of frameproof and traceability codes," IEEE Trans. Inf. Theory, vol.47, no.3, pp.1042–1049, 2001.
[2] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," IEEE Trans. Inf. Theory, vol.44, no.5, pp.1897–1905, 1998.
[3] D.R. Stinson, T. van Trung, and R. Wei, "Secure frameproof codes, key distribution patterns, group testing algorithms and related structures," J. Statistical Planning and Inference, vol.86, pp.595–617, 2000.
[4] L. Tina, L. Jacob, and S. Mattias, "A class of traceability codes," IEEE Trans. Inf. Theory, vol.48, no.7, pp.2094–2096, 2002.
[5] A. Barg, G.R. Blakley, and G.A. Kabatiansky, "Digital fingerprinting codes: Problem statements, constructions, identification of traitors," IEEE Trans. Inf. Theory, vol.49, no.4, pp.852–865, 2003.
[6] R. Safavi-Naini and Y. Wang, "Collusion secure q-ary fingerprinting for perceptual content," LNCS, vol.2320, pp.57–75, 2002.
[7] H.-J. Guth and B. Pfitzmann, "Error- and collusion-secure fingerprinting for digital data," LNCS, vol.1768, pp.134–145, 2000.
[8] G. Tardos, "Optimal probabilistic fingerprint codes," Proc. 35th Annual ACM Symposium on Theory of Computing, pp.116–125, 2003.
[9] M. Hagiwara, G. Hanaoka, and H. Imai, "A short random fingerprinting code against a small number of pirates," LNCS, vol.3857, pp.193–202, 2006.
[10] T. Isogai and H. Muratani, "Reevaluation of Tardos's code," IEICE Technical Report, ISEC2006-96, 2006.
[11] Z.J. Wang, M. Wu, H. Zhao, W. Trappe, and K.J.R. Liu, "Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation," IEEE Tran. Image Process., vol.14, no.6, pp.804–821, 2005.
[12] S. He and M. Wu, "Performance study on multimedia fingerprinting employing traceability codes," LNCS, vol.3710, pp.84–96, 2005.
[13] 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.
[14] I.K. Kang, K. Sinha, and H.-K. Lee, "New digital fingerprint code construction scheme using group-divisible design," IEICE Trans. Fundamentals, vol.E89-A, no.12, pp.3732–3735, Dec. 2006.
[15] I.K. Kang, C.-H. Lee, H.-Y. Lee, J.-T. Kim, and H.-K. Lee, "Averaging attack resilient video fingerprinting," IEEE International Symposium on Circuits and Systems, vol.6, pp.5529–5532, 2005.
[16] J.M. Seol and S.W. Kim, "Scalable fingerprinting scheme using statistically secure anti-collusion code for large scale contents distribution," LNCS, vol.4096, pp.560–569, 2006.
[17] D. Shashanka and P.K. Bora, "Collusion secure scalable video fingerprinting scheme," 15th International Conference on Advanced Computing and Communications, pp.641–647, 2007.
[18] J. Yang and X. Xu, "A robust anti-collusion coding in digital fingerprinting system," IEEE Asia Pacific Conference on Circuits and Systems (APCCAS2006), pp.996–999, 2006.
[19] J.H. Dinitz and D.R. Stinson, Contemporary Design Theory: A Collection of Surveys, John Wiley & Sons, 1992.
[20] S. Engle, "Fingerprinting and the marking assumption," Ecs228 Cryptography for E-Commerce, 2005.
[21] S.J. Johnson and S.R. Weller, "High-rate LDPC codes from unital design," IEEE Global Telecommunications Conference, pp.2036–2040, 2003.
[22] J.D. Key, "Some applications of magma in designs and codes: Oval designs, hermitian unitals and generalized Reed-Muller codes," J. Symbolic Computation, vol.31, no.1/2, pp.37–53, Jan./Feb. 2001.
[23] I.B. Djordjevic and B. Vasic, "Projective geometry LDPC codes for ultralong-haul WDM high-speed transmission," IEEE Photonics Technol. Lett., vol.15, no.5, pp.784–786, 2003.
[24] J.-M. Seol and S.-W. Kim, "Efficient collusion-resilient RFID tag identification using balanced incomplete block design code," Proc. 6th IEEE International Conference on Computer and Information Technology, p.220, 2006.
[25] S. Hou, T. Uehara, T. Satoh, Y. Morimura, and M. Minoh, "Integrating fingerprint with cryptosystem for Internet-based live pay-TV system," Security and Communication Networks (Published online in Wiley InterScience), vol.1, no.6, pp.461–472, 2008.

Shuhui Hou received the B.S. and M.S. degrees in Mathematics from China, in 1993 and 1996, respectively. From 1996 to 2003, she had been with University of Science and Technology Beijing, China. She is currently a Ph.D. candidate in Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University. Her research interests are in cryptography, digital watermarking and digital fingerprinting.

Tetsutaro Uehara received the B.E, M.E., and D.E. degrees from Kyoto University in 1990, 1992 and 1996, respectively. From 1995 to 1996, he was a research associate of the Graduate School of Engineering, Kyoto University. From 1997 to 2003, he was an assistant professor of the Faculty of Systems Engineering, Wakayama University. From 2003 to 2005, he was an associate professor at the Center for Information Technology of the Graduate School of Engineering, Kyoto University. Since 2006, He has been an Associate Professor of Academic Center for Computing and Media Studies, Kyoto University. His research interest covers high-performance computing, system management technology, multimedia streaming technology and literacy education for security. He is a member of IPSJ, IEEJ, JSSST, ISCIE, CIEC, IN-LAW and IEEE.

Takashi Satoh received the B.E., M.E., and D.E. degrees from Tokyo Institute of Technology. Since 1999 he has been working for the Faculty of Environmental Engineering of the University of Kitakyushu, and now he is an associate professor from 2001. His current research interests include cryptography and network security. He is a member of IPSJ, IACR and IEEE.

Yoshitaka Morimura received B.E. degree in information science and master of informatics degree in intelligence science and technology from Kyoto University, in 2004 and 2006. He is a Ph.D. student in Kyoto University. He is engaged in research of video streaming for e-Leaning. He is a student member of IEEE and IPSJ.

Michihiko Minoh is a professor at Academic Center for Computing and Media Studies (ACCMS), Kyoto University, Japan. He received the B.Eng., M.Eng. and D.Eng. degrees in Information Science from Kyoto University, in 1978, 1980 and 1983, respectively. He has been a director of ACCMS since 2006. His research interest includes a variety area of Image Processing, Artificial Intelligence and Multimedia Applications, particularly, model centered frame work for the computer system to help visual communication among humans and information media structure for human communication. He is a member of Information Processing Society of Japan, the IEEE computer society and communication society, and ACM.