Cryptography and Communications (2024) 16:481–506
https://doi.org/10.1007/s12095-023-00682-y

RESEARCH

Frameproof codes, separable codes and B_2 codes: Bounds and constructions

Marcel Fernandez[1] · John Livieratos[2] · Sebastià Martín[3]

Received: 1 December 2022 / Accepted: 21 October 2023 / Published online: 10 November 2023
© The Author(s) 2023

Abstract
In this paper, constructions of frameproof codes, separable codes, and B_2 codes are obtained. For each family of codes, the Lovász Local Lemmais used to establish lower bounds for the codes. The obtained bounds match all best known bounds in the literature. Our strategy allows us to present constructions of the aforementioned codes, by using the variable framework for the Lovász Local Lemma.

Keywords Frameproof codes · Separable codes · B2 codes · Lovasz local lemma, Moser-Tardos algorityhm

Mathematics Subject Classification (2010) 94B65 · 05D40

1 Introduction

The concept of frameproof codes was coined in [1], where they are used to offer protection against illegal redistribution of digital goods. In that scenario, frameproof codes serve as underlying codes in fingerprinting schemes. In these schemes, a distributor prevents illegal redistribution of his/her goods by making delivered copies different. This is achieved by embedding a unique mark in each copy. Having distinguishable copies, clearly rules out

Marcel Fernandez, John Livieratos and Sebastià Martín contributed equally to this work

✉ Sebastià Martín
sebastia.martin@upc.edu

Marcel Fernandez
marcel.fernandez@upc.edu

John Livieratos
johnlivieratos89@gmail.com

[1] Department of Network Engineering, Universitat Politècnica de Catalunya, Jordi Girona, 31, Barcelona 08034, Spain

[2] Department of Mathematics, National and Kapodistrian University of Athens, Panepistimioupolis, 84, Athens, Greece

[3] Department of Mathematics, Universitat Politècnica de Catalunya, Jordi Girona, 31, Barcelona 08034, Spain

plain redistribution. Unfortunately, this strategy is weak against collusion attacks, where a coalition of malicious users compare their copies in order to detect the positions in which the embedded marks differ. The goal of these traitor users is to change the detected positions, in order to create a new pirate copy that masks their identities. This new pirate copy is the one that will be illegally redistributed. On the other hand, the goal of the distributor is to design the set of marks to be embedded in such a way that, given a pirate copy, traitors can be traced back.

Separable codes were proposed in [2] for the case of multimedia fingeprinting, where traitor users can perform averaging attacks. The connection between separable codes and-frameproof codes was discussed in [2, 3]. See also the work in [4]. Multimedia fingerprinting codes are also related to the family of B_2 codes, as was stated in [5]. The concept of B_2 codes [6] (known as 2-signature codes in multiple access communications) has its origins in the work of Sidon [7].

1.1 Our contribution

For the state of the art about bounds for separable codes, B_2 codes and frameproof codes, we refer the reader to the elegant expositions in [8] and [9]. In this paper, by means of the Lovász Local Lemma, we present proofs that match the already known lower bounds in [8] and [9]. We insist in proving bounds through the Lovász Local Lemma, because we can then use the approach that Giotis et al. [10] made of the variable framework developed by Moser and Tardos [11, 12], to devise an algorithm that constructs codes.

Therefore, the main contribution of this paper, along with the alternative proofs of the lower bounds, is the construction of separable codes, B_2 codes, and frameproof codes. In this sense it is a follow-up companion to [8] and [9], where only existence results were discussed. We stress that we also extend the work in [10], in order to establish the computational complexity of the algorithmic construction in a more precise manner.

2 Previous results

We adopt the notation of [8]. Let $\mathcal{Q}$ be an alphabet of size q. A code $\mathcal{C}$ of length n and size M is a subset of n-tuples in $\mathcal{Q}^n$, i.e. $\mathcal{C} = \{c_1, \ldots, c_M\}$. The n-tuples $c_i = (c_i(1), \ldots, c_i(n)) \in \mathcal{C}$, $1 \leq i \leq M$, are called code words. The distance between two code words c_i and c_j is the number of positions in which they differ, $|\{l \in \{1, \ldots, n\} : c_i(l) \neq c_j(l)|$. The minimum distance of a code is the smallest distance between any two distinct code words, and will be denoted by d. In this case we say that $\mathcal{C}$ is an $(n, M, d)_q$ code or $(n, M)_q$ for short.

Given an $(n, M)_q$ code $\mathcal{C}$, we form an $M \times n$ matrix C by writing the M code words as its rows. We denote the row set of C as $\{c_1, \cdots, c_M\}$. A set U of t rows $\{c_{i_1}, \ldots, c_{i_t}\}$ will be denoted as U^t.

If $\mathcal{Q}$ is the finite field $\mathbb{F}_q$, we can take $\mathcal{C}$ to be a vector subspace of $\mathbb{F}_q^n$. Then the size of the code is $M = q^k$, where k is the dimension of the subspace. In this case we say we have an $[n, k, d]_q$ linear code.

The rate of an $(n, M)_q$ code, which is an important parameter, is defined as

$$R = \frac{\log_q M}{n}. \tag{1}$$

In order to obtain families of codes with good asymptotic rate, we will make use of Forney's idea of code concatenation [13]. We take an *inner* code defined over a small alphabet of size q, say $\mathcal{C}_i = (n, M_i)_q$, and we take an *outer* code $\mathcal{C}_o = (N, M_o)_Q$. The size of the alphabet of the outer code $\mathcal{Q}_o$ is taken to be equal to the cardinality of the inner code, that is $|\mathcal{Q}_o| = Q = M_i$. Thus, we can define a bijection between the outer code alphabet and the code words of the inner code, $\phi : \mathcal{Q}_o \to \mathcal{C}_i$. Applying this bijection to all code words of the outer code, we obtain an $(nN, M_o)_q$ code over the small alphabet of size q, which we denote by $\mathcal{C}_i \circ \mathcal{C}_o$. If the rates of the inner and outer code are R_i and R_o respectively, then the rate of the concatenated code $\mathcal{C}_i \circ \mathcal{C}_o$ is $R_i R_o$.

Given $U \subseteq \mathcal{C}$, we define the ith projection set of U as

$$U(i) = \{u_j(i) \in \mathcal{Q} : u_j \in U\},\ 1 \leq i \leq n. \tag{2}$$

Also, we define the *descendant set* of U as

$$\text{desc}(U) = U(1) \times \cdots \times U(n) = \{z \in \mathcal{Q}^n : z(i) \in U(i),\ 1 \leq i \leq n\}. \tag{3}$$

Definition 1 Let $t \geq 2$ be an integer. We say that an $(n, M)_q$ code $\mathcal{C}$ is:

1. a t-frameproof code, t-FP, if for every $U \subset \mathcal{C}$ with $|U| \leq t$, we have $\text{desc}(U) \cap \mathcal{C} = U$.
2. a $\bar{t}$-separable code, $\bar{t}$-SC, if for all distinct $U, V \in \mathcal{C}$ with $|U| \leq t$ and $|V| \leq t$, we have $\text{desc}(U) \neq \text{desc}(V)$.
3. a B_2 code if all sums $u_i + u_j$, $1 \leq i \leq j \leq M$, are different, where the operation $+$ takes place in the field of real numbers.

In a t-FP code, given a subset U of code words of size at most t, there are no other code words in $\text{desc}(U)$ than the code words in U. This is the weakest form of tracing. The reason to study the families of t-FP, $\bar{t}$-SC, and B_2 codes together is because they are closely related as the following lemma shows.

Lemma 2 [2, 5] *Let n, M, q, t be positive integers greater than or equal to 2. We have the following relationships:*

1. *A t-FP code is a $\bar{t}$-SC.*
2. *A $\bar{t}$-SC is a $(t-1)$-FP code.*
3. *In the binary case, a $\bar{2}$-SC$(n, M)_2$ is a $B_2(n, M)_2$ code and vice versa.*

We will denote the largest cardinality of t-FP, $\bar{t}$-SC, and B_2 codes as $F(t, n, q)$, $S(\bar{t}, n, q)$ and $B(n, q)$, respectively.

Let $R_q(n, t)$ be the optimal rate of an $(n, M)_q$ code. As is customary in information theory, we are interested in the asymptotic rate

$$\underline{R}_q(t) = \limsup_{n \to \infty} R_q(n, t). \tag{4}$$

We will use the following notation for the asymptotic rate of t-FP, $\bar{t}$-SC, and B_2 codes:

$$f(t, q) = \limsup_{n \to \infty} \frac{\log_q F(t, n, q)}{n}, \tag{5}$$

$$s(\bar{t}, q) = \limsup_{n \to \infty} \frac{\log_q S(\bar{t}, n, q)}{n}, \tag{6}$$

$$b(q) = \limsup_{n \to \infty} \frac{\log_q B(n, q)}{n}. \tag{7}$$

2.1 Algebraic geometric codes

Algebraic geometric codes (AG) were developed by V.D. Goppa [14], and built from the theory of algebraic curves. Let $\mathcal{C}$ be a code from a curve of genus g, over the field $\mathbb{F}_q$, with N points. If d denotes the minimum distance of the code, it has been shown that the rate of the code satisfies:

$$R \geq 1 - \frac{d}{N} - \frac{g}{N}. \tag{8}$$

Since we are aiming for codes with the highest possible rate, we need g/N to be small. In that regard, Drinfeld and Vlădut [15] gave the following lower bound for g/N

$$\liminf_{g\to\infty} \frac{g}{N} \geq \frac{1}{\sqrt{q}-1}. \tag{9}$$

Several research efforts [16, 17] show that there are explicitly described sequences of curves that achieve the Drinfeld-Vlădut bound. Also, by the results in [18], AG codes having asymptotic rate

$$R \geq 1 - \frac{d}{N} - \frac{1}{\sqrt{q}-1} \tag{10}$$

can be constructed with polynomial complexity.

2.2 Lovász Local Lemma

We start with a sample space Ω. We will define *independent* random variables, say $\mathcal{V} = \{V_1, \ldots, V_m\}$, taking values in Ω. In this context we will have a set of events $\{E_1, \ldots, E_b\}$ that will be considered "bad". We will assume that all events are defined based on $V_1, \ldots, V_m$. The *scope* $\mathrm{sc}(E)$ of an event E is the minimal subset of random variables in $\mathcal{V}$ that determines its occurrence. In the sequel, these events are going to be ordered. The most used version of the LLL is:

Lemma 3 (Symmetric Lovász Local Lemma) *Let $E_1, \ldots, E_b$ be a set of (typically bad) events such that for each E_j:*

- $Pr[E_j] \leq p \in (0, 1)$.
- *E_j is mutually independent of a set of all but at most s of the other events.*

If

$$ep(s+1) \leq 1, \tag{11}$$

then

$$\Pr\left[\bigcap_{i=1}^{b} \overline{E}_i\right] > 0.$$

That is, all bad events can be avoided.

Intuitively, a configuration exists if the probability of each bad event is not too large, and if the mutual independence is relatively small.

Observe that, in Lemma 3, a given event E has to be *mutually independent* of *all* but at most s events. Let us display some machinery, developed in [19], that will allow us to establish claims on mutual independence throughout the paper. We start by defining *mutual independence*.

Definition 4 [19] An event E is *mutually independent* of a set of events $\mathcal{E}$ if for every $E_1, \ldots, E_r \in \mathcal{E}$, $\Pr[E \mid E_1 \cap \cdots \cap E_r] = \Pr[E]$.

As Molloy and Reed state in [19], for mutual independence, "*looks can often be deceiving*", therefore in their own words: "*we appeal to the following fact nearly every time we wish to establish mutual independence*".

The Mutual Independence Principle [19]. Let $\mathcal{V}$ be a set of independent random variables. Suppose that $E_1, \ldots, E_r \in \mathcal{E}$ is a set of events, where each E_i is determined by $F_i \subset \mathcal{V}$. If $F_i \cap (F_{i_1} \cup \cdots \cup F_{i_k}) = \emptyset$, then E_i is mutually independent of $\{E_{i_1}, \ldots, E_{i_k}\}$.

With the Mutual Independence Principle at hand, we can define a *dependency graph*. The vertices are the events $E_1, \ldots, E_b$, and there is an edge between E_i and E_j if they share at least a random variable. The neighborhood of a vertex E_j will be denoted by Γ_j. We will consider that no vertex belongs to its neighborhood. We denote by $s \geq 1$ the maximum degree of the graph. Therefore, $|\Gamma_j| \leq s$ for $j = 1, \ldots, b$. Now, according to the Mutual Independence Principle, an event E_i is considered to be mutually independent with all events not in Γ_j.

Intuitively, the symmetric version of the Lovász Local Lemma is appropriate when all bad events are "alike" in terms of error probability and dependence. If this is not the case, then one must resort to the general version, where each event is treated individually. However, if the bad events can be grouped into sets of events that are "alike", and the number of sets is relatively small, the following version can be very useful.

Lemma 5 *Let $\mathcal{E} = \{E_1, \ldots, E_b\}$ be a set of (typically bad) events such that each E_j is mutually independent of $\mathcal{E} \setminus (\mathcal{S}_j \cup E_j)$, for some $\mathcal{S}_j \subset \mathcal{E}$. If for all $i = 1, \ldots, b$, the following conditions are satisfied:*

1. $\Pr[E_i] \leq 1/4$,
2. $\sum_{E_i \in \mathcal{S}_j} \Pr[E_i] \leq \frac{1}{4}$,

then, with positive probability, none of the events in $\mathcal{E}$ occur.

2.3 The variable framework

Since the appearance of the Lovász Local Lemma, a lot of work has been done in order to develop an algorithm to explicitly obtain the combinatorial objects whose existence was established by the lemma. The algorithmic version, that became a reality with the work in [11, 12], is usually called the variable framework. In this section, we discuss the variable framework approach of [10]. The main idea in [10] is to show that the probability that the variable framework algorithm performs more than N iterations is inverse exponential in N. We make a succinct presentation, and only highlight the aspects in [10] we will need in Section 5, where we extend the work in [10], and obtain an explicit upper bound on the number of "iterations" that the algorithm performs.

As said before, we can represent a code of size M and length n as an $M \times n$ matrix. We start by outlining a general algorithm that constructs a matrix that represents a code whose code words avoid a set of bad events (see Algorithm 1).

Observe that, if and when Algorithm 1 terminates, it produces a code in which none of the bad events occur. This is true because by the condition in the loop of line 2 of BODY, if and when the Algorithm terminates, no bad event in $\mathcal{E}$ occurs. Note also than by line 2 of the RESAMPLE(E_i) procedure, whenever a RESAMPLE call returns, there is no bad event occurring that shares random variables with E_i.

Algorithm 1

INPUT:

Parameters: Integers M (code size), n (code length).
Code alphabet: $\mathcal{Q} = \{\alpha_1, \ldots, \alpha_q\}$.
$M \times n$ matrix of independent random variables $\mathcal{C} = \{X_{1,1}, \ldots, X_{M,n}\}$, taking values in $\mathcal{Q}$, with a certain probability distribution.
Events: Ordered set $\mathcal{E} = \{E_1, \ldots, E_b\}$ of events.

OUTPUT:

Assignment A of values to $X_{i,j}$, such that no bad event in $\mathcal{E}$ occurs.

BODY:

1: Sample variables $X_{i,j}$, $1 \leq i \leq M$, $1 \leq j \leq n$. Let A be the resulting assignment of values.
2: **while** there is a bad event in $\mathcal{E}$ occurring **do**
3: RESAMPLE(E_i), where E_i is the least indexed bad event
4: **end while**
5: Output current assignment A.

RESAMPLE(E_i)

1: Resample the random variables $X_{r,s}$ associated with E_i.
2: **while** there is a *least indexed* bad event E_j, such that $\mathrm{sc}(E_i) \cap \mathrm{sc}(E_j) \neq \emptyset$, occurring under the current assignment **do**
3: RESAMPLE(E_j)
4: **end while**

Let us show how it is proven in [10] that this algorithm terminates fast with positive probability. A RESAMPLE call made in line 2 of the BODY is a *root* call, and one made by line 2 of the RESAMPLE(E_i) routine is a *recursive* call. The computational complexity discussion we will make is based on the number of RESAMPLE calls. Specifically, the authors of [10] prove the following result (we have adapted notation to our context):

Theorem 6 [10] *Let $X_{i,j}$, $1 \leq i \leq M$, $1 \leq j \leq n$ be distinct random variables (arranged as an $M \times n$ matrix), taking values in an alphabet $\mathcal{Q}$. Let $\mathcal{E} = \{E_1, \ldots, E_b\}$ be a set of bad events, where each event is associated to a subset of the random variables $X_{i,j}$. Let $p \geq \Pr[E_i]$, $\forall 1 \leq i \leq b$, and let s be the maximum number of events whose scopes intersect the scope of a given event. Suppose bad events satisfy $ep(s+1) < 1$. Then, the probability that Algorithm 1 executes for at least N rounds is* inverse exponential in *N and, upon termination, the algorithm outputs an $M \times n$ matrix in which no bad event $\mathcal{E}$ occurs.*

Proof (*Sketch*) First, note that Algorithm 1 makes progress after each RESAMPLE call. More precisely, take an arbitrary call, say RESAMPLE(E_i).

(i) If and when RESAMPLE(E_i) terminates, then E_i no longer occurs.
(ii) Let us suppose E_j is not a bad event at the start of RESAMPLE(E_i). If and when RESAMPLE(E_i) ends, then E_j is still not bad.

In other words, after a RESAMPLE call returns, all progress made up to that point is maintained, and there is at least one more bad event that is "fixed".

To continue, let us define a graph in order to represent executions of Algorithm 1. More precisely, the graph is a *labeled rooted forest* whose components are rooted trees with labeled vertices. In our scenario the labels of the vertices of the trees are from the set of events $\mathcal{E}$. A *witness* forest of an execution of Algorithm 1, making at least N RESAMPLE calls, is a representation of the execution detailed in the following way:

1. A node labeled as E_i depicts a RESAMPLE(E_i) call.

2. The labels of the roots correspond to root RESAMPLE calls (line 3 of BODY).
3. A recursive RESAMPLE(E_j) call done in line 3 of RESAMPLE(E_i) is associated to a child labeled as E_j of the node labeled as E_i.

Let $\mathcal{T}$ denote a witness forest having N nodes. Next lemma will be key to prove our result. □

Lemma 7 *Let p denote an upper bound on the probability of bad events in the sense of Lemma 3. Let P_N denote the probability that Algorithm 1 executes at least N RESAMPLE calls. Then*

$$\mathrm{P}_N \leq \sum_{\mathcal{T}:|\mathcal{T}|=N} p^N. \tag{12}$$

The sum is over all witness forests with N nodes.

See [20]. □

Since the sum in (12) is over all witness forests, it follows that to compute P_N we have to count witness forests.

We call a forest *feasible* if:

(i) the labels of the roots are pairwise distinct,
(ii) the labels of the children of a node are pairwise distinct,
(iii) if a vertex labeled by E_j is a child of a vertex labeled by E_i, then $\mathrm{sc}(E_i) \cap \mathrm{sc}(E_j) \neq \emptyset$.

It can be seen that the class of feasible forests includes witness forests. So it will be enough for our purposes to deal with the former. As a matter of fact, as shown in [20], it turns out we can deal with *full unlabeled ordered rooted planar forests*. Informally, to convert a feasible forest into a full unlabeled ordered rooted planar forest, one has to:

- add root nodes labeled conveniently so that the set of labels of roots is the set of bad events,
- add leaves to every node so that the set of labels of each child is the set of bad events whose scope intersect the scope of the event of its label. Thus, each node will have at most $s + 1$ children, since the scope of an event intersects itself,
- then remove all the labels.

Consequently, it suffices to count the number of full $(s + 1)$-ary rooted planar trees with a given number of nodes, say v. This number, that we denote by T_v, has already been computed (see for instance [21, Theorem 5.13]):

$$T_v = \frac{1}{sv+1}\binom{(s+1)v}{v}, \tag{13}$$

and we have the upper bound

$$T_v < A\left(\left(1+\frac{1}{s}\right)^s (s+1)\right)^v, \tag{14}$$

where A is a constant depending only on s.

The number F_N of rooted planar forests with N internal nodes that are composed of $|\mathcal{E}|$ $(s + 1)$-ary rooted planar trees is:

$$F_N = \sum_{\substack{N_1+\cdots+N_{|\mathcal{E}|}=N \\ N_1,\ldots,N_{|\mathcal{E}|}\geq 0}} T_{N_1}\cdots T_{N_{|\mathcal{E}|}}. \tag{15}$$

Now, from (15) and (14) we get:

$$F_N < (AN)^{|\mathcal{E}|}\left(\left(1+\frac{1}{s}\right)^s(s+1)\right)^N < (AN)^{|\mathcal{E}|}(e(s+1))^N. \tag{16}$$

And therefore, (12) together with (16) yield

$$P_N < (AN)^{|\mathcal{E}|}(ep(s+1))^N. \tag{17}$$

Now Theorem 6 follows since $ep(s+1) < 1$ by assumption.

Remark 8 Note that according to the Mutual Independence Principle, in Algorithm 1, RESAMPLE(E_i), line 2, the statement $\mathrm{sc}(E_i) \cap \mathrm{sc}(E_j) \neq \emptyset$ means that the events E_j we are resampling are *not* considered to be mutually independent with E_i.

3 Sketch of proof

Since we are going to provide constructions for t-frameproof codes, $\bar{2}$-separable codes and B_2 codes, using the variable framework of the LLL, the outline of the proofs will be similar. In this section we provide an outline of our strategy in order to clarify our discussion.

Our proof reasoning will observe the following guidelines:

- **Existence**
  - *Definition of random variables and events*. Typically we will arrange a set of random variables as a matrix. A given assignment to these random variables will represent a code of length equal to the number of columns of the matrix and size equal to the number of rows of the same matrix. The events will be defined in terms of subsets of rows of the matrix.
  - *Computing the probability of an event being bad.* This probability will depend only on the number of columns of the matrix (code length).
  - *Computing the number of events not mutually independent with a given event*. This number will depend only on the number of rows of the matrix (code size).
  - *Apply the LLL*. Since the LLL relates probability with mutual independence, we will obtain a relationship between number of columns and number of rows, so that there exists a matrix that avoids all bad events.
- **Construction**
  - *Particularize the variable framework algorithm to the code to be constructed*. We will clearly establish the input parameters of the algorithm.
  - *Impose further restrictions of the LLL condition*. This will allow us to find closed expressions for an upper bound of the computational complexity. Nevertheless this complexity will be exponential in the code length.
  - *Describe a concatenated construction*. Establish outer code parameters, so the overall construction is polynomial in the code length.

4 Lower bounds

The key issue in using the variable framework of the LLL is deciding how to define the random variables and events that are going to be used in order to represent the object one

wishes to obtain. The choice has direct impact in the sampling procedure, which in turn affects the computational complexity.

4.1 Frameproof codes

Since we wish to obtain a code of size M and length n, we take Mn independent random variables $\mathcal{X} = \{X_1, \ldots, X_{Mn}\}$, and arrange them as an $M \times n$ matrix C. If assignments to the rows of the matrix represent code words, then entry X_{ij} is associated with position j of code word i. Let us take a row c of C, and a set of t rows U^t of C, such that $c \notin U^t$. Abusing notation, we also denote the assignment to these random variables as c and U^t respectively. To define bad events, we will use Definition 1. So, for frameproof codes, a *bad* event is an assignment for which $c \in \text{desc}(U^t)$. We denote such an event as $E(c, U^t)$.

Observe that, according to the Mutual Independence Principle, an event $E(c_i, U^t)$ is mutually independent with all events whose scope does not intersect $\text{sc}(E(c_j, V^t))$. In this case, this means that $E(c_i, U^t)$ is mutually independent with all events *not* sharing all the variables of at least a row of C. Thus, given $E(c_i, U^t)$, we need to compute the number s of events, different from $E(c_i, U^t)$, whose scopes intersect with $\text{sc}(E(c_j, V^t))$. In Section 2.2, s was defined as the maximum degree of the dependency graph. Therefore, the number s corresponding to a given event $E(c, U^t)$ is the total number of events, minus the number of events that do not contain neither c nor U^t, minus 1 (corresponding to the event $E(c, U^t)$).

$$s = \binom{M}{1}\binom{M-1}{t} - \binom{M-(t+1)}{1}\binom{M-(t+2)}{t} - 1. \tag{18}$$

According to Pascal's rule,

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}, \qquad \binom{n-1}{k} = \binom{n-2}{k-1} + \binom{n-2}{k}. \tag{19}$$

Therefore,

$$\binom{n}{k} - \binom{n-2}{k} = \binom{n-1}{k-1} + \binom{n-2}{k-1}. \tag{20}$$

By repeated application, we have

$$\begin{aligned}
\binom{n}{k} - \binom{n-3}{k} &= \binom{n-1}{k-1} + \binom{n-2}{k-1} + \binom{n-3}{k-1} &&\vdots \\
\binom{n}{k} - \binom{n-s}{k} &= \binom{n-1}{k-1} + \binom{n-2}{k-1} + \cdots + \binom{n-s}{k-1} &&= \sum_{j=1}^{s} \binom{n-j}{k-1}
\end{aligned}$$

Substituting n by $M-1$, k for t, and s by $t+1$, we obtain

$$\binom{M-1}{t} - \binom{M-1-(t+1)}{t} = \sum_{j=1}^{t+1} \binom{M-1-j}{t-1}. \tag{21}$$

Therefore, from (18) we have

$$\begin{aligned} s+1 &= M\binom{M-1}{t} - (M-(t+1))\binom{M-(t+2)}{t} \\ &= M\binom{M-1}{t} - M\binom{M-(t+2)}{t} + (t+1)\binom{M-(t+2)}{t} \\ &= M\left[\binom{M-1}{t} - \binom{M-(t+2)}{t}\right] + (t+1)\binom{M-(t+2)}{t} \\ &= M\left[\sum_{j=1}^{t+1}\binom{M-1-j}{t-1}\right] + (t+1)\binom{M-(t+2)}{t} \\ &< M(t+1)\binom{M-2}{t-1} + (t+1)\binom{M-(t+2)}{t} \\ &< M(t+1)\frac{M^{t-1}}{(t-1)!} + (t+1)\frac{M^t}{t!} = \frac{(t+1)^2}{t!}M^t \leq \frac{9}{2}M^t. \end{aligned}$$

In the previous calculation we have used the fact that $\binom{n}{k} \leq \frac{n^k}{k!}$.
Therefore, the maximum degree of the dependency graph, s, satisfies

$$s+1 < \frac{9}{2}M^t, \quad \forall t \geq 2. \tag{22}$$

A first approach to the problem is to consider a uniform distribution for the random variables. When doing so, we obtain the following theorem. This result will later be improved by using a refined distribution.

Theorem 9 *Let $\mathcal{Q} = \{0, \cdots, q-1\}$ be an alphabet of size $q \geq 2$. If $t \geq q$, then there exists a t-frameproof code of length n and size:*

$$F(t,n,q) \geq \left\lfloor \frac{1}{(9e/2)^{\frac{1}{t}}(1-(1-\frac{1}{q})^t)^{\frac{n}{t}}} \right\rfloor. \tag{23}$$

Proof Using the above notation, we take the distribution of the random variables X_{ij}, $1 \leq i \leq M$, $1 \leq j \leq n$, to be

$$\Pr(X_{ij}=0) = \cdots = \Pr(X_{ij}=q-1) = \frac{1}{q}. \tag{24}$$

Then, the probability of a bad event $E(c_i, U^t)$ is

$$\Pr[E(c_i,U^t)] = \left(1-\left(1-\frac{1}{q}\right)^t\right)^n \tag{25}$$

Indeed, $\Pr[c_i(l)=\alpha] = \frac{1}{q}$ with $\alpha \in \mathcal{Q}$, and the result follows from the independence of the random variables X_{ij}.

Having computed both the probability of a bad event, and an upper bound on the value of s (22), according to the Lovász Local Lemma, all bad events can be avoided if

$$e\left(1-\left(1-\frac{1}{q}\right)^t\right)^n 9M^t/2 \leq 1 \tag{26}$$

Therefore,

$$M \leq \frac{1}{(9e/2)^{\frac{1}{t}}(1-(1-\frac{1}{q})^t)^{\frac{n}{t}}}, \tag{27}$$

and the theorem follows.

The versatility of the Lovász Local Lemma allows to accommodate for alternative distributions of the random variables. To improve the result in the previous theorem, we are going to follow the approach in [9], where the authors use an interesting non-uniform distribution. Note that, in a related context, a similar approach was used in [22]. The proof of the following theorem follows along the lines of the previous one, and it is given in the Appendix.

Theorem 10 *Let $\mathcal{Q} = \{0, \cdots, q-1\}$ be an alphabet of size $q \geq 2$. If $t+1 \geq q$, then there exists a t-frameproof code of length n and size:*

$$F(t,n,q) \geq \left\lfloor \frac{1}{(9e/2)^{\frac{1}{t}}} \frac{1}{\left(1-(1-\frac{q-1}{t+1})(\frac{q-1}{t+1})^t - \frac{q-1}{t+1}(\frac{t}{t+1})^t\right)^{\frac{n}{t}}} \right\rfloor. \tag{28}$$

Remark 11 According to [9], Theorem 10 is an improvement on Theorem 9 when $q \leq \frac{t}{2}+1$, and $t \geq 8$.

4.2 Separable codes

For the family of separable codes, and in view of Lemma 2, we focus on the interesting and already non-trivial case $t = 2$. This is also the main focus in [8]. To establish a lower bound for $\bar{2}$-separable codes, we again take Mn random variables $X_1, \ldots, X_{Mn}$, and arrange them as an $M \times n$ matrix C, with rows $\{c_1, \ldots, c_M\}$, $c_i = \{X_{i1}, \ldots, X_{in}\}$. As before, this is our representation of a code of size M and length n, where code words correspond to assignments to the random variables associated with a row, that is position j of code word c_i corresponds to matrix entry X_{ij}. We obtain the following result, whose proof is given in the Appendix.

Theorem 12 *Let $\mathcal{Q} = \{0, \cdots, q-1\}$ be an alphabet of size $q \geq 2$. If $n \geq 2$, then there exist $\bar{2}$-separable codes of size*

$$S(\bar{2},n,q) \geq \left\lfloor \frac{1}{(16)^{\frac{1}{3}}} \left(\frac{q^3}{2q-1}\right)^{n/3} \right\rfloor. \tag{29}$$

The following corollary states the asymptotic rate.

Corollary 13 *be Let $\mathcal{Q} = \{0, \cdots, q-1\}$ an alphabet of size $q \geq 2$. If $n \geq 2$, then there exist $\bar{2}$-separable codes of rate*

$$s(\bar{2},q) \geq 1 - \frac{\log_q(2q-1)}{3} \tag{30}$$

Proof Just apply the definition of $s(\bar{2},q)$ in (6).

Note that this is the same rate obtained in Corollary 4 of [8].

4.3 B2 codes

For B_2 codes we have the following result, whose proof will be given in the Appendix.

Theorem 14 *Let $\mathcal{Q} = \{0, \cdots, q-1\}$ be an alphabet of size $q \geq 2$. If $n \geq 2$, then there exist B_2 codes of size*

$$B(n,q) \geq \left\lfloor \frac{q^{n/3}}{2} \right\rfloor. \tag{31}$$

Our result and the one in Theorem 8 of [8] are asymptotically identical, as the following corollary shows.

Corollary 15 *Let $\mathcal{Q} = \{0, \cdots, q-1\}$ be an alphabet of size $q \geq 2$. If $n \geq 2$, then there exist B_2 codes of rate*

$$b(q) \geq \frac{1}{3}. \tag{32}$$

Proof The proof is immediate using (31) in (7).

5 Combinatorial constructions using the variable framework

In this section we provide probabilistic combinatorial constructions, for t-frameproof codes, $\bar{2}$-separable codes and B_2 codes. The constructions are obtained as the output of an algorithm. First, we study the complexity of Algorithm 1 (see Section 2.3).

5.1 Expected number of iterations

We extend the work in [10], and deal with the expected number of RESAMPLE calls made in line 3 of the BODY, and line 3 of the RESAMPLE routine. Let us first give the explicit expression of A in (14). The work in [23] proves that

$$\frac{\sqrt{2\pi}e^{-n}n^{n+1}}{\sqrt{n}} < n! < \frac{\sqrt{2\pi}e^{-n}n^{n+1}}{\sqrt{n-1}},$$

and therefore,

$$\binom{(s+1)v}{v} < \frac{\frac{\sqrt{2\pi}e^{-(s+1)v}((s+1)v)^{(s+1)v+1}}{\sqrt{(s+1)v-1}}}{\frac{\sqrt{2\pi}e^{-sv}(sv)^{sv+1}}{\sqrt{sv}}\frac{\sqrt{2\pi}e^{-v}v^{v+1}}{\sqrt{v}}} \tag{33}$$

$$= \frac{\sqrt{s}(s+1)^v}{\sqrt{2\pi}\sqrt{(s+1)v-1}}\frac{(s+1)^{sv+1}}{s^{sv+1}} \tag{34}$$

$$= f(s,v)\left(\left(1+\frac{1}{s}\right)^s(s+1)\right)^v, \tag{35}$$

$$\text{where } f(s,v) = \frac{s+1}{\sqrt{2\pi}\sqrt{s}\sqrt{(s+1)v-1}}. \tag{36}$$

Now, according to (13),

$$T_v = \frac{1}{sv+1}\binom{(s+1)v}{v} < \frac{f(s,v)}{sv+1}\, e^v (s+1)^v < \frac{s+1}{\sqrt{2\pi} s^2 v} e^v (s+1)^v < \frac{e^v (s+1)^v}{\sqrt{2\pi}(s-1)v}. \tag{37}$$

Observe that constant A in (14) corresponds to taking $v = 1$ in $f(s, v)$, that makes A dependent only on s, and in particular $A < 1$.

Take a given event, say E_i, and let $\mathbb{E}[E_i]$ be the expected number of times E_i is resampled. According to the reasoning in the proof of Theorem 6, we have that a fully $(s+1)$-ary rooted planar tree with v internal nodes serves us to analyze a RESAMPLE call and its recursion.

Let $\mathcal{T}_i$ be the set of witness trees rooted in E_i, and let $\mathcal{T}_i^v$ be the set of witness trees rooted in E_i with v nodes. Finally, let P_T be the probability that a tree T is a witness tree in an execution of Algorithm 1. Then, by the reasoning done in Section 2.3,

$$\mathbb{E}[E_i] \leq \sum_{T \in \mathcal{T}_i} P_T = \sum_{v=1}^{\infty} \Big(\sum_{T \in \mathcal{T}_i^v} P_T \Big). \tag{38}$$

Using Lemma 7, the expectation in (38) can be bounded as follows:

$$\begin{aligned}
\sum_{v=1}^{\infty} \Big(\sum_{T \in \mathcal{T}_i^v} P_T \Big) &< \sum_{v=1}^{\infty} \Big(T_v p^v \Big) \\
&< \sum_{v=1}^{\infty} \left(\frac{1}{sv+1} \binom{(s+1)v}{v} p^v \right) \\
&< \frac{1}{\sqrt{2\pi}(s-1)} \sum_{v=1}^{\infty} \frac{1}{v} \Big(ep(s+1) \Big)^v \\
&= \frac{1}{\sqrt{2\pi}(s-1)} (-\ln(1 - ep(s+1))).
\end{aligned} \tag{39}$$

We have used the well known Taylor expansion $\ln(1-x) = -\sum_{v=1}^{\infty} \frac{x^v}{v}$, $\forall |x| < 1$, and the fact that since a witness tree is a feasible tree, then we can sum over the number of feasible trees. Finally, adding for the total number of possible bad events, we have the following proposition:

Proposition 16 *Let $E_1, \ldots, E_m$ be bad events that are to be avoided. The expected number of* RESAMPLE *calls (both in the main body and recursive routine) in an execution of Algorithm 1 is at most:*

$$\sum_{j=1}^{m} \mathbb{E}[E_j] < \frac{1}{\sqrt{2\pi}} \frac{m}{(s-1)} (-\ln(1 - ep(s+1))), \quad \forall s \geq 2. \tag{40}$$

Proof The proposition follows from (38) and (39).

Corollary 17 *Let $E_1, \ldots, E_m$ the bad events that are to be avoided. If $ep(s+1) \leq 2/3$ and $s \geq 2$, then the expected number of* RESAMPLE *calls (both in the main body and recursive routine) in an execution of Algorithm 1 is at most $\frac{m}{s}$.*

Proof Since $-\ln(1-x) \leq x + x^2$, $\forall x \in [0, 2/3]$, in view of (40), an upper bound on the expected number of steps is

$$\frac{10m}{(s-1)9\sqrt{2\pi}} < \frac{m}{s}, \quad \forall s \geq 2. \tag{41}$$

Remark 18 In their paper, Moser and Tardos [12] also discussed the expectation of the number of times an event is going to be resampled. Adapted to Lemma 5, their result states that

$$\sum_{\text{events } E_i} \mathbb{E}[E_i] \leq \sum_{\text{events } E_i} \frac{2\Pr[E_i]}{1-2\Pr[E_i]}. \tag{42}$$

We will have occasion to use (42) when we discuss the complexity of constructions for separable and B_2 codes.

5.2 Frameproof codes

We begin by giving constructions of t-frameproof codes over an alphabet of size q. We will deal with the interesting case $q = 2$, but to understand the impact of the alphabet size, we start by discussing the case of a larger alphabet $q = t$. Observe that $q = t$ minimizes the denominator of (23).

5.2.1 Plain algorithmic construction

For $q = t$ and in order to obtain compact expressions, it is more convenient to use Theorem 9. We will use the improvement given in Theorem 10 for $q = 2$. Let us impose a stronger restriction to (26),

$$e\left(1-\left(1-\frac{1}{t}\right)^t\right)^n \frac{9}{2}M^t \leq \frac{2}{3}, \tag{43}$$

which leads to

$$n \geq \frac{\ln(27eM^t/4)}{-\ln(1-(1-\frac{1}{t})^t)}. \tag{44}$$

Since $x < -\ln(1-x)$, $\forall x \in (0,1)$, and $(1-1/t)^t \geq 1/4$, $\forall t \geq 2$, then

$$\frac{\ln(27eM^t/4)}{-\ln(1-(1-\frac{1}{t})^t)} < 4\ln(27eM^t/4). \tag{45}$$

Finally, we observe that $4\ln(27eM^t/4) < 6t\ln M - 1$, $\forall t \geq 3$, $\forall M \geq 8$. If $t = 2$, we can directly check that equation (44) is satisfied for $n \geq 6t\ln M - 1$. Then,

$$n \geq \lfloor 6t \ln t \log_t M \rfloor, \ \forall t \geq 2. \tag{46}$$

Proposition 19 *Let $t \geq 2$, $M \geq 8$, and $n = \lfloor 6t \ln t \log_t M \rfloor$. Using Algorithm 1, t-frameproof $(n, M)_t$ codes can be constructed, with an expected number of* RESAMPLE *calls less than $\frac{M}{t}$.*

Proof The codes can be obtained using Algorithm 1 with the following input:

- Integers $t \geq 2$, $M \geq 8$ and $n = \lfloor 6t \ln t \log_t M \rfloor$.
  Alphabet $\mathcal{Q}$ of size t, $\mathcal{Q} := \{0, \ldots, t-1\}$.
- Random variables: $\{X_{ij} : \mathcal{Q} \to \mathcal{Q},\ 1 \leq i \leq M,\ 1 \leq j \leq n\}$.
- Probability mass function: $\Pr(X_{ij} = 0) = \cdots = \Pr(X_{ij} = t-1) = \frac{1}{t}$.
- *Bad* events:
  - Arrange the r.v. X_{ij} as an $M \times n$ matrix C, with $(C)_{ij} = X_{ij}$.
    * Let C_{row} the set of rows of C.
    * Define the set $\{(r, T) \mid T \subset C_{row},\ |T| = t,\ r \in C_{row} \setminus T\}$.

* Order the previous set and denote it by $\mathcal{R}$.
* Let $E(r, T)$ be the bad event $r \in \mathrm{desc}(T)$, in the sense of Section 4.1.

– Define the *ordered* set $\mathcal{E} := \{E(r, T) \mid (r, T) \in \mathcal{R}\}$.

The existence of the code is guaranteed by the reasoning leading to (46). It only remains to prove the statement about the expected number of RESAMPLE calls. Since we have imposed (43), then according to Corollary 17 we have to find an upper bound on $\frac{m}{s}$.
Since the total number of events is $m = \binom{M}{1}\binom{M-1}{t}$, and $s+1 = \binom{M}{1}\binom{M-1}{t} - \binom{M-(t+1)}{1}\binom{M-(t+2)}{t}$, we have

$$\frac{m}{s+1} < \frac{M\binom{M-1}{t}}{M\binom{M-1}{t} - M\binom{M-(t+2)}{t}} = \frac{1}{1 - \frac{\binom{M-(t+2)}{t}}{\binom{M-1}{t}}} < \frac{M-t}{t+1}. \tag{47}$$

This is because

$$\frac{\binom{M-(t+2)}{t}}{\binom{M-1}{t}} = \frac{(M-(t+2))(M-(t+3))\cdots(M-(2t+1))}{(M-1)(M-2)\cdots(M-t)} \tag{48}$$

$$= \prod_{k=1}^{t} \frac{M-(t+1+k)}{M-k} = \prod_{k=1}^{t}\left(1 - \frac{t+1}{M-k}\right) \tag{49}$$

$$< 1 - \frac{t+1}{M-t}. \tag{50}$$

Now,

$$(M-t)\frac{s+1}{s} \le M\frac{t+1}{t} \implies \frac{m}{s} = \frac{m}{s+1}\frac{s+1}{s} \le \frac{M}{t},$$

and in view of (18), the inequality on the left holds because $s \ge t$.

Analogously, for binary codes we have the following result:

Proposition 20 *Let $t \ge 2$, $M \ge 8$, and $n = \lfloor 3t(t+1)\log_2 M \rfloor$. Using Algorithm 1, binary t-frameproof $(n, M)_2$ codes can be constructed, with an expected number of* RESAMPLE *calls less than $\frac{M}{t}$.*

Proof For $q = 2$, we make again a stronger restriction, in this case to (63), and impose

$$e\left[1 - (1 - \frac{q-1}{t+1})(\frac{q-1}{t+1})^t - \frac{q-1}{t+1}(\frac{t}{t+1})^t\right]^n \frac{9}{2}M^t \le \frac{2}{3}. \tag{51}$$

Therefore, the code length n has to satisfy

$$n \ge \frac{\ln(27eM^t/4)}{-\ln\left(1 - \frac{t}{t+1}\left(\frac{1}{t+1}\right)^t - \frac{1}{t+1}\left(\frac{1}{1+\frac{1}{t}}\right)^t\right)}. \tag{52}$$

Since

$$-\ln\left(1 - \frac{t}{t+1}\left(\frac{1}{t+1}\right)^t - \frac{1}{t+1}\left(\frac{1}{1+\frac{1}{t}}\right)^t\right) > -\ln\left(1 - \frac{1}{e(t+1)}\right) > \frac{1}{e(t+1)},$$

then

$$\frac{\ln(27eM^t/4)}{-\ln\left(1-\frac{t}{t+1}\left(\frac{1}{t+1}\right)^t-\frac{1}{t+1}\left(\frac{1}{1+\frac{1}{t}}\right)^t\right)} < e(t+1)\ln(27eM^t/4).$$

Finally, since $e(t+1)\ln(27eM^t/4) < 3t(t+1)\log_2 M - 1$, for all $t \geq 3$, $M \geq 8$, we can safely take $n \geq \lfloor 3t(t+1)\log_2 M \rfloor$. The case $t = 2$ can be checked directly from equation (52).
The codes can be can be constructed using Algorithm 1 with the following input:

- Integers $t \geq 2$, $M \geq 8$, and $n = \lfloor 3t(t+1)\log_2 M \rfloor$.
  Alphabet $\mathcal{Q}$ of size 2, $\mathcal{Q} := \{0, 1\}$.
- Random variables: $\mathcal{C} = \{X_{ij} : \mathcal{Q} \to \mathcal{Q},\ 1 \leq i \leq M, 1 \leq j \leq n\}$.
- Probability mass function: $\Pr(X_{ij} = 0) = \frac{t}{t+1}$, $\Pr(X_{ij} = 1) = \frac{1}{t+1}$.
- *Bad* events:
  - Arrange the r.v. X_{ij} as an $M \times n$ matrix C, with $(C)_{ij} = X_{ij}$.
    * Let C_{row} the set of rows of C.
    * Define the set $\{(r, T) \mid T \subset C_{row},\ |T| = t,\ r \in C_{row} \setminus T\}$.
    * Order the previous set and denote it by $\mathcal{R}$.
    * Let $E(r, T)$ be the bad event $r \in \mathrm{desc}(T)$, in the sense of Section 4.1.
  - Define the *ordered* set $\mathcal{E} := \{E(r, T) \mid (r, T) \in \mathcal{R}\}$.

The claim about the number of iterations is proved in the same manner as in Proposition 19, so we omit it.

Remark 21 We would like to point out that performing the same analysis leading to (46), using (26) instead of (43), would lead to a value of n of the same order of magnitude.

Remark 22 Observe that, according to Proposition 19 and Proposition 20, we can take $M \leq t^{n/(7t\ln t)}$ and $M \leq 2^{n/(3t(t+1))}$, for codes with alphabet size t and 2, respectively. This means that the number of RESAMPLE calls is exponential in the code length. Moreover, consider line 2 in BODY of Algorithm 1. For the algorithm to find the least indexed event, in the worst case, it is needed to go over all events in $\mathcal{E}$, and check if they are bad. There are approximately M^t events in $\mathcal{E}$. Moreover, in line 2 of RESAMPLE, the algorithm must check all events in the neighborhood of the event that has been resampled. In both cases, and given the bound we have proved, the number of events that need to be checked is exponentially large in n. We deal with this situation in the following section.

5.2.2 Polynomial complexity constructions

Let us overcome the drawback stated in Remark 22, and construct codes with complexity polynomial in the code length. To do so, we will resort to concatenated constructions.

The concept of frameproof codes goes as far as the work of Boneh and Shaw in [1]. In that paper, the following result is stated:

Lemma 23 [1] *Let $\mathcal{C}$ be an $(n, M, d)_Q$ code. If $d > n - \frac{n}{t}$, then $\mathcal{C}$ is a t-frameproof code.*

Now, using (10) we obtain the following result.

Lemma 24 *An AG t-frameproof $(n, M, d)_Q$ code can be constructed, for rates*

$$R + \epsilon = \frac{1}{t} - \frac{1}{\sqrt{Q} - 1}, \tag{53}$$

and polynomial complexity $O((n \log_Q n)^3)$.

Proof From Lemma 23 we have that $\frac{d}{n} > 1 - \frac{1}{t}$ is a sufficient condition for the frameproof property. With this in mind, the lemma is a consequence of (10). The result about the complexity is stated in [18].

Proposition 25 [1] *If $\mathcal{C}_i$ is a t-FP code of rate R_i, and $\mathcal{C}_o$ is a t-FP code of rate R_o, then $\mathcal{C}_i \circ \mathcal{C}_o$ is a t-FP code of rate $R_i R_o$.*

In view of the previous proposition, for the inner code we take the constructions for t-frameproof codes (alphabet sizes $q = 2$ and $q = t$) that we have presented in Propositions 19 and 20. In both cases, for the outer code we take an alphabet $\mathcal{Q}$ of size $|\mathcal{Q}| = Q \geq t^\beta$, with $\beta > 2$. Then, from (53) we have:

$$R_o + \epsilon \geq \frac{1}{t} - \frac{1}{t^{\beta/2} - 1}, \text{ i.e., } R_o = \frac{1}{t}\left(1 - o(1)\right). \tag{54}$$

Now, we are in position to state the following theorem:

Theorem 26 *Using the variable framework, with $t \geq 2$, we can construct t-frameproof codes, over an alphabet of size t, of rate*

$$R = \frac{1}{6t^2 \ln t}(1 - o(1)), \tag{55}$$

with polynomial complexity in the code length.

Proof For the outer code we take an AG $(n_o, M_o, d_o)_{Q_o}$ code as given by Lemma 24, over an alphabet $\mathcal{Q}_o$ of size $|\mathcal{Q}_o| = Q_o \geq t^\beta$, $\beta > 2$. Note that Q_o has to be a prime or a prime power, therefore we can choose $Q_o \leq 2\lceil t^\beta \rceil$, according to Bertrand's postulate.

Now, the concatenated construction imposes to take an inner code of size $M_i = Q_o$. According to Proposition 19, we can construct such a code using Algorithm 1, with an expected number of RESAMPLE calls less than $M_i/t = Q_o/t \leq 2\lceil t^\beta \rceil / t < 2t^\beta$. From Lemma 23, we have that $t < n_o$, and therefore, the expected number of RESAMPLE calls is less than $2n_o^\beta$, i.e. polynomial in the code length $n = n_o n_i$. Moreover, from Lemma 24, the complexity of constructing the outer code is also polynomial in the code length. Finally, the claim about the rate is straightforward from (54), taking an inner code of rate

$$R_i = \frac{1}{6t \ln t}, \tag{56}$$

(which is again possible by Proposition 19), and then applying Proposition 25.

For the binary case and $t \geq 3$, using Proposition 20, we have the following theorem, whose proof is analogous to the previous one.

Theorem 27 *Using the variable framework, for $t \geq 3$, we can construct t-frameproof binary codes of rate*

$$R = \frac{1}{3t^2(t+1)}(1 - o(1)), \tag{57}$$

with polynomial complexity in the code length.

5.3 Separable codes and B2 codes

For the case of $\overline{2}$-separable codes and B_2 codes we have the following proposition, whose proof is given in the Appendix.

Proposition 28 *Using the variable framework, we can construct* $\overline{2}$*-SC* $(\left\lfloor \frac{4\log_q M}{3-\log_q(2q-1)} \right\rfloor, M)_q$ *codes, with an expected number of* RESAMPLE *calls less than* $\frac{M}{9}$, $\forall M \geq 16$.

For the binary case we have:

Corollary 29 *By means of the variable framework,* $\overline{2}$*-SC* $(\left\lfloor \frac{4\log_2 M}{3-\log_2 3} \right\rfloor, M)_2$, *binary codes can be constructed. The expected number of* RESAMPLE *calls is less than* $M/9$, $\forall M \geq 16$.

Again, as in the case of frameproof codes, Remark 22 applies, and since M is exponential in the code length n, so are the expected number of RESAMPLE calls. In order to construct codes with polynomial complexity in the code length, we use code concatenation again. For the outer code we have, by Lemma 2, that a t-FP code is a $\bar{t}$-SC code, so we can use Lemma 24 to obtain the following result:

Theorem 30 *Using the variable framework, we can construct* $\bar{2}$*-SC*$(n, M)_q$ *codes, of rate* R *satisfying*

$$R + \epsilon = \frac{1}{8} - \frac{\log_q(2q-1)}{24}, \tag{58}$$

with polynomial complexity in the code length.

Proof For the outer code we take an AG $(n_o, M_o, d_o)_{Q_o}$ code as given by Lemma 24, with $t = 2$, over an alphabet of size $Q_o \geq 2^\beta$, $\beta \geq 4$. Again, according to Bertrand's postulate, we can choose $2^\beta \leq Q_o \leq 2\lceil 2^\beta \rceil$. According to (54), the rate R_o of this code satisfies

$$R_o + \epsilon \geq \frac{1}{2} - \frac{1}{2^{\beta/2}-1} \geq \frac{1}{6}, \ \forall \beta \geq 4. \tag{59}$$

Now, the concatenated construction imposes to take an inner code of size $M_i = Q_o$, over an alphabet of size $q < Q_o$. According to Proposition 28, we can construct such an inner code using Algorithm 1, with an expected number of RESAMPLE calls less than $\frac{2\lceil 2^\beta \rceil}{9}$. From Lemma 23, we have that $t = 2 \leq n_o - 1$ and therefore, the expected number of RESAMPLE calls is less than $2n_o^\beta/9$, i.e., polynomial in the code length $n = n_o n_i$. Moreover, as before, from Lemma 24, the complexity of constructing the outer code is also polynomial in the code length. Finally, the claim about the rate is straightforward from (28), if we take an inner code of rate

$$R_i = \frac{3-\log_q(2q-1)}{4}, \tag{60}$$

(which is again possible by Proposition 28), and then applying Proposition 25.

Corollary 31 *We can construct* $\bar{2}$*-SC*$(n, M)_2$ *codes, of rate* R *satisfying*

$$R + \epsilon = \frac{1}{8} - \frac{\log_2 3}{24}, \quad \forall \epsilon > 0, \tag{61}$$

with polynomial complexity in the code length.

Observe that in the previous theorem we have also constructed $B_2(n, M)_2$ codes, since a $\bar{2}$*-SC*$(n, M)_2$ is a $B_2(n, M)_2$ code and vice versa, according to Lemma 2.

6 Conclusions

In this paper we have presented constructions for t-frameproof codes, $\bar{2}$-separated codes, and B_2 codes, along with lower bounds for the respective code rates. Although the bounds were already known, our proof strategy leads to probabilistic constructions of such codes in polynomial time with respect to the code length.

Frameproof codes first appear in the work of Boneh and Shaw [1]. The reader familiar with [1] would have noticed that ours is a completely different approach with respect to the original one. Whereas in [1], in order to obtain asymptotically good codes, they concatenate a random outer code with an inner code having structure, we concatenate an outer code having structure with an inner code without structure. The reason to do so is that in this way our codes, as opposed to the codes in [1], are decodable with polynomial complexity in the code length, using algebraic list decoding algorithms (see for instance [24]). Moreover, the complexity of constructing our codes is polynomial in the code length. For this reason, both approaches are not comparable.

Separable codes were developed in order to make fingerprinting schemes for multimedia contents resistant to the averaging collusion [3]. There is a vast amount of work in the literature related to both upper and lower bounds. On the other hand, actual constructions are scarce and mostly restricted to codes of short length. We have also presented the first known constructions for the already non trivial case of $t = 2$. Moreover, the codes obtained have asymptotic positive rate. These constructions readily give B_2 codes.

Appendix

Proof of Theorem 10

Recall that we have considered Mn independent random variables $\mathcal{X} = \{X_1, \ldots, X_{Mn}\}$, arranged as an $M \times n$ matrix C.

From [9], we take the distribution of the random variables X_{ij}, $1 \le i \le M$, $1 \le j \le n$, to be

$$\Pr(X_{ij} = 0) = 1 - \frac{q-1}{t+1}, \ \Pr(X_{ij} = 1) = \cdots = \Pr(X_{ij} = q-1) = \frac{1}{t+1}. \tag{62}$$

Now, since the random variables X_{ij} are independent, the probability of a bad event $E(c_i, U^t)$ is

$$\begin{aligned}
\Pr[E(c_i, U^t)] &= \prod_{l=1}^{n} \left(\Pr[c_i(l) = 0] \Pr[0 \in U^t(l)] + \sum_{j=1}^{q-1} \Pr[c_i(l) = j] \Pr[j \in U^t(l)] \right) \\
&= \left[\left(1 - \frac{q-1}{t+1}\right) \left(1 - \left(\frac{q-1}{t+1}\right)^t\right) + (q-1)\frac{1}{t+1} \left(1 - \left(\frac{t}{t+1}\right)^t\right) \right]^n \\
&= \left[1 - \left(1 - \frac{q-1}{t+1}\right) \left(\frac{q-1}{t+1}\right)^t - \frac{q-1}{t+1} \left(\frac{t}{t+1}\right)^t \right]^n .
\end{aligned}$$

With the event probability at hand, as well as an upper bound on the number s computed in (22), we apply the Lovász Local Lemma again. Bad events can be avoided if

$$e \left[1 - \left(1 - \frac{q-1}{t+1}\right) \left(\frac{q-1}{t+1}\right)^t - \frac{q-1}{t+1} \left(\frac{t}{t+1}\right)^t \right]^n \frac{9M^t}{2} \le 1. \tag{63}$$

It follows that $M \leq \frac{1}{(9e/2)^{\frac{1}{t}}} \quad \frac{1}{\left(1-(1-\frac{q-1}{t+1})(\frac{q-1}{t+1})^t - \frac{q-1}{t+1}(\frac{t}{t+1})^t\right)^{\frac{n}{t}}}$.

Proof of Theorem 12

In Section 4.2 we defined Mn random variables $X_1, \ldots, X_{Mn}$, and arranged them as an $M \times n$ matrix C, with rows $\{c_1, \ldots, c_M\}$, $c_i = \{X_{i1}, \ldots, X_{in}\}$. Consider the random variables associated with a set of four rows $V^4 = \{c_{i_1}, c_{i_2}, c_{i_3}, c_{i_4}\}$. As before, we also denote an assignment to these random variables as $\{c_{i_1}, c_{i_2}, c_{i_3}, c_{i_4}\}$. An assignment is bad if $\text{desc}(c_{i_1}, c_{i_2}) = \text{desc}(c_{i_3}, c_{i_4})$, or if $\text{desc}(c_{i_1}, c_{i_3}) = \text{desc}(c_{i_2}, c_{i_4})$, or if $\text{desc}(c_{i_1}, c_{i_4}) = \text{desc}(c_{i_3}, c_{i_2})$. We represent such an event by $E(V^4)$.

To compute the probability of a bad event $E(V^4)$ happening, observe that $\text{desc}(c_{i_1}, c_{i_2}) = \text{desc}(c_{i_3}, c_{i_4})$ if $\{c_{i_1}(k), c_{i_2}(k)\} = \{c_{i_3}(k), c_{i_4}(k)\}$, for $1 \leq k \leq n$. Therefore, we start by computing $\Pr[\{c_{i_1}(k), c_{i_2}(k)\} = \{c_{i_3}(k), c_{i_4}(k)\}]$.
We have

$$\Pr[\{c_{i_1}(k), c_{i_2}(k)\} = \{c_{i_3}(k), c_{i_4}(k)\} \mid (c_{i_1}(k) = c_{i_2}(k))] = \frac{1}{q^2}, \tag{64}$$

and

$$\Pr[\{c_{i_1}(k), c_{i_2}(k)\} = \{c_{i_3}(k), c_{i_4}(k)\} \mid (c_{i_1}(k) \neq c_{i_2}(k))] = \frac{2}{q^2}. \tag{65}$$

Also,

$$\Pr[c_{i_1}(k) = c_{i_2}(k)] = \frac{1}{q} \text{ and } \Pr[c_{i_1}(k) \neq c_{i_2}(k)] = \frac{q-1}{q}. \tag{66}$$

By the law of total probability,

$$\begin{aligned}
&\Pr[\{c_{i_1}(k), c_{i_2}(k)\} = \{c_{i_3}(k), c_{i_4}(k)\}] \\
&= \Pr[\{c_{i_1}(k), c_{i_2}(k)\} = \{c_{i_3}(k), c_{i_4}(k)\} \mid (c_{i_1}(k) = c_{i_2}(k))] \cdot \Pr[c_{i_1}(k) = c_{i_2}(k)] \\
&+ \Pr[\{c_{i_1}(k), c_{i_2}(k)\} = \{c_{i_3}(k), c_{i_4}(k)\} \mid (c_{i_1}(k) \neq c_{i_2}(k))] \cdot \Pr[c_{i_1}(k) \neq c_{i_2}(k)] \\
&= \frac{1}{q^2}\frac{1}{q} + \frac{2}{q^2}\frac{q-1}{q} = \frac{2q-1}{q^3}
\end{aligned} \tag{67}$$

Since the random variables X_{ij} are independent and identically distributed (i.i.d), and we have to take into account the three cases $\text{desc}(c_{i_1}, c_{i_2}) = \text{desc}(c_{i_3}, c_{i_4})$, $\text{desc}(c_{i_1}, c_{i_3}) = \text{desc}(c_{i_2}, c_{i_4})$, $\text{desc}(c_{i_1}, c_{i_4}) = \text{desc}(c_{i_3}, c_{i_2})$, then

$$\Pr[E(V^4)] \leq 3\left(\frac{2q-1}{q^3}\right)^n \tag{68}$$

Remark 32 When computing the probability of a bad event, the reader has noticed in (64) and (66) that, for a given assignment, c_{i_1} can be equal to c_{i_2}. In this case, if the assignments for c_{i_3} and c_{i_4} were to be different, then the event would be considered good. Nevertheless, the event should be considered bad, because we have $\text{desc}(c_{i_1}) = \text{desc}(c_{i_2})$ if and only if $c_{i_1} = c_{i_2}$ and, from Definition 1, the code would not satisfy the separable property.

According to the previous remark, let us consider a set of two rows $V^2 = \{c_{i_1}, c_{i_2}\}$. An assignment is bad if $\text{desc}(c_{i_1}) = \text{desc}(c_{i_2})$. Using (66), we have that

$$\Pr[E(V^2)] = \left(\frac{1}{q}\right)^n. \tag{69}$$

According to the Mutual Independence Principle, an event is mutually independent with all events not intersecting its scope. For a given event, the number of mutually independent events can be computed by adding the number of both mutually independent events of size 4 and mutually independent events of size 2.

Let us start with an event of size 4, say $E(V^4) = \{c_{i_1}, c_{i_2}, c_{i_3}, c_{i_4}\}$. The number of events of size 4 mutually independent with $E(V^4)$ is the total number of events of size 4 minus the number of events of size 4 that do not contain any of the rows $\{c_{i_1}, c_{i_2}, c_{i_3}, c_{i_4}\}$, minus 1 (corresponding to the event $E(V^4)$) :

$$\binom{M}{4} - \binom{M-4}{4} - 1. \tag{70}$$

We can obtain the following upper bound on this number:

$$\binom{M}{4} - \binom{M-4}{4} - 1 = \frac{2M^3 - 21M^2 + 79M - 105}{3} - 1 < \frac{2M^3}{3}. \tag{71}$$

On the other hand, the number of events of size 2 mutually independent with $E(V^4)$ is the total number of events of size 2 minus the number of events of size 2 that do not contain any of the rows $\{c_{i_1}, c_{i_2}, c_{i_3}, c_{i_4}\}$,

$$\binom{M}{2} - \binom{M-4}{2}. \tag{72}$$

We have,

$$\binom{M}{2} - \binom{M-4}{2} = 4M - 10 < 4M. \tag{73}$$

The number of events of size 2 mutually independent with $E(V^2) = \{c_{i_1}, c_{i_2}\}$ is the total number of events of size 2 minus the number of events of size 2 that do not contain any of the rows $\{c_{i_1}, c_{i_2}\}$, minus 1 (corresponding to the event $E(V^2)$) :

$$\binom{M}{2} - \binom{M-2}{2} - 1 = 2M - 4 < 2M. \tag{74}$$

Analogously, the number of events of size 4 mutually independent with $E(V^2) = \{c_{i_1}, c_{i_2}\}$ is the total number of events of size 4 minus the number of events of size 4 that do not contain any of the rows $\{c_{i_1}, c_{i_2}\}$,

$$\binom{M}{4} - \binom{M-2}{4} = \frac{2M^3 - 15M^2 + 37M - 30}{6} < \frac{M^3}{3}. \tag{75}$$

Now, according to Lemma 5, we have that all bad events can be avoided both if

$$\frac{2M^3}{3} 3\left(\frac{2q-1}{q^3}\right)^n + 4M\left(\frac{1}{q}\right)^n \leq \frac{1}{4} \tag{76}$$

and

$$\frac{M^3}{3} 3\left(\frac{2q-1}{q^3}\right)^n + 2M\left(\frac{1}{q}\right)^n \leq \frac{1}{4}. \tag{77}$$

Observe that (77) is holds whenever (76) does.

Now, let us consider two cases:

1. If $2M^3\left(\frac{2q-1}{q^3}\right)^n < \frac{4M}{q^n}$, then (76) is satisfied when $\frac{8M}{q^n} \leq \frac{1}{4}$, i.e., $M \leq \frac{q^n}{32}$.
2. If $2M^3\left(\frac{2q-1}{q^3}\right)^n \geq \frac{4M}{q^n}$, then (76) is satisfied when $4M^3\left(\frac{2q-1}{q^3}\right)^n \leq \frac{1}{4}$, i.e.,

$$M \leq \frac{1}{(16)^{\frac{1}{3}}}\left(\frac{q^3}{2q-1}\right)^{n/3}. \tag{78}$$

Since $\frac{q^n}{32} \geq \frac{1}{(16)^{\frac{1}{3}}}\left(\frac{q^3}{2q-1}\right)^{n/3}$, then (78) implies (76) holds.

Remark 33 Observe that once the events $E(V^2)$ have been considered (and ruled out), we do not need to bother about the following events:

- Let $\{c_{i_1}, c_{i_2}, c_{i_3}\}$ be a set of three rows. An assignment is bad if $\text{desc}(c_{i_1}) = \text{desc}(c_{i_2}, c_{i_3})$, or $\text{desc}(c_{i_2}) = \text{desc}(c_{i_1}, c_{i_3})$, or $\text{desc}(c_{i_3}) = \text{desc}(c_{i_1}, c_{i_2})$, which implies $c_{i_1} = c_{i_2} = c_{i_3}$.
- Let $\{c_{i_1}, c_{i_2}\}$ be a set of two rows. An assignment is bad if $\text{desc}(c_{i_1}) = \text{desc}(c_{i_1}, c_{i_2})$, or $\text{desc}(c_{i_2}) = \text{desc}(c_{i_1}, c_{i_2})$, which implies $c_{i_1} = c_{i_2}$.

Proof of Theorem 14

Our approach is routine by now. To establish a lower bound for B_2 codes, we take Mn random variables X_{ij} arranged as an $M \times n$ matrix C, and associate the rows of the matrix $\{c_1, \ldots, c_M\}$, where $c_i = \{X_{i1}, \ldots, X_{in}\}$, to code words.

First, we need to define bad events. For the $M \times n$ matrix to be a B_2 code, there are two types of sets of assignments we wish to avoid. Given a four row subset $V^4 = \{c_{i_1}, c_{i_2}, c_{i_3}, c_{i_4}\}$, an assignment to the random variables (that again we also denote by $\{c_{i_1}, c_{i_2}, c_{i_3}, c_{i_4}\}$) is bad if $c_{i_1} + c_{i_2} = c_{i_3} + c_{i_4}$, or if $c_{i_1} + c_{i_3} = c_{i_2} + c_{i_4}$, or if $c_{i_1} + c_{i_4} = c_{i_2} + c_{i_3}$. We represent the event of an assignment to V^4 being bad as $E(V^4)$. A three element subset $V^3 = \{c_{i_1}, c_{i_2}, c_{i_3}\}$ is bad if $2c_{i_1} = c_{i_2} + c_{i_3}$, or if $2c_{i_2} = c_{i_1} + c_{i_3}$, or if $2c_{i_3} = c_{i_1} + c_{i_2}$. The event of V^3 being bad will be represented by $E(V^3)$.

Note that, for $V^4 = \{c_{i_1}, c_{i_2}, c_{i_3}, c_{i_4}\}$, the event that $c_{i_1} = c_{i_2} = c_{i_3} = c_{i_4}$ is taken as a bad event. Also, for $V^3 = \{c_{i_1}, c_{i_2}, c_{i_3}\}$, the event that $c_{i_1} = c_{i_2} = c_{i_3}$ is also taken as a bad event. However, an assignment such that there are two equal code words might be possible, so we need to consider the event $V^2 = \{c_{i_1}, c_{i_2}\}$. An assignment to such a V^2 is bad if $2c_{i_1} = 2c_{i_2}$.

Let us compute the probability of bad events. Notice that given $\{c_{i_1}, c_{i_2}, c_{i_3}, c_{i_4}\}$, with $\{c_{i_1}, c_{i_2}, c_{i_3}\}$ fixed, there is only a single value for c_4 such that $c_{i_1} + c_{i_2} = c_{i_3} + c_{i_4}$. Since there are q^3 ways to choose $\{c_{i_1}, c_{i_2}, c_{i_3}\}$, it is clear that

$$\Pr[c_{i_1} + c_{i_2} = c_{i_3} + c_{i_4}] = \frac{q^3}{q^4} = \frac{1}{q}.$$

As in the previous section, we are dealing with three cases: $c_{i_1} + c_{i_2} = c_{i_3} + c_{i_4}$, $c_{i_1} + c_{i_3} = c_{i_2} + c_{i_4}$ and $c_{i_1} + c_{i_4} = c_{i_2} + c_{i_3}$. Since the random variables X_{ij} are i.d.d., the probability of a bad event $E(V^4)$ happening is

$$\Pr[E(V^4)] = 3\frac{1}{q^n} \tag{79}$$

Now, given $\{c_{i_1}, c_{i_2}, c_{i_3}\}$ with c_{i_2}, c_{i_3} fixed, there is only a single value for c_{i_1} that satisfies $2c_{i_1} = c_{i_2} + c_{i_3}$. There are q^2 ways to choose c_{i_2}, c_{i_3}, so

$$\Pr[2c_{i_1} = c_{i_2} + c_{i_3}] = \frac{q^2}{q^3} = \frac{1}{q}.$$

Since we are also dealing with three cases,

$$\Pr[E(V^3)] = 3\frac{1}{q^n}. \tag{80}$$

For $V^2 = \{c_{i_1}, c_{i_2}\}$ we have that $2c_{i_1} = 2c_{i_2}$ if and only if $c_{i_1} = c_{i_2}$, and so,

$$\Pr[E(V^2)] = \left(\frac{1}{q}\right)^n. \tag{81}$$

Let us now tackle the computation of the dependence.

From the Mutual Independence Principle, given an event, we consider as mutually dependent all other events having at least one row of the matrix of random variables C in common.

We count the number of 2-subsets, 3-subsets and 4-subsets that contain a given row. To compute that number, we count the number of 1-subsets, 2-subsets and 3-subsets in a set of $M - 1$ elements, that is

$$\binom{M-1}{1}, \binom{M-1}{2} \text{ and } \binom{M-1}{3}, \text{ respectively.} \tag{82}$$

Now, observe that we need to take into account the number of rows in the event, thus, for events $E(V^4)$, $E(V^3)$ and $E(V^2)$ the condition in Lemma 5 leads to:

$$4\left(\binom{M-1}{3}\frac{3}{q^n} + \binom{M-1}{2}\frac{3}{q^n} + \binom{M-1}{1}\frac{1}{q^n}\right) \leq \frac{1}{4} \tag{83}$$

$$3\left(\binom{M-1}{3}\frac{3}{q^n} + \binom{M-1}{2}\frac{3}{q^n} + \binom{M-1}{1}\frac{1}{q^n}\right) \leq \frac{1}{4} \tag{84}$$

$$2\left(\binom{M-1}{3}\frac{3}{q^n} + \binom{M-1}{2}\frac{3}{q^n} + \binom{M-1}{1}\frac{1}{q^n}\right) \leq \frac{1}{4} \tag{85}$$

Since (85) and (84) are satisfied whenever (83) is, and

$$3\binom{M-1}{3} + 3\binom{M-1}{2} + \binom{M-1}{1} < \frac{M^3}{2},$$

then Lemma 5 leads to:

$$4\frac{M^3}{2}q^n \leq \frac{1}{4} \iff M \leq \frac{q^{n/3}}{2}.$$

Proof of Proposition 28

Observe that, we cannot use the extension of the work in [10] (as in Section 5), since that development is only valid for the symmetric version of the LLL. Therefore, as observed in Remark 18, we can resort to the original work of Moser and Tardos [12], who provide the bound (42) on the expected number of times resamples are performed in Algorithm 1.

We are going to obtain an upper bound on the number

$$S = \sum_{\text{events } E_i} \frac{2\Pr[E_i]}{1-2\Pr[E_i]} = \sum_{\text{events } E(V^4)} \frac{2\Pr[E(V^4)]}{1-2\Pr[E(V^4)]} + \sum_{\text{events } E(V^2)} \frac{2\Pr[E(V^2)]}{1-2\Pr[E(V^2)]}. \tag{86}$$

We observe that the number of events of type $E(V^4)$ is $\binom{M}{4} < \frac{M^4}{24}$, and the number of events of type $E(V^2)$ is $\binom{M}{2} < \frac{M^2}{2}$.

Since $\frac{2x}{1-2x} < 3x$, for all $x < \frac{1}{6}$, and

$$\Pr[E(V^2)] = \frac{1}{q^n} \le \frac{1}{8}, \quad \Pr[E(V^4)] = 3\left(\frac{2q-1}{q^3}\right)^n < \frac{1}{6}, \quad \forall n \ge 3,$$

then, according to (86), we have

$$S < \frac{M^4}{24} 3\Pr[E(V^4)] + \frac{M^2}{2} 3\Pr[E(V^2)] \le \frac{M^4}{8}\left(\frac{2q-1}{q^3}\right)^n + \frac{3M^2}{2q^n}. \tag{87}$$

Now, the bound on M obtained in (78) implies

$$\left(\frac{2q-1}{q^3}\right)^n \le \frac{1}{16M^3}, \quad \frac{1}{q^n} < \left(\frac{2q-1}{q^3}\right)^{n/2} \le \frac{1}{4M^{3/2}},$$

and therefore,

$$S < \frac{M^4}{8}\frac{1}{16M^3} + \frac{3}{2}M^2\frac{1}{4M^{3/2}} = \frac{1}{8}\left(\frac{M}{16} + 3\sqrt{M}\right) \le \frac{M}{9}, \quad \forall M \ge 14,$$

so we can conclude the expected number of RESAMPLE calls is less than $M/9$.

To conclude, from (78), we obtain $n = \frac{\log_q(16M^3)}{3-\log_q(2q-1)} \ge \frac{4\log_q M}{3-\log_q(2q-1)}, \forall M \ge 16$.

The codes can be obtained using Algorithm 1 with the following input:

- Alphabet $\mathcal{Q}$ of size q, $\mathcal{Q} := \{0, \ldots, q-1\}$.
  Integers $M \ge 16$, $n = \left\lfloor \frac{4\log_q M}{3-\log_q(2q-1)} \right\rfloor$.
- Random variables: $\mathcal{C} = \{X_{ij} : \mathcal{Q} \to \mathcal{Q},\ 1 \le i \le M, 1 \le j \le n\}$.
- Probability mass function: $\Pr(X_{ij} = 0) = \cdots = \Pr(X_{ij} = q-1) = \frac{1}{q}$.
- *Bad* events:
  - Arrange the r.v. X_{ij} as an $M \times n$ matrix C, with $(C)_{ij} = X_{ij}$.
    * Let C_{row} the set of rows of C.
    * Define the set $\{T \subset C_{row} : |T| = 4\}$.
    * Order the previous set and denote it by $\mathcal{R}$.
    * Let $E(T)$ be a bad event in the sense of Section 3.2.
  - Define the ordered set $\mathcal{E} := \{E(T) \mid T \in \mathcal{R}\}$.

Acknowledgements We thank the anonymous referees for their comments, that have greatly improved the presentation of the paper, as well as the accuracy of some results.

Author Contributions All authors contributed the same.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. The work of Marcel Fernández has been supported by TCO-RISEBLOCK (PID2019-110224RB-I00) MINECO. The work of Sebastiá Martín has been supported by Ministerio de Ciencia e Innovación, PID2019-109379RB-I00.

Availability of supporting data No supporting data in the paper.

Declarations

Ethical Approval and Consent to participate This manuscript has not been submitted to any other journal for simultaneous consideration.

Consent for publication Yes

Competing interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE transactions on information theory **44**(5), 1897–1905 (1998)
2. Cheng, M., Miao, Y.: On anti-collusion codes and detection algorithms for multimedia fingerprinting. IEEE transactions on information theory **57**(7), 4843–4851 (2011)
3. Cheng, M., Ji, L., Miao, Y.: Separable codes. IEEE Trans Inf Theor **58**(3), 1791–1803 (2012)
4. Blackburn, S.R.: Probabilistic existence results for separable codes. IEEE Trans Inf Theor **61**(11), 5822–5827 (2015)
5. Egorova, E., Fernandez, M., Kabatiansky, G., Lee, M.H.: Signature codes for the a-channel and collusion-secure multimedia fingerprinting codes. In: 2016 IEEE international symposium on information theory (ISIT), pp 3043–3047 (2016)
6. Lindström, B.: Determination of two vectors from the sum. J. Combin. Theor, Series A **6**, 402–407 (1969)
7. Sidon, S.: Ein satz über trigonometrische polynome und seine anwendung in der theorie der fourier-reihen. Mathematische Annalen **106**, 536–539 (1932)
8. Gu, Y., Fan, J., Miao, Y.: Improved bounds for Separable codes and B_2 codes. IEEE communications letters **24**(1), 15–19 (2020)
9. Shangguan, C., Wang, X., Ge, G., Miao, Y.: New bounds for frameproof codes. IEEE transactions on information theory **63**(11), 7247–7252 (2017)
10. Giotis, I., Kirousis, L., Psaromiligkos, K.I., Thilikos, D.M.: On the algorithmic Lovász local lemma and acyclic edge coloring. In: Proceedings of the twelfth workshop on analytic algorithmics and combinatorics (2015). Soc. Ind. Appl. Math
11. Moser, R.A.: A constructive proof of the Lovász local lemma. In: Proceedings 41st annual acm symposium on theory of computing (STOC), pp. 343–350 (2009). ACM
12. Moser, R.A., Tardos, G.: A constructive proof of the general Lovász local lemma. J. ACM (JACM) **57**(2), 11 (2010)
13. Forney, G.D.: Concatenated Codes. MIT Press, Cambridge, MA (1966)
14. Goppa, V.D.: Codes on algebraic curves. Sov. Math.-Dokl. **24**, 170–172 (1981)
15. Vlăduţ, S., Drinfeld, V.: Number of points of an algebraic curve. Functional Analysis and Its Applications - Funct Anal Appl-Engl tr. **17**, 53–54 (1983)
16. Garcia, A., Stichtenoth, H.: A tower of Artin - Schreier extensions of Function Fields attaining the Drinfeld - Vlâdut bound. Inventiones Mathematicae **121**, 211–222 (1995)

17. Garcia, A., Stichtenoth, H.: On the asymptotic behaviour of some towers of function fields over finite fields. J. Num. Theor. **61**(2), 248–273 (1996)
18. Shum, K., Aleshnikov, I., Kumar, P., Stichtenoth, H., Deolalikar, V.: A low-complexity algorithm for the construction of algebraic-geometric codes better than the gilbert-varshamov bound. Information Theory, IEEE Transactions on **47**, 2225–2241 (2001)
19. Molloy, M., Reed, B.: Graph Colouring and the Probabilistic Method. Springer, Springer (2002)
20. Livieratos, J.: Constraint satisfaction problems: Probabilistic approach and applications to social choice theory. PhD thesis, National and Kapodistrian University of Athens, Department of Mathematics (2020)
21. Sedgewick, R., Flajolet, P.: An Introduction to the Analysis of Algorithms. Addison-Wesley, Addison-Wesley (2013)
22. Alon, N., Cohen, G., Krivelevich, M., Litsyn, S.: Generalized hashing and parent-identifying codes. J. Combin. Theory, Series A **104**(1), 207–215 (2003)
23. Sandor, J., Debnath, L.: On certain inequalities involving the constant e and their applications. J. Math. Anal. Appl. **249**(2), 569–582 (2000)
24. Fernandez, M., Moreira, J., Soriano, M.: Identifying traitors using the koetter-vardy algorithm. Information Theory, IEEE Transactions on **57**, 692–704 (2011)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.