

# Constructions of low-degree and error-correcting $\epsilon$ -biased generators\*

Amir Shpilka<sup>†</sup>

## Abstract

In this work we give two new constructions of  $\epsilon$ -biased generators. Our first construction significantly extends a result of Mossel et al. (Random Structures and Algorithms 2006, pages 56-81), and our second construction answers an open question of Dodis and Smith (STOC 2005, pages 654-663). In particular we obtain the following results:

1. For every  $k = o(\log n)$  we construct an  $\epsilon$ -biased generator  $G : \{0, 1\}^m \rightarrow \{0, 1\}^n$  that is implementable by degree  $k$  polynomials (namely, every output bit of the generator is a degree  $k$  polynomial in the input bits). For any constant  $k$  we get that  $n = \Omega(m/\log(1/\epsilon))^k$ , which is nearly optimal. Our result also separates degree  $k$  generators from generators in  $NC_k^0$ , showing that the stretch of the former can be much larger than the stretch of the latter. The problem of constructing degree  $k$  generators was introduced by Mossel et al. who gave a construction only for the case of  $k = 2$ .
2. We construct a family of asymptotically good binary codes such that the codes in our family are also  $\epsilon$ -biased sets for an exponentially small  $\epsilon$ . Our encoding algorithm runs in polynomial time in the block length of the code. Moreover, these codes have a polynomial time decoding algorithm. This answers an open question of Dodis and Smith.

The paper also contains an appendix by Venkatesan Guruswami that provides an explicit construction of a family of error correcting codes of rate  $1/2$  that has efficient encoding and decoding algorithms and whose dual codes are also good codes.

## 1 Introduction

A subset  $S \subset \{0, 1\}^n$  is called an  $\epsilon$ -biased set if its bias with respect to any linear test is at most  $\epsilon$ ; namely, for every non-zero vector  $w \in \{0, 1\}^n$  we have that

---

\*A preliminary version appeared in [Shp06].

<sup>†</sup>Faculty of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel. Email: shpilka@cs.technion.ac.il. This research was supported by the Israel Science Foundation (grant number 439/06).

$|\Pr_{s \in S}[\langle w, s \rangle = 1] - 1/2| \leq \epsilon$ , where  $\langle w, s \rangle$  denotes the inner-product mod 2 of the vectors  $w$  and  $s$ . In other words, for every hyperplane  $H \subset \{0, 1\}^n$  it holds that  $\left| |S \cap H| - \frac{|S|}{2} \right| \leq \epsilon |S|$ . An  $\epsilon$ -biased generator is a mapping  $G : \{0, 1\}^m \rightarrow \{0, 1\}^n$  whose image is an  $\epsilon$ -biased set. A subset  $C \subset \{0, 1\}^n$  is called a **good error correcting code**<sup>1</sup> if it has an exponential (in  $n$ ) size and the Hamming distance between any two of its elements is linear (again, in  $n$ ).

In this paper we give two constructions of  $\epsilon$ -biased sets. The first is a construction of an  $\epsilon$ -biased generator such that each of its output bits is a low degree polynomial. Our second result is a construction of a family of good (efficiently encodable and decodable) error correcting codes that are also  $\epsilon$ -biased sets for an exponentially small  $\epsilon$ .

## 1.1 Background

The notion of  $\epsilon$ -biased sets (or more accurately of an  $\epsilon$ -biased distribution) was first defined by Naor and Naor [NN93] who also gave the first constructions of such distributions and demonstrated the power of  $\epsilon$ -biased sets for several applications. Alternative constructions and related notions appeared in a series of papers [AGHP92, AIK<sup>+</sup>90, RSW93, EGL<sup>+</sup>98, AM95]. Since their first appearance  $\epsilon$ -biased sets have found many applications in different areas of theoretical computer science including: derandomization of algorithms such as fast verification of matrix multiplication [NN93]; construction of almost  $k$ -wise independent distributions [NN93, MNN94]; inapproximability results for quadratic equations over  $\mathbb{F}_2$  [HPS93]; learning theory [AM95]; explicit constructions of Ramsey graphs [Nao92]; explicit constructions of Cayley expanders [AR94, MW04]; construction of efficient low degree tests and short PCPs [BFLS91, FGL<sup>+</sup>96, BSSVW03]; and construction of two-source extractors [Raz05].

In several recent works  $\epsilon$ -biased sets were studied from a different perspective. In [CM01] Cryan and Miltersen ask whether there exist an  $NC^0$  construction of an  $\epsilon$ -biased generator for which  $n$  is super-linear in  $m$ . This question was answered affirmatively by Mossel et al. [MST06] who gave a construction of a generator in  $NC_k^0$  with  $n = m^{\Omega(\sqrt{k})}$ , where  $NC_k^0$  denotes the class of functions in which every output bit depends on at most  $k$  input bits. Mossel et al. also raised the question of constructing  $\epsilon$ -biased generators such that each of their output bits is a degree  $k$  polynomial in the input bits. They were also able to give a construction of a degree 2 generator with a near optimal stretch (i.e.  $n = \Omega(m^2)$  and  $\epsilon = \exp(-O(n))$ ). Note that the maximal stretch of such  $\epsilon$ -biased generators (i.e. in  $NC_k^0$  or of low degree) is much smaller than  $n = \Omega(2^m \cdot \epsilon^2)$  that can be achieved non constructively, or the known stretch of  $n = \Omega(2^{m/2} \cdot \epsilon)$  that was given in [AGHP92].

In [DS05] Dodis and Smith ask “Does there exist an explicitly-constructible ensemble of good codes with small bias and polytime encoding and decoding algorithms (ideally, codes with linear rate and minimum distance, and exponentially small bias)?”. Namely, [DS05] raise the question of constructing a family of good codes that are also  $\epsilon$ -biased sets for an exponentially (in the block length of the code) small  $\epsilon$ . Such a family of codes was needed for the construction of a cryptographic scheme that will enable two parties to securely

---

<sup>1</sup>To be completely accurate we have to speak about family of codes, and we do it in a later section.

correct errors in a shared secret string ([DS05] managed to construct such scheme using other methods).

## 1.2 Our results.

Our first result is a construction of degree  $k$   $\epsilon$ -biased generators of maximal stretch (up to a constant). Namely we give a construction of a degree  $k$  generator from  $m$  bits to  $n = \Omega((m/\log(1/\epsilon))^k)$  bits, for any fixed  $k$ . Thus, for every fixed  $\epsilon$  the output length is  $\Omega(m^k)$ , and clearly the output length cannot exceed  $m^k$  (as there are only  $O(m^k)$  linearly independent polynomials of degree  $k$  in  $m$  variables).

**Theorem 1** *For every integer  $k$  and every large enough<sup>2</sup> integer  $m$  and every  $\epsilon > \exp(-O(\frac{m^{1-\frac{1}{k}}}{k2^k}))$ , there is a mapping  $G : \{0, 1\}^m \rightarrow \{0, 1\}^n$ , where  $n = \Omega((\frac{m}{k2^k \log(1/\epsilon)})^k)$ , such that  $G$  is a degree  $k$  generator with bias at most  $\epsilon$ .*

For a generator  $G : \{0, 1\}^m \rightarrow \{0, 1\}^n$  we define the stretch of  $G$  to be  $n - m$ . As a corollary of Theorem 1 we get a separation between the possible stretch of a degree  $k$  generator and that of a generator constructed in  $NC_k^0$ . Indeed, the theorem gives a degree  $k$  generator with stretch  $m^{k-o(1)}$  (for constant  $k$  and  $\epsilon = 2^{-m^{o(1)}}$ ). In contrast, Theorem 6 of [MST06] shows that the stretch of an  $\epsilon$ -biased generator in  $NC_k^0$  is at most  $O(2^k m^{\lceil k/2 \rceil})$ , for  $\epsilon < 2^{-2k}$ .

Our second result is a construction of a family of  $\epsilon$ -biased good codes. Namely, we give a construction of a family of good codes (constant relative rate, constant relative distance, efficient encoding and decoding algorithms) such that the codes in the family have an exponentially small bias. Thus, our construction answers affirmatively the open question of [DS05]. We note that such a code cannot be linear as it has to fool all linear tests. We give a less formal statement of our result here, and leave the formal statement to Section 4.

**Theorem 2** *(informal) There is a polynomial time constructible generator  $G : \{0, 1\}^m \rightarrow \{0, 1\}^n$ , that has an exponentially small bias (namely  $\epsilon = \exp(-O(n))$ ), such that its image  $C = G(\{0, 1\}^m)$  is a good code (that is,  $C$  has constant relative rate and linear (in  $n$ ) relative distance) that has a polynomial time decoding algorithm that can fix a constant fraction of errors.*

## 1.3 Motivation

### Motivation for studying low degree $\epsilon$ -biased generators.

It is very desirable to give explicit constructions of low complexity for combinatorial objects. For example, in [Hås87, Gol00a, CM01, KL01, MST06, AIK06] questions regarding the existence of objects such as  $\epsilon$ -biased generators, one-way functions and pseudo-random generators in  $NC^0$  were studied. In [MST06] the question of constructing degree  $k$   $\epsilon$ -biased generators, i.e. generators that each of their output bits is a degree  $k$  polynomial in the input

---

<sup>2</sup>More precisely, there exists  $m_0$ , independent of  $k$ , such that for every  $m \geq m_0$ ...

bits, was first raised. As low degree polynomials are a natural “low complexity” class, constructing low degree  $\epsilon$ -biased generators is a natural question. [MST06] gave a construction of a degree 2 generator with a near optimal stretch (i.e.  $n = \Omega(m^2)$  and  $\epsilon = \exp(-O(n))$ ), but were unable to achieve similar results for higher degrees.

**Motivation for studying error-correcting  $\epsilon$ -biased generators.**

Error correcting codes (ECCs) have many applications in theoretical computer science (cf. [Fei95, Gur01, Tre04]). Finding explicit constructions of ECCs is an extensively studied question (cf. [MS77, Tre04]). In recent years the focus in the theoretical computer science community is on giving explicit constructions of ECCs that have additional properties, for example: codes that have efficient list-decoding algorithms (cf. survey of Trevisan [Tre04]), quantum codes (cf. [NC00, KSV02]), codes that are locally testable and codes that are locally decodable (cf. survey of Goldreich [Gol00b]). In [DS05] the question of constructing a family of good codes that are also  $\epsilon$ -biased sets for an exponentially (in the block length of the code) small  $\epsilon$  was raised. As error correcting codes and  $\epsilon$ -biased sets are such important objects it is natural to combine them and to construct an  $\epsilon$ -biased error correcting code.

When constructing ECCs one usually tries to maximize the minimum distance and the rate of the code. On the other hand when constructing an  $\epsilon$ -biased set one tries to minimize the bias and the size of the set. Thus it seems a self contradicting task to construct a set that is both a good code and a “good”  $\epsilon$ -biased set. However, when fixing the rate (i.e. the size of the set) one can think of the following optimization problem: find a set of vectors that is a code of that given rate with the largest minimal distance and the smallest bias.

**1.4 Methods**

Our constructions are similar in spirit to the constructions of  $\epsilon$ -biased generators of Mossel et al. [MST06]. In order to describe their construction we recall that every linear test can be identified with a binary vector of length  $n$ . The weight of the test is the number of non zero coordinates of this vector. [MST06] constructed  $\epsilon$ -biased generators in the following way. They first constructed a generator  $G^{(h)}$  that is (almost) unbiased w.r.t. heavy tests, i.e. the bias of  $G^{(h)}$  w.r.t. any test  $w$  of large weight is small. Then they construct a generator  $G^{(l)}$  that is (almost) unbiased w.r.t. light tests. Their final generator is the XOR, on two independent inputs, of  $G^{(h)}$  and  $G^{(l)}$ . Namely,  $G(x, y) = G^{(h)}(x) \oplus G^{(l)}(y)$ .

In the proofs of both Theorem 1 and Theorem 2 we also separate the construction of the generators to two tasks. First we construct a generator that is (almost) unbiased w.r.t. heavy tests  $G^{(h)}$  and then we construct a generator that is (almost) unbiased w.r.t. light tests  $G^{(l)}$ . As in [MST06] our final generator is the XOR of the two generators on two independent inputs.

A significant difference from [MST06] is in the way that we construct  $G^{(l)}$ . In their work  $G^{(l)}$  was constructed “from scratch”, i.e. there was no connection between the construction of  $G^{(l)}$  and of  $G^{(h)}$ . We show a novel use of error correcting codes that enables us to transform *any* generator that is (almost) unbiased with respect to heavy tests to a generator that is (almost) unbiased w.r.t. light tests. In the heart of this transformation lies the observation

that using the generating matrix of a linear error correcting code of relative rate  $1/2$ , one can construct a linear transformation (from  $\{0, 1\}^n$  to itself) that sends “light” vectors to “heavy” vectors (where “light” and “heavy” depend on the properties of the code). The reason is that every linear code from  $n$  bits to  $2n$  bits (and hence of rate  $1/2$ ) can be easily transformed to a code (with the same parameters) of the form  $x \rightarrow (x, Ax)$  where  $A$  is a linear transformation from  $\{0, 1\}^n$  to itself. Thus if the weight of  $x$  is small, then the weight of  $Ax$  must be large, and so  $A$  has the required property. We note that as we want  $A$  to send vectors of length  $n$  to vectors of length  $n$  we must consider codes of rate  $1/2$ . Using this transformation we are able to reduce the task of constructing an  $\epsilon$ -biased generator to the task of constructing a generator that is unbiased w.r.t. heavy tests (and that may have additional properties).

The construction of  $G^{(h)}$  is also different from the one in [MST06] from the obvious reason that it has to satisfy different properties in both papers. However, its “spirit” is the same in the sense that both constructions rely on the fact that the bias of the sum of many independent random variables is exponentially small (in the number of variables).

## 1.5 Organization

In Section 2 we give the basic notations and definitions. In particular in Subsection 2.1 we give the basic definitions of error correcting codes and recall a construction of good error correcting codes that their dual codes also form good codes. In Subsection 2.2 we give the basic definitions regarding  $\epsilon$ -biased sets and  $\epsilon$ -biased generators and give the proofs of some well known facts. In Section 3 we prove Theorem 1, and in Section 4 we prove Theorem 2.

Appendix A, written by Venkatesan Guruswami, contains an explicit construction of a family of error correcting codes of rate  $1/2$  that has efficient encoding and decoding algorithms and whose dual codes are also good codes. While this result seems to be well known to expert, we were unable to find any explicit statement of it and so we give it here for completeness.

## 2 Preliminaries

We shall denote with  $\log(x)$  the natural logarithm of  $x$ , i.e.  $\log(x) = \log_e(x)$ . We denote  $\exp(x) = e^x$ . We use the notation of  $|I|$  to denote the size of the set  $I$ . For a random variable  $X$  that is distributed according to a distribution  $D$  we denote with  $\mathbb{E}_D[X]$  the expectation of  $X$ . For a matrix  $A$  we denote with  $A^t$  the transpose of  $A$ . For a positive integer  $i$  we denote with  $[i]$  the set  $[i] = \{1, 2, \dots, i\}$ .

For a vector  $v \in \{0, 1\}^n$  we denote with  $v_i$  the  $i$ 'th coordinate of  $v$ . Namely,  $v = (v_1, \dots, v_n)$ . For two vectors  $u, v \in \{0, 1\}^n$  we denote with  $\text{dist}(u, v)$  the hamming distance between  $u$  and  $v$ , i.e.  $\text{dist}(u, v) = |\{i : u_i \neq v_i\}|$ . We also denote with  $\text{wt}(v)$  (the weight of  $v$ ) the number of non-zero coordinates of  $v$ . In other words,  $\text{wt}(v) = \text{dist}(v, \vec{0})$ , where  $\vec{0}$  is the zero vector. For  $v, u \in \{0, 1\}^n$  we denote with  $\langle v, u \rangle$  their inner product modulo 2, i.e.  $\langle v, u \rangle = v_1 \cdot u_1 \oplus \dots \oplus v_n \cdot u_n$ .

For two vectors  $v, u \in \{0, 1\}^n$  we denote  $v \oplus u = (v_1 \oplus u_1, \dots, v_n \oplus u_n)$ , i.e. it is the coordinate-wise XOR of  $v$  and  $u$ . For two multisets  $S_1, S_2 \subseteq \{0, 1\}^n$  we denote  $S_1 \oplus S_2$  to be the multiset  $S_1 \oplus S_2 = \{v \oplus u \mid v \in S_1 \text{ and } u \in S_2\}$ . For two functions  $G_1 : \{0, 1\}^{m_1} \rightarrow \{0, 1\}^n$  and  $G_2 : \{0, 1\}^{m_2} \rightarrow \{0, 1\}^n$  we denote with  $G = G_1 \oplus G_2$  the function  $G(x, y) : \{0, 1\}^{m_1} \times \{0, 1\}^{m_2} \rightarrow \{0, 1\}^n$  satisfying  $G(x, y) = G_1(x) \oplus G_2(y)$ . Usually we write  $\{0, 1\}^{m_1+m_2}$  instead of  $\{0, 1\}^{m_1} \times \{0, 1\}^{m_2}$ .

## 2.1 Error correcting codes

Let  $E : \{0, 1\}^k \rightarrow \{0, 1\}^n$ . Denote with  $C = E(\{0, 1\}^k)$  the image of  $E$ . Then  $C$  is called an  $[n, k, d]$ -code if for any two codewords  $E(v), E(u) \in C$ , where  $u \neq v$ , we have that  $\text{dist}(E(u), E(v)) \geq d$ . We denote with  $R = k/n$  the *relative rate* of  $C$  and with  $\delta = d/n$  the *relative minimal distance* of  $C$ , and say that  $C$  is an  $[R, \delta]$ -code. When  $E$  is a linear mapping we say that  $C$  is a linear code. A map  $D : \{0, 1\}^n \rightarrow \{0, 1\}^k$  can correct  $t$  errors (w.r.t.  $E$ ) if for any  $v \in \{0, 1\}^k$  and any  $w \in \{0, 1\}^n$  such that  $\text{dist}(E(v), w) \leq t$  we have that  $D(w) = v$ . Such a  $D$  is called a *decoding algorithm* for  $C$ . A family of codes  $\{C_i\}$ , where  $C_i$  is an  $[R_i, \delta_i]$ -code of block length  $n_i$ , has *constant rate* if there exists a constant  $0 < R$  such that for all codes in the family it holds that  $R_i \geq R$ . The family has a *linear distance* if there exists a constant  $0 < \delta$  such that for all codes in the family we have  $\delta_i \geq \delta$ . In such a case we say that the family is a family of  $[R, \delta]$  codes. If a family of codes as above has  $\lim_{i \rightarrow \infty} n_i = \infty$ , a constant rate and a linear minimal distance then we say that the family is a family of *good codes* and that the codes in the family are good. Similarly, we say that the family of codes has a decoding algorithm for a fraction  $\tau$  of errors if for each  $C_i$  there is a decoding algorithm  $D_i$  that can decode from  $\tau \cdot n_i$  errors.

When  $C$  is a linear code we define the dual of  $C$  in the following way:

$$C^\perp \triangleq \{y \in \{0, 1\}^n : \forall x \in C \langle y, x \rangle = 0\}.$$

A family of codes is said to have *good dual codes* if the family of the dual codes is good.

For our constructions we shall need families of error correcting codes that have certain properties. For the proof of Theorem 1 we shall need a family of good codes that have efficient encoding and decoding algorithms. The theorem below (which is folklore) guarantees the existence of the required family.

**Theorem 3** *There exist  $\zeta > 0$  and a family of linear error correcting codes  $\{C_n\}$ , such that  $C_n \subset \{0, 1\}^{2n}$  and each  $C_n$  is an  $[\frac{1}{2}, \zeta]$  code. Moreover, for each  $n$  there is a decoding algorithm for  $C_n$ , that can correct  $\zeta/2$  fraction of errors, and that runs in polynomial time. In addition, there is a polynomial time algorithm  $A$  that given  $n$  outputs the generating matrix for  $C_n$ .*

**Proof** [Sketch] The idea of the proof is to concatenate Reed-Solomon codes with good inner codes. We shall only give a sketch of the proof. Given an integer  $n$  let  $k$  be such that  $2^{k-1} \leq n < 2^k$ . Let  $\mathbb{F} = \mathbb{F}_{2^{2k}}$ . Let  $m$  be such that  $n \leq 1.8mk \leq n + 2k$ . Consider a Reed-Solomon code  $C_{out} \subset \mathbb{F}^m$  of rate  $d = \lfloor 0.9m + 1 \rfloor$ . Clearly  $0.9m \leq d \leq 0.9m + 1$ . Let  $C_{in}$  be a

$[\frac{2}{3}, \delta]$ -code of block length  $3k$  that has an efficient algorithm for decoding from  $0 < \delta' < \delta/2$  fraction of errors (we can construct such a code recursively<sup>3</sup>). Note that the rate of  $C_{in}$  is  $2k$ . Consider the concatenated code  $C = C_{out} \circ C_{in}$ . It is a code of rate  $d \cdot 2k$  and block length  $m \cdot 3k < 2n$ . By our choice of  $m$  and  $k$  we have that  $n \leq 1.8mk \leq 2dk \leq 1.8mk + 2k \leq n + 4k$ . Moreover, it is known how to construct a decoding algorithm for  $C$  given decoding algorithms for  $C_{out}$  and  $C_{in}$  (see e.g. [For66, GS02]). In particular  $C$  is a  $[R, \zeta']$ -code for some  $R > 1/2$  and  $\zeta' > 0$ . By taking a sub-code  $C'$  of  $C$  of rate  $n$ , and adding  $2n - 3mk$  zeros to the codewords in  $C'$ , we get a  $[\frac{1}{2}, \zeta]$ -code of rate  $n$  that has efficient algorithms for encoding and for decoding from a  $\zeta/2$  fraction of errors for some constant  $\zeta > 0$ .

We note that the sketch above can be improved to yield codes with better parameters, but we do not make any attempt to optimize the parameters here.  $\square$

For the proof of Theorem 2 we shall need a family of good error correcting codes, that can be efficiently encoded and decoded, such that their dual codes are also good codes. Constructions of such codes are considered to be a known fact, see e.g. [HvLP98, SV90, FR93, GS01]. However, there is a slight problem with these constructions that is usually ignored, which is that the decoding algorithms usually require the code to be represented in a specific form, and it is not clear how to find such a representation efficiently<sup>4</sup>. In appendix A we give a proof due to Guruswami of the following theorem that explicitly shows how to construct the required codes.

**Theorem 4** *There exists a constant  $\zeta \geq 1/30$  such that for every integer  $i \geq 1$  there is a  $[1/2, \eta]$ -code of block length  $n_i = 42 \cdot 8^{i+1}$ , for some  $\eta \geq \zeta$ . Moreover, a representation of this code can be constructed in  $\text{poly}(n_i)$  time that enables polynomial time encoding and deterministic polynomial time decoding from a fraction  $\zeta/2$  of errors. Furthermore, the dual of this linear code is a  $[1/2, \eta']$ -code for some  $\eta' \geq \zeta$ .*

## 2.2 $\epsilon$ -biased sets

**Definition 5** *Let  $S \subseteq \{0, 1\}^n$  be a multi-set. For a non-zero vector  $w \in \{0, 1\}^n$  we denote with  $\text{bias}_w(S)$  the bias of  $S$  w.r.t.  $w$ . That is,*

$$\text{bias}_w(S) \triangleq \left| \frac{1}{2} - \Pr_{s \in S}[\langle w, s \rangle = 1] \right| = \left| \frac{1}{2} - \frac{1}{|S|} \sum_{s \in S} \langle w, s \rangle \right|.$$

*The bias of  $S$  is equal to the maximal bias w.r.t. any non-zero test:*

$$\text{bias}(S) = \max_{\vec{0} \neq w \in \{0, 1\}^n} \text{bias}_w(S).$$

*We say that,  $S$  is  $\epsilon$ -biased if for every  $\vec{0} \neq w \in \{0, 1\}^n$  it holds that  $\text{bias}_w(S) \leq \epsilon$ .*

<sup>3</sup>In this sketch we only show how to get relative rate  $1/2$  and not relative rate of  $2/3$  that is needed for  $C_{in}$ , but the construction is very similar in spirit.

<sup>4</sup>Because of this reason, Theorem 3 of the conference version of this paper [Shp06] is not correct as stated.

In order to define  $\epsilon$ -biased generator it will be convenient to speak about  $\epsilon$ -biased distributions. In the following we identify vectors  $w \in \{0, 1\}^n$  with a subset  $W \subseteq [n]$  in the usual manner ( $i \in W$  if and only if  $w_i = 1$ ).

**Definition 6** Let  $G : \{0, 1\}^m \rightarrow \{0, 1\}^n$  be a map. We denote  $G = (G_1, \dots, G_n)$ , where  $G_i$  is the  $i$ -th output bit of  $G$ . The bias of  $G$  w.r.t. a non-zero vector  $w \in \{0, 1\}^n$  is defined to be

$$\text{bias}_w(G) \triangleq \left| \frac{1}{2} - \Pr_{x \in \{0, 1\}^m} [\oplus_{i \in W} G_i(x) = 1] \right| = \left| \frac{1}{2} - \mathbb{E}_{x \in \{0, 1\}^m} \langle G(x), w \rangle \right|.$$

The bias of  $G$  is equal to the maximal bias w.r.t. any non-zero test:

$$\text{bias}(G) = \max_{\bar{0} \neq w \in \{0, 1\}^n} \text{bias}_w(G).$$

In particular,  $G$  is  $\epsilon$ -biased if for every non-zero  $w$  it holds that  $\text{bias}_w(G) \leq \epsilon$ . Notice that the bias of  $G$  is equal to the bias of the multi-set  $G(\{0, 1\}^m)$ .

The following well known lemma gives information about the bias of the XOR of two independent random variables (or the XOR of two different sets).

**Lemma 7** Let  $G_1, G_2 : \{0, 1\}^m \rightarrow \{0, 1\}^n$  be two independent random variables (w.r.t. the uniform distribution on  $\{0, 1\}^m$ ) taking values in  $\{0, 1\}^n$ . Then for every  $w \in \{0, 1\}^n$  it holds that  $\text{bias}_w(G_1 \oplus G_2) \leq 2\text{bias}_w(G_1)\text{bias}_w(G_2) \leq \min_i \text{bias}_w(G_i)$ . Similarly, if  $S_1, S_2 \subseteq \{0, 1\}^n$  are multisets then we have that  $\text{bias}_w(S_1 \oplus S_2) \leq 2\text{bias}_w(S_1)\text{bias}_w(S_2) \leq \min_i \text{bias}_w(S_i)$ .

The following lemma is an immediate corollary of Lemma 7.

**Lemma 8** Let  $(G^{(i)})_{i \in I}$  be a family of functions from  $\{0, 1\}^m$  to  $\{0, 1\}^n$ . Assume that  $(G^{(i)})_{i \in I}$  are independent random variables w.r.t. the uniform distribution on  $\{0, 1\}^m$ . Then

$$\text{bias}(\oplus_{i \in I} G^{(i)}) \leq 2^{|I|-1} \prod_{i \in I} \text{bias}(G^{(i)}).$$

Similarly, if  $(S_i)_{i \in I}$  is a family of subsets of  $\{0, 1\}^n$  we have that

$$\text{bias}(\oplus_{i \in I} S_i) \leq 2^{|I|-1} \prod_{i \in I} \text{bias}(S_i).$$

As a special case of this lemma we get the well known estimate for the bias of a sum of independent random coins.

**Lemma 9** Let  $X_1, \dots, X_t$  be independent 0/1 random variables. Assume that for some  $0 < \delta < 1/2$  and for every  $i$  we have that  $\delta \leq \Pr[X_i = 1] \leq 1 - \delta$ , then

$$\text{bias}(\oplus_{i=1}^t X_i) \leq \frac{1}{2} (1 - 2\delta)^t.$$

Another basic fact that we shall need is an estimate on the bias of a degree  $k$  polynomial. This is another folklore lemma that is usually attributed to Schwartz and Zippel [Sch80, Zip79], as it resembles their original result.

**Lemma 10 (Schwartz-Zippel)** *Let  $f : \{0, 1\}^m \rightarrow \{0, 1\}$  be a non-constant degree  $k$  polynomial, then*

$$\frac{1}{2^k} \leq \Pr[f(x) = 1] \leq 1 - \frac{1}{2^k}.$$

*Equivalently,*

$$\text{bias}(f(x)) \leq \frac{1}{2} - \frac{1}{2^k}.$$

**Proof** The proof is by induction on the degree  $k$ . For  $k = 1$  the claim is clear as any non constant linear function is unbiased. For general  $k$ , we first note that we can assume w.l.o.g. that  $f$  is multilinear (as otherwise we consider its multilinearization, which is also of degree at most  $k$  and is equal to  $f$  on  $\{0, 1\}^n$ ). Assume w.l.o.g. that  $f$  depends on  $x_1$  (as  $f$  is not constant) and write  $f = x_1 \cdot h(x_1, \dots, x_n) + g(x_2, \dots, x_n)$ , where  $h$  is of degree at most  $k - 1$ . By the induction hypothesis the probability that  $h$  is not zero for a random assignment  $\rho$  to  $(x_2, \dots, x_n)$  is at least  $2^{1-k}$  (if  $h$  is a non zero constant then the probability is 1). For any such  $\rho$  for which  $h(\rho)$  is non zero, there is exactly one value  $\alpha \in B$  for which  $\alpha \cdot h(\rho) + g(\rho) \neq 0$ . Thus the probability that  $f$  is not zero is at least  $2^{-k}$ . Similarly we get that the probability that  $f$  is zero is at least  $2^{-k}$ .  $\square$

### 3 Low degree $\epsilon$ -biased generator

In this section we construct an  $\epsilon$ -bias generator where each of its output bits is a low degree polynomial. Similarly to the construction in the paper of Mossel et al. [MST06] our generator is the combination of two other generators. One generator has a low bias w.r.t. “heavy” tests (tests that involve a large number of output bits) and the other has a low bias against “light” tests (i.e. tests that involve a small number of output bits). The final generator is obtained by XOR-ing the output of the two generators on independent seeds.

We first construct a generator that is unbiased against heavy tests, and then show a general method for transforming generators that are unbiased with respect to heavy tests into generators that are unbiased with respect to light tests.

#### 3.1 Construction for heavy tests

The following theorem gives a construction of a generator that is unbiased w.r.t. heavy tests. We try to give the most general statement so it is a bit cumbersome. The reader should have the following “relaxed” statement in mind: For every  $k, m$  and every  $0 < \epsilon$  (such that  $\epsilon$  is not too small as a function of  $m$ , say  $\epsilon > 2^{-\sqrt{m}}$ ), we can construct a degree  $k$   $\epsilon$ -biased generator (that is unbiased w.r.t. heavy tests) from  $m$  bits to  $n$  bits where  $n = \Omega((m/k \log(1/\epsilon))^k)$ . One difference of the relaxed statement from the formal one is that we need to make all the parameters involved in the construction integers. Another difference is that we aim to make the bound on  $\epsilon$  as small as possible. We now give the formal statement of the theorem.

**Theorem 11** For every integer  $k$ , a large enough integer  $m$  and every  $0 < a, \epsilon$  such that  $\epsilon \geq \exp(-O(\frac{am^{1-\frac{1}{k}}}{k2^k}))$ , let  $\gamma = \gamma_{k,\epsilon,a} = \frac{2a}{2^k \log(1/2\epsilon) + 5a}$ ,  $s_0 = \lfloor \gamma m \rfloor$ ,  $i_0 = \lfloor m/s_0 \rfloor$ . Let  $n = i_0 \binom{s_0}{k}$ . Then there is an explicit map  $G^{(h)} : \{0, 1\}^m \rightarrow \{0, 1\}^n$  with the following properties:

- For every  $w \in \{0, 1\}^n$  such that  $\text{wt}(w) \geq an$  we have that  $\text{bias}_w(G^{(h)}) \leq \epsilon$ .
- Each output bit of  $G^{(h)}$  is a degree  $k$  polynomial in the input bits.

**Proof** We first give a sketch of the proof. Partition the  $m$  input bits to  $k$  sets of roughly equal sizes. For each subset of inputs  $B$  we define a set of output bits such that each of the outputs in the set depends only on the input bits in  $B$ . Thus, output bits that correspond to different sets are independent (as random variables when the input is chosen uniformly at random from  $\{0, 1\}^m$ ). The output bits that depend on a given set of input bits are in 1 – 1 correspondence with multilinear monomials of degree  $k$ . Namely, every output bit is the evaluation of a degree  $k$  monomial on the input bits of the relevant set. Now, given a “heavy” linear combination of the output bits, we can present it as a linear combination of “many” degree  $k$  polynomials, where each polynomial is defined on a different subset of inputs. As these polynomials are defined on distinct sets of variables, they are independent random variables, and by Lemma 9 the bias of their sum is small. We now give the more formal proof.

Denote with  $\mathcal{M}_k[y_1, \dots, y_{s_0}]$  the set of all degree  $k$  multilinear monomials in  $s_0$  variables. The size of  $\mathcal{M}_k$  is  $\binom{s_0}{k}$ . Partition the input bits  $\{x_1, \dots, x_m\}$  to  $i_0$  sets of size  $s_0$ , and to a leftover set of size  $m - i_0 \cdot s_0 < s_0$ . The  $i$ -th set is  $B_i = \{x_{s_0 \cdot (i-1) + 1}, \dots, x_{s_0 \cdot i}\}$ . For every one of the first  $i_0$  sets of the partition we define  $\binom{s_0}{k}$  output bits. Each of the output bits corresponds to a different monomial from  $\mathcal{M}_k$  evaluated on the variables of  $B_i$ . We denote with  $G_i$  the set of output bits corresponding to  $B_i$  and with  $g_{i,M}$  the output bit corresponding to the monomial  $M$ . With these notations we have that

$$g_{i,M}(x_1, \dots, x_m) = M(B_i) = M(x_{s_0 \cdot (i-1) + 1}, \dots, x_{s_0 \cdot i}),$$

$$G_i = (g_{i,M})_{M \in \mathcal{M}_k},$$

$$G^{(h)} = (G_1, \dots, G_{i_0}) = (g_{i,M})_{i \in [i_0], M \in \mathcal{M}_k}.$$

The length of the output of  $G^{(h)}$  is clearly  $i_0 \binom{s_0}{k} = n$ .

For example, let  $m = 7$ ,  $\gamma = 1/2$  and  $k = 2$ . Then  $s_0 = 3$ ,  $i_0 = 2$ ,  $n = 2 \cdot \binom{3}{2} = 6$ ,  $\mathcal{M}_3 = \{y_1 y_2, y_1 y_3, y_2 y_3\}$ ,  $B_1 = \{x_1, x_2, x_3\}$  and  $B_2 = \{x_4, x_5, x_6\}$ . We get that  $G_1 = (x_1 x_2, x_1 x_3, x_2 x_3)$ ,  $G_2 = (x_4 x_5, x_4 x_6, x_5 x_6)$ , and  $G^{(h)} = (x_1 x_2, x_1 x_3, x_2 x_3, x_4 x_5, x_4 x_6, x_5 x_6)$ .

We now show that for every  $w \in \{0, 1\}^n$  such that  $\text{wt}(w) \geq an$  we have that  $\text{bias}_w(G^{(h)}) \leq \epsilon$ . Indeed let  $w$  be such that  $\text{wt}(w) \geq an$ . For convenience we enumerate the coordinates of  $w$  in the same way as the coordinates of  $G^{(h)}$ , that is  $w = (w_{i,M})_{i \in [i_0], M \in \mathcal{M}_k}$ . We also partition  $w$  to  $i_0$  disjoint sets  $w = (w_1, \dots, w_{i_0})$ , where  $w_i = (w_{i,M})_{M \in \mathcal{M}_k}$ . We note that as  $\text{wt}(w) \geq an$  then at least  $\lceil a \cdot \frac{n}{\binom{s_0}{k}} \rceil = \lceil a \cdot i_0 \rceil$  of the  $w_i$ 's are not empty. We now have

that,

$$\langle w, G^{(h)}(x) \rangle = \bigoplus_{i=1}^{i_0} \langle w_i, G_i(x) \rangle = \bigoplus_{i=1}^{i_0} \left( \bigoplus_{M \in \mathcal{M}_k} w_{i,M} g_{i,M}(x) \right) = \bigoplus_{i=1}^{i_0} p_i(B_i),$$

where each  $p_i(B_i)$  is a degree  $k$  polynomial, over  $\mathbb{F}_2$ , in the variables of  $B_i$ . Denote with  $I$  the set of indices for which  $p_i \neq 0$ . As each  $p_i$  is a sum of different degree  $k$  monomials we have that the size of  $I$  is equal to the number of nonempty  $w_i$ -s. Thus,

$$|I| \geq \lceil a \cdot i_0 \rceil \geq a \cdot i_0 = a \left\lfloor \frac{m}{\lceil \gamma m \rceil} \right\rfloor \geq a \left( \frac{m}{\lceil \gamma m \rceil} - 1 \right) \geq a \left( \frac{1}{\gamma} - 1 \right) > 2^{k-1} \log(1/2\epsilon).$$

As the sets  $B_i$  are disjoint, the polynomials  $p_i(B_i)$  for  $i \in I$ , viewed as random variables in the input bits, are independent random variables. By the Schwartz-Zippel Lemma (Lemma 10), we get that the bias of each  $p_i$ , for  $i \in I$  is at most  $\frac{1}{2} - \frac{1}{2^k}$ , and so by Lemma 8 we get that

$$\text{bias}_w(G^{(h)}) \leq \frac{1}{2} \left( 1 - \frac{2}{2^k} \right)^{|I|} < \frac{1}{2} \left( 1 - \frac{1}{2^{k-1}} \right)^{2^{k-1} \log(1/2\epsilon)} \leq \epsilon.$$

It is clear that the complexity of computing this encoding is polynomial in  $n$ . This completes the proof of the Theorem.  $\square$

### 3.2 From heavy to light

In this section we prove a theorem that shows that in order to construct an  $\epsilon$ -biased generator it is sufficient to construct a generator whose output is almost unbiased w.r.t. “heavy” tests. The basic tool in proving this theorem is a linear transformation from  $\{0, 1\}^n$  to itself, that sends all the non-zero vectors in the Hamming ball of radius  $an$  (light vectors) to vectors of weight at least  $bn$  (heavy vectors). Stated differently, this linear transformation has the property that it takes any two vectors that are at distance at most  $an$  and sends them to vectors at distance at least  $bn$ . This definition immediately brings to mind error correcting codes, and indeed the construction of such transformations is based on the generating matrix of a suitable error correcting code.

**Definition 12** *A linear transformation  $A : \{0, 1\}^n \rightarrow \{0, 1\}^n$  is  $(a, b)$ -expanding if for every  $v \in \{0, 1\}^n$  such that  $\text{wt}(v) \leq an$  we have that  $\text{wt}(v^t A) = \text{wt}(A^t v) \geq bn$ . We say that  $A$  is symmetric  $(a, b)$ -expanding if in addition for every  $u \in \{0, 1\}^n$  such that  $\text{wt}(u) \leq an$  we have that  $\text{wt}(Au) \geq bn$ .*

We now show how to construct symmetric expanding linear transformations.

**Theorem 13** *(Expanding transformations) Assume that there exists an explicit construction of a linear  $[\frac{1}{2}, \delta]$ -code  $C$  of block length  $2n$  over  $\mathbb{F}_2$ . Then there is an explicit  $(a, \delta - a)$ -expanding transformation  $A : \{0, 1\}^n \rightarrow \{0, 1\}^n$  that can be constructed in the same time (up to an additive  $O(n^3)$  term) as the generating matrix of the underlying code. If in addition the dual code,  $C^\perp$ , is also a  $[\frac{1}{2}, \delta]$ -code then  $A$  is a symmetric  $(a, \delta - a)$ -expanding.*

**Proof** Let  $C$  be a  $[\frac{1}{2}, \delta]$  code of block length  $2n$  (in particular the rate of  $C$  is  $n$ ). Let  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{2n}$  be the generating matrix of  $C$ . As the relative rate is  $1/2$  we can assume w.l.o.g. that  $G$  has the following form  $G = \begin{pmatrix} I \\ A \end{pmatrix}$  where  $I$  is the  $n \times n$  identity matrix and  $A$  is an  $n \times n$  matrix. Let  $\vec{0} \neq w \in \{0, 1\}^n$  be a vector of weight  $\leq an$ . Then  $\delta n \leq \text{wt}(Gw) = \text{wt}(w) + \text{wt}(Aw) \leq an + \text{wt}(Aw)$ . In particular  $\text{wt}(Aw) \geq \delta n - an$ . Thus the matrix  $A^t$  is  $(a, \delta - a)$ -expanding.

Let us now assume that  $C^\perp$  is also  $[\frac{1}{2}, \delta]$  code. It is easy to see that the matrix  $H = \begin{pmatrix} A^t \\ I \end{pmatrix}$  is a generating matrix for  $C^\perp$ . As before we get that for every non zero  $w$  of weight  $\leq an$ ,  $\delta n \leq \text{wt}(Hw) = \text{wt}(A^t w) + \text{wt}(w) \leq an + \text{wt}(A^t w)$ . Hence,  $\text{wt}(A^t w) \geq \delta n - an$ . Together with the above observation that  $\text{wt}(Aw) \geq \delta n - an$  we get that  $A$  is symmetric  $(a, \delta - a)$ -expanding.  $\square$

By applying Theorem 13 on the codes obtained from Theorem 3 we get the following corollary.

**Corollary 14** *Let  $\zeta$  be as guaranteed in Theorem 3. Then for every  $0 < a < \zeta$  and every integer  $n$  there is an explicit symmetric  $(a, \zeta - a)$ -expanding transformation of dimension  $n$  that can be constructed in time polynomial in its dimension.*

We now show that using expanding linear transformations we can transform any generator that is  $\epsilon$ -biased w.r.t. heavy tests to an  $\epsilon$ -biased generator.

**Theorem 15** *Let  $A$  be an  $(a, a)$ -expanding transformation of dimension  $n$ . Let  $G^{(h)} : \{0, 1\}^m \rightarrow \{0, 1\}^n$  be an  $\epsilon$ -biased generator against tests of weight  $\geq an$ , for some  $\epsilon > 0$ . That is, for every  $w \in \{0, 1\}^n$  with  $\text{wt}(w) \geq an$ , we have that  $\text{bias}_w(G^{(h)}) \leq \epsilon$ . Define  $G^{(l)} : \{0, 1\}^m \rightarrow \{0, 1\}^n$  in the following way  $G^{(l)}(x) = A(G^{(h)}(x))$ . Then for every  $w \in \{0, 1\}^n$  with  $\text{wt}(w) \leq an$ , we get that  $\text{bias}_w(G^{(l)}) \leq \epsilon$ . In particular, if we define  $G : \{0, 1\}^{2m} \rightarrow \{0, 1\}^n$  as*

$$G(x, y) = G^{(h)}(x) \oplus G^{(l)}(y)$$

*then  $\text{bias}(G) \leq \epsilon$ . The map  $G^{(l)}$  can be constructed in time polynomial in the construction time of  $G^{(h)}$  and of  $A$ , and hence so does  $G$ .*

**Proof** Let  $w \in \{0, 1\}^n$  be such that  $\text{wt}(w) \leq an$ . It is clear that

$$\begin{aligned} \text{bias}_w(G^{(l)}) &= \text{bias}_w(A(G^{(h)}(x))) = \left| \frac{1}{2} - \mathbb{E}_{x \in \{0, 1\}^m} \langle A(G^{(h)}(x)), w \rangle \right| \\ &= \left| \frac{1}{2} - \mathbb{E}_{x \in \{0, 1\}^m} \langle G^{(h)}(x), A^t w \rangle \right| = \text{bias}_{A^t w}(G^{(h)}). \end{aligned}$$

As  $A$  is  $(a, a)$ -expanding we have that  $\text{wt}(A^t w) \geq an$ . By the assumption on  $G^{(h)}$  we get that  $\text{bias}_{A^t w}(G^{(h)}) \leq \epsilon$  and so  $\text{bias}_w(G^{(l)}) \leq \epsilon$ . The claim regarding the bias of  $G$  is an immediate corollary of Lemma 7. The claim regarding the construction time of  $G$  is obvious.  $\square$

We note that as  $A$  is a linear transformation then the degree of  $G$  (i.e. the maximal degree of its output bits when viewed as polynomials, over  $\mathbb{F}_2$ , in the input variables), is at most the degree of  $G^{(h)}$ .

### 3.3 Proof of Theorem 1

The proof of Theorem 1 follows from Theorem 11, Corollary 14 and Theorem 15. Indeed, let  $\zeta$  be as in Theorem 3. Set  $a = \zeta/2$ . Given two integers  $0 < k, m$  and  $\epsilon \geq \exp(-O(\frac{am^{1-\frac{1}{k}}}{2^k k}))$ , let  $n = i_0 \binom{s_0}{k}$ , where  $s_0, i_0$  are as in Theorem 11. By Theorem 3 there exists  $[\frac{1}{2}, 2a]$ -codes of block length  $2n$ . Using the notations of Theorem 11 we bound  $n$  from below by

$$n = i_0 \binom{s_0}{k} = \Omega \left( \left( \frac{\gamma m}{k} \right)^k \right) = \Omega \left( \left( \frac{m}{2^k k \log(1/\epsilon)} \right)^k \right).$$

By Corollary 14 we can construct in polynomial time an  $(a, a)$ -expanding transformation of dimension  $n$ . For our  $k, m, n, a, \epsilon$  let  $G^{(h)}$  be the generator obtained from theorem 11. Clearly  $G^{(h)}$  satisfies the conditions of Theorem 15. Let  $G$  be the generator obtained from Theorem 15. It is easy to verify that  $G$  is the desired generator.  $\square$

## 4 Construction of $\epsilon$ -biased good codes

In this section we prove Theorem 2. We first give a formal statement of the theorem. We shall use the notations of Theorem 4.

**Theorem 16** *Let  $a = \zeta/2$ . Then for any large enough integers  $0 \leq j, n$  where  $n = \frac{1}{2}n_j$  there is a polynomial time constructible generator  $G : \{0, 1\}^m \rightarrow \{0, 1\}^n$ , where  $m \geq an/48$ , such that its image  $C = G(\{0, 1\}^m)$  is a code with the following properties.  $C$  has relative rate  $\geq a/48$ ; relative distance  $\geq a^2/24$ ; a polynomial time decoding algorithm that can fix  $a^2/48$  fraction of errors; and the bias of  $C$  (and hence of  $G$ ) is  $\epsilon = \exp(-O(n))$ .*

The construction of  $\epsilon$ -biased good codes follows the same lines as the construction of low-degree  $\epsilon$ -biased generators. The main difference is that we don't have to keep the degree low but rather make sure that the generator outputs a good code. Thus, we will need a generator for heavy tests that outputs a good code and a way of transforming this generator into a truly unbiased generator that also outputs a good code. The difference from the proof of Theorem 1 is in two points. First we will need a different construction of a generator  $G^{(h)}$  for heavy tests (compare to Theorem 11). Then we will need a construction of a *symmetric* expanding transformation (see Definition 12) that will enable us to transform this generator to an  $\epsilon$ -biased good code (compare to Theorem 15). In particular we need to use the family of codes from Theorem 4 instead of the codes guaranteed by Theorem 3.

As before we start by giving a construction of a good code that is also unbiased w.r.t. heavy tests. We then show a general way of transforming codes that are unbiased w.r.t. heavy tests to  $\epsilon$ -biased codes.

## 4.1 Construction of codes that are unbiased w.r.t. heavy tests

**Theorem 17** *Let  $0 < a < 1$  be a constant. Let  $n$  be an integer. Let  $M$  be an integer satisfying  $an/48 \leq M \leq an/6$  and let  $\hat{C}$  be an  $[M, m, d]$  binary linear error correcting code that has a polynomial time encoding algorithm (i.e. its generating matrix can be computed in polynomial time) and a polynomial time decoding algorithm that can correct  $\hat{\alpha} > 0$  fraction of errors. Denote  $\hat{R} = m/M$  and  $\hat{\delta} = d/M$ . Then there is a map  $G^{(h)} : \{0, 1\}^m \rightarrow \{0, 1\}^n$ , that can be constructed in polynomial time, such that its image  $C = G^{(h)}(\{0, 1\}^m)$  has the following properties.*

- $C$  is an  $[R, \delta]$ -code of block length  $n$  for  $R \geq \frac{a}{48}\hat{R}$  and  $\delta \geq \frac{a}{48}\hat{\delta}$ .
- The weight of every  $v \in C$  is bounded from above by  $\text{wt}(v) \leq an/3$ .
- $C$  has a polynomial time encoding algorithm and a polynomial time decoding algorithm that can correct from  $\alpha \geq \frac{a}{48}\hat{\alpha}$  fraction of errors.
- For every  $w \in \{0, 1\}^n$  such that  $\text{wt}(w) \geq an$  we have that  $\text{bias}_w(C) \leq \exp(-O(n))$ .

**Proof** We start with a sketch of the proof. The proof is similar in nature to the proof of Theorem 11. As before we partition the input string to roughly  $m/k$  subsets of size  $k$ , for some  $k$ . For each of the subsets we assign  $2^{k-2}$  distinct output bits that correspond to the values of  $2^{k-2}$  linearly independent polynomials, in  $k$  variables, evaluated on the input bits that belong to the subset. In this way we get roughly  $2^{k-2}m/k$  output bits. We then concatenate (in the sense of string concatenation) to these output bits the encoding of the input bits w.r.t. the code  $\hat{C}$ . This defines the map  $G^{(h)}$  from  $m$  bits to roughly  $2^{k-2}m/k + M$  bits. It remains to show that  $G^{(h)}$  has the required properties. Indeed, the first  $2^{k-2}m/k$  output bits will assure us that the weight of each output word is not too large and that (as in the proof of Theorem 11) the output has a small bias w.r.t. heavy tests. The concatenation of  $\hat{C}$  ensures that the distance between any two output words is linearly large. The decoding property follows from the decodability of  $\hat{C}$ . As before, in order to be completely accurate we have to handle the case that for our  $k$  (that will be later specified)  $m/k$  is not an integer, and the case where  $2^{k-2}m/k > n - M$ . We now give the formal proof.

Let  $k$  be the smallest integer satisfying  $\lfloor m/k \rfloor 2^{k-2} \geq n - M$ . It is clear that  $k$  is a constant depending only on  $a$  and  $\hat{R}$ . Let  $i_0 = \lfloor m/k \rfloor$ ,  $t_0 = \lfloor (n - M)/i_0 \rfloor$  and  $i_1 = n - M - i_0 \cdot t_0$ . Clearly  $i_1 < i_0$ . Let  $x = (x_1, \dots, x_m)$  be our input. Partition the first  $k \lfloor m/k \rfloor = ki_0$  bits to  $i_0$  sets of size  $k$ . The  $i$ -th set in the partition is  $B_i = \{x_{(i-1)k+1}, \dots, x_{ik}\}$ . For each of the  $B_i$ 's we define  $t_0$  output bits, and for the first  $i_1$  sets we define an additional output bit. This gives a total of  $i_0 t_0 + i_1 = n - M$  output bits. Note that this construction completely ignores the last  $m - k \lfloor m/k \rfloor = m - ki_0$  bits of the input. Denote this set of bits with  $B_0$ .

Denote with  $\{\chi_0, \dots, \chi_{2^{k-2}-1}\}$  the following characteristic functions in  $k$  variables:

$$\chi_j(y_1, \dots, y_k) = 1 \Leftrightarrow (y_1 = 1) \wedge (y_2 = 1) \wedge \left( \sum_{i=3}^k y_i 2^{i-3} = j \right).$$

Equivalently, let  $j_0, \dots, j_{k-3}$  be the binary representation of  $j$  when  $j_0$  is the LSB and  $j_{k-3}$  is the MSB (i.e.  $j = \sum_{i=0}^{k-3} j_i 2^i$ ). We have that

$$\chi_j(y_1, \dots, y_k) = y_1 \cdot y_2 \cdot \prod_{i=3}^k (y_i - j_{i-3} + 1) \pmod{2}. \quad (1)$$

It is easy to see that the  $\chi_j$ -s are linearly independent, and that on every input at most one of the  $\chi_j$ -s is non-zero. Note that the degree of every monomial of  $\chi_j$  is at least 2 and at most  $k$ . We denote with  $G_i$  the set of output bits corresponding to  $B_i$  and with  $g_{i,j} \in G_i$  the output bit corresponding to  $\chi_j$ . Namely,

$$g_{i,j}(x_1, \dots, x_m) = \chi_j(x_{k \cdot (i-1) + 1}, \dots, x_{k \cdot i}).$$

With these notations we have that

$$\forall 1 \leq i \leq i_1 \quad G_i = (g_{i,j})_{j=0, \dots, t_0},$$

$$\forall i_1 < i \leq i_0 \quad G_i = (g_{i,j})_{j=0, \dots, t_0-1}.$$

The length of the output is clearly  $i_1(t_0 + 1) + (i_0 - i_1)t_0 = n - M$ . Denote with  $G_{\hat{C}} : \{0, 1\}^m \rightarrow \{0, 1\}^M$  the generating matrix of the code  $\hat{C}$ . In particular, the encoding of the vector  $x = (x_1, \dots, x_m)$  is  $G_{\hat{C}} \cdot x$ . We now define the map  $G^{(h)}$ :

$$G^{(h)} = (G_1, \dots, G_{i_0}, G_{\hat{C}}),$$

that is, on an input  $x$  we first have  $n - M$  output bits that come from  $G_1, \dots, G_{i_0}$ , and the last  $M$  bits are the encoding of  $x$  w.r.t. to the code  $\hat{C}$ . Clearly the output length is  $n$ . Let  $C = G^{(h)}(\{0, 1\}^m)$  be the image of  $G^{(h)}$ . We show that  $C$  has the required properties.

The rate of  $C$  is  $m$  and thus its relative rate  $R$  is

$$R = m/n \geq m/(48M/a) = \frac{a}{48} \hat{R}.$$

It is clear that  $C$  contains  $\hat{C}$  as its last  $M$  bits and so the minimal distance of  $C$  is at least  $\hat{\delta}M$ . Thus the relative minimal distance of  $C$  is:

$$\delta \geq \hat{\delta}M/n \geq \frac{a}{48} \hat{\delta}.$$

In order to give an upper bound on the weight of every  $v \in C$ , we recall that in every  $G_i$  at most one of the output bits is non-zero. Thus the total weight of  $v \in C$  is bounded from above by  $i_0 + M$ . We get that

$$\forall v \in C, \quad \text{wt}(v) \leq i_0 + M = \lfloor m/k \rfloor + M \leq M/k + M \leq 2M \leq an/3.$$

To show the error correcting property we note that if the total number of errors is  $\hat{\alpha}M$ , then in particular the last  $M$  bits of  $C$  contain at most  $\hat{\alpha}M$  errors. As the last  $M$  bits of  $C$

correspond to a codeword in  $\hat{C}$  we can use the decoding algorithm of  $\hat{C}$  to obtain the original message. Thus we can fix at least  $\hat{\alpha}M \geq \frac{a}{48}\hat{\alpha}n$  errors.

It remains to show that  $C$  is  $\epsilon$ -biased w.r.t. words of weight  $\geq an$ . Indeed let  $\text{wt}(w) \geq an$ . As in the proof of Theorem 11 we partition  $w$  to  $i_0 + 1$  disjoint sets  $w = (w_1, \dots, w_{i_0}, w_{\hat{C}})$ , where the number of bits in  $w_i$  is the same as the number of output bits in  $G_i$ . We also write  $w_i = (w_{i,0}, \dots, w_{i,|G_i|-1})$ . From  $\text{wt}(w) \geq an$  we get that the supports of at least

$$\left\lceil \frac{an - M}{2^{k-2}} \right\rceil \geq \left\lceil \frac{a - a/6}{2^{k-2}} n \right\rceil$$

of  $w_1, \dots, w_{i_0}$  are not empty. As in Theorem 11 we get that

$$\begin{aligned} \langle w, G^{(h)}(x) \rangle &= \left( \bigoplus_{i=1}^{i_0} \langle w_i, G_i(x) \rangle \right) \oplus \langle w_{\hat{C}}, G_{\hat{C}}(x) \rangle \\ &= \left( \bigoplus_{i=1}^{i_0} \left( \bigoplus_{g_{i,j} \in G_i} w_{i,j} \chi_{i,j}(B_i) \right) \right) \oplus \langle w_{\hat{C}}, G_{\hat{C}}(x) \rangle \\ &= \left( \bigoplus_{i=1}^{i_0} p_i(B_i) \right) \oplus \langle w_{\hat{C}}, G_{\hat{C}}(x) \rangle, \end{aligned}$$

where  $p_i(B_i)$  is a polynomial over  $\mathbb{F}_2$  in the variables of  $B_i$ . As  $G_{\hat{C}}(x)$  is a linear function in  $\{x_1, \dots, x_m\}$  we have that

$$\langle w_{\hat{C}}, G_{\hat{C}}(x) \rangle = \bigoplus_{i=1}^{i_0} \ell_i(B_i) \oplus \ell_0(B_0),$$

where the  $\ell_i$ 's are linear functions. We thus have that

$$\langle w, G^{(h)}(x) \rangle = \left( \bigoplus_{i=1}^{i_0} p_i(B_i) \right) \oplus \left( \bigoplus_{i=1}^{i_0} \ell_i(B_i) \oplus \ell_0(B_0) \right) = \left( \bigoplus_{i=1}^{i_0} \tilde{p}_i(B_i) \right) \oplus \ell_0(B_0),$$

where each  $\tilde{p}_i$  is a polynomial over  $\mathbb{F}_2$  in the variables of  $B_i$ . We note that if  $\ell_0(B_0)$  is not a constant linear function, then  $\text{bias}_w(G^{(h)}) = 0$ , and if it is a constant function then it does not affect the bias.

Denote with  $I$  the set of indices for which  $p_i \neq 0$ . As each  $p_i$  is a sum of linearly independent polynomials we have that the size of  $I$  is equal to the number of non empty  $w_i$ -s. Since  $\tilde{p}_i = p_i + \ell_i$  and each monomial of  $p_i$  has degree at least 2 (see discussion after Eq. (1)) we get that if  $p_i \neq 0$  then  $\tilde{p}_i \neq 0$  (because  $p_i$  and  $\ell_i$  cannot cancel each other). We conclude that at least  $\left\lceil \frac{a-a/6}{2^{k-2}} n \right\rceil$  of the  $\tilde{p}_i$ 's are non-zero polynomials (of degree at most  $k$ ). As the sets  $B_i$  are disjoint the polynomials  $\tilde{p}_i(B_i)$  for  $i \in I$ , viewed as random variables in the

input bits, are independent random variables. By the Schwartz-Zippel Lemma (Lemma 10), we get that the bias of each  $\tilde{p}_i$ , for  $i \in I$  is at most  $\frac{1}{2} - \frac{1}{2^k}$ , and so by Lemma 8 we get that

$$\text{bias}_w(G^{(h)}) \leq \frac{1}{2} \left(1 - \frac{2}{2^k}\right)^{|I|} \leq \frac{1}{2} \left(1 - \frac{1}{2^{k-1}}\right)^{\lceil \frac{a-a/6}{2^{k-2}}n \rceil} = \exp\left(-O\left(\frac{a}{2^{2k}}n\right)\right) = \exp(-O(n)).$$

□

## 4.2 From heavy to light: keeping the distance large

For the purpose of constructing  $\epsilon$ -biased codes we shall need the more powerful notion of a symmetric expanding transformation. We shall also require that the transformation has an efficient decoding algorithm (in some special sense).

**Definition 18** *An  $(a, b)$ -expanding transformation  $A$  of dimension  $n$  can decode from  $\alpha$  fraction of errors if there exists a decoding algorithm  $D$  such that for any two vectors  $v, \text{err} \in \{0, 1\}^n$ , satisfying  $\text{wt}(v) \leq an$  and  $\text{wt}(\text{err}) \leq \alpha n$ , we have that  $D(Av + \text{err}) = v$ .*

**Theorem 19** *(Symmetric expanding transformations) Let  $C$  be such that both  $C$  and  $C^\perp$  are  $[\frac{1}{2}, \delta]$  codes of block length  $2n$ . Assume that  $C$  has a polynomial time decoding algorithm that can handle  $\alpha$  fraction of errors. Then, for every  $0 < a < \delta$ , there is an explicit symmetric  $(a, \delta - a)$ -expanding transformation  $A$  of dimension  $n$ , that has a polynomial time algorithm for decoding from  $2\alpha - a$  fraction of errors. Moreover, the constructing time of  $A$  is the same (up to  $\pm O(n^3)$ ) as the time for constructing the generating matrix of  $C$ , and the running time of the decoding algorithm is the same (up to  $\pm O(n)$ ) as the running time of the decoding algorithm of  $C$ .*

**Proof** Let  $G$  be the generating matrix of  $C$ . W.l.o.g we can assume that  $G = \begin{pmatrix} I \\ A \end{pmatrix}$ . The proof of Theorem 13 shows that  $A$  is symmetric  $(a, \delta - a)$ -expanding. Thus, it remains to prove the decoding property. Given a vector of the form  $A(v) + \text{err}$ , where  $\text{wt}(v) \leq an$ , and  $\text{wt}(\text{err}) \leq (2\alpha - a)n$ , consider the  $2n$  dimensional vector  $(\vec{0}, A(v) + \text{err})$ , where  $\vec{0}$  is the  $n$ -th dimensional zero vector. The distance of this vector from the vector  $(v, A(v))$  is at most  $an + (2\alpha - a)n = 2\alpha n$ . As  $(v, Av)$  belongs to the image of  $G$  (and hence belongs to  $C$ ) we can apply the given decoding algorithm for  $C$  (that can correct  $\alpha \cdot 2n$  errors) on the word  $(\vec{0}, A(v) + \text{err})$  to get the word  $(v, Av)$  from which we get  $v$ . The claim regarding the running time is obvious. □

Applying Theorem 19 on the codes obtained from Theorem 4 we get the following corollary (using the notations of Theorem 4).

**Corollary 20** *For every  $0 < a < \zeta$  and every large enough integer  $j$  there is an explicit symmetric  $(a, \zeta - a)$ -expanding transformation  $A$ , of dimension  $n = \frac{1}{2}n_j$ , that can be constructed in polynomial time (in  $n$ ), and that has a polynomial time decoding algorithm that can correct  $\zeta - a$  fraction of errors.*

We now show that by using symmetric expanding transformations we can transform a good code that is  $\epsilon$ -biased w.r.t. heavy tests to a good code that is  $\epsilon$ -biased.

**Theorem 21** *Let  $A$  be a symmetric  $(a, a)$ -expanding transformation of dimension  $n$  that can correct  $\beta$  fraction of errors. Let  $G^{(h)} : \{0, 1\}^m \rightarrow \{0, 1\}^n$  be a mapping whose image is a code  $C \subset \{0, 1\}^n$  with the following properties.*

- $C$  is an  $[R, \delta]$  code of block length  $n$ , for some  $R, \delta > 0$ .
- There exists  $0 < \Delta < a/2$  such that for every word  $v \in C$  we have that  $\text{wt}(v) \leq \Delta n$ .
- There exists  $\alpha > 0$  such that  $C$  has a polynomial time decoding algorithm that can correct  $\alpha$  fraction of errors.
- There exists  $\epsilon > 0$  such that for every  $w \in \{0, 1\}^n$  with  $\text{wt}(w) \geq an$ , we have that  $\text{bias}_w(C) \leq \epsilon$ .

Let  $G^{(l)}(x) = A(G^{(h)}(x))$ . Finally let  $G : \{0, 1\}^{2m} \rightarrow \{0, 1\}^n$  be the following generator  $G(x, y) = G^{(h)}(x) \oplus G^{(l)}(y)$ . Let  $\tilde{C} \subset \{0, 1\}^n$  be the image of  $G$  (in other words,  $\tilde{C} = C \oplus A(C)$ ). Then  $\tilde{C}$  has the following properties.

- $\tilde{C}$  has relative rate  $2R$  and relative minimal distance at least  $\min(\delta, a - 2\Delta)$ .
- $\tilde{C}$  has a polynomial time decoding algorithm that can correct  $\min(\beta - \Delta, \alpha)$  fraction of errors.
- $\text{bias}(\tilde{C}) \leq \epsilon$  (and hence  $\text{bias}(G) \leq \epsilon$ ).

**Proof** We prove the properties of  $\tilde{C}$  one by one. From the definition of  $G$  it is clear that if it is one to one then the relative rate of  $\tilde{C}$  is  $2R$ . We now show the minimal distance property which in particular implies that  $G$  is one to one, and hence the claim about the rate follows. Let  $v, u$  be two different codewords in  $\tilde{C}$ . Then we can write  $v = v_1 + Av_2$  and  $u = u_1 + Au_2$  for  $v_1, v_2, u_1, u_2 \in C$ . It is clear that  $\text{dist}(u, v) = \text{dist}(u_1 + Au_2, v_1 + Av_2) = \text{dist}(u_1 - v_1, A(v_2 - u_2))$ . As  $u_1, v_1, u_2, v_2 \in C$  we have that  $\text{wt}(u_1 - v_1) \leq \text{wt}(u_1) + \text{wt}(v_1) \leq 2\Delta n < an$  and similarly that  $\text{wt}(v_2 - u_2) \leq 2\Delta n < an$ . We analyze two cases.

**Case  $v_2 \neq u_2$ :** As  $A$  is symmetric  $(a, a)$ -expanding we have that  $\text{wt}(A(v_2 - u_2)) \geq an$ . Thus,  $\text{dist}(u_1 - v_1, A(v_2 - u_2)) \geq (a - 2\Delta)n$  (this is a trivial bound on the distance of a vector of weight at most  $2\Delta$  and a vector of weight at least  $a$ ). It follows that  $\text{dist}(u, v) \geq (a - 2\Delta)n$ .

**Case  $v_2 = u_2$ :** In this case  $\text{dist}(u, v) = \text{dist}(u_1, v_1) \geq \delta n$ , as  $C$  has minimal distance  $\geq \delta n$  (note that we must have that  $v_1 \neq u_1$ ).

Combining the two cases we get that the relative distance of  $\tilde{C}$  is at least  $\min(a - 2\Delta, \delta)$ .

We now show a decoding algorithm for  $\tilde{C}$ . Let  $u$  be a codeword in  $\tilde{C}$  and let  $v_1, v_2 \in C$  be such that  $u = v_1 + Av_2$ . Let  $\text{err} \in \{0, 1\}^n$  be an error vector of weight  $\text{wt}(\text{err}) \leq \min(\beta - \Delta, \alpha) \cdot n$ . We now show how to recover  $v_1, v_2$  from the corrupted word  $u + \text{err}$ . Let  $\tilde{\text{err}} = \text{err} + v_1$ . As  $\text{wt}(\text{err}) \leq (\beta - \Delta)n$  and  $\text{wt}(v_1) \leq \Delta n$  we have that  $\text{wt}(\tilde{\text{err}}) \leq \beta n$ . By our assumption  $A$  can decode from  $\beta$  fraction of errors (and  $\text{wt}(v_2) < an$ ), thus  $A$  can

recover the value of  $v_2$  from the input  $Av_2 + \widetilde{err}$ . As  $u + err = Av_2 + (v_1 + err) = Av_2 + \widetilde{err}$ , we can recover  $v_2$  from the input  $u + err$ . Given  $v_2$  and the word  $u + err$  we can get the vector  $v_1 + err$  as  $v_1 + err = u + err - Av_2$ . Since  $\text{wt}(err) \leq \alpha n$  we can use the decoding algorithm of  $C$  to get  $v_1$  from the input  $v_1 + err$ . Clearly the running time of this decoding algorithm is polynomial whenever the running time of the decoding algorithms for  $A$  and  $C$  are polynomial. This shows the decoding property.

Finally we notice that as a direct consequence of Theorem 15 we get that  $G$  is an  $\epsilon$ -biased generator (equivalently, that  $\tilde{C}$  is an  $\epsilon$ -biased set). □

### 4.3 Proof of Theorem 16

In order to prove Theorem 2, we apply Theorem 17 on the codes promised by Theorem 4, to get a good code that is unbiased w.r.t. heavy tests. Then we apply Theorem 21 on this code to obtain a good code that is  $\epsilon$ -biased. Details follow.

Let  $\zeta$  and  $\{n_j\}$  be as guaranteed in Theorem 4. Set  $a = \zeta/2$ . Given a positive integer  $n = \frac{1}{2}n_j$ , let  $i$  be the largest integer such that  $n_i \leq \alpha n/6$ . Let  $m = \frac{1}{2}n_i$ , then by the fact that  $n_{i+1} = 8 \cdot n_i$  we get that  $m > \alpha n/96$ . By Theorem 4 the code  $\hat{C} \triangleq C_{n_i}$  is a  $[\frac{1}{2}, 2a]$ -code of block length  $M = n_i$ . In other words,  $\hat{C}$  is a  $[\frac{1}{2}, 2a]$ -code of rate  $m$  and block length  $2m = M \leq \alpha n/6$ , whose dual code is also a  $[\frac{1}{2}, 2a]$ -code of block length  $M$ . Recall that  $\hat{C}$  has a decoding algorithm that can correct at least  $\hat{\alpha} \triangleq a$  fraction of errors. By applying Theorem 17 on the code  $\hat{C}$  we obtain a generator  $G^{(h)} : \{0, 1\}^m \rightarrow \{0, 1\}^n$  that is (almost) unbiased against tests of weight at least  $\alpha n$  and whose image is a code  $C$  with the following parameters: The relative rate of  $C$  is  $m/n \geq \frac{\alpha}{96}$ ; the relative distance of  $C$  is at least  $\frac{\alpha}{48} \cdot 2a = \frac{a^2}{24}$ ; the weight of every codeword of  $C$  is at most  $\frac{\alpha}{3}n$ ;  $C$  has a decoding algorithm that can fix a fraction  $\frac{a^2}{48}$  of errors; and the bias of  $C$  against words of weight at least  $\alpha n$  is at most  $\epsilon = \exp(-O(n))$ .

Now we apply Theorem 21 on the generator  $G^{(h)}$ , where  $A$  be the symmetric  $(a, a)$ -expanding transformation of dimension  $n$  guaranteed by Corollary 20 (by the argument above the relevant parameters are:  $R \geq \alpha/96$ ,  $\delta \geq a^2/24$ ,  $\alpha = a^2/48$ ,  $\beta = a$  and  $\Delta = a/3$ ). We obtain a generator  $G : \{0, 1\}^{2m} \rightarrow \{0, 1\}^n$  whose image is a code  $\tilde{C}$  with the following parameters. The relative rate of  $\tilde{C}$  is  $2m/n \geq \frac{\alpha}{48}$ ; the relative minimal distance of  $\tilde{C}$  is at least  $\min(a^2/24, a/3) = a^2/24$ ;  $\tilde{C}$  has a decoding algorithm that can fix  $\min(2a/3, a^2/48) = a^2/48$  fraction of errors; and the bias of  $\tilde{C}$  (and hence of  $G$ ) is at most  $\epsilon = \exp(-O(n))$ . As the constructions of  $A$  and of  $\hat{C}$ , run in polynomial time, we get that  $C$  and hence  $\tilde{C}$  have polynomial time encoding. Similarly, we get that as  $A$  and  $\hat{C}$  have decoding algorithms that run in polynomial time then so does  $C$  and  $\tilde{C}$ . This completes the proof of Theorem 2. □

## Acknowledgements

I would like to thank Adam Smith for introducing me to the problem of constructing  $\epsilon$ -biased good error correcting codes and for many valuable discussions. Thanks also to Oded Goldreich, Sofya Raskhodnikova and Avi Wigderson for helpful discussions and to Vinay Deolalikar, Mitsuru Hamada, Simon Litsyn, Ryutaroh Matsumoto, Oded Regev, Ronny Roth, Madhu Sudan and Sergey Yekhanin for helpful information regarding algebraic geometry error correcting codes. Thanks to Oded Goldreich for his comments and remarks. A special thank you to Venkat Guruswami for agreeing to write Appendix A and for many useful comments that improved the presentation of the results (as well as pointing an inaccuracy in an earlier version of the paper - Theorem 3 of [Shp06] was not true as stated).

## References

- [AGHP92] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple construction of almost  $k$ -wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992.
- [AIK<sup>+</sup>90] M. Ajtai, H. Iwaniec, J. Komlos, J. Pintz, and E. Szemerédi. Construction of a thin set with small fourier coefficients. *Bulletin of the London Mathematical Society*, 22:583–590, 1990.
- [AIK06] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in  $NC^0$ . *SIAM Journal on Computing*, 36(4):845–888, 2006.
- [AM95] N. Alon and Y. Mansour. epsilon-discrepancy sets and their application for interpolation of sparse polynomials. *Information Processing Letters*, 54(6):337–342, 1995.
- [AR94] N. Alon and Y. Roichman. Random cayley graphs and expanders. *Random Structures and Algorithms*, 5(2):271–285, 1994.
- [BFLS91] L. Babai, L. Fortnow, L. A. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing*, pages 21–31, 1991.
- [BSSVW03] E. Ben-Sasson, M. Sudan, S. Vadhan, and A. Wigderson. Randomness-efficient low degree tests and short PCPs via epsilon-biased sets. In *Proceedings of the 35th STOC*, pages 612–621, 2003.
- [CM01] M. Cryan and P. B. Miltersen. On pseudorandom generators in  $NC^0$ . In *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science*, pages 272–284, 2001.

- [DS05] Y. Dodis and A. Smith. Correcting errors without leaking partial information. In *Proceedings of the 37th Annual STOC*, pages 654–663, 2005.
- [EGL<sup>+</sup>98] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Velickovic. Efficient approximation of product distributions. *Random Structures and Algorithms*, 13(1):1–16, 1998.
- [Fei95] J. Feigenbaum. The use of coding theory in computational complexity. In *Different Aspects of Coding Theory, Proceedings of Symposia on Applied Mathematics*, pages 207–233, 1995.
- [FGL<sup>+</sup>96] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *JACM*, 43(2):268–292, 1996.
- [For66] G. D. Forney. *Concatenated Codes*. M.I.T. Press, 1966.
- [FR93] G. L. Feng and T. R. N. Rao. Decoding algebraic-geometric codes up to the designed minimum distance. *IEEE Transactions on Information Theory*, 39(1):37–45, 1993.
- [Gol00a] O. Goldreich. Candidate one-way functions based on expander graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(90), 2000.
- [Gol00b] O. Goldreich. Short locally testable codes and proofs (survey). *Electronic Colloquium on Computational Complexity (ECCC)*, 5(14), 2000.
- [GP08] V. Guruswami and A. C. Patthak. Correlated Algebraic-Geometric codes: Improved list decoding over bounded alphabets. *Mathematics of Computation*, 77(261):447–473, January 2008.
- [GS92] P. Gemmell and M. Sudan. Highly resilient correctors for multivariate polynomials. *Information Processing Letters*, 43(4):169–174, 1992.
- [GS96] A. Garcia and H. Stichtenoth. On the asymptotic behavior of some towers of function fields over finite fields. *Journal of Number Theory*, 61(2):248–273, 1996.
- [GS01] V. Guruswami and M. Sudan. On representations of algebraic-geometry codes. *IEEE Transactions on Information Theory*, 47(4):1610–1613, 2001.
- [GS02] V. Guruswami and M. Sudan. Decoding concatenated codes using soft information. In *IEEE Conference on Computational Complexity*, pages 148–157, 2002.
- [Gur01] V. Guruswami. *List decoding of error correcting codes*. PhD thesis, MIT, 2001.

- [Hås87] J. Håstad. One-way permutations in  $NC^0$ . *Information Processing Letters*, 26(3):153–155, 1987.
- [HPS93] J. Håstad, S. Phillips, and S. Safra. A well-characterized approximation problem. *Information Processing Letters*, 47(6):301–305, 1993.
- [HvLP98] T. Hoholdt, J. H. van Lint, and R. Pellikaan. *Handbook of Coding Theory*, volume 1, chapter Algebraic Geometry Codes, pages 871–961. Elsevier, 1998.
- [KL01] M. Krause and S. Lucks. Pseudorandom functions in  $TC^0$  and cryptographic limitations to proving lower bounds. *Computational Complexity*, 10(4):297–313, 2001.
- [KSV02] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, 2002.
- [MNN94] R. Motwani, J. Naor, and M. Naor. The probabilistic method yields deterministic parallel algorithms. *JCSS*, 49(3):478–516, 1994.
- [MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes, Part II*. North-Holland, 1977.
- [MST06] E. Mossel, A. Shpilka, and L. Trevisan. On epsilon-biased generators in  $NC^0$ . *Random Structures and Algorithms*, 29(1):56–81, 2006.
- [MW04] R. Meshulam and A. Wigderson. Expanders in group algebras. *Combinatorica*, 24(4):659–680, 2004.
- [Nao92] M. Naor. Constructing Ramsey graphs from small probability spaces. Technical report, IBM Research Report RJ 8810, 1992.
- [NC00] M. Nielsen and I. Chuang. *Quantum computation and quantum information*. Cambridge, 2000.
- [NN93] J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. on Computing*, 22(4):838–856, 1993.
- [Raz05] R. Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual STOC*, pages 11–20, 2005.
- [RSW93] A. A. Razborov, E. Szemerédi, and A. Wigderson. Constructing small sets that are uniform in arithmetic progressions. *Combinatorics, Probability & Computing*, 2:513–518, 1993.
- [SAK<sup>+</sup>01] K. Shum, I. Aleshnikov, P. Vijay Kumar, H. Stichtenoth, and V. Deolalikar. A low-complexity algorithm for the construction of algebraic-geometric codes better than the Gilbert-Varshamov bound. *IEEE Transactions on Information Theory*, 47(6):2225–2241, 2001.

- [Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *JACM*, 27(4):701–717, 1980.
- [Shp06] A. Shpilka. Constructions of low-degree and error-correcting in-biased generators. In *21st Annual IEEE Conference on Computational Complexity*, pages 33–45, 2006.
- [Sti93] H. Stichtenoth. *Algebraic Function Fields and Codes*. Universitext, Springer-Verlag, Berlin, 1993.
- [SV90] A. N. Skorobogatov and S. G. Vladut. On the decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 36(5):1051–1060, 1990.
- [Tre04] L. Trevisan. Some applications of coding theory in computational complexity. *Quaderni di Matematica*, (13):347–424, 2004.
- [WB86] L. R. Welch and E. R. Berlekamp. Error correction of algebraic block codes. *US Patent Number 4,633,470*, December 1986.
- [Zip79] R. Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and algebraic computation*, pages 216–226. 1979.

## A A note on explicit unique-decodable algebraic-geometric codes

This appendix was written by Venkatesan Guruswami<sup>5</sup> and it contains a proof for Theorem 4. The claim in the theorem seems to be a known fact to experts in coding theory, however, we did not find a complete proof of it in the literature and so we give one here.

Our goal is to demonstrate an explicit (i.e., deterministic polynomial time constructible) family of asymptotically good binary codes of rate  $1/2$  that can be decoded from a constant fraction of errors in deterministic polynomial time and whose dual is also asymptotically good. The construction is based on algebraic-geometric codes defined over an asymptotically good tower of function fields. This yields codes over a constant-sized extension field of  $\mathbb{F}_2$ , which can then be converted to binary codes by expressing elements of the extension field in a self-dual basis.

We begin with a description of a deterministic decoding algorithm to correct  $(d^* - g - 1)/2$  errors in an algebraic-geometric codes of designed distance  $d^*$  and genus  $g$ . It is a straightforward generalization of the Gemell-Sudan [GS92] description of the Welch-Berlekamp decoder [WB86] for Reed-Solomon codes.

We do not attempt to define all algebro-geometric notions we will use, and instead point the reader to the excellent text by Stichtenoth [Sti93] for background on function fields and

---

<sup>5</sup>Department of Computer Science and Engineering, University of Washington, Seattle, WA 98185. Email: `venkat@cs.washington.edu`.

algebraic-geometric codes. Let  $K/\mathbb{F}_q(X)$  be an algebraic function field, and let  $P_1, P_2, \dots, P_n$  and  $Q$  be  $n+1$  rational places of  $K$ . (Geometrically, these correspond to points on the curve all of whose coordinates belong to  $\mathbb{F}_q$ .) For a positive integer  $\alpha$ , let  $L(\alpha Q)$  be the  $\mathbb{F}_q$ -linear space of all functions in  $K$  that have at most  $\alpha$  poles at  $Q$ , and no poles elsewhere. By the Riemann-Roch theorem, the dimension of  $L(\alpha Q)$  is at least  $\alpha - g + 1$  where  $g \geq 0$  is the genus of  $K$ . Define  $D = \{P_1, \dots, P_n\}$ . Consider the code defined by

$$C(D, \alpha Q) = \{ \langle f(P_1), f(P_2), \dots, f(P_n) \rangle : f \in L(\alpha Q) \} ,$$

whose codewords correspond to evaluations of functions in  $L(\alpha Q)$  at the rational places  $P_1, \dots, P_n$ . The following well-known result summarizes the properties of the above code.

**Lemma 22** *The code  $C = C(D, \alpha Q)$  is an  $\mathbb{F}_q$ -linear  $[n, k, d]$  code with  $k \geq \alpha - g + 1$  and  $d \geq d^* = n - \alpha$  ( $d^*$  is called the designed distance of the code). If  $\alpha > 2g - 2$ , then we have  $k = \alpha - g + 1$ . The dual code  $C^\perp$  has minimum distance at least  $\alpha - (2g - 2)$ .*

Note that the generator matrix of  $C(D, \alpha Q)$  consists of the evaluations of a basis  $\phi_1, \dots, \phi_k$  of the space  $L(\alpha Q)$  at the places  $P_1, \dots, P_n$ . We now show that if we know the evaluations of the basis of a slightly large space compared to  $L(\alpha Q)$  then this suffices to decode the code up to about  $(d^* - g)/2$  errors. For convenience we assume  $\alpha > 2g - 2$  in what follows, so that the dimension of  $L(\alpha Q)$  is exactly  $k = \alpha - g + 1$ . We will also assume, again for convenience, that  $\alpha < n - 3g$ .

**Theorem 23** *For  $\ell = \lceil \frac{n+\alpha}{2} \rceil + g$ , let  $\mathcal{B} = \{\phi_1, \phi_2, \dots, \phi_\ell\}$  be a basis of  $L(\ell Q)$  with increasing pole orders at  $Q$  (this implies that  $\phi_1, \dots, \phi_k$  form a basis of  $L(\alpha Q)$ ). There is a deterministic polynomial time algorithm that given the evaluations of  $\mathcal{B}$  at  $P_1, \dots, P_n$ , and input  $y \in \mathbb{F}_q^n$ , determines the unique codeword of  $C(D, \alpha Q)$  within Hamming distance  $\lfloor (d^* - g - 1)/2 \rfloor$  if one exists.*

**Proof** Let  $e^* = \lfloor \frac{d^* - g - 1}{2} \rfloor$ . Suppose that  $f \in L(\alpha Q)$  is such that  $f(P_i) \neq y_i$  for  $e \leq e^*$  values of  $i \in \{1, 2, \dots, n\}$ . Clearly such a  $f$ , if one exists, must be unique, since the distance of the  $C(D, \alpha Q)$  is at least  $d^*$ . Let  $E = \{i : f(P_i) \neq y_i\}$  denote the set of error locations. By the Riemann-Roch theorem, there exists a non-zero  $\xi \in L((e+g)Q)$  such that  $\xi(P_i) = 0$  for every  $i \in E$  (this follows by considering the dimension of  $L((e+g)Q - \sum_{i \in E} P_i)$ , which is at least 1 by Riemann-Roch). Define  $\psi = f\xi$ . Clearly  $\psi(P_i) - y_i\xi(P_i) = 0$  for every  $i$ . By definition  $\psi \in L((\alpha + e + g)Q)$ . It follows that there exist non zero  $\tilde{\psi} \in L((\alpha + g + e^*)Q)$  and  $\tilde{\xi} \in L((e^* + g)Q)$  such that

$$\tilde{\psi}(P_i) - y_i\tilde{\xi}(P_i) = 0 \text{ for } i = 1, 2, \dots, n . \quad (2)$$

By our assumption that  $\alpha < n - 3g$ ,  $e^* + g > 2g - 2$ , so  $\dim(L((e^* + g)Q)) = e^* + 1$ , and  $\dim(L((\alpha + g + e^*)Q)) = \alpha + e^* + 1$ . One can therefore find  $\tilde{\psi}, \tilde{\xi}$  satisfying above by writing  $\tilde{\psi} = \sum_{i=1}^{\alpha+e^*+1} a_i\phi_i$  and  $\tilde{\xi} = \sum_{j=1}^{e^*+1} b_j\phi_j$  and solving the homogeneous linear system given in Eq. (2) for the unknowns  $a_i, b_j$  in  $O(n^3)$  time. (We can set up such a system over  $\mathbb{F}_q$  since we know the evaluations of the basis functions  $\phi_i$  at the places  $P_1, \dots, P_n$ .) Note that we can find a non-zero solution because we know that one exists.

We will now prove that  $\tilde{\psi}/\tilde{\xi} = f$ . Indeed, consider  $\nu = \tilde{\psi} - f\tilde{\xi}$ . For every  $i \notin E$ ,  $\nu(P_i) = \tilde{\psi}(P_i) - f(P_i)\tilde{\xi}(P_i) = \tilde{\psi}(P_i) - y_i\tilde{\xi}(P_i) = 0$ . Thus  $\nu$  has at least  $n - e \geq n - e^*$  zeroes. On the other hand,  $\nu \in L((\alpha + e^* + g)Q)$  and thus has at most  $(\alpha + e^* + g)$  poles. It follows that if  $n - e^* > \alpha + e^* + g$ , or in other words if  $2e^* < d^* - g$ , then  $\nu = 0$ , and thus  $f = \tilde{\psi}/\tilde{\xi}$ .

One can thus compute  $f(P_i)$  for  $i = 1, 2, \dots, n$  and thus the codeword of  $C(D, \alpha Q)$  encoding  $f$ . We can also find the message  $f$  by interpolation: write  $f = \sum_{i=1}^k c_i \phi_i \in L(\alpha Q)$  and compute the coefficients  $c_i$  by solving a linear system.  $\square$

**Theorem 24** *For every prime power  $r \geq 8$ , and every integer  $m \geq 1$ , there is a rate  $1/2$  linear code over  $\mathbb{F}_{r^2}$  of block length  $n_m = r^{m+1}(r-1)$  with relative distance at least  $\frac{1}{2} - \frac{1}{r-1}$ . Moreover a representation of this code can be constructed in  $\text{poly}(n_m)$  time that enables polynomial time encoding and deterministic polynomial time decoding from a fraction  $1/10$  of errors. Furthermore, the dual of this linear code has relative distance at least  $1/3$ .*

**Proof** For every prime power  $r$ , Garcia and Stichtenoth [GS96] present an explicit tower of function fields over  $\mathbb{F}_{r^2}$ , defined by a sequence of Artin-Schreier extensions of degree  $r$ , which exhibits the optimal trade-off between genus and number of rational places (meeting the so-called Drinfeld-Vlăduț bound). For every positive integer  $m$ , they construct a function field  $F_m/\mathbb{F}_{r^2}(X_0)$  with at least  $n_m + 1 = (r^2 - r)r^m + 1$  rational places, including a fully ramified place  $P_\infty^{(m)}$  above “infinity” (namely, the pole of  $X_0$  in  $\mathbb{F}_{r^2}(X_0)$ ) that can serve the role of  $Q$  to define the algebraic-geometric code, with the remaining  $n_m$  rational places serving as evaluation points  $P_1, P_2, \dots, P_{n_m}$  of the code. The genus of  $F_m$  is at most  $r^{m+1}$ . Shum *et al* [SAK<sup>+</sup>01] present an algorithm to compute in time polynomial in  $n_m, \ell$ , a basis of  $L(\ell P_\infty^{(m)})$  together with evaluations of the basis functions at the rational places  $P_1, P_2, \dots, P_{n_m}$ . (The reader can also find a synopsis of this construction in [GP08, Sec. 7].)

For the choice  $\alpha_m = n_m/2 + g - 1$  and  $D = \{P_1, \dots, P_{n_m}\}$ , the code  $C_m = C(D, \alpha_m P_\infty^{(m)})$  has dimension  $n_m/2$ , and relative distance at least  $(n_m - \alpha_m)/n_m > 1/2 - g/n_m \geq \frac{1}{2} - \frac{1}{r-1}$ . Since we know how to compute the requisite basis for the spaces  $L(\ell P_\infty^{(m)})$  efficiently by [SAK<sup>+</sup>01], Theorem 23 implies an efficient unique decoding algorithm for the code  $C_m$  that can correct  $\lfloor n_m/4 \rfloor - g \geq n_m/4 - n_m/(r-1) - 1$  errors. For  $r \geq 8$ , this exceeds a fraction  $1/10$  of errors.

By Lemma 22, the dual of the code has distance at least  $\alpha - (2g - 2) > n_m/2 - g \geq n_m \left( \frac{1}{2} - \frac{1}{r-1} \right) \geq n_m/3$ .  $\square$

Theorem 24 guarantees the sort of code we are after, except for our requirement of binary codes. This is now easily achieved, by picking  $r$  to a power of 2, and expressing elements in  $\mathbb{F}_{r^2}$  as  $2 \log_2 r$  bit strings with respect to a *self-dual basis*. For a prime  $p$ , a self-dual basis of  $\mathbb{F}_{p^s}$  over  $\mathbb{F}_p$  is a basis  $\{\beta_1, \beta_2, \dots, \beta_s\}$  of  $\mathbb{F}_{p^s}$  as a vector space over  $\mathbb{F}_p$  such that  $\text{Tr}(\beta_i \beta_j) = 0$  for  $1 \leq i < j \leq s$ , and  $\text{Tr}(\beta_i^2) = 1$  for  $1 \leq i \leq s$ . (Here  $\text{Tr} : \mathbb{F}_{p^s} \rightarrow \mathbb{F}_p$  is the Trace map:  $\text{Tr}(x) = x + x^p + x^{p^2} + \dots + x^{p^{s-1}}$ .) It is known that a self-dual basis exists if  $p$  is even; we will use  $p = 2$ . The following lemma asserts that using a self-dual basis to express elements of  $\mathbb{F}_{p^s}$  over  $\mathbb{F}_p$  respects taking duals of codes. The proof is standard and omitted.

**Lemma 25** *Let  $C \subseteq \mathbb{F}_{2^s}^n$  be a linear code. Define  $\text{SD}(C) \subseteq \mathbb{F}_2^{ns}$  to be the binary linear code obtained by expressing the symbols from  $\mathbb{F}_{2^s}$  in codewords of  $C$  as elements of  $\mathbb{F}_2^s$  corresponding to the coefficients of that symbol with respect to a fixed self-dual basis of  $\mathbb{F}_{2^s}$  over  $\mathbb{F}_2$ . Then*

$$\text{SD}(C^\perp) = \left(\text{SD}(C)\right)^\perp.$$

The following lemma is obvious:

**Lemma 26** *If  $C \subseteq \mathbb{F}_{2^s}^n$  is a linear code and  $\text{SD}(C) \subseteq \mathbb{F}_2^{ns}$  the associated binary code defined above, then the rate of  $\text{SD}(C)$  is the same as that of  $C$ , and the relative distance of  $\text{SD}(C)$  is at least  $1/s$  times the relative distance of  $C$ . Also, an algorithm to correct a fraction  $\gamma$  of errors for  $C$  implies an algorithm to correct a fraction  $\gamma/s$  of errors for  $\text{SD}(C)$ .*

Using the above two lemmas, we can convert the family of codes guaranteed by Theorem 24 over the field  $\mathbb{F}_{64}$  into a family of polynomial time constructible rate  $1/2$  binary linear codes decodable in deterministic polynomial time from a fraction  $1/60$  of errors and whose duals have relative distance at least  $1/18$ . This gives us the binary linear codes we were after.