Matching Vector Codes [*]

Zeev Dvir[†]
Princeton University
zeev.dvir@gmail.com

Parikshit Gopalan
MSR SVC
parik@microsoft.com

Sergey Yekhanin
MSR SVC
yekhanin@microsoft.com

Abstract

An (r, δ, ϵ)-locally decodable code encodes a k-bit message x to an N-bit codeword $C(x)$, such that for every $i \in [k]$, the i-th message bit can be recovered with probability $1 - \epsilon$, by a randomized decoding procedure that queries only r bits, even if the codeword $C(x)$ is corrupted in up to δN locations.

Recently a new class of locally decodable codes, based on families of vectors with restricted dot products has been discovered. We refer to those codes as Matching Vector (MV) codes. Several families of $(r, \delta, \Theta(r\delta))$-locally decodable MV codes have been obtained. While codes in those families were shorter than codes of earlier generations, they suffered from having large values of $\epsilon = \Omega(r\delta)$, which meant that r-query MV codes could only handle error-rates below $\frac{1}{r}$. Thus larger query complexity gave shorter length codes but at the price of less error-tolerance. No MV codes of super-constant number of queries capable of tolerating a constant fraction of errors were known to exist.

In this paper we present a new view of matching vector codes and uncover certain similarities between MV codes and classical Reed Muller codes. Our view allows us to obtain deeper insights into the power and limitations of MV codes. Specifically,

1. We show that existing families of MV codes can be enhanced to tolerate a large constant fraction of errors, independent of the number of queries. Such enhancement comes at a price of a moderate increase in the number of queries;
2. Our construction yields the first families of matching vector codes of super-constant query complexity that can tolerate a constant fraction of errors. Our codes are shorter than Reed Muller LDCs for all values of $r \leq \log k/(\log\log k)^c$, for some constant c;
3. We show that any MV code encodes messages of length k to codewords of length at least $k2^{\Omega(\sqrt{\log k})}$. Therefore MV codes do not improve upon Reed Muller LDCs for $r \geq (\log k)^{\Omega(\sqrt{\log k})}$.

[*] Earlier versions of this work have appeared as an ECCC report [Gop09], and a FOCS'2010 paper [DGY10].

[†] Research partially supported by NSF grants CCF-0832797 and DMS-0835373.

1 Introduction

Classical error-correcting codes allow one to encode a k-bit message x into an N-bit codeword $C(x)$, in such a way that x can still be recovered even if $C(x)$ gets corrupted in a number of coordinates. The disadvantage of classical error-correction is that one needs to read all or most of the (corrupted) codeword to recover any information about x. Suppose that one is only interested in recovering one or a few bits of x. In this case, more efficient schemes are possible allowing one to read only a small number of code positions. Such schemes are known as Locally Decodable Codes (LDCs). Locally decodable codes allow reconstruction of an arbitrary bit x_i, from looking only at $r \ll N$ randomly chosen coordinates of $C(x)$. While initial applications of locally decodable codes have been to data transmission and storage, they have found applications in other areas such as complexity theory and cryptography. See the survey [Yek11] for more details. Below is a slightly informal definition of LDCs:

An (r, δ, ϵ)-locally decodable code encodes k-bit messages x to N-bit codewords $C(x)$, such that for every $i \in [k]$, the bit x_i can be recovered with probability $1 - \epsilon$, by a randomized decoding procedure that makes only r queries, even if the codeword $C(x)$ is corrupted in up to δN locations.

We would like to have LDCs that have small values of r, N and ϵ and a large value of δ. However typically the parameters are not regarded as equally important. In applications of LDCs to data transmission and storage one wants δ to be a large constant (ideally close to $1/4$), and the codeword length N to be small. At the same time the exact number of queries r is not very important provided that it is much smaller than k. Similarly the exact value of $\epsilon < 1/2$ is not important since one can easily amplify ϵ to be close to 0, by running the decoding procedure few times and taking a majority vote. In applications of LDCs in cryptography one thinks of $\delta > 0$ and $\epsilon < 1/2$ as constants whose values are of low significance and focuses on the trade-off between r and N, with emphasis on very small values of r such as $r = 3$ or $r = 4$.

1.1 Three generations of locally decodable codes

The notion of locally decodable codes was explicitly discussed in various places in the early 1990s, most notably in [BFLS91, Sud92, PS94]. Katz and Trevisan [KT00] were the first to provide a formal definition of LDCs (see also [STV99]) and prove lower bounds on their length. Their bounds were improved in [GKST02, KdW04] where a tight (exponential) lower bound for the length of 2-query LDCs was obtained. Further lower bounds on the length of LDCs were obtained in [WdW05, Woo07]. The best lower bounds for the length of r-query LDCs currently have the form $\tilde{\Omega}\left(n^{1+1/(\lceil r/2 \rceil - 1)}\right)$ [Woo07]. They are very far form matching the best upper bounds.

One can informally classify the known families of locally decodable codes into three generations based on the technical ideas that underlie them. The first generation captures codes based on the idea of (low-degree) multivariate polynomial interpolation. All such codes [BFLS91, KT00, BIK05, CGKS98] are (directly or indirectly) based on classical (generalized) Reed Muller (RM) codes [MS77]. The code consists of evaluations of low degree polynomials in $\mathbb{F}_q[z_1, \ldots, z_n]$, at all points of $\mathbb{F}_q^n$, for some finite field $\mathbb{F}_q$. The decoder recovers the value of the unknown polynomial at a point by shooting a line in a random direction and decoding along it using noisy polynomial interpolation [BF90, Lip90, STV99]. The method behind these constructions is very general. It yields locally decodable codes of all possible query complexities, (i.e., one can choose r to be an arbitrary non-decreasing function of k) that tolerate a constant fraction of errors. (We say that an r-query code C *tolerates* δ fraction of errors if C is (r, δ, ϵ)-locally decodable for some $\epsilon < 1/2$.)

The second generation of LDCs [BIKR02] (and later [WY05]) combined the earlier ideas of polynomial interpolation with a clever use of recursion to show that Reed-Muller type codes are not the shortest possible for constant values of query complexity $r \geq 3$. Codes of the second generation are

$(r, \delta, \Theta(r\delta))$-locally decodable. Thus the fraction of noise handled by these codes decays linearly with r. No LDCs of the second generation with $r = \omega(1)$ and $\delta = \Omega(1)$ are known to exist.

The latest (third) generation of LDCs was initiated in [Yek08] and developed further in [Rag07, KY09, Efr09, IS10, MFL$^+$10]. New codes are obtained through an argument involving a mixture of combinatorial and algebraic ideas, where the key ingredient is a design of a large family of low dimensional (matching) vectors with constrained dot products. Recently an important progress in constructions of LDCs of the third generation has been accomplished in [Efr09] where the first constructions of codes from matching vectors modulo composites (rather than primes) were considered. In what follows we refer to LDCs of the third generation as Matching Vector (MV) codes.

To date several families of $(r, \delta, \Theta(r\delta))$-locally decodable MV codes have been obtained. While codes in those families were dramatically shorter than codes of earlier generations, similarly to codes of [BIKR02, WY05] they suffered from having large values of $\epsilon = \Omega(r\delta)$. Thus as the number of queries increased, the length N became smaller as a function of k, but at the price of a reduction in the error-rate that the code could handle. Codes with constant query complexity could only tolerate tiny amounts of error, and no MV codes with $r = \omega(1)$ capable of tolerating a constant fraction of errors were known to exist.

The reason that previous constructions all gave $\epsilon = \Omega(r\delta)$ lay in the reliance on the *smoothness* of the decoder to prove its correctness. The proofs proceeded by showing that each of the r queries made by the decoder is smooth, meaning that it is distributed (close to) uniformly over the bits of the codeword. By the union bound, if a δ fraction is corrupted, then we are unlikely to query any of these locations. This argument clearly will not work once the error rate exceeds $1/r$. Indeed a recent result [GM11] shows that for 3-query LDCs, correcting more than $1/3$ fraction of errors requires exponential length.

1.2 Our results

In this work we develop a new view of matching vector codes and uncover certain similarities between MV codes and classical Reed Muller codes. Our view allows us to obtain a deeper insight into the power and limitations of MV codes.

1. We show that existing families of MV codes can be enhanced to tolerate a nearly $1/8$ fraction of errors, independent of the value of r, at a price of a moderate increase in the number of queries[1]. Specifically, for every constant $t \geq 2$, we obtain a family of binary $\left(t^{O(t)}, \delta, 4\delta(1 + O(1/\ln t))\right)$-locally decodable codes of length essentially identical to the length of $\left(2^{O(t)}, \delta, 2^{O(t)}\delta\right)$-locally decodable codes of [Efr09]. These codes encode messages of length k into codewords of length $\exp\exp\left((\log k)^{1/t}(\log\log k)^{1-1/t}\right)$.

2. We obtain the first families of (binary) matching vector codes of super-constant query complexity that can tolerate a constant fraction of errors, close to $1/8$. Our codes are shorter than Reed Muller LDCs for all values of $r \leq \log k/(\log\log k)^c$, for some constant c.

3. The parameters of a MV code are determined by the parameters of the underlying family of matching vectors. We obtain new upper and lower bounds on the parameters of such families and conclude that any MV code encodes messages of length k to codewords of length at least $k2^{\Omega(\sqrt{\log k})}$. Therefore MV codes do not improve upon Reed Muller locally decodable codes for $r \geq (\log k)^{\Omega(\sqrt{\log k})}$.

[1] It is interesting to contrast our work with the work of Woodruff [Woo08] who obtained a non-linear transformation that (in certain circumstances) allows one to reduce LDC codeword length at a price of a *loss* in the value of δ.

1.3 Our techniques

Our constructions are centered around a new view of MV codes that flashes out some intrinsic similarities between MV codes and RM codes. In our view a MV code consists of a linear subspace of polynomials in $\mathbb{F}_q[z_1, \ldots, z_n]$, evaluated at all points of $\mathbb{C}_m^n$, where $\mathbb{C}_m$ is a certain multiplicative subgroup of $\mathbb{F}_q^*$. The decoding algorithm is similar to traditional local decoders for RM codes. The decoder shoots a line in a certain direction and decodes along it. The difference is that the monomials which are used are not of low-degree, they are chosen according to a matching family of vectors. (Two collections of vectors $\mathcal{U}, \mathcal{V} \subseteq \mathbb{Z}_m^n$ form a matching family if for every $\mathbf{u}_i \in \mathcal{U}$ there is a unique $\mathbf{v}_i \in \mathcal{V}$ such that $(\mathbf{u}_i, \mathbf{v}_i) = 0$, while other dot products $(\mathbf{u}_j, \mathbf{v}_i)$ belong to a small set $S \subseteq \mathbb{Z}_m \setminus \{0\}$.) Further, the lines for decoding are *multiplicative*, a notion that we will define shortly.

Constructions of locally decodable codes from matching vectors have previously been considered in [Yek08, Rag07, Efr09, IS10]. In this work we show that if the family of matching vectors underlying the MV code is *bounded* (meaning that dot products between all vectors $\mathbf{u} \in \mathcal{U}$ and $\mathbf{v} \in \mathcal{V}$ are small in $\mathbb{Z}_m$ with respect to the natural total ordering); then the restriction of a codeword of the MV code to a multiplicative line yields an evaluation of a low degree polynomial. Therefore one can apply existing techniques for noisy polynomial interpolation in the decoding process and tolerate a large fraction of errors. We show how the currently best known families of matching vectors (due to Grolmusz [Gro00]) can be turned into bounded families. We also give a simple construction of bounded families of matching vectors.

We also initiate a systematic study of families of matching vectors and prove upper bounds on their sizes. For the case when m is a prime, our bounds are obtained via classical results on the expansion of incidence graphs of points and hyperplanes in finite geometries. Our bounds for composites are obtained via reductions to the prime case. These bounds in turn imply that any matching vector code must stretch messages of length k to codewords of length $k2^{\Omega(\sqrt{\log k})}$ for large enough k, regardless of the query complexity.

1.4 Concurrent work

Some of our ideas were independently rediscovered by Ben-Aroya et al. [BET10a]. See [BET10a, section Related work] for an accurate account of the relation between the papers. Some results of Ben-Aroya et al. are stronger than ours. Specifically, we show how one can locally decode binary MV codes from nearly 1/8 fraction of errors (1/4 for codes over large alphabets). Ben-Aroya et. al [BET10a] show how one can decode binary MV codes from nearly 1/4 fraction of errors (1/2 for codes over large alphabets). They also consider local list decoding of MV codes.

The idea behind the improved unique-decoder of [BET10a] is that if we take a sufficiently large (but constant) number of multiplicative lines then the average agreement with the codeword along a line is likely to exceed 1/2. We run our decoder (proposition 7) along every line we picked. Each invocation returns a candidate value of the desired message symbol. Ben-Aroya et. al [BET10a] show that if we use Forney's GMD decoding technique [For66] where one assigns weights to each output of the decoder based on the number of errors along the corresponding line then the symbol with the largest weight is with high probability the correct symbol.

1.5 Subsequent work

Our main result shows that increasing the query complexity from $O\left(2^t\right)$ to $t^{O(t)}$ in matching vector codes allows one to handle an error rate independent of t. This result has been strengthened first in [BET10b], and later in [SY11] where it was shown that (similarly to Reed Muller codes) matching vector codes in fact require only a constant multiplicative increase in the query complexity in order to tolerate a fixed constant fraction of errors.

Finally, we would like to point out a recent technically unrelated work of Kopparty et al. [KSY11] that introduces a new class of locally decodable codes termed *multiplicity codes* that vastly improve upon RM codes (and MV codes) in the regime of polynomially high query complexity.

1.6 Outline

We start section 3 with formal definitions of locally decodable codes and matching families of vectors. We introduce the concept of a bounded matching family and show how any such family yields an LDC tolerating a large fraction of errors. In section 4 we present two constructions of bounded matching families. In section 5 we put the results of sections 3 and 4 together to obtain new upper bounds on the length of MV codes. In section 6 we obtain a collection of upper bounds on the size of matching families of vectors. In section 7 we translate the results of section 6 into lower bounds on the length of MV codes.

2 Notation

We use the following standard mathematical notation:

- $[k] = \{1, \ldots, k\}$;
- $\mathbb{F}_q$ is a finite field of q elements. $\mathbb{F}_q^*$ is the multiplicative group of $\mathbb{F}_q$;
- For a polynomial $f \in \mathbb{F}_q[z_1, \ldots, z_h]$ we denote by $\mathrm{supp}(f)$ the set of monomials with non zero coefficients in f, where a monomial $z_1^{e_1} \cdots z_h^{e_h}$ is identified with the integer h-tuple $(e_1, \ldots, e_h)$;
- $\mathbb{Z}_m$ is the ring of integers modulo an integer m. $\mathbb{Z}_m^*$ is the set of invertible elements of $\mathbb{Z}_m$;
- $d(\mathbf{x}, \mathbf{y})$ denotes the Hamming distance between vectors $\mathbf{x}$ and $\mathbf{y}$;
- $(\mathbf{u}, \mathbf{v})$ stands for the dot product of vectors $\mathbf{u}$ and $\mathbf{v}$;
- For a vector $\mathbf{w} \in \mathbb{Z}_m^n$ and an integer $l \in [n]$, let $\mathbf{w}(l)$ denote the l-th coordinate of $\mathbf{w}$;
- A D-evaluation of a function f defined over D, is a vector of values of f at all points of D.
- We write $\exp(x)$ to denote $2^{O(x)}$.

3 Matching vector codes: the framework

In this section we formally define locally decodable codes and matching families of vectors. We review the existing construction of LDCs from matching families, casting it in a new language that makes explicit certain intrinsic similarity between MV codes and RM codes. We then introduce the concept of a *bounded* matching family and show how MV codes based on these families can be decoded from large amounts of error.

Definition 1 *A q-ary code $C : \mathbb{F}_q^k \rightarrow \mathbb{F}_q^N$ is said to be (r, δ, ϵ)-locally decodable if there exists a randomized decoding algorithm $\mathcal{A}$ such that*

1. *For all* $\mathbf{x} \in \mathbb{F}_q^k$, $i \in [k]$ *and* $\mathbf{y} \in \mathbb{F}_q^N$ *such that* $d(C(\mathbf{x}), \mathbf{y}) \leq \delta N : \Pr[\mathcal{A}^{\mathbf{y}}(i) = \mathbf{x}(i)] \geq 1 - \epsilon$, *where the probability is taken over the random coin tosses of the algorithm* $\mathcal{A}$.

2. $\mathcal{A}$ *makes at most* r *queries to* $\mathbf{y}$.

A locally decodable code is called linear if C is a linear transformation over $\mathbb{F}_q$. Our constructions of locally decodable codes are linear. While our main interest is in binary codes we deal with codes over larger alphabets as well.

Definition 2 *Let* $S \subseteq \mathbb{Z}_m \setminus \{0\}$. *We say that families* $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ *and* $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ *of vectors in* $\mathbb{Z}_m^n$ *form an* S*-matching family if the following two conditions are satisfied:*

- *For all* $i \in [k]$, $(\mathbf{u}_i, \mathbf{v}_i) = 0$;
- *For all* $i, j \in [k]$ *such that* $i \neq j$, $(\mathbf{u}_j, \mathbf{v}_i) \in S$.

We now show how one can obtain a MV code out of a matching family. We start with some notation.

- We assume that q is a prime power, $m \mid q-1$, and denote a subgroup of $\mathbb{F}_q^*$ of order m by $\mathbb{C}_m$;
- We fix some generator g of $\mathbb{C}_m$;
- For $\mathbf{w} \in \mathbb{Z}_m^n$, we define $g^{\mathbf{w}} \in \mathbb{C}_m^n$ by $\left(g^{\mathbf{w}(1)}, \ldots, g^{\mathbf{w}(n)}\right)$;
- For $\mathbf{w}, \mathbf{v} \in \mathbb{Z}_m^n$ we define the multiplicative line $M_{\mathbf{w},\mathbf{v}}$ through $\mathbf{w}$ in direction $\mathbf{v}$ to be the multi-set
$$M_{\mathbf{w},\mathbf{v}} = \left\{ g^{\mathbf{w}+\lambda\mathbf{v}} \mid \lambda \in \mathbb{Z}_m \right\}; \tag{1}$$
- For $\mathbf{u} \in \mathbb{Z}_m^n$, we define the monomial $\mathrm{mon}_{\mathbf{u}} \in \mathbb{F}_q[z_1, \ldots, z_n]/(z_1^m = 1, \ldots, z_n^m = 1)$ by
$$\mathrm{mon}_{\mathbf{u}}(z_1, \ldots, z_n) = \prod_{\ell \in [n]} z_\ell^{\mathbf{u}(\ell)}. \tag{2}$$

3.1 The general encoding/decoding framework

Observe that for any $\mathbf{w}, \mathbf{u}, \mathbf{v} \in \mathbb{Z}_m^n$ and $\lambda \in \mathbb{Z}_m$ we have
$$\mathrm{mon}_{\mathbf{u}}\left(g^{\mathbf{w}+\lambda\mathbf{v}}\right) = g^{(\mathbf{u},\mathbf{w})}\left(g^{\lambda}\right)^{(\mathbf{u},\mathbf{v})}. \tag{3}$$

The formula above implies that the $M_{\mathbf{w},\mathbf{v}}$-evaluation of a monomial $\mathrm{mon}_{\mathbf{u}}$ is a $\mathbb{C}_m$-evaluation of a (univariate) monomial
$$g^{(\mathbf{u},\mathbf{w})} y^{(\mathbf{u},\mathbf{v})} \in \mathbb{F}_q[y]. \tag{4}$$
This observation is the foundation of our decoding algorithms. We now specify the encoding procedure and outline the main steps taken by decoding procedures described later on (propositions 3, 7). Let $\mathcal{U}, \mathcal{V}$ be an S-matching family in $\mathbb{Z}_m^n$, where $|\mathcal{U}| = |\mathcal{V}| = k$.

Encoding: We encode a message $(\mathbf{x}(1), \ldots, \mathbf{x}(k)) \in \mathbb{F}_q^k$ by the $\mathbb{C}_m^n$-evaluation of the polynomial
$$F(z_1, \ldots, z_n) = \sum_{j=1}^{k} \mathbf{x}(j) \cdot \mathrm{mon}_{\mathbf{u}_j}(z_1, \ldots, z_n). \tag{5}$$

Notice that $F = F_{\mathbf{x}}$ is a function of the message $\mathbf{x}$ (we will omit the subscript and treat $\mathbf{x}$ as fixed throughout this section).

Basic decoding: The input to the decoder is a (corrupted) $\mathbb{C}_m^n$-evaluation of F and an index $i \in [k]$.

1. The decoder picks $\mathbf{w} \in \mathbb{Z}_m^n$ uniformly at random;
2. The decoder recovers the noiseless restriction of F to $M_{\mathbf{w},\mathbf{v}_i}$. To accomplish this the decoder may query the (corrupted) $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F at m or fewer locations.

To see that noiseless $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F uniquely determines $\mathbf{x}(i)$ note that by formulas (3), (4) and (5) the $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F is a $\mathbb{C}_m$-evaluation of a polynomial

$$f(y) = \sum_{j=1}^{k} \mathbf{x}(j) \cdot g^{(\mathbf{u}_j,\mathbf{w})} y^{(\mathbf{u}_j,\mathbf{v}_i)} \in \mathbb{F}_q[y]. \tag{6}$$

Further observe that the properties of the S-matching family $\mathcal{U}, \mathcal{V}$ and (6) yield

$$f(y) = \mathbf{x}(i) \cdot g^{(\mathbf{u}_i,\mathbf{w})} + \sum_{s \in S} \left(\sum_{j \,:\, (\mathbf{u}_j,\mathbf{v}_i)=s} \mathbf{x}(j) \cdot g^{(\mathbf{u}_j,\mathbf{w})} \right) y^s. \tag{7}$$

It is evident from the above formula that $supp(f) \subseteq S \cup \{0\}$ and

$$\mathbf{x}(i) = f(0)/g^{(\mathbf{u}_i,\mathbf{w})}. \tag{8}$$

We now describe several local decoders that follow the general paradigm outlined above.

3.2 The simplest decoder

The proposition below gives the simplest local decoder. The current form has first appeared in [Efr09].

Proposition 3 *Let $\mathcal{U}, \mathcal{V}$ be a family of S-matching vectors in $\mathbb{Z}_m^n$, $|\mathcal{U}| = |\mathcal{V}| = k$, $|S| = s$. Suppose $m \mid q-1$, where q is a prime power; then there exists a q-ary linear code encoding k-long messages to m^n-long codewords that is $(s+1, \delta, (s+1)\delta)$-locally decodable for all δ.*

Proof: The encoding procedure has already been specified by formula (5). To recover the value $\mathbf{x}(i)$

1. The decoder picks $\mathbf{w} \in \mathbb{Z}_m^n$ at random, and queries the (corrupted) $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F at $(s+1)$ consecutive locations $\left\{ g^{\mathbf{w}+\lambda\mathbf{v}_i} \mid \lambda \in \{0, \ldots, s\} \right\}$ to obtain values $c_0, \ldots, c_s$.
2. The decoder recovers the unique sparse univariate polynomial $h(y) \in \mathbb{F}_q[y]$ with $supp(h) \subseteq S \cup \{0\}$ such that for all $\lambda \in \{0, \ldots, s\}$, $h(g^\lambda) = c_\lambda$. (The uniqueness of $h(y)$ follows from standard properties of Vandermonde matrices.)
3. Following the formula (8) the decoder returns $h(0)/g^{(\mathbf{u}_i,\mathbf{w})}$.

The discussion above implies that if all $(s+1)$ locations queried by the decoder are not corrupted then $h(y)$ is indeed the noiseless restriction of F to $M_{\mathbf{w},\mathbf{v}_i}$, and decoder's output is correct. It remains to note that each individual query of the decoder samples a uniformly random location and apply the union bound. ■

Remark 4 In the proposition above we interpolate the polynomial $h(y)$ to recover its free coefficient. In certain cases (relying on special properties of the integer m and the set S) it may be possible to recover the free coefficient in ways that do not require complete interpolation and thus save on the number of queries. This general idea has been used in [Yek08, Efr09] for the case of three-query codes, and in [IS10, MFL$^+$10]. In appendix B we present some new general sufficient conditions that allow for such a saving.

3.3 Improved decoding using bounded matching families

We now introduce the concept of a bounded matching family of vectors and show how MV codes based on bounded matching families can be decoded from large amounts of error. In what follows we identify $\mathbb{Z}_m$ with the subset $\{0, \ldots, m-1\}$ of real numbers. This imposes a total ordering on $\mathbb{Z}_m$, $0 < 1 < \ldots < m-1$ and allows us to compare elements of $\mathbb{Z}_m$ with reals.

Definition 5 *Let b be a positive real. A set $S \subseteq \mathbb{Z}_m$ is b-bounded if for all $s \in S$, $s < b$.*

Definition 6 *Let b be a positive real. An S-matching family $\mathcal{U}, \mathcal{V}$ in $\mathbb{Z}_m^n$ is b-bounded if the set S is b-bounded.*

The proposition below gives the first local decoder for MV codes that is capable of tolerating large amounts of error. Our constructions of MV codes in section 5 rely on it.

Proposition 7 *Let $0 < \sigma < 1$. Let $\mathcal{U}, \mathcal{V}$ be a σm-bounded family of S-matching vectors in $\mathbb{Z}_m^n$, $|\mathcal{U}| = |\mathcal{V}| = k$. Suppose $m \mid q-1$, where q is a prime power; then there exists a q-ary linear code encoding k-long messages to m^n-long codewords that is $(m, \delta, 2\delta/(1-\sigma))$-locally decodable for all δ.*

Proof: The encoding procedure has already been specified by (5). To recover the value $\mathbf{x}(i)$,

1. The decoder picks $\mathbf{w} \in \mathbb{Z}_m^n$ at random, and queries the (corrupted) $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F at all m locations $\left\{g^{\mathbf{w}+\lambda\mathbf{v}_i} \mid \lambda \in \mathbb{Z}_m\right\}$ to obtain values $c_0, \ldots, c_{m-1}$.
2. The decoder recovers the univariate polynomial $h(y) \in \mathbb{F}_q[y]$ of degree less than σm such that for all but at most $(m - \sigma m)/2$ values of $\lambda \in \mathbb{Z}_m$, $h(g^\lambda) = c_\lambda$. If such an h does not exist the decoder encounters a failure, and returns 0. Note that $\deg h < \sigma m$ implies that $h(y)$ is unique, if it exists. The search for $h(y)$ is an instance of the Reed-Solomon unique decoding problem. It can be done efficiently using the Berlekamp-Welch algorithm [MS77].
3. Following the formula (8) the decoder returns $h(0)/g^{(\mathbf{u}_i,\mathbf{w})}$.

The discussion above implies that if the $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F is corrupted in at most $(m - \sigma m)/2$ locations, then $h(y)$ is indeed the noiseless restriction of F to $M_{\mathbf{w},\mathbf{v}_i}$, and the decoder's output is correct. It remains to note that each individual query of the decoder samples a uniformly random location and thus by Markov's inequality the probability that more than $(m - \sigma m)/2$ of decoder's queries go to corrupted locations is at most $2\delta/(1-\sigma)$. ■

Remark 8 Following the initial publication of our work, in [SY11] it was shown that the query complexity of codes from proposition 7 can be reduced from m to $O(|S|)$.

3.4 From q-ary to binary codes

Proposition 7 yields non-binary locally decodable codes. As we remarked earlier our main interest is in binary LDCs. The next lemma extends proposition 7 to produce binary codes. The idea behind the proof is fairly standard and involves concatenation with a good binary error correcting code.

Lemma 9 *Let $0 < \sigma < 1$. Let $\mathcal{U}, \mathcal{V}$ be a σm-bounded family of S-matching vectors in $\mathbb{Z}_m^n$, $|\mathcal{U}| = |\mathcal{V}| = k$. Suppose $m \mid q-1$, where $q = 2^b$. Further suppose that there exists a binary linear code C_{inner} of distance μB encoding b-bit messages to B-bit codewords; then there exists a binary linear code C encoding kb-bit messages to $m^n B$-bit codewords that is $(mB, \delta, 2\delta/(\mu - \mu\sigma))$-locally decodable for all δ.*

Proof: Observe that the condition of the lemma is strictly stronger than the condition of proposition 7. Thus the implication of proposition 7 holds. Let C_{outer} be the 2^b-ary $(m, \delta, 2\delta/(1-\sigma))$-LDC encoding k-long messages to m^n-long codewords. We define the code C to be the concatenation [MS77] of C_{outer} and C_{inner}. In order to decode a single bit, the decoder recovers the symbol of the large alphabet that the bit falls into.

Recall that in order to recover a single coordinate of the message the decoder of C_{outer} queries the corrupted encoding at m points of a multiplicative line, and then solves the Reed Solomon decoding problem (i.e., finds the unique univariate polynomial of degree less than σm that agrees with the observed sequence of values in all but at most $(1-\sigma)m/2$ locations).

The decoder of C acts similarly. Firstly, it entirely reads m corrupted B-bit codewords of the inner code that store (encoded) coordinate values of the outer code along a randomly chosen multiplicative line. Secondly, it decodes a binary code that is a concatenation of a Reed Solomon code of degree less than σm over $\mathbb{F}_{2^b}$ and a binary code C_{inner} up to half of its minimum distance. Decoding is correct provided that the total number of errors in mB locations read is at most $(1-\sigma)\mu m B/2$. Decoding can be done efficiently provided that C_{inner} has an efficient decoder.

It remains to note that each individual query of the decoder samples a uniformly random location and thus by Markov's inequality the probability that more than $(1-\sigma)\mu m B/2$ of decoder's queries go to corrupted locations is at most $2\delta/(\mu - \mu\sigma)$. ■

Proposition 7 allows one to obtain LDCs over large alphabets that tolerate δ up to $1/4$ (by setting σ close to zero). Lemma 9 allows one to obtain *binary* LDCs that tolerate δ up to $1/8$ (by setting σ close to zero, and μ close to $1/2$).

4 Matching vectors: constructions

In this section we present constructions of bounded matching families of vectors. Our first construction (lemma 13) is based on an existing matching family due to Grolmusz [Gro00]. We argue that an appropriate scaling turns Grolmusz's family into a bounded family. Later in section 5 we use this construction to obtain MV codes that improve upon LDCs of [Efr09] in terms of the amount of noise that they can tolerate, and improve upon classical r-query RM LDCs in terms of codeword length for all $r \leq \log k/(\log\log k)^c$. Our second construction (lemma 16) is self-contained. It improves on the first construction for large values of m, and yields MV codes that match RM LDCs for certain values of $r > \log k/(\log\log k)^c$.

4.1 Grolmusz family of matching vectors

Definition 10 (Canonical set) *Let $m = \prod_{i=1}^{t} p_i$ be a product of distinct primes. The* canonical set *in $\mathbb{Z}_m$ is the set of all non-zero s such that for every $i \in [t]$, $s \in \{0,1\} \bmod p_i$.*

Basic parameters of Grolmusz's family are given by the following lemma. The construction follows the lines of Grolmusz's construction of a set system with restricted intersections modulo composites [Gro00, Gro02], but with some differences. We use an approach suggested by Sudan [Sud09] to go directly from polynomials to matching vectors without constructing set-systems, which gives a slight improvement in parameters. We defer the proof to appendix A.

Lemma 11 *Let $m = \prod_{i=1}^{t} p_i$ be a product of distinct primes. Let w be a positive integer. Let $\{e_i\}$, $i \in [t]$ be integers such that for all i, we have $p_i^{e_i} > w^{1/t}$. Let $d = \max_i p_i^{e_i}$ and $h \geq w$ be arbitrary.*

Let S be the canonical set; then there exists an $\binom{h}{w}$-sized family of S-matching vectors in $\mathbb{Z}_m^n$, where $n = \binom{h}{\leq d}$.

We now argue that a canonical set can be made bounded via scaling by an invertible element.

Lemma 12 *Let $m = \prod_{i=1}^t p_i$ be a product of distinct primes. Let S be the canonical set in $\mathbb{Z}_m$. There exists an $\alpha \in \mathbb{Z}_m^*$ such that the set αS is σm-bounded for any $\sigma > \sum_{i \in [t]} 1/p_i$.*

Proof: We start with some notation.

- For every $i \in [t]$, define the integer $\hat{p_i} = m/p_i$;
- Let $\alpha \in \mathbb{Z}_m^*$ be the unique element such that for all $i \in [t], \alpha = \hat{p_i} \bmod p_i$.

Observe that for any $i, j \in [t]$,

$$\left(\alpha^{-1}\hat{p_i}\right) \bmod p_j = \begin{cases} 1, & \text{if i=j;} \\ 0, & \text{otherwise.} \end{cases}$$

Let $s \in S$ be arbitrary. Set $I = \{i \in [t] \mid p_i \text{ does not divide } s\}$. Observe that

$$s = \alpha^{-1} \sum_{i \in I} \hat{p_i},$$

since the identity above clearly holds modulo each of the $\{p_i\}$. Therefore

$$\alpha s = \sum_{i \in I} \hat{p_i} \leq m \sum_{i \in [t]} 1/p_i.$$

■

The argument above shows that any S-matching family $\mathcal{U}, \mathcal{V}$ where S is the canonical set can be turned into a bounded one by scaling all vectors in $\mathcal{V}$ by an invertible element provided $\sum_{i \in [t]} 1/p_i < 1$. Note that such scaling does not change the set $\mathcal{U}$, and hence the corresponding MV code. It also does not change the set of points queried by the decoder (of proposition 7), since for an invertible $\alpha \in \mathbb{Z}_m$, and an arbitrary $\mathbf{v} \in \mathbb{Z}_m^n$ multiplicative lines in the directions $\mathbf{v}$ and $\alpha\mathbf{v}$ are the same. Combining lemma 12 with lemma 11 we obtain

Lemma 13 *Let $m = \prod_{i=1}^t p_i$ be a product of distinct primes. Let w be a positive integer. Let $\{e_i\}$, $i \in [t]$ be integers such that for all i, we have $p_i^{e_i} > w^{1/t}$. Let $d = \max_i p_i^{e_i}$, and $h \geq w$ be arbitrary. Then there exists an $\binom{h}{w}$-sized σm-bounded family of matching vectors in $\mathbb{Z}_m^n$, where $n = \binom{h}{\leq d}$ and σ is an arbitrary real number larger than $\sum_{i \in [t]} 1/p_i$.*

4.2 Simple construction of matching vectors

In this section we give an elementary construction of a bounded family of matching vectors. The construction works for both prime and composite moduli. The family improves upon the family of lemma 13 for large values of m. In what follows we use $\mathbb{Z}_{\geq 0}$ to denote the set of non-negative integers.

Definition 14 *Let $b(m', n)$ denote the number of vectors $\mathbf{w} \in \mathbb{Z}_{\geq 0}^n$ such that $\|\mathbf{w}\|_2^2 = m'$.*

Thus $b(m', n)$ counts the number of integer points on the surface of the n-dimensional ball of radius $\sqrt{m'}$ in the positive orthant.

Lemma 15 *Let $m' < m$ and $n \geq 2$ be arbitrary positive integers. There exists a $b(m', n-1)$-sized $(m'+1)$-bounded family of matching vectors in $\mathbb{Z}_m^n$.*

Proof: Let $k = b(m', n-1)$ and let $\mathbf{w}_1, \ldots, \mathbf{w}_k$ be the vectors in $\mathbb{Z}_{\geq 0}^{n-1}$ such that $\|\mathbf{w}_i\|_2^2 = m'$. For each $\mathbf{w}_i$, we define vectors in $\mathbb{Z}^n$ by

$$\mathbf{u}_i = (1, -\mathbf{w}_i), \ \mathbf{v}_i = (m', \mathbf{w}_i).$$

We claim that the resulting family of vectors is a $\{1, \ldots, m'\}$-matching family. To prove this, observe that $(\mathbf{u}_i, \mathbf{v}_j) = m' - (\mathbf{w}_i, \mathbf{w}_j)$. If $i = j$, then $(\mathbf{w}_i, \mathbf{w}_j) = \|\mathbf{w}_i\|_2^2 = m'$ whereas if $i \neq j$; then by Cauchy-Schwartz

$$(\mathbf{w}_i, \mathbf{w}_j) \leq \|\mathbf{w}_i\|_2 \|\mathbf{w}_j\|_2 = m'.$$

In fact the inequality must be strict since $\mathbf{w}_i$ and $\mathbf{w}_j$ both lie on the surface of the same ball, hence they are not collinear. But since their inner product lies in $\mathbb{Z}_{\geq 0}$, we conclude that $(\mathbf{w}_i, \mathbf{w}_j) \in \{0, \ldots, m'-1\}$, hence $(\mathbf{u}_i, \mathbf{v}_j) \in \{1, \ldots, m'\}$. Now note that since $m > m'$, the inner products do not change modulo m. ■

The lemma below follows by combining lemma 15 with some crude lower bounds for $b(m', n-1)$.

Lemma 16 *Let $m' < m$ and $n \geq 2$ be arbitrary positive integers. There exists a k-sized $(m'+1)$-bounded family of matching vectors in $\mathbb{Z}_m^n$, where*

$$k = \tfrac{1}{m'+1} \left(\tfrac{m'}{n-1}\right)^{(n-1)/2} \qquad \text{for } m' \geq n, \tag{9}$$

$$k = \binom{n-1}{m'} \qquad \text{for } m' < n. \tag{10}$$

Proof: To prove (9), we set $d = \left\lfloor \sqrt{m'/(n-1)} \right\rfloor$. For every vector $\mathbf{w} \in \{0, \ldots, d\}^{n-1}$, we have $0 \leq \|\mathbf{w}\|^2 \leq (n-1)d^2 \leq m'$. By the pigeonhole principle, there exists some $m'' \in \{0, \ldots, m'\}$ such that $b(m'', n-1) \geq (d+1)^{n-1}/(m'+1)$, which by lemma 15 yields an $(m'+1)$-bounded matching family of size

$$k \geq \frac{1}{m'+1} \left(\left\lfloor \sqrt{\frac{m'}{n-1}} \right\rfloor + 1 \right)^{n-1} \geq \frac{1}{m'+1} \left(\frac{m'}{n-1} \right)^{(n-1)/2}.$$

Note that the condition $m' \geq n$ is only needed to ensure that the bound in meaningful.

To prove (10), we observe that $b(m', n-1) \geq \binom{n-1}{m'}$ by taking all vectors in $\{0,1\}^{n-1}$ of Hamming weight exactly m'. The bound follows from lemma 15. ■

5 Upper bounds for the length of MV codes

In this section we combine the results of the previous sections to derive upper bounds on MV codes. A combination of lemma 9 and lemma 13 yields

Lemma 17 *Let $m = \prod_{i=1}^t p_i$ be a product of distinct odd primes. Let w be a positive integer. Suppose integers $\{e_i\}$, $i \in [t]$ are such that for all i, we have $p_i^{e_i} > w^{1/t}$. Let $d = \max_i p_i^{e_i}$, and $h \geq w$ be arbitrary. Let σ is an arbitrary real number larger than $\sum_{i \in [t]} 1/p_i$. Suppose $m \mid q - 1$, where $q = 2^b$. Further* *suppose that there exists a binary code C_{inner} of distance μB encoding b-bit messages to B-bit codewords; then there exists a binary linear code C encoding $\left(\binom{h}{w}\cdot b\right)$-bit messages to $\left(m^{\binom{h}{\le d}}\cdot B\right)$-bit codewords that is $(mB, \delta, 2\delta/(\mu - \mu\sigma))$-locally decodable for all δ.*

In what follows we estimate asymptotic parameters of our codes.

Lemma 18 *For all integers $t \ge 2$ and $k \ge 2^t$ there exists a binary linear code encoding k-bit messages to*

$$N = \exp\exp\left((\log k)^{1/t}(\log\log k)^{1-1/t}t\ln t\right)$$

-bit codewords that is $\left(t^{O(t)}, \delta, 4\delta(1 + O(1/\ln t))\right)$-locally decodable for all δ.

Proof: The proof follows by appropriately setting the parameters in lemma 17.

1. By [Sho05, theorem 5.7] there exists a universal constant c such that the range $[(c/2)t\ln t, ct\ln t]$ contains at least t distinct odd primes $p_1, \ldots, p_t$;
2. Note that $\sum_{i\in[t]} 1/p_i = O(1/\ln t)$;
3. Set $m = \prod_{i\in[t]} p_i$. Clearly, $m = t^{O(t)}$;
4. Set b to be the smallest positive integer such that $m \,\big|\, 2^b - 1$ Clearly, $b \le m - 1 = t^{O(t)}$. Set $q = 2^b$;
5. A standard greedy argument (that is used to prove the classical Gilbert-Varshamov bound [MS77]) implies that there is a universal constant c' such that for all integers $s \ge 1$, there exists a binary linear code of distance $(1/2 - c'/\sqrt{s})s^2$ encoding s-bit messages to s^2-bit codewords. We set C_{inner} to be a binary linear code that encodes b-bit messages to $B = t^{O(t)}$-bit codewords and has distance μB, for $\mu \ge \left(1/2 - c'/t^{\Omega(t)}\right)$;
6. We now assume that there exists a positive integer w which is a multiple of t such that $k = w^{w/t}$. Clearly, we have $w = \Theta(t\log k/\log\log k)$;
7. Following lemma 17 for every $i \in [t]$, let e_i be the smallest integer such that $p_i^{e_i} > w^{1/t}$. Let $d = \max_i p_i^{e_i}$. Clearly, $d = O(w^{1/t}t\ln t)$;
8. Set $h = c''\left\lceil w^{1+1/t}\right\rceil$, where c'' is an appropriately chosen (constant) positive integer ensuring that $h \ge d$;
9. Observe that $\binom{h}{w}\cdot b \ge (h/w)^w \ge k$;
10. Note that $\binom{h}{\le d} \le d(eh/d)^d$;
11. Set $N = m^{\binom{h}{\le d}}\cdot B \le t^x$, where $x = O(t)(ew)^{O(w^{1/t}t\ln t)}$;
12. We combine lemma 17 with inequalities that we proved above and make basic manipulations to obtain a binary linear code encoding k-bit messages to $\exp\exp\left((\log k)^{1/t}(\log\log k)^{1-1/t}t\ln t\right)$-bit codewords that is $\left(t^{O(t)}, \delta, 4\delta(1 + O(1/\ln t))\right)$-locally decodable for all δ.

13. Finally, we note that the assumption about $k = w^{w/t}$, for some w can be safely dropped. If k does not have the required shape, we pad k-bit messages with zeros to get messages of length k', where k' has the shape $w^{w/t}$ and then apply the procedure above. One can easily check that such padding requires a sub-quadratic blow up in the message length and therefore does not affect asymptotic parameters.

■

Setting t to be a constant in lemma 18 yields

Theorem 19 *For every constant* $t \geq 2$ *and all large enough* k*, there exists a binary linear code encoding* k*-bit messages to* $\exp\exp\left((\log k)^{1/t}(\log\log k)^{1-1/t}\right)$*-bit codewords that is* $\left(t^{O(t)}, \delta, 4\delta(1+O(1/\ln t))\right)$*-locally decodable for all* δ*.*

For every constant $t \geq 2$, theorem 19 gives a family of $\left(t^{O(t)}, \delta, 4\delta(1+O(1/\ln t))\right)$-locally decodable codes of length essentially identical to the length of the shortest known $\left(2^{O(t)}, \delta, 2^{O(t)}\delta\right)$-locally decodable codes of [Efr09]. Our codes can tolerate much larger amounts of noise, (i.e., for large values of t our codes tolerate approximately $1/8$ fraction of errors, while the fraction of errors tolerated by codes from earlier work drops to zero rapidly.) The improvement comes at a price of a moderate increase in the number of queries.

The following theorem gives asymptotic parameters of our codes in terms of r and k.

Theorem 20 *For every large enough integer* r *and every* $k \geq r$*, there exists a binary linear code encoding* k*-bit messages to*

$$\exp\exp\left((\log k)^{O(\log\log r/\log r)}(\log\log k)^{1-\Omega(\log\log r/\log r)}\log r\right) \tag{11}$$

bit codewords that is $(r, \delta, 4\delta(1+O(1/\ln\ln r)))$*-locally decodable for all* δ*.*

Proof: The proof follows by setting parameters in lemma 18. Set t to be the largest integer such that $t^{O(t)} \leq r$, where the constant in O-notation is the same as the one in lemma 18. Assuming r is sufficiently large we have $t = \Theta(\log r/\log\log r)$. One can also check that $k \geq r$ implies that the condition of lemma 18 is satisfied. An application of the lemma concludes the proof. ■

5.1 MV codes over characteristic zero

We remark here that the entire construction and analysis of MV codes described in the preceding sections (apart from the parts dealing with reduction to the binary case) work also if the underlying field, $\mathbb{F}_q$, is replaced with the complex number field $\mathbb{C}$. The only property we used in $\mathbb{F}_q$ is that it contains an element of order m, which trivially holds over $\mathbb{C}$ for every m. This implies the existence of *linear* LDCs with essentially the same parameters as above also over the complex numbers (the definition of LDCs over an arbitrary field is the same as for finite fields, we simply allow the decoder to preform field arithmetic operations on its inputs). Once one has a linear code over the complex numbers, it is straightforward to get a code over the reals by writing each complex number as a pair of real numbers.

We find this interesting since, prior to MV codes, there were no known constructions of LDC's over characteristic zero (apart from trivial 2-query codes of exponential stretch). In fact, there are no known constructions, apart from MV codes, even over finite fields with very large characteristic (RM codes require the characteristic to be at most the codeword length). Even though this might seem like an esoteric setting, LDCs over characteristic zero did came up in some recent works in connection to arithmetic circuit complexity [DS06, Dvi10, BDWY11].

5.2 Comparison to Reed Muller codes

Theorem 20 yields the first family of locally decodable codes (other than RM codes) that have super-constant query complexity and tolerate a constant fraction of errors. In this section we provide a comparison between RM codes and our codes.

A Reed Muller locally decodable code [KT00, Yek11] is specified by three integer parameters. Namely, a prime power (alphabet size) q, number of variables n, and the degree $d < q - 1$. The q-ary code consists of $\mathbb{F}_q^n$-evaluations of all polynomials in $\mathbb{F}_q[z_1, \ldots, z_n]$ of total degree at most d. Such code encodes $k = \binom{n+d}{d}$-long messages to q^n-long codewords. Provided that $d < \sigma(q-1)$, the code is $(q-1, \delta, 2\delta/(1-\sigma))$-locally decodable for all δ. If q is a power of 2 non-binary RM LDCs can be turned into binary via concatenation (in a manner similar to the one used in lemma 9). If one does concatenation with an asymptotically good code of relative distance μ one gets a binary linear code encoding k-bit messages to N-bit codewords that is $(r, \delta, 2\delta/(\mu - \mu\sigma))$-locally decodable for all δ, where

$$k = \binom{n+d}{d} \log q, \quad N = \Theta(q^n \log q), \quad r = \Theta(q \log q). \tag{12}$$

One can get various asymptotic families of RM LDCs by specifying an appropriate relation between n and d and letting these parameters grow to infinity. Increasing d relative to n yields shorter codes of larger query complexity.

Example 21 Setting $d = n$, $q = cn$ (for a constant c), and letting n grow while concatenating with asymptotically good binary codes of relative distance μ one gets a family of binary LDCs that encode k-bit messages to $k^{\Theta(\log \log k)}$-bit codewords and are $(\Theta(\log k \log \log k), \delta, 2\delta/(\mu - 2\mu/c))$-locally decodable for all δ.

We now argue that RM LDCs are inferior to codes of theorem 20 (with respect to codeword length) for all $r \leq \log k/(\log \log k)^c$, where c is a universal constant. To arrive at such a conclusion we need a lower bound on the length of RM LDCs. Let d, n, and q be such that formulas (12) yield an r-query LDC, where r belongs to the range of our interest. We necessarily have $d \leq n$ (otherwise $r > \log k$). Thus

$$k = \binom{n+d}{d} \log q \leq (2en/d)^d \log q \leq n^{O(d)}, \tag{13}$$

and $n \geq k^{\Omega(1/d)}$. Therefore writing $\exp(x)$ to denote $2^{\Omega(x)}$, we have

$$N \geq \exp \exp \left(\log k/d\right) \geq \exp \exp \left(\log k/r\right). \tag{14}$$

Note that when r is a constant then already 3-query codes of [Efr09] improve substantially upon (14). To conclude the argument one needs to verify that there exists a constant c such that for every nondecreasing function $r(k)$, where $r(k)$ grows to infinity, and satisfies $r(k) \leq \log k/(\log \log k)^c$, for all sufficiently large k the right hand side of (14) evaluates to a larger value than (11).

Remark 22 It is interesting to observe that while MV codes of theorem 20 improve upon RM LDCs only for $r \leq \log k/(\log \log k)^c$, one can get MV codes that (asymptotically) match RM LDCs of example 21 combining lemma 16 (where m has the shape $2^b - 1$, $n = m + 1$ and $m' = n/2$) with lemma 9.

6 Matching Vectors: limitations

Let $k(m,n)$ denote the size of the largest family of S-matching vectors in $\mathbb{Z}_m^n$ where we allow S to be an arbitrary subset of $\mathbb{Z}_m \setminus \{0\}$. The rate of any locally decodable code obtained via propositions 3 or 7 is at most $k(m,n)/m^n$. Our goal in this section is to prove upper bounds on $k(m,n)$. In section 7 we translate these bounds into lower bounds on the length of MV codes.

There is a large body of work in combinatorics on the closely related problem of set-systems with restricted modular intersections. The problem there is to bound the size of the largest set family $\mathcal{F}$ on $[n]$, where the sets in $\mathcal{F}$ have cardinality 0 modulo some integer m, while their intersections have non-zero cardinality modulo m. The classical result in this area is the modular Ray-Chaudhuri-Wilson theorem [BF98] showing that when m is a prime (or a prime power), an upper bound of $n^{O(m)}$ holds. It is known that such a bound does not apply when m is composite [Gro00, Gro02]. The best upper bound for general m shows that $|\mathcal{F}| \leq 2^{n/2}$ [Sga99].

We start by bounding $k(m,n)$ in the prime case.

6.1 The prime case

We present two bounds for the prime case. The first bound (theorem 23) is based on the linear algebra method [BF98] and is tight when p is a constant. The second bound (lemma 28) is nearly tight when n is constant.

Theorem 23 *For any positive integer n and any prime p, we have*

$$k(p,n) \leq 1 + \binom{n+p-2}{p-1}. \tag{15}$$

Proof: Let $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$, $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ be a family of S-matching vectors of $\mathbb{F}_p^n$, for some $S \subseteq \mathbb{F}_p^*$. For each $i \in [k]$, we consider the polynomial

$$P_i(z_1, \ldots, z_n) = 1 - \left(\sum_{l=1}^{n} \mathbf{v}_i(l) \cdot z_l\right)^{p-1}$$

in the ring $\mathbb{F}_p[z_1, \ldots, z_n]$. It is easy to see that $P_i(\mathbf{u}_i) = 1$, whereas $P_i(\mathbf{u}_j) = 0$ for all $j \neq i$. This implies that the k polynomials $\{P_i\}_{i=1}^k$ are linearly independent. But these polynomials all lie in an $\mathbb{F}_p$ vector-space of dimension $1 + \binom{n+p-2}{p-1}$, since they are spanned by the monomial 1 and all monomials of degree exactly $p-1$ in $z_1, \ldots, z_n$. ■

Note that equation (10) shows that for constant p and growing n, the above bound is asymptotically tight.

Our second bound comes from translating the problem of constructing matching vectors into a problem about points and hyperplanes in projective space. The $n-1$ dimensional projective geometry $\mathrm{PG}(\mathbb{F}_p, n-1)$ over $\mathbb{F}_p$ consist of all points in $\mathbb{F}_p^n \setminus \{0^n\}$ under the equivalence relation $\lambda\mathbf{v} \equiv \mathbf{v}$ for $\lambda \in \mathbb{F}_p^*$. Projective hyperplanes are specified by vectors $\mathbf{u} \in \mathbb{F}_p^n \setminus \{0^n\}$ under the equivalence relation $\lambda\mathbf{u} \equiv \mathbf{u}$ for $\lambda \in \mathbb{F}_p^*$; such a hyperplane contains all points $\mathbf{v}$ where $(\mathbf{u}, \mathbf{v}) = 0$.

We define a bipartite graph $H(U,V)$ where the vertices on the left correspond to all hyperplanes in $\mathrm{PG}(\mathbb{F}_p, n-1)$, vertices on the right correspond to all points in $\mathrm{PG}(\mathbb{F}_p, n-1)$ and $\mathbf{u}$ and $\mathbf{v}$ are adjacent if $(\mathbf{u}, \mathbf{v}) = 0$. For $X \subseteq U$ and $Y \subseteq V$, we define $N(X)$ and $N(Y)$ to be their neighborhoods. We use $N(\mathbf{u})$ for the neighborhood of $\mathbf{u}$.

Definition 24 *Let n be a positive integer and p be a prime. Let U be the set of hyperplanes in $\mathrm{PG}(\mathbb{F}_p, n-1)$. We say that a set $X \subseteq U$ satisfies the* unique neighbor property *if for every $\mathbf{u} \in X$, there exists $\mathbf{v} \in N(\mathbf{u})$ such that $\mathbf{v}$ is not adjacent to $\mathbf{u}'$ for any $\mathbf{u}' \in X \setminus \{\mathbf{u}\}$.*

Lemma 25 *Let n be a positive integer and p be a prime. Let U be the set of hyperplanes in $\mathrm{PG}(\mathbb{F}_p, n-1)$. There exists a set $X \subseteq U$, $|X| = k$ satisfying the unique neighbor property if and only if there exists a k-sized family of $\mathbb{Z}_p^*$-matching vectors in $\mathbb{Z}_p^n$.*

Proof: Assume that $X = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ satisfies the unique neighbor property. Let $Y = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ be such that $\mathbf{v}_i$ is a unique neighbor of $\mathbf{u}_i$. This implies that $(\mathbf{u}_i, \mathbf{v}_i) = 0$ and $(\mathbf{u}_j, \mathbf{v}_i) \neq 0$ for $i \neq j$. Thus X, Y gives a $\mathbb{Z}_p^*$-matching vector family in $\mathbb{Z}_p^n$.

For the converse, let us start with a k-sized matching vector family $\mathcal{U}, \mathcal{V}$ in $\mathbb{Z}_p^n$. In case $k = 1$ the lemma holds trivially. We claim that if $k \geq 2$ then $\mathbf{u} \in \mathcal{U}$ implies that $\lambda\mathbf{u} \notin \mathcal{U}$ for any $\lambda \in \mathbb{F}_p^* \setminus \{1\}$. This is true since $(\mathbf{u}, \mathbf{v}) = 0$ implies $(\lambda\mathbf{u}, \mathbf{v}) = 0$, which would violate the definition of a matching vector family. Thus we can associate each $\mathbf{u} \in \mathcal{U}$ with a distinct hyperplane in $\mathrm{PG}(\mathbb{F}_p, n-1)$. Similarly, we can associate every $\mathbf{v} \in \mathcal{V}$ with a distinct point in $\mathrm{PG}(\mathbb{F}_p, n-1)$. It is easy to see that $\mathbf{v}_i$ is a unique neighbor of $\mathbf{u}_i$, hence the set $\mathcal{U}$ satisfies the unique neighbor property. ■

Corollary 26 *Let n be a positive integer and p be a prime. Let U be the set of hyperplanes in $\mathrm{PG}(\mathbb{F}_p, n-1)$. The size of the largest set $X \subseteq U$ that satisfies the unique neighbor property is exactly $k(p, n)$.*

The expansion of the graph $H(U, V)$ was analyzed by Alon using spectral methods [Alo86, theorem 2.3]. We use the rapid expansion of this graph to bound the size of the largest matching vector family.

Lemma 27 *Let $n \geq 2$ be an integer and p be a prime. Let U (V) be the set of hyperplanes (points) in $\mathrm{PG}(\mathbb{F}_p, n-1)$. Let $u = \frac{p^n-1}{p-1} = |U| = |V|$. For any nonempty set $X \subseteq U$ with $|X| = x$,*

$$|N(X)| \geq u - u^{\frac{n}{n-1}}/x. \tag{16}$$

Lemma 28 *Let n be a positive integer and p be a prime; then*

$$k(p, n) \leq 4p^{n/2} + 2. \tag{17}$$

Proof: If $n = 1$, inequality (17) holds trivially. We assume $n \geq 2$. Let $\mathcal{U} \subseteq U$, $\mathcal{V} \subseteq V$ be a matching family of size $k(p, n)$. Pick $X \subseteq \mathcal{U}$ of size $x > 0$. By (16),

$$|N(X)| \geq u - u^{\frac{n}{n-1}}/x.$$

Since every point in $\mathcal{U} \setminus X$ must contain a unique neighbor from $V \setminus N(X)$, we have

$$|\mathcal{U} \setminus X| \leq |V \setminus N(X)| \leq \frac{u^{\frac{n}{n-1}}}{x} \;\Rightarrow\; |\mathcal{U}| \leq \frac{u^{\frac{n}{n-1}}}{x} + x. \tag{18}$$

Note that the inequality in the right hand side of (18) holds for all positive integers x. Picking $x = \left\lceil u^{\frac{n}{2(n-1)}} \right\rceil$ gives

$$|\mathcal{U}| \leq 2\left\lceil u^{\frac{n}{2(n-1)}} \right\rceil \leq 2\left(\frac{p^n}{p-1}\right)^{\frac{n}{2(n-1)}} + 2 = 2\left(\frac{p}{p-1}\right)^{\frac{n}{2(n-1)}} p^{n/2} + 2 \leq 4p^{n/2} + 2,$$

where the last inequality is a simple calculation. ■

Equation (9) shows that for $n = O(1)$ we have $k(p, n) = \Omega\left(p^{(n-3)/2}\right)$. Thus lemma 28 is nearly tight when n is a constant and p grows to infinity. Note that for this setting of parameters, the linear-algebra bound (15) only gives $k(p, n) \leq O(p^{n-1})$.

6.2 The prime power case

Lemma 29 *Let n be a positive integer, p be a prime and $e \geq 2$. We have*

$$k(p^e, n) \leq p^{(e-1)n} k(p, n+1).$$

Proof: Assume for contradiction that we have a matching family $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}, \mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ of size $k > p^{(e-1)n} k(p, n+1)$ in $\mathbb{Z}_{p^e}^n$. For every $i \in [k]$, write $\mathbf{u}_i = \mathbf{u}'_i + p^{e-1}\mathbf{u}''_i$ where $\mathbf{u}'_i \in \mathbb{Z}_{p^{e-1}}^n$ and $\mathbf{u}''_i \in \mathbb{Z}_p^n$. By the pigeonhole principle, there are $k' > k(p, n+1)$ values of i which give the same vector $\mathbf{u}'_i \in \mathbb{Z}_{p^{e-1}}^n$, assume for convenience that the corresponding vectors in $\mathcal{U}$ are $\mathbf{u}_1, \ldots, \mathbf{u}_{k'}$ with matching vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{k'}$. We will use these vectors to construct a matching vector family of size $k' > k(p, n+1)$ in $\mathbb{Z}_p^{n+1}$, which gives a contradiction.

For each $i \in [k']$, we extend $\mathbf{u}''_i$ to a vector $\bar{\mathbf{u}}_i$ by appending 1 in the last coordinate. For every $i \in [k']$, write $\mathbf{v}_i = \mathbf{v}'_i + p\mathbf{v}''_i$ where $\mathbf{v}'_i \in \mathbb{Z}_p^n$ and $\mathbf{v}''_i \in \mathbb{Z}_{p^{e-1}}^n$. We extend $\mathbf{v}'_i$ to a vector $\bar{\mathbf{v}}_i$ by appending $(\mathbf{u}'_i, \mathbf{v}_i)/p^{e-1} \in \mathbb{Z}_p$ in the last coordinate (we will show that this ratio is in fact integral).

We claim that for all $i \in [k']$, $(\bar{\mathbf{u}}_i, \bar{\mathbf{v}}_i) = 0 \bmod p$. To see this, observe that

$$(\bar{\mathbf{u}}_i, \bar{\mathbf{v}}_i) = (\mathbf{u}''_i, \mathbf{v}'_i) + (\mathbf{u}'_i, \mathbf{v}_i)/p^{e-1}. \tag{19}$$

But we have

$$(\mathbf{u}_i, \mathbf{v}_i) = (\mathbf{u}'_i, \mathbf{v}_i) + p^{e-1}(\mathbf{u}''_i, \mathbf{v}_i) \equiv (\mathbf{u}'_i, \mathbf{v}_i) + p^{e-1}(\mathbf{u}''_i, \mathbf{v}'_i) = 0 \bmod p^e$$

From this we conclude that $(\mathbf{u}'_i, \mathbf{v}_i) \equiv 0 \bmod p^{e-1}$, and that $(\mathbf{u}''_i, \mathbf{v}'_i) + (\mathbf{u}'_i, \mathbf{v}_i)/p^{e-1} = 0 \bmod p$. From equation (19), we conclude that $(\bar{\mathbf{u}}_i, \bar{\mathbf{v}}_i) = 0 \bmod p$. Next we claim that $(\bar{\mathbf{u}}_j, \bar{\mathbf{v}}_i) \neq 0 \bmod p$ for $i \neq j \in [k']$. We have

$$(\bar{\mathbf{u}}_j, \bar{\mathbf{v}}_i) = (\mathbf{u}''_j, \mathbf{v}'_i) + (\mathbf{u}'_i, \mathbf{v}_i)/p^{e-1} \tag{20}$$

But, since $\mathbf{u}'_i = \mathbf{u}'_j$, we also have

$$(\mathbf{u}_j, \mathbf{v}_i) = (\mathbf{u}'_j, \mathbf{v}_i) + p^{e-1}(\mathbf{u}''_j, \mathbf{v}_i) \equiv (\mathbf{u}'_i, \mathbf{v}_i) + p^{e-1}(\mathbf{u}''_j, \mathbf{v}'_i) \not\equiv 0 \bmod p^e$$

which implies that $(\mathbf{u}''_j, \mathbf{v}'_i) + (\mathbf{u}'_i, \mathbf{v}_i)/p^{e-1} \not\equiv 0 \bmod p$. This shows that the vectors $\{\bar{\mathbf{u}}_j\}_{j=1}^{k'}$, $\{\bar{\mathbf{v}}_i\}_{i=1}^{k'}$ give a matching vector family of size $k' > k(p, n+1)$, which is a contradiction. ■

6.3 The composite case

Lemma 30 *Let m, n, and q be arbitrary positive integers such that $q|m$ and $(q, m/q) = 1$; then*

$$k(m, n) \leq (m/q)^n \, k(q, n).$$

Proof: Let us write $m/q = r$. Let $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$, $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ be a family of S-matching vectors of $\mathbb{Z}_m^n$, for some $S \subseteq \mathbb{Z}_m \setminus \{0\}$. For each vector $\mathbf{u} \in \mathbb{Z}_m^n$ we define the vectors $\mathbf{u}' \equiv \mathbf{u} \bmod q \in \mathbb{Z}_q^n$ and $\mathbf{u}'' \equiv \mathbf{u} \bmod r \in \mathbb{Z}_r^n$. Similarly, for each $\mathbf{v} \in \mathbb{Z}_m^n$ we define the vectors $\mathbf{v}' \equiv \mathbf{v} \bmod q \in \mathbb{Z}_q^n$ and $\mathbf{v}'' \equiv \mathbf{v} \bmod r \in \mathbb{Z}_r^n$. From the definition of a matching vector family, we have that

- For all $i \in [k]$, $(\mathbf{u}'_i, \mathbf{v}'_i) = 0$ and $(\mathbf{u}''_i, \mathbf{v}''_i) = 0$;
- For all $i, j \in [k]$ such that $i \neq j$, $(\mathbf{u}'_j, \mathbf{v}'_i) \neq 0$ or $(\mathbf{u}''_j, \mathbf{v}''_i) \neq 0$.

Assume $k > (m/q)^n\, k(q,n)$. By the pigeonhole principle, there exists a vector $\mathbf{u} \in \mathbb{Z}_r^n$ such that $\mathbf{u}_j'' = \mathbf{u}$ holds for $k' > k(q,n)$ values of $j \in [k]$. Let us assume that these values are $1, \ldots, k'$. Note that for any $i, j \in [k']$ we have $(\mathbf{u}_j'', \mathbf{v}_i'') = (\mathbf{u}_i'', \mathbf{v}_i'') = 0$. Hence, by the definition of a matching family, we must have

- For all $i \in [k']$, $(\mathbf{u}_i', \mathbf{v}_i') = 0$;
- For all $i, j \in [k']$ such that $i \neq j$, $(\mathbf{u}_j', \mathbf{v}_i') \neq 0$.

Thus vectors $\{\mathbf{u}_1', \ldots, \mathbf{u}_{k'}'\}$ and $\{\mathbf{v}_1', \ldots, \mathbf{v}_{k'}'\}$ form a matching family mod q of size larger than $k(q,n)$ which gives a contradiction. ■

Theorem 31 *Let m and n be arbitrary positive integers. Suppose p is a prime divisor of m; then*

$$k(m,n) \leq 5\frac{m^n}{p^{(n-1)/2}}.$$

Proof: Let p^e be the largest power of p which divides m. By lemmas 30, 29 and 28, we get

$$k(m,n) \leq \left(\frac{m}{p^e}\right)^n p^{(e-1)n}\left(4p^{(n+1)/2} + 2\right) \leq 5\frac{m^n}{p^{(n-1)/2}}$$

■

The above bound is weak when n and p are constants. Below we give another bound which handles the case of small n. We start with the case when $n = 1$.

Lemma 32 *Let $m \geq 2$ be an arbitrary positive integer; then*

$$k(m,1) \leq m^{O(1/\log\log m)} = m^{o_m(1)}.$$

Proof: Let $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}, \mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ be a family of $\mathbb{Z}_m \setminus \{0\}$-matching vectors in $\mathbb{Z}_m^1$. We treat every vector $\mathbf{u} \in \mathcal{U}$ as an integer and observe that for any $i \neq j \in [k]$, $\gcd(\mathbf{u}_i, m) \neq \gcd(\mathbf{u}_j, m)$. (Otherwise $(\mathbf{u}_i, \mathbf{v}_i) = 0$ would yield $(\mathbf{u}_j, \mathbf{v}_i) = 0$.) An application of a standard bound [HW85], saying the number of distinct divisors of m is at most $m^{O(1/\log\log m)}$ concludes the proof. ■

We now proceed to the case of general n.

Theorem 33 *Let m and n be arbitrary positive integers; then*

$$k(m,n) \leq m^{n-1+o_m(1)}.$$

Proof: Let $\mathcal{U} = \{\mathbf{u}_j\}_{j\in[k]}, \mathcal{V} = \{\mathbf{v}_i\}_{i\in[k]}$ be a family of $\mathbb{Z}_m \setminus \{0\}$-matching vectors in $\mathbb{Z}_m^n$. Given a vector $\mathbf{u} \in \mathbb{Z}_m^n$, we define the $\mathbb{Z}_m$-orbit of $\mathbf{u}$ to be the set of all vectors that can be written as $\lambda\mathbf{u}$ for $\lambda \in \mathbb{Z}_m$. The orbits are not disjoint. We claim that all of $\mathbb{Z}_m^n$ can be covered by no more than $m^n/\phi(m)$ orbits, where $\phi(\cdot)$ denotes the Euler function. Furthermore, we claim that each orbit can contribute at most $k(m,1)$ vectors to $\mathcal{U}$.

Let $U \subseteq \mathbb{Z}_m^n$ denote the set of all vectors $\mathbf{u}$ such that the GCD of all coordinates of $\mathbf{u}$ is in $\mathbb{Z}_m^*$. Any vector $\mathbf{u}' \in \mathbb{Z}_m^n$ can be written as $\lambda\mathbf{u}$ for $\mathbf{u} \in U$ and $\lambda \in \mathbb{Z}_m$. Thus the orbits of vectors in U cover all of $\mathbb{Z}_m^n$. For $\mathbf{u}, \mathbf{u}' \in U$, we say that $\mathbf{u}' \equiv \mathbf{u}''$ if $\mathbf{u}''$ lies in the $\mathbb{Z}_m$ orbit of $\mathbf{u}'$. It is easy to see that this is indeed an equivalence relation on U, which divides U into equivalence classes of size $\phi(m)$. Here the equivalence class of a vector $\mathbf{u}$ is $\mathbb{Z}_m^* \cdot \mathbf{u}$. Thus if we pick $U' \subseteq U$ which contains a single representative of each equivalence class, then the orbits of U' contain all of $\mathbb{Z}_m^n$. Thus we have $|U'| = \frac{|U|}{\phi(m)} \leq \frac{m^n}{\phi(m)}$.

Now consider the orbit of a vector $\mathbf{u} \in \mathcal{U}$. Assume that it contributes the vectors $\mathbf{u}_1 = \lambda_1 \mathbf{u}, \ldots, \mathbf{u}_t = \lambda_t \mathbf{u}$ to $\mathcal{U}$ where $\lambda_i \in \mathbb{Z}_m$. Assume that the matching vectors in $\mathcal{V}$ are $\mathbf{v}_1, \ldots, \mathbf{v}_t$. Then it is easy to see that $\mathcal{U}' = \{\lambda_1, \ldots, \lambda_t\}$ and $\mathcal{V}' = \{(\mathbf{u}, \mathbf{v}_1), \ldots, (\mathbf{u}, \mathbf{v}_t)\}$ are a matching vector family in one dimension, so that $t \leq k(m, 1)$. Thus we conclude that

$$k(m,n) \leq \frac{m^n}{\phi(m)} k(m,1) \leq m^{n-1+o_m(1)}$$

using a standard lower bound [HW85], $\phi(m) \geq \Omega(m/\log\log m)$ and lemma 32. ■

The upper bound of lemma 28 (that applies only when m is prime) is substantially stronger than the bounds of theorems 31 and 33. We feel that latter bounds can be improved. Specifically, we propose the following

Conjecture 34 *Let m and n be arbitrary positive integers; then $k(m,n) \leq O\left(m^{n/2}\right)$.*

We discuss the implications of the conjecture for the lower bounds for MV codes in remark 37.

7 Lower bounds for MV codes

We now translate the bounds on matching vector families from the previous section to bounds on the encoding length of matching vector codes. We argue that any family of (non-binary) matching vector codes, (i.e., codes that for some m and n, encode $k(m,n)$-long messages to m^n-long codewords) has a multiplicative encoding blow-up (*stretch*) of at least $2^{\Omega(\sqrt{\log k})}$.

Theorem 35 *Consider an infinite family of Matching Vector codes $C_\ell : \mathbb{F}_q^k \to \mathbb{F}_q^N$ for $\ell \in \mathbb{N}$, where $k = k(\ell)$ and $N = N(\ell)$ go to infinity with ℓ. For large enough ℓ, we have*

$$N \geq k 2^{0.4\sqrt{\log k}}.$$

Proof: For each ℓ, we have a family of matching vectors in $\mathbb{Z}_m^n$ where m, n depend on ℓ. We have $N = m^n$ while $k \leq k(m,n)$. First assume that $n > \sqrt{\log N}$. Then by theorem 31 with p a prime divisor of m, we have

$$k \leq \frac{5m^n}{p^{(n-1)/2}} \leq \frac{5N}{2^{0.5\sqrt{\log N} - 1/2}} \leq \frac{N}{2^{0.4\sqrt{\log N}}}$$

where the last inequality holds for large enough N, and hence for all large ℓ. Hence assume that $n \leq \sqrt{\log N}$ so that $m \geq 2^{\sqrt{\log N}}$. As ℓ goes to infinity, N and hence m go to infinity. So for large enough ℓ, theorem 33 gives $k(m,n) \leq m^{n-1+o_m(1)} \leq m^{n-0.9}$. Hence

$$k \leq \frac{m^n}{m^{0.9}} \leq \frac{N}{2^{0.9\sqrt{\log N}}}.$$

Thus $k \leq \frac{N}{2^{0.4\sqrt{\log N}}}$ for large enough ℓ. This implies that $N \geq k 2^{0.4\sqrt{\log k}}$ for large enough ℓ. ■

One can generalize theorem 35 to get a similar statement for binary MV codes (i.e., codes obtained by a concatenation of a non-binary MV code with an asymptotically good binary code).

Theorem 36 *Let $\{m_\ell\}$ and $\{n_\ell\}$, $\ell \in \mathbb{N}$ be two arbitrary sequences of positive integers, such that ${m_\ell}^{n_\ell}$ monotonically grows to infinity. Consider an infinite family of binary codes $C_\ell : \mathbb{F}_2^{k_\ell} \to \mathbb{F}_2^{N_\ell}$ for $\ell \in \mathbb{N}$, where each code C_ℓ is obtained via a concatenation of a MV code encoding $k(m_\ell, n_\ell)$-long messages to $m_\ell^{n_\ell}$-long codewords over $\mathbb{F}_{q_\ell}$, (here $q_\ell = 2^t$ is the smallest integer such that $m_\ell \mid 2^t - 1$) with an asymptotically good binary code of some fixed rate; then for large enough ℓ the stretch of C_ℓ is at least $2^{\Omega(\sqrt{\log k_\ell})}$.*

Proof: Pick a sufficiently large value of ℓ. Consider two cases

- $n_\ell \neq 1$. First observe that $k(m_\ell, n_\ell) \geq m_\ell$, e.g., take $\mathcal{U} = \{\mathbf{u}_\alpha\}_{\alpha \in \mathbb{Z}_m}$, $\mathcal{V} = \{\mathbf{v}_\beta\}_{\beta \in \mathbb{Z}_m}$, where

  $$\mathbf{u}_\alpha = (1, \alpha, 0, \ldots, 0) \text{ and } \mathbf{v}_\beta = (\beta, -1, 0, \ldots, 0).$$

  Next note that by theorem 35 the stretch of the non-binary code is at least $2^{\Omega\left(\sqrt{\log k(m_\ell, n_\ell)}\right)}$, and the concatenation with a binary code can only increase the stretch. Finally note that the dimension k_ℓ of the binary code satisfies

  $$k_\ell \leq k(m_\ell, n_\ell) m_\ell \leq k^2(m_\ell, n_\ell).$$

  Thus $2^{\Omega\left(\sqrt{\log k(m_\ell, n_\ell)}\right)} \geq 2^{\Omega(\sqrt{\log k_\ell})}$, for an appropriately chosen constant in Ω-notation.

- $n_\ell = 1$. Set $k' = k(m_\ell, n_\ell)$. Be lemma 32, $k' = m_\ell^{o(1)}$. Note that $k_\ell = k' t$ and $N_\ell = \Omega(m_\ell \cdot t)$, for some $t \leq m_\ell$. These conditions yield $N_\ell \geq \Omega\left(k_\ell^{3/2}\right)$.

■

7.1 Comparison with RM LDCs

Here we observe that it is possible to construct binary RM LDCs that have a stretch of $2^{O(\sqrt{\log k})}$ and query complexity of $(\log k)^{O(\sqrt{\log k})}$. By formula (12) the rate of an RM LDC specified by parameters n and $d = q/2$ is given by

$$k/N \geq \Omega\left(\binom{n+d}{d} / q^n\right).$$

Combining this with $\binom{n+d}{d} \geq (d/n)^n$, and using $d = q/2$ we get

$$k/N \geq \Omega((1/2n)^n).$$

Thus in order to have rate above $1/2^{O(\sqrt{\log k})}$ it suffices to have

$$n = O\left(\sqrt{\log k} / \log \log k\right). \tag{21}$$

Given k we choose n to be the largest integer satisfying (21). Next we choose d to be the smallest integer satisfying $k \leq \binom{n+d}{d} \log q$. One can easily check that this yields $d = (\log k)^{O(\sqrt{\log k})}$, giving an RM LDC with desired parameters.

Remark 37 It is not hard to verify that if conjecture 34 holds; then any MV code must have length $N = \Omega(k^2)$. This would imply that RM LDCs improve on MV codes once $r \geq \log^2 k$, (by an argument similar to the one above). Since MV codes improve on RM codes for $r \leq \log k / (\log \log k)^{O(1)}$, this would give a clearer picture of the comparison between the two families of codes.

8 Conclusions

In this work we developed a new view of matching vector codes and uncovered certain similarities between MV codes and classical Reed Muller codes. Our view allowed us to obtain a deeper insight into the power and limitations of MV codes. We showed that similarly to Reed Muller codes MV codes constitute a rich code class containing codes of both constant and growing query complexities, capable of tolerating large amounts of noise. We showed that for query complexity $r \leq \log k/(\log\log k)^{O(1)}$ MV codes are superior to RM LDCs and for $r \geq (\log k)^{\Omega(\sqrt{\log k})}$ MV codes are inferior to RM codes. There are many questions that are left open by our work. We elaborate on some of them.

- It is very interesting to see if one can get MV codes that improve upon RM codes for values of $r \geq \log k/(\log\log k)^{O(1)}$. This calls for constructions of bounded matching families in $\mathbb{Z}_m^n$, where m is comparable to (or larger than) n.
- It is very interesting to prove (or disprove) conjecture 34. A simple case where it is open is when n is a constant and m is a product of two nearly equal primes.
- Our results show that MV codes share many properties of RM codes. We would like to know if MV codes are (or can be made) locally correctable [BIW07, Dvi10]. Note that to date, RM LDCs constitute the only known class of locally correctable codes.

One aspect of locally decodable codes that has not been discussed in this paper is the strong explicitness of the code construction.

Acknowledgements

We would like to thank anonymous reviewers whose detailed comments have helped us substantially improve the presentation.

References

[Alo86] Noga Alon. Eigenvalues, geometric expanders, sorting in rounds, amd Ramsey theory. *Combinatorica*, 6:207–219, 1986.

[BBR94] David A. Barrington, Richard Beigel, and Steven Rudich. Representing Boolean functions as polynomials modulo composite numbers. *Computational Complexity*, 4:67–382, 1994.

[BDWY11] Boaz Barak, Zeev Dvir, Avi Wigderson, and Amir Yehudayoff. Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes. In *43nd ACM Symposium on Theory of Computing (STOC)*, 2011.

[BET10a] Avraham Ben-Aroya, Klim Efremenko, and Amnon Ta-Shma. Local list decoding with a constant number of queries. In *51st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 715–722, 2010.

[BET10b] Avraham Ben-Aroya, Klim Efremenko, and Amnon Ta-Shma. A note on amplifying the error-tolerance of locally decodable codes. In *Electronic Colloquium on Computational Complexity (ECCC)*, TR10-134, 2010.

[BF90] Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In *7th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 415 of Lecture Notes in Computer Science, pages 37–48. Springer, Berlin, Heidelberg, 1990.

[BF98] Laszlo Babai and Peter Frankl. *Linear algebra methods in combinatorics.* 1998.

[BFLS91] Laszlo Babai, Lance Fortnow, Leonid Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *23rd ACM Symposium on Theory of Computing (STOC)*, pages 21–31, 1991.

[BIK05] Amos Beimel, Yuval Ishai, and Eyal Kushilevitz. General constructions for information-theoretic private information retrieval. *Journal of Computer and System Sciences*, 71:213–247, 2005.

[BIKR02] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Jean-Francios Raymond. Breaking the $O\left(n^{1/(2k-1)}\right)$ barrier for information-theoretic private information retrieval. In *43rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 261–270, 2002.

[BIW07] Omer Barkol, Yuval Ishai, and Enav Weinreb. On locally decodable codes, self-correctable codes, and t-private PIR. In *International Workshop on Randomization and Computation (RANDOM)*, pages 311–325, 2007.

[CGKS98] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. *Journal of the ACM*, 45:965–981, 1998.

[DGY10] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. In *51st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 705–714, 2010.

[DS06] Zeev Dvir and Amir Shpilka. Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434, 2006.

[Dvi10] Zeev Dvir. On matrix rigidity and locally self-correctable codes. In *25th IEEE Computational Complexity Conference (CCC)*, pages 102–113, 2010.

[Efr09] Klim Efremenko. 3-query locally decodable codes of subexponential length. In *41st ACM Symposium on Theory of Computing (STOC)*, pages 39–44, 2009.

[For66] David Forney. Generalized minimum distance decoding. *IEEE Transactions on Information Theory*, 12:125–131, 1966.

[GKST02] Oded Goldreich, Howard Karloff, Leonard Schulman, and Luca Trevisan. Lower bounds for locally decodable codes and private information retrieval. In *17th IEEE Computational Complexity Conference (CCC)*, pages 175–183, 2002.

[GM11] Anna Gal and Andrew Mills. Three query locally decodable codes with higher correctness require exponential length. In *28th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 673–684, 2011.

[Gop06] Parikshit Gopalan. *Computing with Polynomials over Composites.* PhD thesis, Georgia Institute of Technology, 2006.

[Gop09] Parikshit Gopalan. A note on Efremenko's locally decodable codes. Electronic Colloquium on Computational Complexity (ECCC) TR09-069, 2009.

[Gro00] Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit Ramsey graphs. *Combinatorica*, 20:71–86, 2000.

[Gro02] Vince Grolmusz. Constructing set-systems with prescribed intersection sizes. *Journal of Algorithms*, 44:321–337, 2002.

[HW85] G. Hardy and E. Wright. *An introduction to the theory of numbers.* Clarendon Press, Oxford, 1985.

[IS10] Toshiya Itoh and Yasuhiro Suzuki. New constructions for query-efficient locally decodable codes of subexponential length. *IEICE Transactions on Information and Systems*, pages 263–270, 2010.

[KdW04] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *Journal of Computer and System Sciences*, 69:395–420, 2004.

[KSY11] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. In *43nd ACM Symposium on Theory of Computing (STOC)*, 2011.

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *32nd ACM Symposium on Theory of Computing (STOC)*, pages 80–86, 2000.

[KY09] Kiran S. Kedlaya and Sergey Yekhanin. Locally decodable codes from nice subsets of finite fields and prime factors of Mersenne numbers. *SIAM Journal on Computing*, 38:1952–1969, 2009.

[Lip90] Richard Lipton. Efficient checking of computations. In *7th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 415 of Lecture Notes in Computer Science, pages 207–215. Springer, Berlin, Heidelberg, 1990.

[LN83] Rudolf Lidl and Harald Niederreiter. *Finite Fields.* Cambridge University Press, Cambridge, 1983.

[MFL$^+$10] Yeow Meng Chee, Tao Feng, San Ling, Huaxiong Wang, and Liangfeng Zhang. Query-efficient locally decodable codes of subexponential length. In *Electronic Colloquium on Computational Complexity (ECCC)*, TR10-173, 2010.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes.* North Holland, Amsterdam, New York, 1977.

[PS94] Alexander Polishchuk and Daniel Spielman. Nearly-linear size holographic proofs. In *26th ACM Symposium on Theory of Computing (STOC)*, pages 194–203, 1994.

[Rag07] Prasad Raghavendra. A note on Yekhanin's locally decodable codes. In *Electronic Colloquium on Computational Complexity (ECCC)*, TR07-016, 2007.

[Sga99] Jiri Sgall. Bounds on pairs of families with restricted intersections. *Combinatorica*, 19:555–566, 1999.

[Sho05] V. Shoup. *A computational introduction to number theory and algebra.* Cambridge University Press, Cambridge, 2005.

[STV99] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. In *39th ACM Symposium on Theory of Computing (STOC)*, pages 537–546, 1999.

[Sud92] Madhu Sudan. *Efficient checking of polynomials and proofs and the hardness of approximation problems.* PhD thesis, University of California at Berkeley, 1992.

[Sud09] Madhu Sudan. Personal communication, 2009.

[SY11] Shubhangi Saraf and Sergey Yekhanin. Noisy interpolation of sparse polynomials, and applications. In *26th IEEE Computational Complexity Conference (CCC)*, 2011.

[WdW05] Stephanie Wehner and Ronald de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *32nd International Colloquium on Automata, Languages and Programming (ICALP)*, volume 3580 of Lecture Notes in Computer Science, pages 1424–1436. Springer, Berlin, Heidelberg, 2005.

[Woo07] David Woodruff. New lower bounds for general locally decodable codes. In *Electronic Colloquium on Computational Complexity (ECCC)*, TR07-006, 2007.

[Woo08] David Woodruff. Corruption and recovery-efficient locally decodable codes. In *International Workshop on Randomization and Computation (RANDOM)*, pages 584–595, 2008.

[WY05] David Woodruff and Sergey Yekhanin. A geometric approach to information theoretic private information retrieval. In *20th IEEE Computational Complexity Conference (CCC)*, pages 275–284, 2005.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM*, 55:1–16, 2008.

[Yek11] Sergey Yekhanin. Locally decodable codes. *Foundations and trends in theoretical computer science*, 2011. to appear.

A Grolmusz matching family

Let $m = \prod_{i=1}^{t} p_i$ be a product of distinct primes. Recall that the canonical set in $\mathbb{Z}_m$ is the set of all non-zero s such that for every $i \in [t]$, $s \in \{0, 1\} \bmod p_i$. Our goal here is to prove the following

Lemma 11 *Let $m = \prod_{i=1}^{t} p_i$ be a product of distinct primes. Let w be a positive integer. Suppose integers $\{e_i\}$, $i \in [t]$ are such that for all i, we have $p_i^{e_i} > w^{1/t}$. Let $d = \max_i p_i^{e_i}$, and $h \geq w$ be arbitrary. Let S be the canonical set modulo m; then there exists an $\binom{h}{w}$-sized family of S-matching vectors in $\mathbb{Z}_m^n$, where $n = \binom{h}{\leq d}$.*

Our construction of the matching family is modeled along the lines of Grolmusz's construction of a set system with restricted intersections modulo composites [Gro00, Gro02]. His construction uses the low-degree OR representations of Barrington et al. [BBR94]. However, we will use Lemma 39 to bypass the set system and go directly to the matching family from polynomials. In addition to being more direct, this also gives a slightly larger collection of vectors.

Definition 38 *Let $S \subseteq \mathbb{Z}_m \setminus \{0\}$. We say that a set of polynomials $\mathcal{F} = \{f_1, \dots, f_k\} \subseteq \mathbb{Z}_m[z_1, \dots, z_h]$ and a set of points $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_k\} \subseteq \mathbb{Z}_m^h$ form a polynomial S-matching family of size k if*

- *For all $i \in [k]$, $f_i(\mathbf{x}_i) = 0$;*
- *For all $i, j \in [k]$ such that $i \neq j$, $f_j(\mathbf{x}_i) \in S$.*

Let $\mathcal{F}, \mathcal{X}$ be a k-sized polynomial matching family. For $i \in [k]$, let $\operatorname{supp}(f_i)$ denote the set of monomials in the support of the polynomial f_i. We define $\operatorname{supp}(\mathcal{F}) = \bigcup_{i=1}^{k} \operatorname{supp}(f_i)$ and $\dim(\mathcal{F}) = |\operatorname{supp}(\mathcal{F})|$. The following lemma was observed by Sudan [Sud09].

Lemma 39 *An k-sized polynomial S-matching family $\mathcal{F}, \mathcal{X}$ over $\mathbb{Z}_m$ yields a k-sized S-matching family $\mathcal{U}, \mathcal{V}$ in $\mathbb{Z}_m^n$, where $n = \dim(\mathcal{F})$.*

Proof: Let $\mathrm{mon}_1, \dots, \mathrm{mon}_n$ be the set of monomials in $\operatorname{supp}(\mathcal{F})$. For every $j \in [k]$ we have

$$f_j(z_1 \dots, z_h) = \sum_{l=1}^{n} c_{jl} \cdot \mathrm{mon}_l.$$

We define the vector $\mathbf{u}_j$ to be the n-dimensional vector of coefficients $\{c_{jl}\}$ of the polynomial f_j. Similarly, for $i \in [k]$, we define the vector $\mathbf{v}_i$ to be the vector of evaluations of monomials $\mathrm{mon}_1, \dots, \mathrm{mon}_n$ at the point $\mathbf{x}_i$. It is easy to check that for all $i, j \in [k]$, $(\mathbf{u}_j, \mathbf{v}_i) = f_j(\mathbf{x}_i)$ and hence the sets $\mathcal{U}, \mathcal{V}$ indeed form an S-matching family. ■

In what follows we assume that parameters $m, t, \{p_i\}_{i \in [t]}, \{e_i\}_{i \in [t]}, w, h$, and the set S satisfy the precondition of lemma 11. Theorem 2.16 in [Gop06] yields

Lemma 40 *For every $i \in [t]$, there is an explicit multilinear polynomial $f_i(z_1, \dots, z_h) \in \mathbb{Z}_{p_i}[z_1, \dots, z_h]$ where $\deg f_i \leq p_i^{e_i} - 1$ such that for $\mathbf{x} \in \{0, 1\}^h$, we have*

$$f_i(\mathbf{x}) \equiv \begin{cases} 0 \bmod p_i, & \textit{if } \sum_{l=1}^{h} \mathbf{x}(l) \equiv w \bmod p_i^{e_i}, \\ 1 \bmod p_i, & \textit{otherwise.} \end{cases}$$

Corollary 41 *There is an explicit multilinear polynomial $f(z_1, \ldots, z_h) \in \mathbb{Z}_m[z_1, \ldots, z_n]$, $\deg(f) \le \max_{i\in[t]} (p_i^{e_i} - 1)$ such that for all $\mathbf{x} \in \{0,1\}^h$, we have*

$$f(\mathbf{x}) = \begin{cases} 0 \bmod m, & \textit{if } \sum_{l=1}^{h} \mathbf{x}(l) = w, \\ s \bmod m, \text{ for } s \in S, & \textit{if } \sum_{l=1}^{h} \mathbf{x}(l) < w, \end{cases}$$

where coordinates of $\mathbf{x}$ *are summed as integers.*

Proof: Define the polynomial f so that for all $i \in [t]$, $f(z_1, \ldots, z_h) \equiv f_i(z_1, \ldots, z_h) \bmod p_i$. Clearly, $\deg(f) \le \max_{i\in[t]} \deg f_i$. We claim that the polynomial f satisfies the above requirement. Observe that by the Chinese remainder theorem and lemma 40

$$f(\mathbf{x}) = 0 \bmod m \quad \text{iff} \quad \text{for all } i \in [t],\ \sum_{l=1}^{h} \mathbf{x}(l) \equiv w \bmod p_i^{e_i}.$$

The right hand side can be equivalently written as

$$\sum_{l=1}^{h} \mathbf{x}(l) \equiv w \bmod \prod_i p_i^{e_i}.$$

Note that for all $i \in [t]$, $p_i^{e_i} > w^{1/t}$. Hence $m = \prod_i p_i^{e_i} > w$. Thus whenever the integer sum $\sum_{l=1}^{h} \mathbf{x}(l) < w$, we have $\sum_{l=1}^{h} \mathbf{x}(l) \not\equiv w \bmod m$, which proves the claim. ■

Proof of lemma 11: For every $T \subseteq [h]$ of size w, define the polynomial f_T wherein the polynomial f from corollary 41, we set $z_j = 0$ for $j \notin T$ (but z_j stays untouched for $j \in T$). Define $\mathbf{x}_T \in \{0,1\}^h$ to be the indicator of the set T. Viewing vectors $\mathbf{x} \in \{0,1\}^h$ as indicator vectors $\mathbf{x}_L$ for sets $L \subseteq [h]$, it is easy to check that for all $T, L \in [h]$, $f_T(\mathbf{x}_L) = f(\mathbf{x}_{L\cap T})$. Combining this with Corollary 41 gives

- For all $T \subseteq [h]$, where $|T| = w$, $f_T(\mathbf{x}_T) = f(\mathbf{x}_T) \equiv 0 \bmod m$,
- For all $T \ne L \subseteq [h]$, where $|T| = |L| = w$, $f_T(\mathbf{x}_L) = f(\mathbf{x}_{L\cap T}) \in S \bmod m$,

where the second bullet follows from the observation that $|L \cap T| \le w - 1$. Thus the set of polynomials $\mathcal{F} = \{f_T\}_{T\subseteq[h],|T|=w}$ and points $\mathcal{X} = \{\mathbf{x}_T\}_{T\subseteq[h],|T|=w}$ form a polynomial S-matching family.

It is clear that $k = |\mathcal{F}| = \binom{h}{w}$. To bound n, we note that $\deg(f) \le d$ and f is multilinear. Thus we can take $\mathrm{supp}(\mathcal{F})$ to be the set of all multilinear monomials in $z_1, \ldots, z_h$ of degree at most d. Thus $\dim(\mathcal{F}) = \binom{h}{\le d}$. ■

B Improving the locality of the decoder

As we have mentioned in remark 4 is certain cases, relying on the special properties of the integer m and the set $S \subseteq \mathbb{Z}_m \setminus \{0\}$ it is possible to reduce the query complexity of the decoder given by proposition 3 below $|S| + 1$. In this section we present a general sufficient condition that makes such a reduction possible, generalizing the argument from [Yek08]. We restrict our attention to fields of characteristic 2.

Definition 42 *Fix an integer $t \ge 1$ and let $m = 2^t - 1$. Given an integer s, define* $\mathrm{wt}(s)$ *to be the number of non-zero digits in its base 2 expansion. For a set $S \subseteq \mathbb{Z}_m \setminus \{0\}$, define* $\mathrm{wt}(S) = \max_{s\in S} \mathrm{wt}(s)$. *Finally, given a polynomial $f(y) = \sum_i f_i y^i \in \mathbb{F}_{2^t}[y]$, define its linearity degree as* $\mathrm{ldeg}(f) = \max_{i:f_i\ne 0} \mathrm{wt}(i)$.

Note that $\mathbb{F}_{2^t}$ is a t-dimensional vector space over $\mathbb{F}_2$, thus we can talk about subspaces of $\mathbb{F}_{2^t}$. The following lemma can be found in [LN83].

Lemma 43 *Let $f(y) = \sum_i f_i y^i \in \mathbb{F}_{2^t}[y]$. Let $L \subseteq \mathbb{F}_{2^t}$ be a subspace. Suppose $\dim L \geq \operatorname{ldeg}(f) + 1$; then*

$$\sum_{a \in L \setminus \{0\}} f(a) = f(0). \tag{22}$$

The following proposition gives a general condition allowing for a reduction in the query complexity of the decoder given by proposition 3. We use the notation from section 3.

Proposition 44 *Let $q = 2^t$ for $t \geq 1$ and let $m = q - 1$. Let $\mathcal{U}, \mathcal{V}$ be a family of S-matching vectors in $\mathbb{Z}_m^n$, where $|\mathcal{U}| = |\mathcal{V}| = k$ and $\operatorname{wt}(S) \leq \ell - 1$. There exists a q-ary linear code encoding k-long messages to m^n-long codewords that is $(2^\ell - 1, \delta, (2^\ell - 1)\delta)$-locally decodable for all δ.*

Proof: The encoding procedure is specified by formula (5). Recall that g is a generator of $\mathbb{C}_m = \mathbb{F}_{2^t}^*$. To recover the value $\mathbf{x}_i$,

1. The decoder picks a linear subspace $L = \{a_0, a_1, \ldots, a_{2^\ell - 1}\} \subseteq \mathbb{F}_{2^t}$ of dimension ℓ. Let integers $\lambda_1, \ldots, \lambda_{2^\ell - 1}$ be such that $g^{\lambda_j} = a_j$ for all $j \in \{1, \ldots, 2^\ell - 1\}$.

2. The decoder picks $\mathbf{w} \in \mathbb{Z}_m^n$ uniformly at random, and queries the (corrupted) $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F at the locations $\left\{g^{\mathbf{w} + \lambda_j \mathbf{v}_i} \mid j \in \{1, \ldots, 2^\ell - 1\}\right\}$ to obtain the values $c_1, \ldots, c_{2^\ell - 1}$.

3. The decoder returns $\sum_{j=1}^{2^\ell - 1} c_j / g^{(\mathbf{u}_i, \mathbf{w})}$.

To prove the correctness of the decoder, observe that by equation (7), the noiseless $M_{\mathbf{w},\mathbf{v}_i}$-evaluation of F is given by

$$f(y) = \mathbf{x}(i) \cdot g^{(\mathbf{u}_i, \mathbf{w})} + \sum_{s \in S} \left(\sum_{j \,:\, (\mathbf{u}_j, \mathbf{v}_i) = s} \mathbf{x}(j) \cdot g^{(\mathbf{u}_j, \mathbf{w})} \right) y^s.$$

Each query location is uniformly random over the codeword, hence with probability $(2^\ell - 1)\delta$ none of the locations are corrupted. Assuming this holds, we have $c_j = f(a_j)$ for all j. Since $\operatorname{supp}(f) \subseteq S \cup \{0\}$ and $\operatorname{wt}(S) \leq \ell - 1$ we have $\operatorname{ldeg}(f) \leq \ell - 1$. Hence by lemma 43,

$$\sum_j c_j = \sum_{a \in L \setminus \{0\}} f(a) = f(0) = \mathbf{x}(i) \cdot g^{(\mathbf{u}_i, \mathbf{w})}.$$

■

Remark 45 Proposition 44 generalizes the code construction from [Yek08, Rag07]. Those works choose m to be a Mersenne prime, and S to be the multiplicative subgroup generated by 2 in $\mathbb{Z}_m^*$. Such choice clearly yields $\operatorname{wt}(S) = 1$. By proposition 44 this translates to a large saving in query complexity.