Explicit Capacity-Achieving List-Decodable Codes
or
Decoding Folded Reed-Solomon Codes up to their Distance

Venkatesan Guruswami[*] Atri Rudra[†]

Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195

Abstract

For every $0 < R < 1$ and $\varepsilon > 0$, we present an explicit construction of error-correcting codes of rate R that can be list decoded in polynomial time up to a fraction $(1 - R - \varepsilon)$ of errors. These codes achieve the "capacity" for decoding from *adversarial* errors, i.e., achieve the *optimal* trade-off between rate and error-correction radius. At least theoretically, this meets one of the central challenges in coding theory.

Prior to this work, explicit codes achieving capacity were not known for *any* rate R. In fact, our codes are the first to beat the error-correction radius of $1 - \sqrt{R}$, that was achieved for Reed-Solomon codes in [11], for all rates R. (For rates $R < 1/16$, a recent breakthrough by Parvaresh and Vardy [14] improved upon the $1 - \sqrt{R}$ bound; for $R \to 0$, their algorithm can decode a fraction $1 - O(R \log(1/R))$ of errors.)

Our codes are simple to describe — they are certain *folded Reed-Solomon codes*, which are in fact *exactly* Reed-Solomon (RS) codes, but viewed as a code over a larger alphabet by careful bundling of codeword symbols. Given the ubiquity of RS codes, this is an appealing feature of our result, since the codes we propose are not too far from the ones in actual use.

The main insight in our work is that some carefully chosen folded RS codes are "compressed" versions of a related family of Parvaresh-Vardy codes. Further, the decoding of the folded RS codes can be reduced to list decoding the related Parvaresh-Vardy codes. The alphabet size of these folded RS codes is polynomial in the block length. This can be reduced to a (large) constant using ideas concerning "list recovering" and expander-based codes from [9, 10]. Concatenating the folded RS codes with suitable inner codes also gives us polytime constructible binary codes that can be efficiently list decoded up to the Zyablov bound.

[*] Research supported in part by NSF grant CCF-0343672 and a Sloan Research Fellowship.

[†] Research supported by NSF grant CCF-0343672.

1 Introduction

1.1 Background and Context

Error-correcting codes enable reliable communication of messages over a noisy channel by cleverly introducing redundancy into the message to encode it into a codeword, which is then transmitted on the channel. This is accompanied by a decoding procedure that recovers the correct message even when several symbols in the transmitted codeword are corrupted. In this work, we focus on the adversarial or worst-case model of errors — we do not assume anything about how the errors and error locations are distributed beyond an upper bound on the total number of errors that may be caused. The central trade-off in this theory is the one between the amount of redundancy needed and the fraction of errors that can be corrected. The redundancy is measured by the *rate* of the code, which is the ratio of the the number of information symbols in the message to that in the codeword — thus, for a code with encoding function $E : \Sigma^k \to \Sigma^n$, the rate equals k/n. The *block length* of the code equals n, and Σ is its *alphabet*.

The goal in decoding is to find, given a noisy received word, the actual codeword that it could have possibly resulted from. If we target correcting a fraction ρ of errors (ρ will be called the error-correction radius), then this amounts to finding codewords within (normalized Hamming) distance ρ from the received word. We are guaranteed that there will be a unique such codeword provided the distance between *every* two distinct codewords is at least 2ρ, or in other words the relative distance of the code is at least 2ρ. However, since the relative distance δ of a code must satisfy $\delta \leqslant 1 - R$ where R is the rate of the code (by the Singleton bound), the best trade-off between ρ and R that unique decoding permits is $\rho = \rho_U(R) = (1 - R)/2$. But this is an overly pessimistic estimate of the error-correction radius, since the way Hamming spheres pack in space, for *most* choices of the received word there will be at most one codeword within distance ρ from it even for ρ much greater than $\delta/2$. Therefore, *always* insisting on a unique answer will preclude decoding most such received words owing to a few pathological received words that have more than one codeword within distance roughly $\delta/2$ from them.

A notion called list decoding, that dates back to the late 1950's [3, 19], provides a clean way to get around this predicament, and yet deal with worst-case error patterns. Under list decoding, the decoder is required to output a list of all codewords within distance ρ from the received word. Let us call a code C *(ρ, L)-list decodable* if the number of codewords within distance ρ of any received word is at most L. To obtain better trade-offs via list decoding, we need (ρ, L)-list decodable codes where L is bounded by a polynomial function of the block length, since this an *a priori* requirement for polynomial time list decoding. How large can ρ be as a function of R for which such (ρ, L)-list decodable codes exist? A standard random coding argument shows that we can have $\rho \geqslant 1 - R - o(1)$ over large enough alphabets, cf. [20, 4], and a simple counting argument shows that ρ must be at most $1 - R$. Therefore the *list decoding capacity*, i.e., the information-theoretic limit of list decodability, is given by the trade-off $\rho_{\text{cap}}(R) = 1 - R = 2\rho_U(R)$. Thus list decoding holds the promise of correcting *twice* as many errors as unique decoding, for *every* rate.

The above-mentioned list decodable codes are non-constructive. In order to realize the potential of list decoding, one needs explicit constructions of such codes, and on top of that, polynomial time algorithms to perform list decoding. After essentially no progress in this direction in over 30 years, the work of Sudan [17] and improvements to it in [11], achieved efficient list decoding up to $\rho_{\text{GS}}(R) = 1 - \sqrt{R}$ errors for an important family of codes called Reed-Solomon codes. Note that $1 - \sqrt{R} > \rho_U(R) = (1-R)/2$ for every rate R, $0 < R < 1$, so this result showed that list decoding can be effectively used to go beyond the unique decoding radius for every rate (see Figure 1). The ratio $\rho_{\text{GS}}(R)/\rho_U(R)$ approaches 2 for rates $R \to 0$, enabling error-correction when the fraction of errors approaches 100%, a feature that has found numerous applications outside coding theory, see for example [18], [6, Chap. 12].

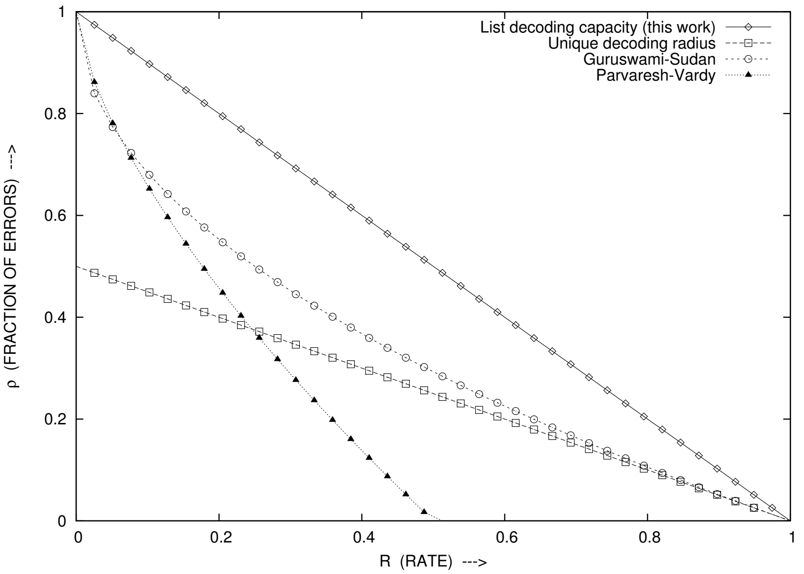

Figure 1: Error-correction radius ρ plotted against the rate R of the code for known algorithms. The best possible trade-off, i.e., capacity, is $\rho = 1 - R$, and our work achieves this.

Unfortunately, the improvement provided by [11] over unique decoding diminishes for larger rates, which is actually the regime of greater practical interest. For rates $R \to 1$, the ratio $\frac{\rho_{\text{GS}}(R)}{\rho_U(R)}$ approaches 1, and already for rate $R = 1/2$ the ratio is at most 1.18. Thus, while the results of [17, 11] demonstrated that list decoding always, for every rate, enables correcting more errors than unique decoding, they fell short of realizing the full quantitative potential of list decoding.

The bound $\rho_{\text{GS}}(R)$ stood as the best known error-correction radius for efficient list decoding for several years. In fact constructing (ρ, L)-list decodable codes of rate R for $\rho > \rho_{\text{GS}}(R)$ and polynomially bounded L, regardless of the complexity of actually performing list decoding to radius ρ, itself was elusive. Some of this difficulty was due to the fact that $1 - \sqrt{R}$ is the largest radius for which small list size can be shown generically, via the so-called Johnson bound to argue about the number of codewords in Hamming balls using only information on the relative distance of the code, cf. [5].

In a recent breakthrough paper [14], Parvaresh and Vardy presented codes which are list-decodable beyond the $1 - \sqrt{R}$ radius for low rates R. The codes they suggest are variants of Reed-Solomon (RS) codes obtained by evaluating $m \geqslant 1$ correlated polynomials at elements of the underlying field (with $m = 1$ giving RS codes). For any $m \geqslant 1$, they achieve the error-correction radius $\rho_{\text{PV}}^{(m)}(R) = 1 - \sqrt[m+1]{m^m R^m}$. For rates $R \to 0$, choosing m large enough, they can list decode up to radius $1 - O(R\log(1/R))$, which approaches the capacity $1 - R$. However, for $R \geqslant 1/16$, the best choice of m (the one that maximizes $\rho_{\mathrm{PV}}^{(m)}(R)$) is in fact $m = 1$, which reverts back to RS codes and the error-correction radius $1 - \sqrt{R}$. (See Figure 1 where the bound $1 - \sqrt[3]{4R^2}$ for the case $m = 2$ is plotted — except for very low rates, it gives a small improvement over $\rho_{\mathrm{GS}}(R)$.) Thus, getting arbitrarily close to capacity for some rate, as well as beating the $1 - \sqrt{R}$ bound for every rate, both remained open[1].

1.2 Our Result

In this work, we get arbitrarily close to the list decoding capacity $\rho_{\mathrm{cap}}(R)$ for every rate. In other words, we give explicit codes of rate R together with polynomial time list decoding up to a fraction $1 - R - \varepsilon$ of errors for every rate R and arbitrary $\varepsilon > 0$. This realizes the full quantitative promise of list decoding by correcting twice as many errors as unique decoding for every rate (Figure 1).

Description of our codes: Our capacity-achieving codes are very closely related to Reed-Solomon codes. Recall that a Reed-Solomon code of block length n over a field $\mathbb{F}$ is obtained encoding a message f, viewed as a low-degree polynomial, by its evaluations $f(x_0), f(x_1), \ldots, f(x_{n-1})$ at n distinct points in $\mathbb{F}$. We begin with such a RS code, with some moderate, easily met conditions on the size of the field $\mathbb{F}$, and an appropriate choice and ordering $x_0, \ldots, x_{n-1}$ of the evaluation points. We make use of the structure of the evaluation points $\{x_0, \ldots, x_{n-1}\}$ chosen from $\mathbb{F}$, unlike the previous works [17, 11, 14] that worked for an *arbitrary* set of evaluation points.

The encoding function of our code first encodes according to the RS code, partitions the resulting codeword symbols into n/m intervals of size m, and treats the tuple of m field elements in each of these n/m intervals as a symbol over the larger alphabet $\mathbb{F}^m$. This gives a code over $\mathbb{F}^m$ of the same rate as the original RS code.

Thus, our code is really just a Reed-Solomon code, but viewed as a code over a larger alphabet by a simple bundling of codeword symbols. We call these codes *folded Reed-Solomon codes*, following the nomenclature used by Krachkovsky [12] who studied correcting phased error bursts for RS codes.[2]

Our result can also be stated as giving decoding algorithms for certain Reed-Solomon codes that corrects a much larger fraction of errors than the one in [11] if the errors happen in large, *phased* bursts (the actual errors can be adversarial). We believe this is an appealing feature of our result, since Reed-Solomon codes are ubiquitous, and developing improved decoding algorithms for codes already in use has greater likelihood of potential practical use than proposing radically new coding schemes.

[1] Independent of our work, Alex Vardy constructed a variant of the code defined in [14] which could be list decoded with fraction of errors more than $1 - \sqrt{R}$ for all rates R. However, his construction does not achieve the list decoding capacity.

[2] The paper by Krachkovsky [12] presents an algorithm to correct a fraction $\frac{m}{m+1}(1 - R)$ of errors with probability $1 - o(1)$ in such folded RS codes, for the noise model where whenever an error occurs, the original symbol is replaced by a uniformly random element from the alphabet. This was also the noise model considered in the works [1, 2], where algorithms correcting close to a fraction $1 - R$ of such errors are presented for some special RS codes. This noise model, however, is unrealistic. In fact, under such a model, we can reduce the error-correction problem to decoding from erasures by a simple trick (this was pointed out to us by Piotr Indyk), and decoding from a fraction $1 - R$ of erasures with rate R is easy.

Our codes are also closely related to the codes constructed by Parvaresh and Vardy [14], and in fact their decoding proceeds by a black-box reduction to decoding certain codes in the latter family. See section 2 for more details.

1.3 Other consequences

Our result extends easily to the problem of *list recovering* (see Definition 3). The biggest advantage here is that we are able to achieve a rate that is independent of the size of the input lists. This is an extremely useful feature in concatenated code constructions. We are able to use this to reduce the alphabet size needed to achieve capacity, and also obtain results for binary codes. We briefly describe these results below.

To get within ε of capacity, the folded RS codes we construct have alphabet size $n^{\Omega(1/\varepsilon^2)}$ where n is the block length. By concatenating list-recoverable FRS codes of rate close to 1 with suitable inner codes followed by redistribution of symbols using an expander graph (similar to a construction for linear-time unique decodable codes in [10]), we can get within ε of capacity with codes over an alphabet of size $2^{O(\varepsilon^{-3}\log(1/\varepsilon))}$. While this is much larger than the $\mathrm{poly}(1/\varepsilon)$ alphabet size achieved by the non-constructive random coding arguments, it is still a constant independent of the block length.

For binary codes, the list decoding capacity is known to be $\rho_{\mathrm{bin}}(R) = H^{-1}(1-R)$ where $H(\cdot)$ denotes the binary entropy function [8]. We do not know explicit constructions of binary codes that approach this capacity. However, using our folded RS codes in a natural concatenation scheme, we give polynomial time constructible binary codes of rate R that can be list decoded up to a fraction $\rho_{\mathrm{Zyab}}(R)$ of errors, where $\rho_{\mathrm{Zyab}}(R)$ is the "Zyablov bound". See Figure 2 for a plot of these bounds.

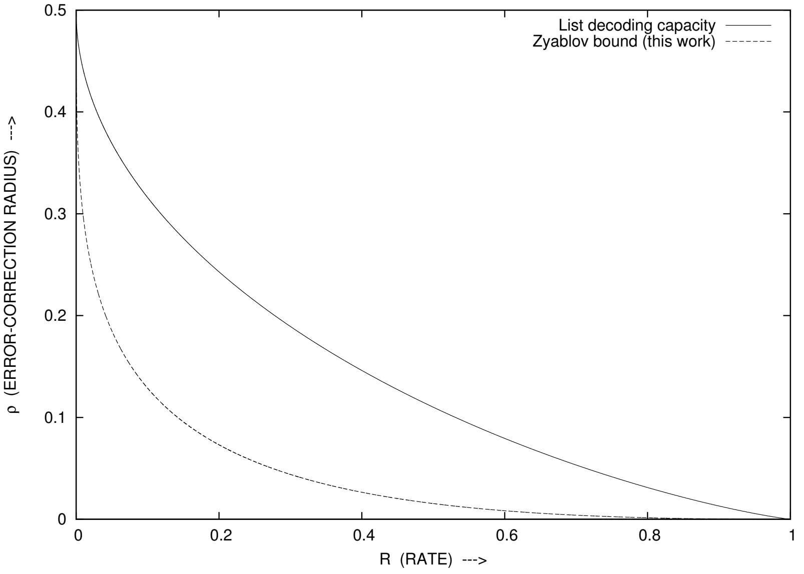

Figure 2: Error-correction radius ρ of our algorithm for binary codes plotted against the rate R. The best possible trade-off, i.e., capacity, is $\rho = H^{-1}(1-R)$, and is also plotted.

1.4 Organization of the paper

We start with an overview of our proof in Section 2. We then fix some notation and recall some well known facts in Section 3. We prove a crucial lemma (which allows us to choose evaluation points with special properties) in Section 4. We formally define the code we are working with and the polynomial time list decoding algorithm in Section 5. Section 6 deals with the extensions of our main result to get list recovery which in turn helps us get codes over smaller alphabets. We conclude with some open questions in Section 7.

2 Overview of Proof Technique

We now briefly describe the main ideas that enable improved decoding of certain folded RS (henceforth, FRS) codes. Our work uses the structure of the evaluation points $\{x_0, \ldots, x_{n-1}\}$ chosen from $\mathbb{F}$, unlike the previous works [17, 11, 14] that worked for an *arbitrary* set of evaluation points. Consider a FRS code as Section 1.2 where for each location i of the FRS code, m field elements $x_{mi}, x_{mi+1}, \ldots, x_{mi+m-1}$ of the RS encoding are bundled together, for $0 \leqslant i < n/m$ (for some m that divides n). See Figure 3 for an example when $m = 4$.

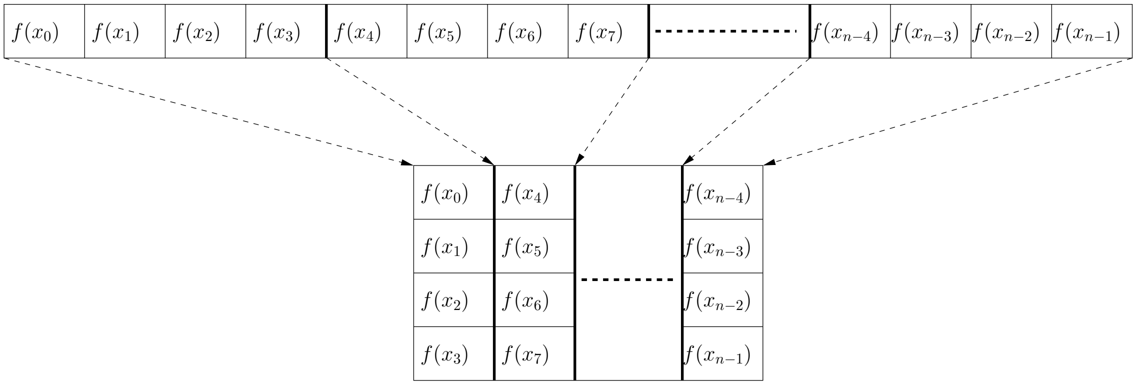

Figure 3: Folding of the Reed Solomon code evaluated on points $\{x_0, x_1, \cdots, x_{n-1}\}$ with parameter $m = 4$.

We will ensure that the elements bundled together satisfy $x_{mi+j} = \alpha x_{mi+j-1}$ for some $\alpha \in \mathbb{F}$, for $0 \leqslant i < n/m$ and $1 \leqslant j < m$. The choice of α is crucial, and here comes our first main idea — α will be chosen so that for some irreducible polynomial $E(X)$ of degree k, where k is the dimension of the original RS code, and some integer d, every polynomial $f \in \mathbb{F}[X]$ of degree at most $(k-1)$ satisfies the identity $f(\alpha X) = f(X)^d \mod E(X)$ in $\mathbb{F}[X]$. Further, assume that $\alpha^m = 1$ (this will neither be needed nor true in our case but it simplifies the description) — this implies $x_{mi} = \alpha x_{mi+m-1}$ for $0 \leqslant i < n/m$.

The reason to pick α as above is to relate FRS codes to the Parvaresh-Vardy (PV) codes [14]. The basic idea in the latter codes is to encode a polynomial f by the evaluations of $s \geqslant 2$ polynomials $f_0 = f, f_1, \ldots, f_{s-1}$ where $f_i(X) = f_{i-1}(X)^d \mod E(X)$ for an appropriate power d — let us call s the order of such a code. An FRS code with bundling using an α as above is in fact exactly the

PV code of order $s = m$ for the set of evaluation points $\{x_0, x_m, x_{2m}, \ldots, x_{(n/m-1)m}\}$. This is nice as it shows that PV codes can meet the Singleton bound (since FRS codes do), but as such does not lead to any better codes for list decoding.

Here comes our second main idea. Let us compare the FRS code to a PV code of order 2 (instead of order m) for the set of evaluation points $\{x_0, x_1, \ldots, x_{n-1}\}$. We find that in the PV encoding of f, each $f(x_i)$ appears exactly twice (once as $f(x_i)$ and another time as $f_1(\alpha^{-1}x_i)$), whereas it appears only once in the FRS encoding. (See Figure 4 for an example when $m = 4$ and $s = 2$.) In other words, the PV and FRS codes have the same information, but the rate of the FRS codes is bigger by a factor of 2. Decoding the FRS codes from a fraction ρ of errors reduces to correcting the same fraction ρ of errors for the PV code. But the rate vs. error-correction radius trade-off is better for the FRS code since it has twice the rate of the PV code.

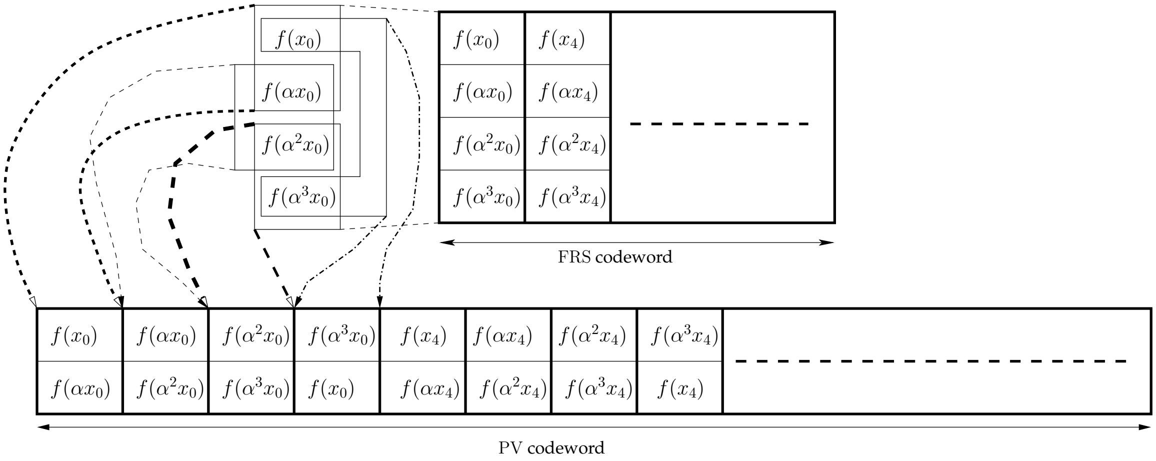

Figure 4: The correspondence between a folded Reed-Solomon code (with $m = 4$) evaluated over $\{x_0, x_4, x_8, \cdots, x_{n-4}\}$ and the Parvaresh Vardy code (of order $s = 2$) evaluated over $\{x_0, x_1, x_2, \cdots, x_{n-1}\}$. Recall that by the choice of α, $x_{4i+j} = \alpha^j x_{4i}$ for $0 \leqslant i < n/4$ and $0 < j \leqslant 3$; $\alpha^4 = 1$ and $f_1(X) = f(\alpha X)$. The correspondence for the first block in the FRS codeword and the first four blocks in the PV codeword is shown explicitly in the left corner of the figure.

In other words, our folded RS codes are chosen such that they are "compressed" forms of suitable PV codes, and thus have better rate than the corresponding PV code for a similar error-correction performance. This is where our gain is, and using this idea we are able to construct folded RS codes of rate R that are list decodable up to radius roughly $1 - \sqrt[s+1]{R^s}$ for any $s \geqslant 1$. Picking s large enough lets us get within any desired ε from capacity.

3 Preliminaries and Notation

In this section, we fix the notations used in this paper and recall some well known results. We will use p to denote a prime number and for any prime power q, we will use $\mathbb{F}_q$ to denote the finite field on q elements. The ring of polynomials over $\mathbb{F}_q$ will be denoted by $\mathbb{F}_q[X]$. The multiplicative group of $\mathbb{F}_q$ will be denoted by $\mathbb{F}_q^*$. For any element $\alpha \in \mathbb{F}_q^*$, its order in $\mathbb{F}_q$ is the smallest positive integer l such that $\alpha^l = 1$. We will be denote the order of α by $\mathrm{ord}_{\mathbb{F}_q}(\alpha)$. Note that $\mathrm{ord}_{\mathbb{F}_q}(\alpha)$ always divides $q-1$ and when $\mathrm{ord}_{\mathbb{F}_q}(\alpha) = q-1$, we say that α is a generator of $\mathbb{F}_q^*$.

For integers $m, n \geqslant 1$, a vector $v \in ((\mathbb{F}_q)^m)^n$ will be denoted as $\langle (v_0^i, v_1^i, \cdots, v_{m-1}^i) \rangle_{i=1}^n$. Further, for two vectors $u, v \in ((\mathbb{F}_q)^m)^n$, we will define their agreement as follows:

$$\mathsf{agr}(u,v) = \ \mid \{i \mid (u_0^i, u_1^i, \cdots, u_{m-1}^i) = (v_0^i, v_1^i, \cdots, v_{m-1}^i)\} \mid .$$

Let T be a subset of $\mathbb{F}_q$. For any integer $\ell \geqslant 0$ and $\alpha \in \mathbb{F}_q$, we will define a related set

$$\phi_{\alpha,\ell}(T) = \bigcup_{\beta \in T} \{\beta, \alpha\beta, \cdots, \alpha^{\ell}\beta\} . \tag{1}$$

For positive integers m, s with $s \leqslant m$, we define a map $\psi_{m,s} : ((\mathbb{F}_q)^m)^{n'} \to ((\mathbb{F}_q)^s)^{(m-s+1)n'}$ for arbitrary $n' \geqslant 1$ as follows:

$$\psi_{m,s}\left(\left\langle (b_0^i, b_1^i, \cdots, b_{m-1}^i)\right\rangle_{i=1}^{n'}\right) = \left\langle \ \left\langle (b_j^i, b_{j+1}^i, \cdots, b_{j+s-1}^i)\right\rangle_{j=0}^{m-s} \ \right\rangle_{i=1}^{n'} \tag{2}$$

Definition 1 (Orbit-avoiding sets). *For an integer $m \geqslant 1$ and $\alpha \in \mathbb{F}_q$, the (m,α)-orbit of any $\beta \in \mathbb{F}_q$ is defined to be the set $\{\beta, \alpha\beta, \cdots, \alpha^{m-1}\beta\}$. A set $T \subseteq \mathbb{F}_q$ is called (m,α)-orbit-avoiding if for every $\beta, \gamma \in T$ with $\beta \neq \gamma$, the (m,α)-orbits of β and γ are disjoint.*

We will need the following well-known and useful fact.

Fact 1. *Let q be a prime power. For any polynomial $f(X) \in \mathbb{F}_q[X]$ and integer $i \geqslant 1$, $(f(X))^{q^i} = f(X^{q^i})$.*

We will be working with irreducible polynomials. In particular, we will be interested in irreducible *binomials*. An exact characterization of these objects is known.

Theorem 1. *([13, Chapter 3]) Let $k \geqslant 2$ be an integer and $\alpha \in \mathbb{F}_q^*$. Then the binomial $X^k - \alpha$ is irreducible in $\mathbb{F}_q[X]$ if and only if the following conditions hold:*

1. *each prime factor of k divides $\mathrm{ord}_{\mathbb{F}_q}(\alpha)$ and $\gcd\big(k, \frac{q-1}{\mathrm{ord}_{\mathbb{F}_q}(\alpha)}\big) = 1$;*
2. *$q \equiv 1 \pmod 4$ if $k \equiv 0 \pmod 4$.*

Finally, for positive integers r, s, we will use the following shorthand:

$$F_r(s) = \prod_{i=0}^{s} (1 + \frac{i}{r}).$$

4 A Result on Polynomials over Finite Fields

In this section we will prove the following result.

Theorem 2. *Let p be an arbitrary prime, $b > a \geqslant 1$ be integers coprime to p, and $m \geqslant 1$ be an arbitrary integer. Let $t \geqslant m^2 - m$ be an integer that is not divisible by p, and furthermore is even if $p > 2$. Let q be any power of p such that $q \equiv 1 \mod (t \cdot a \cdot b)$. Let γ be any generator of $\mathbb{F}_q^*$. Finally, define*

$$k = \frac{a(q-1)}{b} \quad \textit{and} \quad e = \frac{(q^a - 1)b}{a(q-1)}\ . \tag{3}$$

Then the following hold:

1. *The polynomial $E(X) = X^k - \gamma$ is irreducible over $\mathbb{F}_q$.*
2. *For any polynomial $f(X) \in \mathbb{F}_q[X]$ of degree at most $k-1$ and $\ell \geqslant 1$,*

   $$(f(X))^{q^{a\ell}} \mod (E(X)) = f(\gamma^{e\ell}X).$$

   Further, if $1 \leqslant \ell < m^2 - m$ then the above map is non-trivial, that is, $\gamma^{e\ell} \neq 1$. In other words, $\mathrm{ord}_{\mathbb{F}_q}(\gamma^e) \geqslant m^2 - m$.

***Proof*.** We begin with the proof of the first claim, for which we will invoke Theorem 1. First note that the order of γ is $q-1$. Let $q = C \cdot t \cdot a \cdot b + 1$, for some integer $C \geqslant 1$. Note that then $k = Cta^2$ which implies that all prime factors of k divide $q - 1 = Ctab$. Further as $(q-1)/\mathrm{ord}_{\mathbb{F}_q}(\gamma) = 1$, $\gcd(k, (q-1)/\mathrm{ord}_{\mathbb{F}_q}(\gamma)) = 1$. Thus, the first condition of Theorem 1 is satisfied. Now, if $p = 2$, then all of C, t, a, b are odd and thus, k is odd which implies that the second condition of Theorem 1 is satisfied. If $p \neq 2$, then t is even. Now if 4 divides k, then either 4 divides t or at least one of a or C is even — in both cases 4 divides $q-1$. Thus, in all cases, the conditions of Theorem 1 are satisfied which proves the first claim.

Let $f(X)$ be any polynomial in $\mathbb{F}_q[X]$ of degree at most $k-1$. Fix $\ell \geqslant 1$. We need to show the following:

$$(f(X))^{q^{a\ell}} - f(\gamma^{e\ell}X) \equiv 0 \pmod{E(X)} \tag{4}$$

We will show the above by induction on ℓ. For the base case of $\ell = 1$, consider the following equality which follows from Lemma 1:

$$(f(X))^{q^a} - f(\gamma^e X) = f(X^{q^a}) - f(\gamma^e X).$$

Now note that $X^{q^a} - \gamma^e X = X(X^{q^a-1} - \gamma^e)$ divides the right hand side of the above equality. Further, $E(X) = X^k - \gamma$ divides $X^{q^a-1} - \gamma^e = X^{ke} - \gamma^e$, which proves (4) for the case when $\ell = 1$. Now assume that (4) holds for all $\ell \leqslant j$. Define $g(X) = f(\gamma^{ej}X)$. The following sequence of equalities shows that (4) is true for $j+1$ and thus, completes the induction.

$$\begin{aligned} f(\gamma^{e(j+1)}X) &= g(\gamma^e X) \\ &= (g(X))^{q^a} \mod (E(X)) \\ &= \left(f(\gamma^{ej}X)\right)^{q^a} \mod (E(X)) \\ &= \left((f(X))^{q^{aj}}\right)^{q^a} \mod (E(X)) \\ &= (f(X))^{q^{a(j+1)}} \mod (E(X)). \end{aligned}$$

The first and the third equalities follow from the definition of $g(X)$ while the second and the fourth equalities follow from the induction hypothesis.

Now let ℓ satisfy $1 \leqslant \ell < m^2 - m$. Then

$$\begin{aligned}\frac{e\ell}{q-1} &= \frac{(q^a-1)b\ell}{a(q-1)^2}\\ &= \frac{b\ell(1+q+q^2+\cdots q^{a-1})}{a(q-1)}\\ &= \frac{b\ell(D(q-1)+a)}{a(q-1)} \text{ (for some } D \geqslant 0)\\ &= \frac{\ell(CDtb+1)}{Cta}.\end{aligned}$$

The second equality follows from the identity $1+q+q^2+\cdots+q^{a-1} = (q^a-1)/(q-1)$ while the third follows from the fact that $1+q+q^2+\cdots+q^{a-1} = a \mod (q-1)$. Now, the last fraction is not integral as the denominator is $0 \mod (t)$ while the numerator is $\ell \mod (t) \not\equiv 0 \mod (t)$ as $0 < \ell < m^2 - m \leqslant t$. Thus, $q-1$ does not divide $e\ell$. Since $o(\gamma) = q-1$, for any l such that $\gamma^l = 1$, $q-1$ has to divide l. It follows that $\gamma^{e\ell} \neq 1$. ■

Remark 1. Let K be the extension field $\mathbb{F}_q[X]/(E(X))$ — its elements are in one-one correspondence with polynomials of degree less than k over $\mathbb{F}_q$. Let $\Gamma : K \to K$ be such that for every $f(X) \in K$, $\Gamma(f(X)) = f(G(X))$ for some polynomial G over $\mathbb{F}_q$. (In the above, we had $\Gamma(f(X)) = f(X)^{q^a} \mod (E(X))$ and $G(X) = \gamma^e X$; as a polynomial over K, $\Gamma(Z) = Z^{q^a}$, and hence had degree q^a.) Any such map Γ is an $\mathbb{F}_q$-linear function on K, and is therefore a *linearized* polynomial, cf. [13, Chap. 3, Sec. 4], which has only terms with exponents that are powers of q (including $q^0 = 1$). It turns out that for our purposes Γ cannot have degree 1, and so it must have degree at least q. The large degree of the map Γ is in turn the cause for the large list size that we need for list decoding.

5 Folded Reed-Solomon codes and their decoding

5.1 The construction

For any $T \subseteq \mathbb{F}_q$, let $\mathrm{RS}_T[n,k,n-k+1]_{\mathbb{F}_q}$ denote the Reed-Solomon code of dimension k and length $n = |T|$ over $\mathbb{F}_q$. Recall that the codewords in this code are $\langle f(\beta)\rangle_{\beta\in T}$ for every polynomial $f(X)$ of degree at most $k-1$.

Definition 2 (Folded Reed-Solomon Codes). *For an integer parameter $m \geqslant 1$, an $\alpha \in \mathbb{F}_q^* \setminus \{1\}$ and an (m,α)-orbit-avoiding set $T \subseteq \mathbb{F}_q$, we define an (m,α)-folded Reed-Solomon code, denoted by $\mathrm{FRS}_{k,m,\alpha,T,\mathbb{F}_q}$, as follows. The codeword corresponding to every polynomial $f(X)$ of degree at most $k-1$ is given by*

$$E_{\mathrm{FRS}} = \left\langle \; \left(f(\beta), f(\alpha\beta), \cdots, f(\alpha^{m-1}\beta)\right) \; \right\rangle_{\beta\in T}$$

The parameters of the code are as follows.

Proposition 3. *The above defines a code over $\mathbb{F}_{q^m}$ of block length $n' = |T|$, rate k'/n' where $k' = k/m$, and minimum distance $d' = n' - \lceil k' \rceil + 1$.*

***Proof*.** The claims about the block length and rate are obvious. If two distinct codewords agree in at least $\lceil k' \rceil$ places, then the corresponding two polynomials of degree at most $k-1$ agree in at least $m\lceil k' \rceil \geqslant k$ points in $\mathbb{F}_q$ which is not possible. ■

5.2 The Parvaresh-Vardy code

We now recall the the recent code of Parvaresh-Vardy [14]. Their construction starts with a RS code $\mathrm{RS}_T[n, k, n-k+1]_{\mathbb{F}_q}$ for some $T \subseteq \mathbb{F}_q$. For integer parameters $s \geqslant 1$, $d_1 < d_2 < \cdots < d_{s-1}$ and an irreducible polynomial $E(X)$ over $\mathbb{F}_q$ of degree k , we will denote the Parvaresh-Vardy code by $\mathrm{CRS}_{k,T,s,\mathbf{d},E(X)}$, where $\mathbf{d} = \langle d_1, d_2, \cdots, d_{s-1} \rangle$.[3] The codeword corresponding to a polynomial $f(X)$ of degree at most $k-1$ is given by

$$E_{\mathrm{CRS}}(f) = \langle (f(\beta), g_1(\beta), g_2(\beta), \cdots, g_{s-1}(\beta)) \rangle_{\beta \in T}$$

where $g_i(X) = (f(X))^{d_i} \mod (E(X))$ for every $1 \leqslant i < s$. For low rates, their code can tolerate errors beyond the radius achieved by Guruswami and Sudan for Reed-Solomon codes [11].

Theorem 4. *[14] The Parvaresh-Vardy code, $\mathrm{CRS}_{k,T,s,\mathbf{d},E(X)}$, where $\mathbf{d} = \langle d_1, d_2, \cdots, d_{s-1} \rangle$, can be list decoded up to a radius of*

$$|T| - \left\lceil \sqrt[s+1]{(k-1)^s |T| F_r(s)} + \frac{1}{r} \right\rceil \tag{5}$$

in time polynomial in $|T|$, where $r \geqslant s$ is an arbitrary integer, provided

$$\frac{d_i}{d_{i-1}} > \left\lceil r \sqrt[s+1]{\frac{|T|}{k-1} F_r(s)} \right\rceil = \Delta_s, \textit{ for every } 1 \leqslant i < s \textit{ (with the convention } d_0 = 1\textit{);} \tag{6}$$

and

$$d_{s-1} \textit{ is at most a polynomial in } |T|. \tag{7}$$

In particular the list decoding algorithm requires $q^{O(1)} m^{O(1)} r^{O(s)} + (d_{s-1})^{O(1)} k^{O(1)} (\log q)^{O(1)}$ operations[4] in $\mathbb{F}_q$. Further, the algorithm always returns a list of size at most $d_{s-1}\Delta_s$.

5.3 Connection between FRS and CRS codes

In this subsection, we show that for certain setting of parameters, there is a bijection between the FRS and CRS codes. The following is a crucial realization.

[3] The acronym CRS stands for Correlated Reed-Solomon code, since the Parvaresh-Vardy code encodes several dependent polynomials, which are correlated in a carefully chosen way, by their joint evaluations at field elements.

[4] The total run time is never explicitly stated in [14] but the (loose) upper bound on the the number of operations stated above follows immediately from looking at the paper.

Lemma 5. *Let $m, s \geqslant 1$ be integers such that $m > s$. Let a, b, q, k, e and $E(X) = X^k - \gamma$ be as defined in Theorem 2 with γ being a generator of $\mathbb{F}_q^*$. Further, define $\alpha = \gamma^e$ and $Q = q^a$. Finally, let $T \subseteq \mathbb{F}_q^*$ be an (m, α)-orbit-avoiding set. If E_{FRS} and E_{CRS} are the encoding functions of $\mathrm{FRS}_{k,m,\alpha,T,\mathbb{F}_q}$ and $\mathrm{CRS}_{k,\phi_{\alpha,m-s}(T),s,\mathbf{d}^{\mathbf{Q}},E(X)}$ (where $\mathbf{d}^{\mathbf{Q}} = \langle Q, Q^2, \cdots, Q^{s-1}\rangle$) respectively, then*

$$\psi_{m,s} \circ E_{\mathrm{FRS}} = E_{\mathrm{CRS}} \ .$$

***Proof*.** If $f(X)$ is a polynomial of degree at most $k-1$, then its encoding in $\mathrm{FRS}_{k,m,\alpha,T,\mathbb{F}_q}$ is given by

$$E_{\mathrm{FRS}}(f) = \big\langle \big(f(\beta), f(\alpha\beta), f(\alpha^2\beta), \cdots, f(\alpha^{m-1}\beta)\big)\big\rangle_{\beta \in T} \, .$$

Now consider the image of $E_{\mathrm{FRS}}(f)$ under $\psi_{m,s}$:

$$\begin{aligned}
\psi_{m,s}(E_{\mathrm{FRS}}(f)) &= \Big\langle \Big\langle \ \big(f(\alpha^j\beta), f(\alpha^{j+1}\beta), \cdots, f(\alpha^{j+s-1}\beta)\big) \ \Big\rangle_{j=0}^{m-s} \Big\rangle_{\beta \in T} & (8)\\
&= \big\langle \big(f(\beta'), f(\alpha\beta'), \cdots, f(\alpha^{s-1}\beta')\big)\big\rangle_{\beta' \in \phi_{\alpha,m-s}(T)} & (9)\\
&= \big\langle \big(f(\beta'), g_1(\beta'), \cdots, g_{s-1}(\beta')\big)\big\rangle_{\beta' \in \phi_{\alpha,m-s}(T)} & (10)\\
&= E_{\mathrm{CRS}}(f) & (11)
\end{aligned}$$

(8) and (9) follow from the definitions of $\psi_{(\cdot,\cdot)}$ and $\phi_{(\cdot,\cdot)}$ in (2) and (1) respectively. (10) follows from Theorem 2 and the fact that $g_i(X) = (f(X))^{Q^i} \mod (E(X))$. (11) follows from the definition of $\mathrm{CRS}_{k,\phi_{\alpha,m-s}(T),s,\mathbf{d}^{\mathbf{Q}},E(X)}$. The proof is complete. ■

The following is an easy consequence of the definition of $\psi_{(\cdot,\cdot)}$ and Lemma 5.

Proposition 6. *Let $m, s, T, E_{\mathrm{FRS}}, E_{\mathrm{CRS}}$ be as defined in Lemma 5. If $\mathbf{r} \in ((\mathbb{F}_q)^m)^{|T|}$ then for any polynomial $f(X)$ of degree at most $k-1$,*

$$\mathsf{agr}\,(\psi_{m,s}(\mathbf{r}), E_{\mathrm{CRS}}(f)) = (m-s+1)\mathsf{agr}\,(\mathbf{r}, E_{\mathrm{FRS}}(f)) \ .$$

5.4 List decoding FRS codes

In this section we show that for certain setting of parameters, list decoding of the folded RS code can be reduced to list decoding of Parvaresh-Vardy codes (with different parameters). The big gain will be that we can work with a much lower agreement for the FRS code, thanks to Proposition 6.

Theorem 7. *Let p be an arbitrary prime, $b > a \geqslant 1$ be integers coprime to p, and $m \geqslant 1$ be an arbitrary integer. For this choice of p, m, a, b, let q, k, e and $E(X) = X^k - \gamma$ be defined as in Theorem 2 where γ is a generator of $\mathbb{F}_q^*$. Let $T \subseteq \mathbb{F}_q^*$ be an (m, α)-orbit-avoiding set of* maximum *size, where $\alpha = \gamma^e$. Then the folded RS code $\mathrm{FRS}_{k,m,\alpha,T,\mathbb{F}_q}$ has alphabet size q^m, block length n' and rate k'/n' where*

$$k = mk' \quad \text{and} \quad \frac{q-1}{m} \geqslant n' = |T| \geqslant \frac{(q-1)}{m}\Big(1 - \frac{1}{m}\Big) \ . \tag{12}$$

Furthermore, for every integer s, $1 \leqslant s < m$, and every r such that $q > 4r^2$ and $r \geqslant s$, there is a polynomial time list decoding algorithm for $\mathrm{FRS}_{k,m,\alpha,T,\mathbb{F}_q}$ that successfully list decodes up to a radius of

$$n' - \left\lceil \sqrt[s+1]{\Big(\frac{m}{m-s+1}\Big)^s n'(k')^s F_r(s)} + \frac{1}{r(m-s+1)} \right\rceil . \tag{13}$$

The algorithm runs in time $q^{O(1)}m^{O(1)}r^{O(s)} + q^{O(as)}$ and always outputs a list of size at most q^{as}.

***Proof*.** The claims about the rate and alphabet size of the FRS code are clear. To bound its block length $n' = |T|$, we will estimate the size of any maximum (m, α)-orbit-avoiding subset of $\mathbb{F}_q^*$. This will also indicate why such a set can be found efficiently. We can define a relation R_α on $\mathbb{F}_q^*$ where $(x, y) \in R_\alpha$ iff xy^{-1} is a power of α. This is an equivalence relation whose equivalence classes all have size $\mathrm{ord}_{\mathbb{F}_q}(\alpha)$, and are the orbits (under action of multiplication by α) of $v = (q-1)/\mathrm{ord}_{\mathbb{F}_q}(\alpha)$ distinct elements $\zeta_1, \dots, \zeta_v \in \mathbb{F}_q^*$. Any (m, α)-orbit-avoiding set T of maximum size leaves out at most $(m-1)$ elements in each of these equivalence classes. Therefore,

$$n' = |T| \geqslant \frac{(q-1) - v(m-1)}{m} = \frac{q-1}{m}\Big(1 - \frac{m-1}{\mathrm{ord}_{\mathbb{F}_q}(\alpha)}\Big) \geqslant \frac{q-1}{m}\Big(1 - \frac{1}{m}\Big)$$

where the last step follows since $\mathrm{ord}_{\mathbb{F}_q}(\alpha) \geqslant m^2 - m$ (by Theorem 2). The upper bound $n' \leqslant (q-1)/m$ is obvious. Regarding constructibility of such a maximum (m, α)-orbit-avoiding set, $T = \{\zeta_i \alpha^{mj} \mid 1 \leqslant i \leqslant v,\ 0 \leqslant j < \lfloor \frac{\mathrm{ord}_{\mathbb{F}_q}(\alpha)}{m} \rfloor\}$ is an explicit description of such a set.

We now specify the decoding algorithm which proceeds by a reduction to decoding a related CRS code. Define $Q = q^a$ and $\mathbf{d^Q} = \langle Q, Q^2, \cdots, Q^{s-1}\rangle$. Let $\mathbf{r} \in ((\mathbb{F}_q)^m)^{n'}$ be the received word. Use the list decoder for $\mathrm{CRS}_{k,\phi_{\alpha,m-s}(T),s,\mathbf{d^Q},E(X)}$ on the vector $\psi_{m,s}(\mathbf{r})$ and return the list of polynomials which is returned by the algorithm.

To see why this works, let $E_{\mathrm{FRS}}(f)$ be the codeword in $\mathrm{FRS}_{k,m,\alpha,T,\mathbb{F}_q}$ corresponding to the polynomial $f(X)$ is degree at most $k-1$. If

$$\mathsf{agr}(\mathbf{r}, E_{\mathrm{FRS}}(f)) \geqslant \left\lceil \sqrt[s+1]{\left(\frac{m}{m-s+1}\right)^s (k')^s n' F_r(s)} + \frac{1}{r(m-s+1)} \right\rceil, \tag{14}$$

then the list decoding algorithm for $\mathrm{CRS}_{k,\phi_{\alpha,m-s}(T),s,\mathbf{d^Q},E(X)}$ should output f. Indeed,

$$\begin{aligned}
\mathsf{agr}(\psi_{m,s}(\mathbf{r}), E_{\mathrm{CRS}}(f)) &= (m-s+1)\mathsf{agr}(\mathbf{r}, E_{\mathrm{FRS}}(f)) \\
&\geqslant \left\lceil \sqrt[s+1]{m^s (k')^s n'(m-s+1) F_r(s)} + \frac{m-s+1}{r(m-s+1)} \right\rceil && (15) \\
&= \left\lceil \sqrt[s+1]{(mk')^s |\phi_{\alpha,m-s}(T)| F_r(s)} + \frac{1}{r} \right\rceil && (16) \\
&> \left\lceil \sqrt[s+1]{(k-1)^s |\phi_{\alpha,m-s}(T)| F_r(s)} + \frac{1}{r} \right\rceil. && (17)
\end{aligned}$$

The first steps follows from Proposition 6. (15) follows from using (14). (16) follows from the fact that $|\phi_{\alpha,m-s}(T)| = (m-s+1)n'$. (17) follows from the fact that $mk' = k > k-1$.

Comparing the bound on the agreement in (17) with the the bound (5) that the Parvaresh-Vardy list decoder can correct, we conclude that f would indeed be a part of the list returned by the Parvaresh-Vardy list decoder, provided we verify conditions (6) and (7), which we do next. For the code $\mathrm{CRS}_{k,\phi_{\alpha,m-s}(T),s,\mathbf{d^Q},E(X)}$ with $\mathbf{d^Q} = \langle d_1 = Q, d_2 = Q^2, \dots, d_{s-1} = Q^{s-1}\rangle$, note that $d_i/d_{i-1} = Q^i/Q^{i-1} = Q = q^a > \left\lceil r \sqrt[s+1]{\frac{|\phi_{\alpha,m-s}(T)|}{k-1} F_r(s)} \right\rceil$. This is because $q^a \geqslant q > 4r^2 \geqslant (2r)^{(s+1)/s}$, $|\phi_{\alpha,m-s}(T)| \leqslant q-1$, and $\sqrt[s+1]{F_r(s)} \leqslant 2$. Moreover, $d_{s-1} = q^{a(s-1)}$ is polynomial in $|\phi_{\alpha,m-s}(T)| = \Theta(q)$.

We now look at the running time of the above scheme. Applying $\psi_{m,s}(\cdot)$ to the received word certainly does not requires more than $q^{O(1)}$ steps. As $d_{s-1} = q^{a(s-1)}$ and $k \leqslant q$, the number of $\mathbb{F}_q$ operations required by the list decoding algorithm of Theorem 4 is no more than $q^{O(1)} m^{O(1)} r^{O(s)} + q^{O(as)}$.

Finally, we consider the size of the list returned by the algorithm, which by Theorem 4 is at most $Q^{s-1} \left\lceil r \sqrt[s+1]{\frac{|\phi_{\alpha,m-s}(T)|}{k-1}} F_r(s) \right\rceil < Q^{s-1} Q = q^{as}$ as claimed. ■

Remark 2. The above decoding bound can be improved to $n' - \sqrt[s+1]{n'(k')^s F_r(s)}$ in the case when the field element α used in the folding satisfies $\mathrm{ord}_{\mathbb{F}_q}(\alpha) = m$.[5] In this case, the (m, α)-orbit of the elements of T partition $\mathbb{F}_q^*$, and the encoding comprises of evaluation of the message polynomial f at these orbits. For such a folded RS code, we can reduce the decoding problem to a CRS code of block length mn' with the agreement increasing m fold. This is done by using all m *cyclic* s-element windows of each orbit (instead of the $(m - s + 1)$ linear s-element windows we used above), as we outlined in Section 2 using an example for the case $m = 4$ and $s = 2$. However, implementing this adds a strong further constraint on the order of α which we must cater to in Theorem 2. To avoid this complication, we just work with any α with large $\mathrm{ord}_{\mathbb{F}_q}(\alpha)$ and lose a factor $\frac{m}{m-s+1}$ in the rate.

Corollary 8. *Let p be a prime, and s, m, r be positive integers such that $m > s$. Let $b > a \geqslant 1$ be integers not divisible by p. Then for some $\frac{a}{b} \leqslant r_0 \leqslant \frac{am}{b(m-1)}$, there is an* explicit *family of folded Reed-Solomon codes over fields of characteristic p with rate r_0 and which can be list decoded up to a fraction*

$$1 - \left(\sqrt[s+1]{\left(\frac{m r_0}{m - s + 1} \right)^s F_r(s)} + o(1) \right) \tag{18}$$

of errors. A code of block length N in the family has alphabet size $O((mN)^m)$, and the list decoding algorithm outputs a list of size at most $(mN)^{O(as)}$ and runs in time $(mN)^{O(as)}$.

***Proof*.** Let q be any power of p that satisfies the conditions of Theorem 2, and let k, e be as defined in (3). Let $\alpha = \gamma^e$ where γ is a generator of $\mathbb{F}_q^*$ and let $T \subseteq \mathbb{F}_q^*$ be an (m, α)-orbit avoiding set of maximum size n' that satisfies (12). We will now apply Theorem 7 to the folded RS code $\mathrm{FRS}_{k,m,\alpha,T,\mathbb{F}_q}$. We note that there are infinitely many possible choices for q and thus we get a family of codes.

Let $k' = k/m$ be the "dimension" of the FRS code. The rate of this code equals

$$r_0 = \frac{k'}{n'} = \frac{k}{n'm} = \frac{a}{b} \cdot \frac{(q-1)}{mn'} .$$

By the bound on n' in (12), we have $(q - 1)(1 - 1/m) \leqslant mn' \leqslant (q - 1)$. Thus, the rate r_0 satisfies

$$\frac{a}{b} \leqslant r_0 \leqslant \frac{am}{b(m-1)}. \tag{19}$$

(18) follows from plugging the upper bound on $r_0 = k'/n'$ from (19) in (13) (after normalizing the error bound). The claim about the alphabet size, list size and decoding complexity follow from Theorem 7 since the underlying field size q satisfies $q = O(mn')$ where n' is the block length and the fact that the $q^{O(as)}$ term dominates the running time. ■

[5] Farzad Parvaresh and Alex Vardy (personal communication) also made this observation about our construction.

Remark 3. Note that setting $m = s$ above gives the same error-correction radius as the one achieved by Parvaresh-Vardy codes. In fact for $m = s$, as was observed in Section 2, the folded Reed-Solomon codes are in fact Parvaresh Vardy codes (defined on the same set of evaluation points). As a special case, for $m = s = 1$, the folded Reed-Solomon (as well as Parvaresh Vardy) codes are just Reed-Solomon codes.

5.5 Achieving the list decoding capacity

For fixed s, as m and r grow, the fraction of errors that can be corrected for the codes of Corollary 8 approaches $1-(r_0)^{s/(s+1)}$. Parvaresh and Vardy [14] obtained a decoding radius of $1-(sr_0)^{s/(s+1)}$ — the extra factor of s multiplying the rate r_0 had a detrimental effect on their performance unless r_0 was very small. In contrast, by picking s larger and larger, the fraction of errors we can correct approaches the optimal bound of $1 - r_0$. We formalize this next by instantiating suitable parameters in Corollary 8 and conclude our main result. We did not attempt to optimize the exact choice of parameters in the calculations below.

Theorem 9 (Main). *For every R, $0 < R < 1$, every $\varepsilon > 0$, and every prime p, there is an* explicit *family of folded Reed-Solomon codes over fields of characteristic p that have rate at least R and which can be list decoded up to a fraction $(1 - R - \varepsilon)$ of errors in polynomial time. The alphabet size of a code of block length N in the family is $N^{O(\varepsilon^{-2}\log^2(1/R))}$ and the list decoding algorithm runs within time (and outputs a list of size at most) $N^{O(\varepsilon^{-2}R\log(1/R))}$.*

***Proof*.** We begin by specifying some choice of parameters. Let $\delta = \varepsilon/8$. We will use:

$$s = \left\lceil \frac{\ln(1/R)}{\ln(1+\delta)} \right\rceil \tag{20}$$

$$m = s^2 - 1 \tag{21}$$

$$r = s^3 \tag{22}$$

Let a and b be the integers coprime to p such that

$$R \leqslant \frac{a}{b} \leqslant R + \delta \; . \tag{23}$$

Clearly such a, b exist with $b = O(1/\delta) = O(1/\varepsilon)$.

Applying Corollary 8 with the above choice of a, b and using the bounds on the rate r_0 in Corollary 8, we have a family of folded RS code with rate r_0 satisfying

$$R \leqslant r_0 \leqslant \frac{(R+\delta)}{(1-1/m)} \; . \tag{24}$$

Codes in this family can be list decoding up to a fraction

$$1 - \left(\sqrt[s+1]{\left(\frac{mr_0}{m-s+1}\right)^s F_r(s)} + o(1) \right)$$

of errors in polynomial time. To get the fraction $(1 - R - \varepsilon)$ of errors claimed in the theorem, we will be done if we can show that

$$\left(\sqrt[s+1]{\left(\frac{m}{m-s+1}\right)^{s} (r_0)^{s}\, F_r(s)} + o(1) \right) \leqslant R + \varepsilon. \tag{25}$$

To show this, we have

$$\begin{aligned}
\sqrt[s+1]{\left(\frac{m}{m-s+1}\right)^{s} (r_0)^{s}\, F_r(s)} &= \left((1+\frac{1}{s}) F_r(s) r_0 \right)^{\frac{s}{s+1}} && (26)\\
&\leqslant \left((1+\frac{1}{s}) \left(1 + \frac{s^2+s}{2s^3} + O(\frac{1}{s^6})\right) \frac{R+\delta}{(1-1/m)} \right)^{\frac{s}{s+1}} && (27)\\
&= \left(\left(1 + \frac{3}{2s} + O(\frac{1}{s^2})\right) (R+\delta) \right)^{\frac{s}{s+1}} && (28)\\
&< \left(1 + \frac{3}{2s} + O(\frac{1}{s^2})\right) (R+\delta) \left(\frac{1}{R+\delta}\right)^{\frac{1}{s+1}} && (29)\\
&= \left(1 + \frac{3}{2s} + O(\frac{1}{s^2})\right) (R+\delta)(1+\delta) && (30)\\
&\leqslant R + 6\delta. && (31)
\end{aligned}$$

(26) follows by substituting the value of m from (21). (27) follows by using the estimate

$$F_r(s) = 1 + \frac{s^2+s}{2r} + O(\frac{1}{r^2})$$

(using r from (22)) and (24). (28) follows by multiplying the different factors and collecting all the $O(\frac{1}{s^2})$ terms together. (29) follows from the fact that $s/(s+1) < 1$ and the first term in (28) is greater than 1. (30) is by substituting the value of s from (20). The last step follows by using the fact that $1/s \leqslant \ln(1+\delta) \leqslant \delta$ and that δ is very small.

Further, as δ is a constant, $o(1) \leqslant \delta$. Recalling the value of $\delta = \varepsilon/8$ proves the inequality (25).

Also, from Corollary 8 the alphabet size of the codes is $O((mn')^m)$. Recalling the value of m from (21), this is at most $(n')^{O(\varepsilon^{-2}\log^2(1/R))}$. Similarly, from Corollary 8 the running time of the list decoding algorithm as well as the size of the output list are both $(mn')^{O(as)}$. Recalling that $a = O(R/\varepsilon)$ and $s = O\left(\frac{\log(1/R)}{\log(1+\delta)}\right) = O(\varepsilon^{-1}\log(1/R))$, this quantity is at most $(n')^{O(\varepsilon^{-2}R\log(1/R))}$. The proof is complete. ■

6 Extensions and Codes over Smaller Alphabets

6.1 Extension to list recovering

We now present a very useful generalization of the list decoding result of Theorem 7 to the setting of *list recovering*. Under the list recovering problem, one is given as input for each codeword position, not just one but a set of several, say l, alphabet symbols. The goal is to find and output all codewords which agree with some element of the input sets for several positions. Codes for which this more general problem can be solved turn out to be extremely valuable as outer codes in concatenated code constructions. In short, this is because one can pass a set of possibilities from decodings of the inner codes and then list recover the outer code with those sets as the input. If we only had a list-decodable code at the outer level, we will be forced to make a unique choice in decoding the inner codes thus losing valuable information.

Definition 3 (List Recovering). *A code $C \subseteq \Sigma^n$ is said to be (ζ, l, L)-list recoverable if for every sequence of sets $S_1, \ldots, S_n$ where each $S_i \subseteq \Sigma$ has at most l elements, the number of codewords $c \in C$ for which $c_i \in S_i$ for at least ζn positions $i \in \{1, 2, \ldots, n\}$ is at most L.*

A code $C \subseteq \Sigma^n$ is said to (ζ, l)-list recoverable in polynomial time if it is $(\zeta, l, L(n))$-list recoverable for some polynomially bounded function $L(\cdot)$, and moreover there is a polynomial time algorithm to find the at most $L(n)$ codewords that are solutions to any $(\zeta, l, L(n))$-list recovering instance.

We remark that when $l = 1$, $(\zeta, 1, \cdot)$-list recovering is the same as list decoding up to a $(1 - \zeta)$ fraction of errors. List recovering has been implicitly studied in several works; the name itself was coined in [9].

Theorem 7 can be generalized to list recover the folded RS codes. Specifically, for a FRS code with parameters as in Theorem 7, for an arbitrary constant $l \geqslant 1$, we can (ζ, l)-list recover in polynomial time provided

$$\zeta n' \geqslant \left\lceil \sqrt[s+1]{\left(\frac{m}{m-s+1}\right)^s n' l (k')^s F_r(s)} + \frac{1}{r(m-s+1)} \right\rceil . \tag{32}$$

We briefly justify this claim. The (ζ, l)-list recovering problem for the FRS code can be reduced to the (ζ, l)-list recovering problem for the related CRS code of block length $n = n'(m - s + 1)$ exactly as in Theorem 7. For the latter CRS code, the Parvaresh-Vardy algorithm can be generalized in a straightforward way to (ζ, l)-list recover provided $\zeta n \geqslant \lceil \sqrt[s+1]{(k)^s n l F_r(s)} + 1/r \rceil$, and one can check that (32) implies this condition. The generalization of the PV algorithm is straightforward: instead of one interpolation condition for each symbol of the received word, we just impose $|S_i| \leqslant l$ many interpolation conditions for each position $i \in \{1, 2, \ldots, n\}$ (where S_i is the i'th input set in the list recovering instance). The number of interpolation conditions is at most nl, and so replacing $n = |T|$ by nl in the bound (5) guarantees successful decoding. This simple generalization to list recovering is a positive feature of all interpolation based decoding algorithms [17, 11, 14] beginning with the one due to Sudan [17].

Picking $r \gg s$ and $m \gg s$ in (32), we get (ζ, l)-list recover with rate R for $\zeta \geqslant \left(lR^s\right)^{1/(s+1)}$. Now comes the remarkable fact: we can pick a suitable $s \gg l$ and perform (ζ, l)-list recovering with agreement parameter $\zeta \geqslant R + \varepsilon$ which is independent of l! We state the formal result below (Theorem 9 is a special case when $l = 1$). We skip the details which are very similar to the proof of Theorem 9.

Theorem 10. *For every integer $l \geqslant 1$, for all R, $0 < R < 1$ and $\varepsilon > 0$, and for every prime p, there is an* explicit *family of folded Reed-Solomon codes over fields of characteristic p that have rate at least R and which can be $(R + \varepsilon, l)$-list recovered in polynomial time. The alphabet size of a code of block length N in the family is $N^{O(\varepsilon^{-2}(l+\log(1/R))^2)}$.*

Remark 4 (Soft Decoding)**.** The decoding algorithm for folded RS codes from Theorem 7 can be further generalized to handle soft information, where for each codeword position i the decoder is given as input a non-negative weight $w_{i,z}$ for each possible alphabet symbol z. The weights $w_{i,z}$ can be used to encode the confidence information concerning the likelihood of the the i'th symbol of the codeword being z. For any $\varepsilon > 0$, for suitable choice of parameters, our codes of rate R over alphabet Σ have a soft decoding algorithm that outputs all codewords $c = \langle c_1, c_2, \ldots, c_N \rangle$ that satisfy

$$\sum_{i=1}^{N} w_{i,c_i} \geqslant \left((1+\varepsilon)(RN)^s \Big(\sum_{i=1}^{N} \sum_{z \in \Sigma} w_{i,z}^{s+1} \Big) \right)^{1/(s+1)} .$$

For $s = 1$, this soft decoding condition is identical to the one for Reed-Solomon codes in [11].

6.2 Binary codes decodable up to Zyablov bound

The optimal list recoverability of the folded RS codes plays a crucial role in establishing the following result concerning list decoding binary codes.

Theorem 11. *For all $0 < R, r < 1$ and all $\varepsilon > 0$, there is a polynomial time constructible family of binary linear codes of rate at least $R \cdot r$ which can be list decoded in polynomial time up to a fraction $(1-R)H^{-1}(1-r) - \varepsilon$ of errors.*

***Proof*.** We will construct binary codes with the claimed property by concatenating two codes C_1 and C_2. For C_1, we will use a folded RS code over a field of characteristic 2 with block length n_1, rate at least R, and which can be $(R+\varepsilon, l)$-list recovered in polynomial time for $l = \lceil 10/\varepsilon \rceil$. Let the alphabet size of C_1 be 2^M where $M = O(\varepsilon^{-4} \log n)$. For C_2, we will use a binary linear code of dimension M and rate at least r which is (ρ, l)-list decodable for $\rho = H^{-1}(1-r-\varepsilon)$. Such a code is known to exist via a random coding argument that employs the semi-random method [8]. Also, a greedy construction of such a code by constructing its M basis elements in turn is presented in [8] and this process takes $2^{O(M)}$ time. We conclude that the necessary inner code can be constructed in $n^{O(1/\varepsilon^4)}$ time. The code C_1, being a folded RS code over a field of characteristic 2, is $\mathbb{F}_2$-linear, and therefore when concatenated with a binary linear inner code such as C_2, results in a binary linear code. The rate of the concatenated code is at least $R \cdot r$.

The decoding algorithm proceeds in a natural way. Given a received word, we break it up into blocks corresponding to the various inner encodings by C_1. Each of these blocks is list decoded up to a radius ρ, returning a set of at most l possible candidates for each outer codeword symbol. The outer code is then $(R+\varepsilon, l)$-list recovered using these sets, each of which has size at most l, as input. To argue about the fraction of errors this algorithm corrects, we note that the algorithm fails to recover a codeword only if on more than a fraction $(1-R-\varepsilon)$ of the inner blocks the codeword differs from the received word on more than a fraction ρ of symbols. It follows that the algorithm correctly list decodes up to a radius $(1-R-\varepsilon)\rho = (1-R-\varepsilon)(H^{-1}(1-r)-\varepsilon)$. Since $\varepsilon > 0$ was arbitrary, we get the claimed result. ■

Optimizing over the choice of inner and outer codes rates r, R in the above results, we can decode up to the Zyablov bound, see Figure 2.

Remark 5. In particular, decoding up to the Zyablov bound implies that we can correct a fraction $(1/2-\varepsilon)$ of errors with rate $\Omega(\varepsilon^3)$, improving the rate of $\Omega(\varepsilon^3/\log^2(1/\varepsilon))$ achieved in [7]. However, our construction and decoding complexity are $n^{O(1/\varepsilon^4)}$ whereas these are at most $f(\varepsilon)n^c$ for an absolute constant c in [7]. Also, we bound the list size needed in the worst-case by $n^{\tilde{O}(1/\varepsilon^3)}$, while the list size needed in the construction in [7] is $2^{\tilde{O}(\log(1/\varepsilon))}$.

6.3 Capacity-Achieving codes over smaller alphabets

Our result of Theorem 9 has two undesirable aspects: both the alphabet size and worst-case list size output by the list decoding algorithm are a polynomial of large degree in the block length. We now show that the alphabet size can be reduced to a constant, albeit a large one, that depends only on the distance ε to capacity.

Theorem 12. *For every R, $0 < R < 1$, every $\varepsilon > 0$, there is a polynomial time constructible family of codes over an alphabet of size $2^{O(\varepsilon^{-3}\log(1/\varepsilon))}$ that have rate at least R and which can be list decoded up to a fraction $(1 - R - \varepsilon)$ of errors in polynomial time.*

Proof (Sketch). The theorem can be proved using the code construction scheme used in [10] for linear time unique decodable codes with optimal rate (with different components appropriate for list decoding plugged in). We only sketch the basic ideas here. The idea is to concatenate two codes C_{out} and C_{in}, and then redistribute the symbols of the resulting codeword using an expander graph. Let $\delta = \varepsilon/5$. The outer code C_{out} will be a code of rate $(1 - 2\delta)$ over an alphabet Σ of size $n^{(1/\delta)^{O(1)}}$ that can be $(1 - \delta, O(1/\delta))$-list recovered in polynomial time, as guaranteed by Theorem 10. That is, the rate of C_{out} will be close to 1, and it can be (ζ, l)-list recovered for large l and $\zeta \to 1$.

The inner code C_{in} will be a $((1 - R - 4\delta), O(1/\delta))$-list decodable code with near-optimal rate, say rate at least $(R+3\delta)$. Such a code is guaranteed to exist over an alphabet of size $O(1/\delta^2)$ using random coding arguments. A naive brute-force for such a code, however, is too expensive, since we need a code with $|\Sigma| = n^{\Omega(1)}$ codewords. Guruswami and Indyk [9], see also [6, Sec. 9.3], prove that there is a small (quasi-polynomial sized) sample space of *pseudolinear codes* in which most codes have the needed property. Furthermore, they also present a deterministic polynomial time construction of such a code (using derandomization techniques), see [6, Sec. 9.3.3].

The concatenation of C_{out} and C_{in} gives a code of rate $\approx R$ over an alphabet of size $O(1/\delta^2)$. Moreover, given a received word of the concatenated code, one can find all codewords which agree with the received word on a fraction $R+4\delta$ of locations in at least $(1-\delta)$ of the inner blocks. Indeed, we can do this by list decoding each of the inner blocks to a radius of $(1-R-4\delta)$ returning up to $l = O(1/\delta)$ possibilities for each block, and then $(1 - \delta, l)$-list recovering C_{out}.

The last component in this construction is a $D = O(1/\delta^3)$-regular Ramanujan graph which is designed to convert an overall agreement on $R + 5\delta$ fraction of the symbols to an agreement of at least $R + 4\delta$ on most (specifically a fraction $(1 - \delta)$) of the inner blocks of the concatenated code. In other words, the expander redistributes symbols in a manner that “smoothens” the distributions of errors evenly among the various inner blocks (except for possibly a δ fraction of the blocks). The expander graph incurs no loss in rate, but increases the alphabet size to $(1/\delta^2)^{O(1/\delta^3)} = 2^{O(\varepsilon^{-3}\log(1/\varepsilon))}$. ■

7 Concluding Remarks

We close with some remarks and open questions. The description of folded RS codes in Definition (2) bears some resemblance to certain extractors constructed in [16]. In the Shaltiel-Umans extractor, the output of the weak-random source is treated as a low-degree d-variate polynomial f over $\mathbb{F}_q$, and f is thought of as a map $f : \mathbb{F}_{q^d} \to \mathbb{F}_q$ (by identifying the vector space $\mathbb{F}_q^d$ with the extension field $\mathbb{F}_{q^d}$). The random seed is treated as a point $\beta \in \mathbb{F}_{q^d}$ and the output of the extractor is $(f(\beta), f(\alpha\beta), \ldots, f(\alpha^{m-1}\beta))$ where α is a generator of $\mathbb{F}_{q^d}^*$. Can some of the techniques in this work and [14] be used to construct simple extractors based on univariate polynomials?

We have solved the qualitative problem of achieving list decoding capacity. Our work could be improved with some respect to some parameters. The size of the list needed to perform list decoding to a radius that is within ε of capacity grows as $n^{O(1/\varepsilon^2)}$ where n is the block length of the code. It remains an open question to bring this list size down to a constant independent of n (the existential random coding arguments work with a list size of $O(1/\varepsilon)$). We managed to reduce the alphabet size needed to approach capacity to a constant independent of n. However, this involved a brute-force search for a rather large code, and the alphabet size of $2^{\tilde{O}(1/\varepsilon^3)}$ we obtained is far from the $\mathrm{poly}(1/\varepsilon)$ bound that the non-constructive arguments achieve. Obtaining a "direct" algebraic construction over a constant-sized alphabet, such as the generalization of the Parvaresh-Vardy framework to algebraic-geometric codes in [7, 15], might help in addressing these two issues.

Finally, constructing binary codes that approach list decoding capacity remains open.

Acknowledgments

We thank Charanjit Jutla, Farzad Parvaresh, Anindya Patthak, Madhu Sudan, and Alex Vardy for discussions and comments.

References

[1] Daniel Bleichenbacher, Aggelos Kiayias, and Moti Yung. Decoding of interleaved Reed-Solomon codes over noisy data. In *Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 97–108, 2003.

[2] Don Coppersmith and Madhu Sudan. Reconstructing curves in three (and higher) dimensional spaces from noisy data. In *Proceedings of the 35th ACM Symposium on Theory of Computing (STOC)*, pages 136–142, June 2003.

[3] Peter Elias. List decoding for noisy channels. *Technical Report 335, Research Laboratory of Electronics, MIT*, 1957.

[4] Peter Elias. Error-correcting codes for list decoding. *IEEE Transactions on Information Theory*, 37:5–12, 1991.

[5] Venkatesan Guruswami. Limits to list decodability of linear codes. In *Proceedings of the 34th ACM Symposium on Theory of Computing (STOC)*, pages 802–811, 2002.

[6] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes*. Lecture Notes in Computer Science 3282. Springer, 2004.

[7] Venkatesan Guruswami. Algebaric-geometric generalizations of the Parvaresh-Vardy codes. *Manuscript*, July 2005.

[8] Venkatesan Guruswami, Johan Håstad, Madhu Sudan, and David Zuckerman. Combinatorial bounds for list decoding. *IEEE Transactions on Information Theory*, 48:1021–1035, May 2002.

[9] Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 658–667, 2001.

[10] Venkatesan Guruswami and Piotr Indyk. Linear time encodable/decodable codes with near-optimal rate. *IEEE Transactions on Information Theory*, 51(10):3393–3400, October 2005.

[11] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999.

[12] Victor Y. Krachkovsky. Reed-Solomon codes for correcting phased error bursts. *IEEE Transactions on Information Theory*, 49(11):2975–2984, November 2003.

[13] Rudolf Lidl and Harald Niederreiter. *Introduction to Finite Fields and their applications*. Cambridge University Press, Cambridge, MA, 1986.

[14] Farzad Parvaresh and Alexander Vardy. Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In *Proceedings of the 43nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 285–294, 2005.

[15] Anindya Patthak. Reducing the alphabet size of the Parvaresh-Vardy code using Algebraic Geometry codes. *Manuscript*, September 2005.

[16] Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *Journal of the ACM*, 52(2):172–216, March 2005.

[17] Madhu Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997.

[18] Madhu Sudan. List decoding: Algorithms and applications. *SIGACT News*, 31:16–27, 2000.

[19] John M. Wozencraft. List Decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, 48:90–95, 1958.

[20] Victor V. Zyablov and Mark S. Pinsker. List cascade decoding. *Problems of Information Transmission*, 17(4):29–34, 1981 (in Russian); pp. 236-240 (in English), 1982.