Subquadratic Time Encodable Codes Beating the Gilbert-Varshamov Bound

Anand Kumar Narayanan and Matthew Weidner

***Abstract*—We construct explicit algebraic geometry codes built from the Garcia-Stichtenoth function-field tower beating the Gilbert-Varshamov bound for alphabet sizes at least 19^2. Messages are identified with functions in certain Riemann-Roch spaces associated with divisors supported on multiple places. Encoding amounts to evaluating these functions at degree-one places. By exploiting algebraic structures particular to the Garcia-Stichtenoth tower, we devise an intricate deterministic $\omega/2 < 1.19$ runtime exponent encoding and $1 + \omega/2 < 2.19$ expected runtime exponent randomized (unique and list) decoding algorithms. Here $\omega < 2.373$ is the matrix multiplication exponent. If $\omega = 2$, as widely believed, the encoding and decoding runtimes are respectively nearly linear and nearly quadratic. Prior to this work, encoding time of code families beating the Gilbert-Varshamov bound were quadratic or worse.**

***Index Terms*—Algebraic geometry codes, error correcting codes, explicit constructions, Gilbert-Varshamov bound.**

I. Introduction

A. Codes Beating the Gilbert-Varshamov Bound

ERROR-CORRECTING codes enable reliable transmission of information over an erroneous channel. A (block) error-correcting code of block length N over a finite alphabet Σ of size Q is a subset $\mathcal{C} \subseteq \Sigma^N$. The rate R at which information is transmitted through the channel after encoding the original message with the code $\mathcal{C}$ is defined as $\log_Q(|\mathcal{C}|)/N$. The minimum distance d of the code $\mathcal{C}$, defined as the minimum Hamming distance among all distinct pairs of elements (codewords) in the code $\mathcal{C}$, quantifies the number of errors that $\mathcal{C}$ can correct. A code with minimum distance d can tolerate $(d-1)/2$ errors. The relative distance δ is defined as $\delta := d/N$. A code $\mathcal{C}$ is linear if the alphabet is a finite field $\mathbb{F}_Q$ (with Q elements) and $\mathcal{C}$ is an $\mathbb{F}_Q$-linear subspace of $\mathbb{F}_Q^N$. One typically desires codes to transmit information at a high rate while still being able to correct a large fraction of errors. That is, one wants codes with large rate and large relative distance. However, for a fixed small Q, rate and relative distance are competing quantities with a tradeoff. The Gilbert-Varshamov bound assures, for every Q, $R > 0$, $0 < \delta \leq 1 - 1/Q$ and small positive ϵ, the existence of an infinite family of codes with increasing block length over an alphabet of size Q with rate R and relative distance δ bounded by

$$R + H_Q(\delta) \geq 1 - \epsilon, \tag{1}$$

where H_Q is the Q-ary entropy function [1], [2]. Random linear codes, where one chooses a random subspace of N-tuples over a finite field meet the bound with high probability. In fact, Varshamov proved the bound using the probabilistic method with random linear codes. Testing if a given linear code meets the Gilbert-Varshamov bound comes down to approximating the minimum distance, an intractable task unless NP equals RP [3], [4]. Hence constructing codes meeting or beating the Gilbert-Varshamov bound remained a long-standing open problem, until the advent of algebraic geometry codes.

Goppa proposed algebraic geometry codes obtained from curves over finite fields as a generalization of Reed-Solomon codes [5]. Messages are identified with functions on the curve in the Riemann-Roch space corresponding to a chosen divisor with support disjoint from a large set of $\mathbb{F}_Q$-rational points on the curve. Evaluations of functions in the Riemann-Roch space at these $\mathbb{F}_Q$-rational points on the curve are taken as the codewords of the code. The rate of the code is the ratio of the dimension of the Riemann-Roch space to the number of $\mathbb{F}_Q$-rational points. The Riemann-Roch theorem gives a bound on the dimension of the code and yields the following tradeoff between rate and relative distance:

$$R + \delta \geq 1 - \frac{g}{N}, \tag{2}$$

where g denotes the genus of the curve. This spurred an effort to construct curves over finite fields where the fraction g/N of the genus to the number of $\mathbb{F}_Q$-rational points is as low as possible. Researchers had to contend with the lower bound $g/N \geq 1/(\sqrt{Q} - 1)$ of Drinfeld-Vlăduţ [6]. In seminal papers, Ihara [7] and Tsfasman, Vlăduţ, and Zink [8] constructed curves meeting the Drinfeld-Vlăduţ bound when the underlying finite field size Q is a square, leading to the Tsfasman-Vlăduţ-Zink bound

$$R + \delta \geq 1 - \frac{1}{\sqrt{Q} - 1}. \tag{3}$$

Remarkably, for $Q \geq 7^2$, the Tsfasman-Vlăduţ-Zink bound is better than the Gilbert-Varshamov bound! This is a rare occasion where an explicit construction yields better parameters than guaranteed by randomized arguments. Garcia and

This work was supported in part by NSF grant #CCF-1423544, Chris Umans' Simons Foundation Investigator grant, the European Union's H2020 Programme under grant agreement number ERC-669891, and Caltech's Rita A. and Øistein Skjellum SURF Fellowship.

A. K. Narayanan is with the Laboratoire d'Informatique de Paris 6 and Institut de Mathématiques de Jussieu–Paris Rive Gauche, Sorbonne Université, UPMC Campus, 75005 Paris, France (email: anand.narayanan@lip6.fr).

Matthew Weidner was with the Department of Mathematics, California Institute of Technology, Pasadena, CA 91125 USA. He is now with the Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA 15213 USA (email: maweidne@andrew.cmu.edu).

Manuscript received August 11, 2018.

Stichtenoth described an explicit tower of function fields which meet the Drinfeld-Vlăduţ bound, hence yield codes matching the Tsfasman-Vlăduţ-Zink bound [9]. The curves in the Garcia-Stichtenoth tower are the primary objects of study in our paper. An outstanding open problem in this area is to explicitly construct codes meeting or beating the Gilbert-Varshamov bound over small alphabets, in particular binary codes ($Q = 2$). It is known that algebraic geometry codes beat the Gilbert-Varshamov bound over $\mathbb{F}_Q$ for any prime power $Q \geq 49$ which is not prime and not 125 [10].

B. Linear Time Encodable Codes Meeting the Gilbert-Varshamov Bound

For codes to find use in practice, one often requires fast encoding and decoding algorithms in addition to satisfying a good tradeoff between rate and minimum distance. An encoding algorithm maps a given message to a codeword. A decoding algorithm takes a possibly corrupted codeword, called the received word, and outputs the message that induced it, provided the number of errors is within the designed tolerance.

A natural question, which remains unresolved, is if there exist linear time encodable and decodable codes meeting or beating the Gilbert-Varshamov bound. One cannot look to random linear codes to resolve this problem, for they require quadratic runtime to encode and are NP-hard to decode [11]. In a breakthrough, Speilman introduced expander codes (explicit codes built from certain expander graphs) thereby proving the existence of linear time encodable and decodable asymptotically "good" codes [12]. A family of codes with increasing block length is deemed good if (in the limit) the rate and relative distance are simultaneously bounded away from zero. A code being good is a weaker condition than meeting the Gilbert-Varshamov bound. Guruswami and Indyk constructed linear time encodable and decodable expander codes approaching the Gilbert-Varshamov bound [13]. However, the closer one wishes to approach the Gilbert-Varshamov bound, the larger the alphabet size of the code; to approach $R+\delta \geq 1-\epsilon$, the alphabet sizes are exponential in $1/\epsilon$. Druk and Ishai constructed linear time encodable codes meeting the Gilbert-Varshamov bound, but these codes are likely NP-hard to decode [14].

C. Main Results

Our main result is the explicit algebraic construction of sub-quadratic time encodable codes beating the Gilbert-Varshamov bound, along with an efficient decoding algorithm.

Theorem I.1. *For every square prime power Q and rate $R \in \left(0, 1-\frac{2\sqrt{Q}+1}{Q-\sqrt{Q}}\right)$, there exists an infinite sequence of codes over $\mathbb{F}_Q$ of increasing length N with rate R and relative distance δ satisfying*

$$R+\delta \geq 1-\frac{2\sqrt{Q}+1}{Q-\sqrt{Q}}.$$

Further, there exists deterministic algorithms to

- *pre-compute a representation of the code at the encoder and decoder in $O(N^{3/2}\log^3 N)$ time; this representation occupies $O(N)$ space*
- *encode a message in $O(N^{\omega/2})$ time, where ω is the matrix multiplication exponent*

and Las Vegas randomized algorithms to

- *decode close to half the designed distance in $O(N^{1+\omega/2}\log^2 N)$ expected time*
- *list decode up to*

$$N\left(1-\sqrt{2\left(R+2\frac{2\sqrt{Q}+1}{Q-\sqrt{Q}}\right)}-\frac{2\sqrt{Q}+1}{Q-\sqrt{Q}}\right)$$

errors with list size at most $\sqrt{\frac{2(\sqrt{Q}-1)}{2+R(\sqrt{Q}-1)}}$ in $O(N^{1+\omega/2}\log^2 N)$ expected time (requiring additional pre-processing taking $O(N^\omega)$ time and $O(N^2)$ space).

For $Q \geq 19^2$, the tradeoff assured by Theorem I.1 is better than the Gilbert-Varshamov bound. The encoding time is linear if the matrix multiplication exponent ω is indeed 2 as widely conjectured. The best known bound for ω yields an encoding time exponent of 1.19 [15].

The pre-processing step can be thought of as computing a succinct representation of the code and is performed at the encoder and decoder independently. If one desires, the pre-processing and encoding time can be made to approach linear time at the cost of needing larger alphabets to beat the Gilbert-Varshamov bound. Our construction is parametrized by an integer $k \geq 2$ and yields the tradeoff

$$R+\delta \geq 1-\frac{k\sqrt{Q}+k-1}{Q-\sqrt{Q}}$$

with pre-computation requiring time $O(N^{3/k}\log^3 N)$, the resulting succinct representation requiring $O(N^{2/k})$ space, and encoding requiring time $O(N^{1+(\omega-2)/k})$; see Theorem V.1. For decoding, we get pre-computation requiring time $O(N^\omega)$, the resulting succinct representation requiring $O(N^2)$ space, and decoding requiring expected time $O(N^{2+(\omega-2)/k}\log^2 N)$; see Theorem VI.1. Theorem I.1 corresponds to $k = 2$. Table I gives a comparison of encoding for $k = 2, 3$. The likeness to the Tsfasman-Vlăduţ-Zink bound (3), which is obtained if one is allowed to substitute $k = 1$, is striking.

D. Applications

As with other algebraic geometry codes, our codes may be used as outer codes in concatenation to obtain long binary codes. In particular, concatenating with Walsh-Hadamard inner codes yields balanced binary codes (or equivalently small bias spaces) [16, § 3.2]. Outside coding theory, our codes have several complexity theoretic implications: efficient secret sharing schemes and interactive proof protocols to name a few. Chen and Cramer [17] initiated the use of algebraic geometry codes in secret sharing and in secure multi-party computation. Their scheme retains the salient features of Shamir's [18] secret sharing (to the extent possible) yet only requires small alphabets for the shares. Our codes fit seamlessly in their framework resulting in a significant speed up. The runtime

exponent (in the number of players) with our codes is $\omega/2$ for secret sharing and $1+\omega/2$ for secret recovery. If $w=2$, we get nearly linear time secret sharing and nearly quadratic time secret recovery. The speed up is in the computational complexity of encoding and decoding (resharing and recombination) per round of secret sharing, the number of rounds (communication complexity) is unchanged. The Chen-Cramer scheme has spawned several extensions and improvements in the ensuing decade, including multiplicative ramp secret sharing schemes [19] (for communication efficient secure multi-party computation), high information rate ramp schemes [20, § 4.3] (for threshold secure computation) and secret sharing schemes relying on nested algebraic geometry codes ([21, Thm 22] and [22, Prop 21]). In [19], communication efficient secure multi-party computation is obtained because the number of bits communicated per round of resharing is reduced since the size of the shares is reduced. In [22], communication efficient secret sharing schemes are obtained, which reduce the communication required to reconstruct the secret. Our codes are applicable across these schemes and result in a speedup similar to that for the Chen-Cramer scheme.

Exciting recent developments in interactive proofs are promising grounds for applying our codes. In Delegated Computation, a remote server runs a computation for a client and tries to prove interactively that it indeed correctly performed the computation. This scenario was modelled [23], [24] as an interactive proof system where the honest prover (server) is limited to polynomial time computation and the verifier (client) is limited to nearly linear time computation. In a recent breakthrough [24], constant round protocols under this framework were described using *Probabilistically Checkable Interactive Proofs* (PCIPs), an interactive version of PCPs where the verifier only reads a few bits of the transcripts. Independently [25], to improve the efficiency of PCPs by adding rounds of interaction (in the random oracle model), *Interactive Oracle Protocols* (IOPs) were introduced, which are equivalent to PCIPs. Constant rate and constant query IOPs were recently constructed using tensor products of algebraic geometry codes from Garcia-Stichtenoth towers [26, § 5.2, Thm 7.1, Lem 7.2]. Taking tensor products of our codes improves the efficiency of these IOPs. In particular, we reduce the efficiency exponent c from $c>3$ to $c>3/2$ in [26, Lem 7.2] by constructing asymptotically good systematic subcodes of our codes (see § V-D). To this end, we tailored these systematic subcodes in a manner that they can be encoded and (equivalently) checked (that is, decide if a given word is a codeword) with deterministic runtime exponent $3/2$. We anticipate further fruitful applications of our code to these interactive protocols in the future.

E. Code Construction: Riemann-Roch spaces from Shifting

We next recount (one point) algebraic geometry codes before sketching our construction. Stichtenoth's textbook [27] is an excellent reference. Let $\mathcal{X}$ be a smooth projective (not necessarily plane) curve over a finite field $\mathbb{F}_Q$ and $\mathbb{F}_Q(\mathcal{X})$ the associated function field. A set $\mathcal{P}$ of $\mathbb{F}_Q$-rational points on $\mathcal{X}$ (or equivalently, degree 1 places in $\mathbb{F}_Q(\mathcal{X})$) will serve as the code places. A divisor is chosen, typically of the form rP_∞ where P_∞ is a place in $\mathbb{F}_Q(\mathcal{X})$ away from $\mathcal{P}$ and $r\in\mathbb{N}$. The Riemann-Roch space $\mathcal{L}(rP_\infty)$ (consisting of functions in $\mathbb{F}_Q(\mathcal{X})$ whose poles are confined to P_∞ and have order bounded by r) is identified with the message space. The code is the evaluation of $\mathcal{L}(rP_\infty)$ at the places in $\mathcal{P}$. The rate is determined by the dimension of $\mathcal{L}(rP_\infty)$ as an $\mathbb{F}_Q$-linear space. The Riemann-Roch theorem then yields bounds on the rate and relative distance, thereby quantifying the performance of the code. Encoding messages requires efficient algorithms to construct and evaluate functions from the Riemann-Roch space. This can be accomplished in polynomial time due to algorithms of Huang and Ierardi (for smooth projective plane curves) [28] and Hess (for smooth projective curves) [29], [30]. However, such generic algorithms are far from linear. We focus on building fast algorithms tailored to the Garcia-Stichtenoth tower.

Take $Q=q^2$ where q is a prime power. The Garcia-Stichtenoth tower over $\mathbb{F}_{q^2}$ is the sequence of function fields defined by $F_0=\mathbb{F}_{q^2}(x_0)$, and $F_{i+1}=F_i(x_{i+1})$ where x_{i+1} satisfies the relation

$$x_{i+1}^q+x_{i+1}=\frac{x_i^q}{x_i^{q-1}+1}.$$

Let $P_\infty^{(n)}$ denote the unique pole of x_0 in F_n. In a series of works, Aleshnikov, Deolalikar, Kumar, Shum and Stichtenoth [31]–[33] described the splitting of places in F_n and established a pole-cancelling algorithm to compute a basis for the Riemann-Roch spaces $\mathcal{L}(rP_\infty^{(n)})$. This culminated in a quadratic time algorithm (with nearly cubic time preprocessing) to encode these codes.

From the n^{th} function field F_n, we construct codes of block length $q^n(q^2-q)$. Let a small integer parameter $k\geq 2$ be chosen. Assume for ease of exposition that k divides n.

Code places: Let $\Omega=\{\alpha\in\mathbb{F}_{q^2}\mid \alpha^q+\alpha=0\}$, which has size q. The code places are all the places in F_n which are zeros of $x_0-\alpha$ for $\alpha\in\mathbb{F}_{q^2}\setminus\Omega$; there are $q^n(q^2-q)$ such places, all of which are $\mathbb{F}_{q^2}$-rational.

Message Space: We begin by constructing functions in the lower function field $F_{n/k}$ that are regular. By regular, we mean that their poles in F_n are confined to $P_\infty^{(n)}$. We devise a procedure called *shifting* that translates functions in $F_{n/k}$ to F_n. It is quite simple and, in spirit, just relabelling the subscripts so that each x_j becomes x_{j+i} for some chosen positive integer i. The symmetry of the defining equations of the Garcia-Stichtenoth tower allows us to determine the pole divisor in F_n of shifts of regular functions from $F_{n/k}$. The poles of the shifts are not confined to $P_\infty^{(n)}$. However, by taking products of carefully chosen shifted functions, we can bound the new poles arising outside $P_\infty^{(n)}$. We thus construct mostly regular functions in F_n by taking products of shifts of regular functions in $F_{n/k}$.

In summary, given a positive integer r, we can construct a large number of functions in a Riemann-Roch space of the form $\mathcal{L}(G+rP_\infty^{(n)})$ where G is a small pole divisor. We take the span of these constructed functions to be our message space. Enough functions are constructed to yield large rate

codes, and properties of curves applied to the divisor $G+rP_\infty^{(n)}$ yield a lower bound on the minimum distance.

Pre-computation: To aid in rapid encoding and decoding, we first pre-compute a set of regular functions in $F_{n/k}$. In particular, we pre-compute the evaluations at all code places of a basis for all regular functions in $F_{n/k}$ of a certain bounded pole degree. This Riemann-Roch space computation is performed using an algorithm of Shum et. al. [32] and has runtime exponent $3/k$.

Encoding Algorithm: Given such a set of regular functions in $F_{n/k}$, we compute basis functions in our message space by taking products of shifted functions. Given a message, which is a tuple over $\mathbb{F}_{q^2}$, encoding amounts to evaluating the corresponding linear combination of the basis functions simultaneously at the code places. We devise a Baby-Step Giant-Step algorithm to perform this multipoint evaluation. The runtime of the encoding step depends on the parameter k. For $k = 2$, the crux of the computation is square matrix multiplications, resulting in an encoding algorithm with runtime exponent $\omega/2$. For larger values of k, the crux is rectangular matrix multiplications of shape determined by k, and the runtime is again subquadratic. In particular, larger values of k give rise to faster encoding algorithms.

Decoding Algorithms: We tailor the Shokrollahi-Wasserman algorithm [34] to our code setting to uniquely decode close to half the relative distance and list decode beyond that. The Shokrollahi-Wasserman algorithm first interpolates a polynomial with coefficients in a Riemann-Roch space such that each message sufficiently close to the received word is a root. Then the roots of the interpolated polynomial in the message Riemann-Roch space are enumerated. Finally the encoding algorithm is used to verify and output the messages in the enumeration that are indeed sufficiently close to the received word.

To adapt their algorithm to our setting, we identify an appropriate Riemann-Roch space (which incidentally is an extension of the message space) as the coefficient space of the interpolation polynomial. Determining the interpolation polynomial now boils down to solving a linear system. We observe that computing matrix-vector products corresponding to this linear system is virtually identical to encoding messages, a task accomplished in subquadratic time. Invoking Wiedemann's algorithm [35] (a Las Vegas randomized iterative method involving matrix-vector products) to solve the linear system, we obtain the interpolation polynomial in subcubic expected time.

To unique decode, we restrict the interpolation polynomial to have degree one. The root finding step is trivial and the algorithm corrects errors up to nearly half the designed distance.

To correct beyond half the designed distance we allow interpolation polynomials of degree greater than one. We perform root finding in quadratic time provided an extra pre-processing step requiring quadratic storage. The resulting algorithm corrects

$$N\left(1-\sqrt{2\left(R+2\frac{kq+k-1}{q(q-1)}\right)}-2\frac{kq+k-1}{q(q-1)}\right)$$

errors with list size at most $\sqrt{\frac{2(q-1)}{2+R(q-1)}}$. For $k = 2$, with list size at most 2, it corrects at least as many errors as guaranteed in the trade off in Theorem I.1.

F. Organization:

In §II, we recount results from [31] on the splitting of places in the Garcia-Stichtenoth tower and establish notation. In §III, we define the shifting operation and construct mostly regular functions in F_n from regular functions in $F_{n/k}$. The code sequences derived from mostly regular functions are defined and their parameters established in §IV. In §V we develop the subquadratic time encoding algorithm using fast matrix multiplication. The decoding algorithms are presented in §VI.

II. Splitting of Places in the Garcia-Stichtenoth Tower

In this section, we recall some notation and results from [31] (see also [32], [33]) on the splitting of places in the Garcia-Stichtenoth tower. For standard terminology from algebraic function fields and algebraic-geometry codes, we recommend the textbook [27].

In $F_0 = \mathbb{F}_{q^2}(x_0)$, x_0 has a unique pole, which we denote by $P_\infty^{(0)}$. This place is totally ramified in every field extension F_n/F_0, hence there is a unique place lying above $P_\infty^{(0)}$ in F_n; we denote this place by $P_\infty^{(n)}$. Let $\Omega = \{\alpha \in \mathbb{F}_{q^2} \mid \alpha^q + \alpha = 0\}$, which has size q. For $\alpha \in \mathbb{F}_{q^2}$, let $P_\alpha^{(0)}$ denote the unique zero of $x_0 - \alpha$ in F_0. The splitting of places $P_\alpha^{(0)}$ is best understood by distinguishing if α is in Ω.

1) When $\alpha \in \mathbb{F}_{q^2} \setminus \Omega$, $P_\alpha^{(0)}$ splits completely in every field extension F_n/F_0, yielding q^n $\mathbb{F}_{q^2}$-rational places. As α varies, we get $(q^2-q)q^n$ rational places in F_n, which we take to be the set of code places.
2) When $\alpha \in \Omega \setminus \{0\}$, $P_\alpha^{(0)}$ is totally ramified in every field extension F_n/F_0, hence there is a unique place lying above it in F_n; we denote this place by $P_\alpha^{(n)}$. The most interesting place is $P_0^{(0)}$. For $t \geq 1$, let $S_t^{(t-1)}$ denote the unique place in F_{t-1} that is a zero of x_{t-1}. We sometimes treat $S_t^{(t-1)}$ as a singleton set instead of a place. We have $S_1^{(0)} = P_0^{(0)}$, and $S_u^{(u-1)}$ lies over $S_t^{(t-1)}$ whenever $u \geq t$.
   In the field extension F_t/F_{t-1}, $S_t^{(t-1)}$ splits completely. Specifically, for each $\alpha \in \Omega$, there is a unique place of F_t which is a simultaneous zero of x_{t-1} and $x_t - \alpha$, and these are all of the places lying above $S_t^{(t-1)}$. We let $S_t^{(t)}$ denote the set of all places lying above $S_t^{(t-1)}$ in F_t besides $S_{t+1}^{(t)}$, i.e., $S_t^{(t)}$ contains the simultaneous zero of x_{t-1} and $x_t - \alpha$ for each $\alpha \in \Omega \setminus \{0\}$. For $u \geq t$, we let $S_t^{(u)}$ denote the set of all places of F_u lying above a place in $S_t^{(t)}$. For $t \leq t' \leq u+1$, we let $S_{t,t'}^{(u)} := \bigcup_{i=t}^{t'} S_i^{(u)}$. These places are also explained and pictured in [32, Fig 5].
   Let $u \geq t-1$. If $u < 2t$, then all of the places in $S_t^{(u)}$ are unramified (but not necessarily split) in the field extension F_{u+1}/F_u. If $u \geq 2t$, then all of the places in $S_t^{(u)}$ are totally ramified in F_{u+1}/F_u.

If Q is a place of F_n and $x \in F_n$, we let $v_Q(x)$ denote the valuation of x at Q. We define the weight of $x \in F_n$ to be $-v_{P_\infty^{(n)}}(x)$.

III. Mostly Regular Functions Through Shifting

For the remainder of the paper, fix an integer parameter $k \geq 2$. In this section, given a positive integer r, through shifting we construct a mostly regular function $f_r \in F_n$ of weight precisely $r + \sum_{i=1}^{k} q^{n-(i-1)\lceil n/k \rceil + 1}$ from regular functions in $F_{n/k}$. The discrepancy of f_r from being regular will be quantified by a pole divisor G of degree at most $q^n(kq+k-1)$. That is, there is a pole divisor G of said degree and weight $\sum_{i=1}^{k} q^{n-(i-1)\lceil n/k \rceil + 1}$ such that for all r,

$$f_r \in \mathcal{L}(G + r(P_\infty^{(n)})) \setminus \mathcal{L}(G + (r-1)(P_\infty^{(n)})).$$

Once a choice of a regular function of each weight in $F_{n/k}$ used by our construction is fixed, the functions f_r are uniquely determined.

A. Shifting

First, we define the shifting operation and determine the poles in F_n of functions arising out of shifting regular functions in $F_{n/k}$.

Let $f = x_m^{e_m} \cdots x_0^{e_0}$ be a monomial in F_m. For $i \geq 0$, we define the *shift of f by i* to be the element

$$f[i] := x_{m+i}^{e_m} \cdots x_i^{e_0} \in F_{m+i}.$$

We extend the definition of shift $\mathbb{F}_{q^2}$-linearly to all of F_m. Soon (Prop. III.1) we show $f[i]$ is well defined.

We will use the following notations throughout the paper. For a function field E and an element $x \in E$, let $(x)^E$ and $(x)_\infty^E$ denote the principal divisor and pole divisor of x as an element of E. In the case $E = F_n$ for some n, we use the abbreviations $(x)^{(n)}$ and $(x)_\infty^{(n)}$ in place of $(x)^{F_n}$ and $(x)_\infty^{F_n}$. For a finite extension of function fields E'/E, let $\mathrm{Con}_E^{E'}$ denote the corresponding conorm map; this is the unique homomorphism from the divisor group of E to the divisor group of E' such that for all places Q of E,

$$\mathrm{Con}_E^{E'}(Q) = \sum_{Q'|Q} e(Q'|Q) \cdot Q',$$

where the sum runs over all places Q' of $F_{E'}$ lying over Q and where $e(Q'|Q)$ denotes the ramification index of Q' over Q. We have the identities

$$(x)^{E'} = \mathrm{Con}_E^{E'}\left((x)^E\right)$$
$$(x)_\infty^{E'} = \mathrm{Con}_E^{E'}\left((x)_\infty^E\right)$$

for all $x \in E$. Also, for all divisors D of E, we have $\deg \mathrm{Con}_E^{E'}(D) = [E' : E] \deg(D)$ ([27, Lem 1.4.12]).

Proposition III.1. *(a) For any $f \in F_m$ and $i \geq 0$, $f[i]$ is well-defined, i.e., it does not depend on the representation of f as a sum of monomials.*

(b) Let $f \in F_m$ be regular of weight r. Then:

- *$f[i]$ has weight r.*
- *$f[i]$ is regular at $S_{i,m+i+1}^{(m+i)}$.*
- *For $t \in [0, i-1]$, for all $P \in S_t^{(m+i)}$, we have*

$$v_P(f[i]) = \begin{cases} -r & \text{if } t \leq \frac{m+i}{2} \\ -rq^{2t-(m+i)} & \text{if } t > \frac{m+i}{2}. \end{cases}$$

- $\deg\left((f[i])_\infty^{(m+i)}\right) = rq^i$.

Proof. For all j, we have the isomorphism

$$\phi_j : F_j \xrightarrow{\sim} F_j$$
$$x_k \mapsto x_{j-k}^{-1},$$

which is its own inverse [32, p. 2237]. It is easy to see that

$$f[i] = \phi_{m+i}(\phi_m(f)),$$

proving (a).

To prove (b), we use the fact that ϕ_j induces bijections $S_t^{(j)} \leftrightarrow S_{j-t}^{(j)}$ for each $t \in [0, j]$, together with a correspondence $P_\infty^{(j)} \leftrightarrow S_{j+1}^{(j)}$ [32, p. 2237]. Thus letting $f \in F_m$ be regular of weight r, $\phi_m(f)$ is regular at all places (including $P_\infty^{(j)}$) except for a pole of order r at $S_{j+1}^{(j)}$. Then by the ramification behavior of the tower, $\phi_m(f) \in F_{m+i}$ is regular at all places (including $P_\infty^{(j)}$) except for $S_{m,m+i+1}^{(m+i)}$. In particular, for $t \in [m, m+i+1]$, for all $P \in S_t^{(m+i)}$, we have

$$v_P(\phi_m(f)) = \begin{cases} -r & \text{if } t \geq \frac{m+i}{2} \\ -rq^{m+i-2t} & \text{if } t < \frac{m+i}{2} \end{cases}$$

Then $f[i] = \phi_{m+i}(\phi_m(f))$ easily has the first three properties in (b). The fourth property follows either from computing the total pole degree directly, or from using the above properties of the conorm map to compute

$$\begin{aligned} \deg\left((f[i])_\infty^{(m+i)}\right) &= \deg\left((\phi_m(f))_\infty^{(m+i)}\right) \\ &= [F_{m+i} : F_m] \deg\left((\phi_m(f))_\infty^{(m)}\right) \\ &= q^i \deg\left((f)_\infty^{(m)}\right) \\ &= rq^i. \end{aligned}$$

□

B. Construction of Mostly Regular Functions

We begin by dealing with the case $r \in [0, q^n - 1]$. Write $r = r_1 q^{n-\lceil n/k \rceil} + r_2 q^{n-2\lceil n/k \rceil} + \cdots + r_{k-1} q^{n-(k-1)\lceil n/k \rceil} + r_k$, with $r_1, \ldots, r_{k-1} \in [0, q^{\lceil n/k \rceil} - 1]$ and $r_k \in [0, q^{n-(k-1)\lceil n/k \rceil} - 1]$. For $i \in [1, k-1]$, let $\bar{f}_i \in F_{\lceil n/k \rceil}$ be regular of weight $q^{\lceil n/k \rceil + 1} + r_i$, and let $\bar{f}_k \in F_{n-(k-1)\lceil n/k \rceil}$ be regular of weight $q^{n-(k-1)\lceil n/k \rceil + 1} + r_k$. Set

$$f_r := \prod_{i=1}^{k} \bar{f}_i[(i-1)\lceil n/k \rceil].$$

The following proposition shows that f_r is mostly regular with weight precisely $r + \sum_{i=1}^{k} q^{n-(i-1)\lceil n/k \rceil + 1}$.

Proposition III.2. *There exists a pole divisor G of degree at most $q^n(kq+k-1)$ and weight $\sum_{i=1}^{k} q^{n-(i-1)\lceil n/k \rceil + 1}$ such that for all $r \in [0, q^n - 1]$,*

$$f_r \in \mathcal{L}(G + r(P_\infty^{(n)})) \setminus \mathcal{L}(G + (r-1)(P_\infty^{(n)})).$$

Proof. Using proposition III.1, it is easy to see that f_r has weight $r + \sum_{i=1}^{k} q^{n-(i-1)\lceil n/k\rceil+1}$ and is regular outside of $S^{(n)}_{0,(k-1)\lceil n/k\rceil-1}$. It remains to bound the pole orders at the places in $S^{(n)}_{0,(k-1)\lceil n/k\rceil-1}$.

Let $\bar{f}_1, \ldots, \bar{f}_k$ be as in the definition of f_r. For $i \in [1, k-1]$, $\bar{f}_i$ has weight less than $q^{\lceil n/k\rceil+1} + q^{\lceil n/k\rceil}$ in $F_{\lceil n/k\rceil}$. Thus by Prop. III.1, the pole divisor of $\bar{f}_i[(i-1)\lceil n/k\rceil]$ in $F_{i\lceil n/k\rceil}$ satisfies

$$(\bar{f}_i[(i-1)\lceil n/k\rceil])^{(i\lceil n/k\rceil)}_{\infty} - (q^{\lceil n/k\rceil+1} + r_i)P^{(i\lceil n/k\rceil)}_{\infty} \leq (q+1)q^{\lceil n/k\rceil} \sum_{t=0}^{(i-1)\lceil n/k\rceil-1} q^{\max\{0,2t-i\lceil n/k\rceil\}} S^{(i\lceil n/k\rceil)}_t.$$

Here we use $S^{(i\lceil n/k\rceil)}_t$ as a shorthand for the divisor $\sum_{P\in S^{(i\lceil n/k\rceil)}_t} P$. Let G_i denote the divisor on the right-hand side. By direct computation or by the same trick used in the proof of Proposition III.1, we have $\deg(G_i) = (q+1)(q^{i\lceil n/k\rceil} - q^{\lceil n/k\rceil})$.

Next, $\bar{f}_k$ has weight less than $q^{n-(k-1)\lceil n/k\rceil+1} + q^{n-(k-1)\lceil n/k\rceil}$ in $F_{n-(k-1)\lceil n/k\rceil}$. Hence again

$$(\bar{f}_k[(k-1)\lceil n/k\rceil])^{(n)}_{\infty} - (q^{n-(k-1)\lceil n/k\rceil+1} + r_k)P^{(n)}_{\infty} \leq (q+1)q^{n-(k-1)\lceil n/k\rceil} \sum_{t=0}^{(k-1)\lceil n/k\rceil-1} q^{\max\{0,2t-n\}} S^{(i\lceil n/k\rceil)}_t.$$

Let G_k denote the divisor on the right-hand side. As above, we have $\deg(G_k) = (q+1)(q^n - q^{n-(k-1)\lceil n/k\rceil})$.

Using the pole divisors computed above, it is easy to see that

$$(f_r)^{(n)}_{\infty} - v_{P^{(n)}_{\infty}}(f_r)P^{(n)}_{\infty} \leq \sum_{i=1}^{k-1} \mathrm{Con}^{F_n}_{F_{i\lceil n/k\rceil}} G_i + G_k.$$

Define

$$G = \left(\sum_{i=1}^{k} q^{n-(i-1)\lceil n/k\rceil+1}\right) P^{(n)}_{\infty} + \sum_{i=1}^{k-1} \mathrm{Con}^{F_n}_{F_{i\lceil n/k\rceil}} G_i + G_k.$$

Then the above remarks show that

$$f_r \in \mathcal{L}(G + r(P^{(n)}_{\infty})) \setminus \mathcal{L}(G + (r-1)(P^{(n)}_{\infty})),$$

and

$$\begin{aligned}\deg(G) &= \sum_{i=1}^{k} q^{n-(i-1)\lceil n/k\rceil+1} \\ &\quad + \sum_{i=1}^{k-1}(q+1)(q^n - q^{n-(i-1)\lceil n/k\rceil}) \\ &\quad + (q+1)(q^n - q^{n-(k-1)\lceil n/k\rceil}) \\ &\leq k(q+1)q^n - q^n = q^n(kq+k-1).\end{aligned}$$

□

For general $r \geq 0$, say $r = sq^n + t$ with $t \in [0, q^n - 1]$, set

$$f_r := x_0^s f_t.$$

Then we again have $f_r \in \mathcal{L}(G + r(P^{(n)}_{\infty})) \setminus \mathcal{L}(G + (r-1)(P^{(n)}_{\infty}))$ because x_0 is regular of weight q^n.

IV. Code Sequences Beating the Gilbert-Varshamov Bound

Define an $\mathbb{F}_{q^2}$-linear map $\psi : \mathbb{F}_{q^2}^{\mathbb{N}^0} \to \bigcup_r \mathcal{L}(G + rP^{(n)}_{\infty})$ by sending the r-th basis vector to f_r (we zero-index the basis vectors). Because the f_r have distinct weights, the strict triangle inequality implies that ψ is injective.

Each f_r is regular at all of the code places. Hence we can speak of the evaluation map $\mathrm{ev} : \bigcup_r \mathcal{L}(G + rP^{(n)}_{\infty}) \to \mathbb{F}_{q^2}^{q^n(q^2-q)}$, which maps a function to the tuple of its values at the code places.

Proposition IV.1. *Let $K \in [1, q^n(q^2-q-kq-k+1)]$. Then the map $(\mathrm{ev}\circ\psi)|_{\mathbb{F}_{q^2}^K} \colon \mathbb{F}_{q^2}^K \to \mathbb{F}_{q^2}^{q^n(q^2-q)}$ is injective, and its image defines an $[N, K, D]$ code, where $N = q^n(q^2-q)$ and*

$$D \geq N - K - q^n(kq+k-1) + 1.$$

Proof. This follows from the above bound on $\deg(G)$ and a standard argument about algebraic geometry codes. For completeness, we give the proof in full.

Let $D^* := N - K - q^n(kq+k-1)+1$. Suppose that for some nonzero $v \in \mathbb{F}_{q^2}^K$, the N-tuple $\mathrm{ev}(\psi(v))$ has less than D^* nonzero coordinates. Let $M > N - D^*$ be the number of coordinates which are zero. We already know that $\psi(v) \in \mathcal{L}(G + (K-1)P^{(n)}_{\infty})$. By definition of M, there are code places $P_1, \ldots, P_M$ at which $\psi(v)$ is zero. Then $\psi(v)$ lies in the Riemann-Roch space $\mathcal{L}(E)$, where

$$E = G + (K-1)P^{(n)}_{\infty} - \sum_{i=1}^{M} P_i.$$

But $\deg(E) = \deg(G) + K - 1 - M < q^n(kq+k-1) + K - 1 - N + D^* = 0$ by Proposition III.2 and the definition of D^*, so $\mathcal{L}(E) = \{0\}$ and $\psi(v) = 0$. But we said above that ψ is injective, so this is a contradiction.

To see that $(\mathrm{ev}\circ\psi)|_{\mathbb{F}_{q^2}^K}$ is injective, note that for $K \leq q^n(q^2-q-kq-k+1)$, we have $D^* \geq 1$, hence the above argument shows that at least one coordinate of $\mathrm{ev}(\psi(v))$ is nonzero. □

The above proposition implies that for any $n \geq 0$, we can define codes of the above form with length $q^n(q^2-q)$ over $\mathbb{F}_{q^2}$ whose rate R and relative distance δ satisfy

$$R + \delta \geq 1 - \frac{kq+k-1}{q^2-q},$$

with many choices of rate. For all k, this exceeds the Gilbert-Varshamov bound for large enough q.

V. Subquadratic Time Encoding

The encoding task is: given a message $v \in \mathbb{F}_{q^2}^{q^n(q^2-q-kq-k+1)}$, output $\mathrm{ev}(\psi(v))$. For simplicity, we assume throughout this section that k divides n; this affects the runtime by a factor of at most $\mathrm{poly}(q)$, which is a constant in our context. Our goal in this section is to prove the following result.

Theorem V.1. *Assume $k \mid n$. For the codes described in Proposition IV.1, there exist deterministic algorithms to:*

- *pre-compute a representation of the code at the encoder using $O((n/k)^3 q^4 (q^n)^3)$ operations over $\mathbb{F}_q$; this representation occupies $O(q^2 (q^n)^{2/k} \log q)$ space*
- *encode a message using $O(kq^4 (q^n)^{1+\frac{\omega-2}{k}})$ operations over $\mathbb{F}_q$, where ω is the matrix multiplication exponent.*

Taking $k = 2$ implies the encoding portion of Theorem I.1, noting that there we treat q as a constant and instead take $N = q^n(q^2 - q)$ to be the parameter of interest. See Table I for a comparison of these runtimes for $k = 2, 3$.

Our approach is to first write the encoding of a vector $w \in \mathbb{F}_{q^2}^{q^{i(n/k)}}$ with respect to $F_{i(n/k)}$ in terms of some encodings with respect to $F_{(i-1)(n/k)}$ and $F_{n/k}$. We then use a Baby-Step Giant-Step algorithm and fast matrix multiplication to build up the encoding of v starting from encodings with respect to $F_{n/k}$.

A. Pre-computation

Pre-compute the evaluations of some $g_0, \dots, g_{q^{n/k}-1} \in F_{n/k}$ at the code places of $F_{n/k}$, where each g_s is regular of weight $q^{n/k+1} + s$. This can be done using the deterministic algorithm in [32] which has runtime roughly cubic in the number of code places. We then need to store $O(q^{n/k} \cdot q^{n/k}(q^2 - q)) = O(q^2 (q^n)^{2/k})$ elements of $\mathbb{F}_q$.

B. Subquadratic Time Encoding with Fast Matrix Multiplication

We begin by considering encoding for $v \in \mathbb{F}_{q^2}^{q^n}$. Encoding messages of length greater than q^n will be dealt with at the end of this subsection.

For $i \in [0, k]$, $w \in \mathbb{F}_{q^2}^{q^{i(n/k)}}$, and P a code place of $F_{i(n/k)}$, we set $w(P) := \psi_i(w)(P)$, where ψ_i is the function ψ corresponding to $F_{i(n/k)}$.

Proposition V.2. *Let $i \in [1, k]$, let $w \in \mathbb{F}_{q^2}^{q^{i(n/k)}}$, and let P be a code place of $F_{i(n/k)}$. Uniquely write*

$$w = \sum_{\ell=0}^{q^{n/k}-1} \iota_\ell(w^{(\ell)})$$

for $w^{(\ell)} \in \mathbb{F}_{q^2}^{q^{(i-1)(n/k)}}$, where $\iota_\ell : \mathbb{F}_{q^2}^{q^{(i-1)(n/k)}} \hookrightarrow \mathbb{F}_{q^2}^{q^{i(n/k)}}$ is the vector space embedding sending the j-th basis vector to the $(j + \ell q^{(i-1)(n/k)})$-th basis vector. Let P' denote the place obtained by restricting P to $F_{(i-1)(n/k)}$, and let P'' denote the place of $F_{n/k}$ at which x_0 has value $x_{(i-1)(n/k)}(P)$, x_1 has value $x_{(i-1)(n/k)+1}(P)$, etc. Then P' and P'' are code places, and

$$w(P) = \sum_{\ell=0}^{q^{n/k}-1} w^{(\ell)}(P') g_\ell(P'').$$

Proof. By [9, Lemma 3.9], the code places of F_m are precisely the places at which each x_t has value in $\mathbb{F}_{q^2} \setminus \Omega$, subject to the relations defining the Garcia-Stichtenoth tower. Both P' and P'' have this form, so they are code places.

Next, by the definition of the f_r, it is easy to see that

$$\psi_i(w) = \sum_{\ell=0}^{q^{n/k}-1} \psi_{i-1}(w^{(\ell)}) \left(g_\ell[(i-1)(n/k)]\right).$$

The claimed equation follows immediately. □

Observe that $w(P)$ looks like an element of a product of two matrices. We can write down an explicit matrix product as follows.

Fix $i \in [1, k]$ and $\alpha \in \mathbb{F}_{q^2} \setminus \Omega$. Let $\mathcal{P}_\alpha$ denote the set of code places P of $F_{i(n/k)}$ for which $x_{(i-1)(n/k)}(P) = \alpha$, let $\mathcal{P}'_\alpha$ denote the set of code places Q of $F_{(i-1)(n/k)}$ such that $x_{(i-1)(n/k)}(Q) = \alpha$, and let $\mathcal{P}''_\alpha$ denote the set of code place R of $F_{n/k}$ such that $x_0(R) = \alpha$. Then it is easy to see that for any $Q \in \mathcal{P}'_\alpha$ and $R \in \mathcal{P}''_\alpha$, there is a unique place $P \in \mathcal{P}_\alpha$ such that $P' = Q$ and $P'' = R$, where P' and P'' are as in the above proposition. Conversely, if $P \in \mathcal{P}_\alpha$, then $P' \in \mathcal{P}'_\alpha$ and $P'' \in \mathcal{P}''_\alpha$.

Easily $|\mathcal{P}'_\alpha| = q^{(i-1)(n/k)}$ and $|\mathcal{P}''_\alpha| = q^{n/k}$. Let $Q^\alpha_1, \dots, Q^\alpha_{q^{i(n/k)}}$ be an enumeration of $\mathcal{P}'_\alpha$, and let $R^\alpha_1, \dots, R^\alpha_{q^{n/k}}$ be an enumeration of $\mathcal{P}''_\alpha$.

Proposition V.3. *Let $i \in [1, k]$, and let $w \in \mathbb{F}_{q^2}^{q^{i(n/k)}}$. Write $w = \sum_{\ell=0}^{q^{n/k}-1} \iota_\ell(w^{(\ell)})$ as in Proposition V.2. For each $\alpha \in \mathbb{F}_{q^2} \setminus \Omega$, define a matrix A^α of shape $q^{(i-1)(n/k)} \times q^{n/k}$ and a matrix B^α of shape $q^{n/k} \times q^{n/k}$ by*

$$A^\alpha_{st} = w^{(t-1)}(Q^\alpha_s) \qquad B^\alpha_{st} = g_{s-1}(R^\alpha_t).$$

Then for every code place P of $F_{i(n/k)}$, letting $\alpha = x_{(i-1)(n/k)}(P)$ and letting s, t be such that $P' = Q^\alpha_s$ and $P'' = R^\alpha_t$, we have $w(P) = (A^\alpha B^\alpha)_{st}$.

Proof. This is just a restatement of Proposition V.2. □

Using this proposition, it is not too difficult to define an algorithm MATRIX-ENCODE which encodes $v \in \mathbb{F}_{q^2}^{q^n}$ using a series of $k(q^2 - q)$ matrix multiplications of shape

$$\left(q^{n-n/k} \times q^{n/k}\right) \times \left(q^{n/k} \times q^{n/k}\right). \tag{4}$$

Encoding for messages of length longer than q^n: To encode $v \in \mathbb{F}_{q^2}^{q^n(q^2-q-kq-k+1)}$, we just need to make $q^2 - (k+1)q$ calls to MATRIX-ENCODE, using the fact that $f_{sq^n+t} = x_0^s f_t$.

C. Complexity of the Encoding Algorithm

Performing the pre-computation using the algorithm in [32] requires at most $(n/k)^3 q^{3n/k+4} = O((n/k)^3 q^4 (q^n)^{3n/k})$ multiplications and divisions in $\mathbb{F}_{q^2}$. Storing the evaluations requires space $O(q^{2n/k+2}) = O(q^2 (q^n)^{2/k})$, since there are $q^{n/k}$ functions g_s and $O(q^{n/k+2})$ code places of $F_{n/k}$.

Next, we compute the runtime of the encoding function for general $v \in \mathbb{F}_{q^2}^{q^n(q^2-q-kq-k+1)}$. The runtime of each call to MATRIX-ENCODE is just the runtime of $k(q^2 - q)$ matrix multiplications of shape (4). Then the runtime to encode general $v \in \mathbb{F}_{q^2}^{q^n(q^2-q-kq-k+1)}$ is the runtime of $k(q^2 - q)^2 = O(kq^4)$ such multiplications. Thus using fast square matrix multiplication, we can encode v using

$$O\left(kq^4 q^{n-2n/k} (q^{n/k})^\omega\right) = O\left(kq^4 (q^n)^{1+\frac{\omega-2}{k}}\right)$$

operations over $\mathbb{F}_{q^2}$, where ω is the exponent of (square) matrix multiplication. This complete the proof of Theorem V.1.

procedure MATRIX-ENCODE($v \in \mathbb{F}_{q^2}^{q^n}$)
  $W_k \leftarrow \{v\}$
  for i from k to 1 **do**
    $W_{i-1} \leftarrow \emptyset$
    for $w \in W_i$ **do**
      Write $w = \sum_{\ell=0}^{q^{n/k}-1} \iota_\ell(w^{(\ell)})$ as in Prop. V.2
      Add all $w^{(\ell)}$ to W_{i-1}
    end for
  end for
  for i from 1 to k **do**
    for $\alpha \in \mathbb{F}_{q^2} \setminus \Omega$ **do**
      for $w \in W_i$ **do**
        Construct the matrix A_w^α corresponding to w in Prop. V.3, using the $w^{(\ell)}(Q)$ computed in iteration $i-1$ (when $i = 1$, just use the values of the scalars $w^{(\ell)}$)
      end for
      Let $\bar{A}^\alpha$ be the matrix made of all A_w^α stacked vertically, and let B^α be as in Prop. V.3
      Multiply $\bar{A}^\alpha$ by B^α, thus computing $w(P)$ for all $w \in W_i$ and code places P of $F_{i(n/k)}$
    end for
  end for
end procedure

Algorithm 1. The algorithm MATRIX-ENCODE. It inputs $v \in \mathbb{F}_{q^2}^{q^n}$ and outputs $\mathrm{ev}(\psi(v))$.

Using the best known bound $\omega \leq 2.373$ [15], we attain the runtime $O\left(kq^4(q^n)^{1+0.373/k}\right)$. When $k \geq 3$, we can instead use fast $(M^2 \times M) \times (M \times M)$ rectangular matrix multiplication; letting ω' be the exponent of such multiplication, we get an algorithm running in time $O(kq^4(q^n)^{1+(\omega'-3)/k})$. We compare the parameters for $k = 2$ and 3 in Table I below.

TABLE I
COMPARISON OF ENCODING TIMES AND CODE QUALITY FOR $k = 2, 3$.

k	2	3
Preprocessing time	$O(N^{1.5}\log^3 N)$	$O(N\log^3 N)$
Encoding time with $\omega = 2$	$O(N)$	$O(N)$
Encoding time with $\omega \approx 2.37$	$O(N^{1.19})$	$O(N^{1.13})$
Encoding time with $\omega' \approx 3.34$	N/A	$O(N^{1.12})$
Smallest q beating G-V bound	19	32

Here N is the code length, ω is the exponent of square matrix multiplication, and ω' is the exponent of $(M^2 \times M) \times (M \times M)$ rectangular matrix multiplication. Runtime dependence on q is absorbed in the asymptotic notation as q is a constant for each family of codes.

D. Systematic Subcodes for Interactive Oracle Proofs

We sketch a construction of systematic subcodes that lowers the efficiency exponent of the IOPs in [26, Lem 7.2] from $c > 3$ to $c > 3/2$. For a positive integer n, let G_n be the generator matrix for an instance of our code over F_n which has rate and relative distance at least $1/4$; this exists so long as q is sufficiently large as a function of k. For any particular $\alpha \in \mathbb{F}_{q^2} \setminus \Omega$, there are q^n rows of G_n corresponding to points with $x_{n/2} = \alpha$; let H denote G_n restricted to these rows. Then after rearranging rows and columns, we have $H = A \otimes B$, where A is the restriction of $G_{n/2}$ to rows corresponding to points with $x_{n/2} = \alpha$, and B is the restriction of $G_{n/2}$ to rows corresponding to points with $x_0 = \alpha$. Letting A and B have ranks r_1 and r_2, by column reducing $G_{n/2}$ (in different ways for A and B), we can take A and B to begin with the diagonal blocks I_{r_1} and I_{r_2}. Thus after applying column operations to G_n, we can assume that H begins with the diagonal block $I_{r_1 r_2}$. It follows that there is a systematic subcode of G_n's code with dimension $r_1 r_2$ and relative distance at least $1/4$. Furthermore, this subcode can be encoded in time $O(N^{\omega/2})$ with preprocessing time $O(N^{3/2}\log^3 N)$, as with our original code.

It remains to show that we can always choose α so that $r_1 r_2$ is a positive constant fraction of N. The sum of $r_1 r_2$ across all α is just the rank of G_n, which is $K \geq N/4$. Thus there exists α for which $r_1 r_2 / N \geq 1/(4q^2)$, as desired. We can find such an α during the preprocessing step in time $O(N^{3/2})$, since computing $r_1 r_2$ for each α just requires finding the ranks of $2(q^2 - q)$ matrices of size $q^{n/2} \times q^{n/2}$.

VI. FAST DECODING ALGORITHMS

Reed-Solomon codes are widespread in practice partly due to fast algebraic decoding algorithms: the Gorenstein-Zierler decoder, rational approximation using the Euclidean algorithm, the Berlekamp-Massey algorithm and fast Fourier decoders, to name a few. In particular, Reed-Solomon codes can be uniquely decoded in subquadratic time up to the unique decoding limit. In a breakthrough, Sudan designed an algorithm to list decode Reed-Solomon codes beyond half the minimum distance [36]. List decoding is a relaxation of unique decoding where the decoder is allowed to output a list of messages and is deemed successful if the message sent is in the list. Shokrollahi and Wasserman soon generalized Sudan's algorithm to algebraic geometry codes [34]. Shortly thereafter, Guruswami and Sudan designed list decoders for both Reed-Solomon codes and algebraic geometry codes that improved on the error correction of previously known algorithms [37]. A novelty they introduced was to use multiplicities in the interpolation step.

We present a unique-decoding algorithm that corrects a fraction of errors close to half the relative distance and a list decoding algorithm to correct beyond that. The algorithms are presented as specializations of the Shokrollahi-Wasserman algorithm to the Garcia-Stichtenoth tower. In particular, we obtain the unique-decoding algorithm as a special case of the list decoding algorithm.

Our results are as follows.

Theorem VI.1. *For the codes described in Proposition IV.1 with dimension K and length $N = q^n(q^2 - q)$, there exist randomized Las Vegas algorithms to:*

- *pre-compute a representation of the code at the decoder in $O(N^{3/2}\log^3 N)$ time; this representation occupies $O(N^{2/k})$ space*
- *uniquely decode up to $\frac{1}{2}\left(N - K - 1 - 4\frac{kq+k-1}{q^2-q}N\right)$ errors in $O(N^{2+(\omega-2)/k}\log^2 N)$ expected time, where ω is the matrix multiplication exponent*

- *for the list decoding algorithm, additionally pre-compute a matrix for the interpolation step in $O(N^\omega)$ time, occupying $O(N^2)$ space*
- *list decode up to*

$$N\left(1-\sqrt{2\left(R+2\frac{kq+k-1}{q(q-1)}\right)}-2\frac{kq+k-1}{q(q-1)}\right)$$

errors with list size at most $\sqrt{\frac{2(q-1)}{2+R(q-1)}}$ in $O(N^{2+(\omega-2)/k}\log^2 N)$ expected time.

Setting $k=2$ yields the decoding portion of Theorem I.1.

A. Pre-computing a Representation of the Code

This pre-computation step is the same as for the encoding algorithm (see Section V-A) and is deterministic. It is needed so that we may call our encoding algorithm as a sub-routine. Additional pre-computation for the list decoding algorithm is discussed in Section VI-E.

B. Modified Shokrollahi-Wasserman Algorithm

Consider codes of block length $N=q^n(q^2-q)$ and dimension K constructed in §4 with parameter k. Let $P_1,P_2,\ldots,P_N$ denote the code places, which are places of F_n, and let $y=(y_1,y_2,\ldots,y_N)\in\mathbb{F}_{q^2}^N$ denote the received word, where y_i is the (possibly errored) evaluation at P_i. Let ℓ be a bound on the number of messages allowed in the list. Let B be an agreement parameter (determined later), that is, we need to correct fewer than $N-B$ errors. The algorithm first interpolates a nonzero polynomial

$$H(T):=u_0+u_1T+\ldots+u_{\ell-1}T^{\ell-1}+u_\ell T^\ell\in F_n[T]$$

in an indeterminate T such that every message in the list (that is, every message whose encoding agrees with the received word at more than B evaluation places) is a root of $H(T)$. Then the roots of $H(T)$ that are in the message Riemann-Roch space $\mathcal{L}(G+(K-1)P_\infty^{(n)})$ are enumerated as a list of candidate messages. The encoding algorithm is finally used to check which messages sufficiently agree with the received word and indeed belong in the list.

To construct such a polynomial $H(T)$, it suffices that

$$H(y_i)(P_i)=\left(\sum_{j=0}^{\ell}u_j(P_i)y_i^j\right)=0,\forall i\in\{1,2,\ldots,N\} \quad (5)$$

and that

$$u_j\in\mathcal{L}(G+w_jP_\infty^{(n)}),\forall j\in\{0,1,\ldots,\ell\} \quad (6)$$

where $w_j:=B-(\ell+1)\deg(G)-(K-1)j$. The latter constraint $u_j\in\mathcal{L}(G+w_jP_\infty^{(n)})$ ensures that when we substitute a function $f\in\mathcal{L}(G+(K-1)P_\infty^{(n)})$, the resulting function $H(f)$ lies in

$$\mathcal{L}\left((\ell+1)G+(B-(\ell+1)\deg(G))P_\infty^{(n)}\right),$$

hence has a pole divisor of degree at most B.

Suppose ℓ and B are such that $B\geq(N+1)/(\ell+1)+\ell(K-1+2\deg(G))/2+\deg(G)-1$. We claim that we can construct a nonzero polynomial $H(T)$ satisfying the constraints (5) and (6). We attempt to populate each coefficient space $\mathcal{L}(G+w_jP_\infty^{(n)})$ with enough of the functions constructed in §3. In particular, we enforce the second constraint (6) by insisting u_j be in the span of $\{f_0,f_1,\ldots,f_{w_j}\}\subseteq\mathcal{L}(G+w_jP_\infty^{(n)})$. (If $w_j<0$, we take $u_j=0$.) Writing each u_j as an unknown linear combination of $\{f_0,f_1,\ldots,f_{w_j}\}$, the first constraint (5) is an $\mathbb{F}_{q^2}$-linear system in at least $w_0+w_1+\ldots+w_\ell+\ell+1$ variables and N constraints. The condition on ℓ,B ensures

$$\begin{aligned}&w_0+w_1+\ldots+w_\ell+\ell+1\\&=(\ell+1)(B-(\ell+1)\deg(G)-\ell(K-1)/2+1)\\&\geq N+1>N,\end{aligned}$$

proving our claim.

Henceforth, fix $\ell:=\lfloor\sqrt{2N/(K-1+2\deg(G))}\rfloor$ and $B:=\lceil\sqrt{2N(K-1+2\deg(G))}+\deg(G)\rceil$. To prove that we can construct $H(T)$ satisfying the constraints for these values of ℓ and B, observe that

$$\begin{aligned}&(N+1)/(\ell+1)+\ell(K-1+2\deg(G))/2+\deg(G)-1\\&\leq N/(\ell+1)+\ell(K-1+2\deg(G))/2+\deg(G)\\&\leq\frac{N}{\sqrt{2N/(K-1+2\deg(G))}}+\frac{\sqrt{2N(K-1+2\deg(G))}}{2}\\&\quad+\deg(G)\\&\leq B,\end{aligned}$$

hence the above claim applies.

Say $f\in\mathcal{L}(G+(K-1)P_\infty^{(n)})$ agrees with $(y_1,y_2,\ldots,y_N)$ at more than B places. Then $H(f)$ has a zero at more than B places yet pole degree at most B. Hence $H(f)=0$ and f is indeed a root of $H(T)$. Thus we can tolerate fewer than $N-B$ errors.

From the proof of proposition III.2, $1/(q-1)<\deg(G)/N\leq\frac{kq+k-1}{q(q-1)}$. Thus with list size at most $\sqrt{\frac{2(q-1)}{2+R(q-1)}}$, we decode up to

$$N\left(1-\sqrt{2\left(R+2\frac{kq+k-1}{q(q-1)}\right)}-\frac{kq+k-1}{q(q-1)}\right)$$

errors.

In the subsequent subsections, we show how to interpolate $H(T)$ and find its roots in the message space in subcubic time. Our code constructions are stated (for instance in Theorem I.1) as families of codes of increasing block length N for each q and R. Hence q and R shall be treated as constants independent of N in the subsequent complexity estimates. In particular, the list size bound ℓ will be treated as a constant independent of N.

C. Fast Interpolation using Black-Box Linear Algebra

Consider the matrix

$$M := \begin{bmatrix} f_0(P_1) & \dots & f_{w_0}(P_1) & y_1 f_0(P_1) & \dots \\ f_0(P_2) & \dots & f_{w_0}(P_2) & y_2 f_0(P_2) & \dots \\ \vdots & \ddots & \vdots & \vdots & \ddots \\ f_0(P_N) & \dots & f_{w_0}(P_N) & y_N f_0(P_N) & \dots \end{bmatrix}$$
$$\begin{bmatrix} y_1 f_{w_1}(P_1) & \dots\dots\dots & y_1^\ell f_0(P_1) & \dots & y_1^\ell f_{w_\ell}(P_1) \\ y_2 f_{w_1}(P_2) & \dots\dots\dots & y_2^\ell f_0(P_2) & \dots & y_2^\ell f_{w_\ell}(P_2) \\ \vdots & \ddots\ddots\ddots & \vdots & \ddots & \vdots \\ y_N f_{w_1}(P_N) & \dots\dots\dots & y_N^\ell f_0(P_N) & \dots & y_N^\ell f_{w_\ell}(P_N) \end{bmatrix}$$

corresponding to the linear system in (5). Given a column vector

$$a := (a_{0,0}, \dots, a_{0,w_0}, a_{1,0}, \dots, a_{1,w_1}, \dots\dots, a_{\ell,0}, \dots, a_{\ell,w_\ell})^t$$

over $\mathbb{F}_{q^2}$, the matrix-vector product

$$Ma = \left(\sum_{j=0}^{w_0} a_{0,j} f_j(P_i) + y_i \sum_{j=0}^{w_1} a_{1,j} f_j(P_i) + \dots + y_i^\ell \sum_{j=0}^{w_\ell} a_{\ell,j} f_j(P_i),\ i = 1, 2, \dots, N \right)^t$$

can be computed in subquadratic time as follows. For each $b \in \{0, 1, \dots, \ell\}$,

$$\sum_{j=0}^{w_b} a_{b,j} f_j(P_i), \forall i \in \{1, 2, \dots, N\}$$

can be computed in $O(N^{1+(\omega-2)/k})$ time using the encoding algorithm in § V by viewing it as encoding the function $\sum_{j=0}^{w_b} a_{b,j} f_j$. Taking the inner product with $(y_1^b, y_2^b, \dots, y_N^b)^t$ yields

$$y_i^b \sum_{j=0}^{w_b} a_{b,j} f_j(P_i), \forall i \in \{1, 2, \dots, N\}$$

in linear time. Adding up these terms obtained for $b \leq \ell$ yields Ma in $O(N^{1+(\omega-2)/k})$ time. We invoke Wiedemann's algorithm [35] to find a nonzero solution to $Mx = 0$ and obtain the interpolation polynomial. Since matrix-vector products take $O(N^{1+(\omega-2)/k})$ time, the linear system is solved in $O(N^{2+(\omega-2)/k} \log^2 N)$ expected time.

D. Unique Decoding

We obtain our unique-decoding algorithm by choosing $\ell = 1$ instead of the above choice of l. Since $H(T)$ is now degree one, the root finding step is trivial and involves just one division of functions. For $\ell = 1$, we may choose $B = \frac{1}{2}(N + K) + 2\deg(G)$, allowing us to correct $\frac{1}{2}(N - K - 1) - 2\deg(G)$ errors. The unique decoding bound assures that the code can correct $(N - K - \deg(G) - 1)/2$ errors, that is, a $(1 - R - \deg(G)/N)/2$ fraction of errors. Our algorithm is guaranteed to correct a $(1 - R - 4\deg(G)/N)/2$ fraction of errors, falling short by the small term $(3/2)\deg(G)/N \leq (3/2)(kq + k - 1)/(q^2 - q)$. There are ways (analogous to [38], [39]) to modify the algorithm and correct up to the unique decoding assurance. We refrain from detailing the changes since list decoding subsumes such improvements.

A question, which we leave open, is if unique decoding could be performed in subquadratic expected time. To this end, one might consider the matrix

$$M = \begin{bmatrix} f_0(P_1) & f_1(P_1) & \dots & f_{w_0}(P_1) \\ f_0(P_2) & f_1(P_2) & \dots & f_{w_0}(P_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_0(P_N) & f_1(P_N) & \dots & f_{w_0}(P_N) \end{bmatrix}$$
$$\begin{bmatrix} y_1 f_0(P_1) & y_1 f_1(P_1) & \dots & y_1 f_{w_1}(P_1) \\ y_2 f_0(P_2) & y_2 f_1(P_2) & \dots & y_2 f_{w_1}(P_2) \\ \vdots & \vdots & \ddots & \vdots \\ y_N f_0(P_N) & y_N f_1(P_N) & \dots & y_N f_{w_1}(P_N) \end{bmatrix}$$

corresponding to the linear system of the interpolation step. The bottleneck in unique decoding is computing a nonzero element in the null space of M. If M were to have sublinear (that is, $o(N)$) displacement rank (see [40] for definition), then by [41] this task can be accomplished in sub quadratic time. Displacement ranks of interpolation matrices arising in list decoding were bounded by Olshevsky and Shokrollahi [42, § 5]. However, their bounds apply only to codes from plane curves, and it is not immediate if their techniques imply sublinear displacement rank for M.

E. Root Finding for List Decoding

The main algorithmic challenge left in list decoding is root finding: to enumerate all the roots of $H(T)$ in $\mathcal{L}(G + (K - 1)P_\infty^{(n)})$. Our strategy is to first pick a place P in F_n of degree just greater than $\deg(G) + K - 1$, thus ensuring that the evaluation map from $\mathcal{L}(G + (K - 1)P_\infty^{(n)})$ to the residue field at P is injective. Then find the roots of the reduction of $H(T)$ at P and lift the roots to $\mathcal{L}(G + (K - 1)P_\infty^{(n)})$.

Such a P can be found in nearly linear time by Artin-Schreier theory. Let $D = \deg(G) + K \leq N$. Pick an $\alpha_0 \in \{\alpha \in \mathbb{F}_{q^{2D}} \mid \alpha^q + \alpha \neq 0\} \bigcap \left(\mathbb{F}_{q^{2D}} \setminus \mathbb{F}_{q^{2(D-1)}}\right)$. To find such an α_0, choose α_0 to be a root of a random degree D irreducible polynomial over $\mathbb{F}_{q^2}$ (in time nearly linear in D using the algorithm of Couveignes and Lercier [43]). With probability at least $1 - 1/q \geq 1/2$, $\alpha_0^q + \alpha_0 \neq 0$. Once such an α_0 is found, we look for a place in F_n above $(x_0 - \alpha_0)$. Observing $\alpha_0^q/(\alpha_0^{q-1} + 1)$ is the fraction of the norm α_0^{q+1} and trace $\alpha_0^q + \alpha_0$ of α_0 down to $\mathbb{F}_{q^{2(D-1)}}$, we see $\alpha_0^q/(\alpha_0^{q-1} + 1) \in \mathbb{F}_{q^{2(D-1)}}$. Since the trace from $\mathbb{F}_{q^{2D}}$ down to $\mathbb{F}_{q^{2(D-1)}}$ is surjective,

$$x_1^q + x_1 = \frac{\alpha_0^q}{\alpha_0^{q-1} + 1}$$

has a solution $x_1 = \alpha_1 \in \mathbb{F}_{q^{2D}}$. Such an α_1 can be found either using Hilbert's theorem 90 or using a generic root finding algorithm. The root finding algorithm of Kaltofen-Shoup [44, Alg E, Thm 1] implemented using the Kedlaya-Umans modular composition algorithm [45] takes expected time $O(N^{1+o(1)})$. Further, $\alpha_1 \in \{\alpha \in \mathbb{F}_{q^{2D}} | \alpha^q + \alpha \neq 0\}$ since $\alpha_0^q/(\alpha_0^{q-1} + 1) \neq 0$. We can thus iterate this process up the tower and find a place $P := (\alpha_0 : \alpha_1 : \dots : \alpha_n : 1)$. Since we

insist $\alpha_0 \in \mathbb{F}_{q^{2D}} \setminus \mathbb{F}_{q^{2(D-1)}}$, the degree of P is indeed D. The computation of the point P may be moved to pre-processing (taking $O(N^{1+o(1)})$ expected time and $O(N \log N)$ storage).

Given P, we reduce the coefficients of $H(T)$ modulo P. To perform this reduction in time quadratic in N, we first pre-compute and store $\{f_0(P), f_1(P), \ldots, f_{w_0}(P)\}$ in time $O(N^{1+\omega/k} \log N + N^2 \log N \log\log N)$ and storage space $O(N^2)$, as follows. We assume the f_r's are defined in terms of the basis for regular functions in $F_{n/k}$ returned by the algorithm in [32]. This algorithm writes each regular function in $F_{n/k}$ as an $\mathbb{F}_{q^2}$-linear combination of a fixed set of $O(N^{1/k} \log N)$ functions, each of which can be written as an $O(\log N)$ size arithmetic circuit in terms of $x_0, x_1, \ldots, x_{n/k}$ (see [32, Theorem 6]). We can evaluate all of these functions at each of the places $(\alpha_0 : \alpha_1 : \ldots : \alpha_{n/k} : 1), (\alpha_{n/k} : \alpha_{n/k+1} : \ldots : \alpha_{2n/k} : 1), \ldots, (\alpha_{(k-1)n/k} : \alpha_{(k-1)n/k+1} : \ldots : \alpha_n : 1)$ of $F_{n/k}$ in time $O(N^{1+1/k} \log^3 N \log\log N)$, noting that because all α_i are not roots of $\alpha^q + \alpha = 0$, using the obvious circuit will only require arithmetic operations over $\mathbb{F}_{q^{2D}}$ (i.e., no pole cancelling is required). Representing these values as a $2D \times O(N^{1/k} \log N)$ matrix over $\mathbb{F}_{q^2}$, we can then multiply by the $O(N^{1/k} \log N) \times q^{n/k}$ matrix over $\mathbb{F}_{q^2}$ which writes the regular functions in $F_{n/k}$ in terms of these functions, yielding the evaluations of a basis for $F_{n/k}$ at each of the places above in time $O(N^{1+\omega/k} \log N)$. To finish the pre-computation, we take the $w_0 + 1 = O(N)$ products corresponding to the definitions of the f_r's at P, each of which involves k multiplications over $\mathbb{F}_{q^{2D}}$; in total, this takes time $O(N^2 \log N \log\log N)$.

Once $H(T)$ is reduced, since its degree is a constant independent of N, all its roots can be enumerated in time nearly linear in N using the Kaltofen-Shoup root finding algorithm [44] implemented using the Kedlaya-Umans modular composition algorithm.

The lifting of roots modulo P to the message space takes time quadratic in N with pre-processing requiring runtime exponent ω and storage quadratic in N. To this end, pre-compute a new basis $\{e_0, e_1, \ldots, e_{K-1}\}$ of Span$\{f_0, f_1, \ldots, f_{K-1}\}$. The basis $\{e_0, e_1, \ldots, e_{K-1}\}$ is chosen such that the D by K matrix L over $\mathbb{F}_{q^2}$ whose i^{th} column is $e_i(P)$ (written in a fixed basis for $\mathbb{F}_{q^{2D}}$ over $\mathbb{F}_{q^2}$) is lower triangular. Such a basis can be found by column reduction of the corresponding matrix whose i^{th} column is $f_i(P)$, with runtime exponent ω. In addition to L, we store the matrix R expressing $\{e_0, e_1, \ldots, e_{K-1}\}$ as an $\mathbb{F}_{q^2}$-linear combination of $\{f_0, f_1, \ldots, f_{K-1}\}$. Now given a residue modulo P, by solving the linear system corresponding to L, we can find an $\mathbb{F}_{q^2}$-linear combination of $\{e_0, e_1, \ldots, e_{K-1}\}$ that evaluates to that residue (if one exists). This takes quadratic time since L is in lower triangular form. The matrix R then expresses this lift as an $\mathbb{F}_{q^2}$-linear combination of $\{f_0, f_1, \ldots, f_{K-1}\}$, as desired.

ACKNOWLEDGEMENT

We would like to thank the reviewers for their suggestions.

REFERENCES

[1] E. N. Gilbert, "A comparison of signalling alphabets," *The Bell System Technical Journal*, vol. 31, no. 3, pp. 504–522, May 1952.

[2] R. R. Varshamov, "Estimate of the number of signals in error correcting codes," *Dokl. Acad. Nauk SSSR*, vol. 117, pp. 739–741, 1957.

[3] I. Dumer, D.Micciancio, and M. Sudan, "Hardness of approximating the minimum distance of a linear code," *IEEE Transactions on Information Theory*, vol. 49-1, pp. 22–37, 2003.

[4] A. Vardy, "The intractability of computing the minimum distance of a code," *IEEE Transactions on Information Theory*, vol. 43-6, pp. 1757–1766, 1997.

[5] V. D. Goppa, "Codes on algebraic curves," *Soviet Math. Dokl.*, vol. 24, no. 1, pp. 170–172, 1981.

[6] V. G. Drinfeld and S. G. Vlăduţ, "The number of points of an algebraic curve," *Func. Anal.*, vol. 17, pp. 53–54, 1983.

[7] Y. Ihara, "Some remarks on the number of rational points of algebraic curves over finite fields," *Journal of the Faculty of Science, University of Tokyo, Section IA Mathematics*, vol. 28, no. 3, pp. 721–724, 1981.

[8] M. A. Tsfasman, S. G. Vlăduţ, and T. Zink, "Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound," *Mathematische Nachrichten*, vol. 109, no. 1, pp. 21–28, 1982. [Online]. Available: http://dx.doi.org/10.1002/mana.19821090103

[9] A. Garcia and H. Stichtenoth, "On the asymptotic behaviour of some towers of function fields over finite fields," *Journal of Number Theory*, vol. 61, no. 2, pp. 248 – 273, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022314X9690147X

[10] A. Bassa, P. Beelen, A. Garcia, and H. Stichtenoth, "An improvement of the Gilbert-Varshamov bound over nonprime fields," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 3859–3861, July 2014.

[11] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Transactions on Information Theory*, vol. 24, pp. 384–386, 1978.

[12] D. Spielman, "Linear time encodable and decodable error-correcting codes," *IEEE Transactions on Information Theory*, vol. 46-6, pp. 1723–1731, 1996.

[13] V. Guruswami and P. Indyk, "Linear-time encodable/decodable codes with near-optimal rate," *IEEE Transactions on Information Theory*, vol. 51-10, pp. 3393–3400, 2005.

[14] E. Druk and Y. Ishai, "Linear-time encodable codes meeting the Gilbert-Varshamov bound and their cryptographic applications," in *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ser. ITCS '14. New York, NY, USA: ACM, 2014, pp. 169–182. [Online]. Available: http://doi.acm.org/10.1145/2554797.2554815

[15] F. Le Gall, "Powers of tensors and fast matrix multiplication," in *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ser. ISSAC '14. New York, NY, USA: ACM, 2014, pp. 296–303. [Online]. Available: http://doi.acm.org/10.1145/2608628.2608664

[16] A. Ben-Aroya and A. Ta-Shma, "Constructing small-bias sets from algebraic-geometric codes," *Theory of Computing*, vol. 9(5), p. 252273, 2013.

[17] H. Chen and R. Cramer, "Algebraic geometric secret sharing schemes and secure multi-party computations over small fields," *Advances in Cryptology - CRYPTO*, vol. 4117, 2006.

[18] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22(11), pp. 612–613, 1979.

[19] H. Chen, R. Cramer, R. de Haan, and I. C. Pueyo, "Strongly multiplicative ramp schemes from high degree rational points on curves," *Advances in Cryptology - EUROCRYPT*, pp. 451–470, 2008.

[20] H. Chen, R. Cramer, S. Goldwasser, R. de Haan, and V. Vaikuntanathan, "Secure computation from random error correcting codes," *Advances in Cryptology - EUROCRYPT*, pp. 291–310, 2007.

[21] O. Geil, S. Martin, U. Martinez-Penas, R. Matsumoto, and D. Ruano, "On asymptotically good ramp secret sharing schemes," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E100A(12), p. 26992708, 2017.

[22] U. Martìnez-Peñas, "Communication efficient and strongly secure secret sharing schemes based on algebraic geometry codes," *IEEE Transactions on Information Theory*, vol. 64(6), p. 41964206, 2018.

[23] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, "Delegating computation: Interactive proofs for muggles," *Journal of the ACM*, vol. 62-4, pp. 27(1)–27(64), 2015.

[24] O. Reingold, G. N. Rothblum, and R. D. Rothblum, "Constant-round interactive proofs for delegating computation," *ACM Symposium on Theory of Computing, STOC*, pp. 49–62, 2016.

[25] E. Ben-Sasson, A. Chiesa, and N. Spooner, "Interactive oracle proofs," *Theory of Cryptography Conference, TCC*, pp. 31–60, 2016.

[26] E. Ben-Sasson, A. Chiesa, A. Gabizon, M. Riabzev, and N. Spooner, "Interactive oracle proofs with constant rate and query complexity," Preprint available at https://eprint.iacr.org/2016/324.pdf, 2017.

[27] H. Stichtenoth, *Algebraic Function Fields and Codes*, 2nd ed. Springer Publishing Company, Incorporated, 2008.
[28] M. D. Huang and D. Ierardi, "Efficient algorithms for the Riemann-Roch problem and for addition in the Jacobian of a curve," in *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, Oct 1991, pp. 678–687.
[29] F. Hess, "Computing Riemann-Roch spaces in algebraic function fields and related topics," *Journal of Symbolic Computation*, vol. 33, no. 4, pp. 425 – 445, 2002.
[30] V. Guruswami and M. Sudan, "On representations of algebraic-geometry codes," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1610–1613, May 2001.
[31] I. Aleshnikov, P. V. Kumar, K. W. Shum, and H. Stichtenoth, "On the splitting of places in a tower of function fields meeting the Drinfeld-Vlăduţ bound," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1613–1619, May 2001.
[32] K. W. Shum, I. Aleshnikov, P. V. Kumar, H. Stichtenoth, and V. Deolalikar, "A low-complexity algorithm for the construction of algebraic-geometric codes better than the Gilbert-Varshamov bound," *IEEE Transactions on Information Theory*, vol. 47, no. 6, pp. 2225–2241, Sep 2001.
[33] K. Shum, "A low-complexity construction of algebraic geometric codes better than the Gilbert-Varshamov bound," Ph.D. dissertation, University of Southern California, 12 2000.
[34] M. A. Shokrollahi and H. Wasserman, "List decoding of algebraic-geometric codes," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 432–437, Mar 1999.
[35] D. Wiedemann, "Solving sparse linear equations over finite fields," *IEEE Transactions on Information Theory*, vol. 32-1, pp. 54–62, 1986.
[36] M. Sudan, "Decoding of Reed Solomon codes beyond the error-correction bound," *Journal of Complexity*, vol. 13, no. 1, pp. 180 – 193, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885064X97904398
[37] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1757–1767, Sep 1999.
[38] G. L. Feng and T. R. N. Rao, "Decoding algebraic-geometric codes up to the designed minimum distance," *IEEE Transactions on Information Theory*, vol. 39(1), p. 3745, 1993.
[39] S. Sakata, J. Justesen, Y. Madelung, H. E. Jensen, and T. Hoholdt, "Fast decoding of algebraic-geometric codes up to the designed minimum distance," *IEEE Transactions on Information Theory*, vol. 41, pp. 1672–1677, 1995.
[40] T. Kailath, S.-T. Kung, and M. Morf, "Displacement ranks of a matrix," *Bull. Amer. Math. Soc. (N.S.)*, vol. 1-5, pp. 769–773, 1979.
[41] A. Boston, C.-P. Jeannerod, and E. Schost, "Solving structured linear systems with large displacement rank," *Theoret. Comput. Sci.*, vol. 407(1-3), pp. 155–181, 2008.
[42] V. Olshevsky and A. Shokrollahi, "A displacement approach to decoding algebraic codes," *Contemporary mathematics*, pp. 265–292, 2001.
[43] J.-M. Couveignes and R. Lercier., "Fast construction of irreducible polynomials over finite fields," *Israel Journal of Mathematics*, vol. May, pp. 1–29, 2012.
[44] E. Kaltofen and V. Shoup, "Fast polynomial factorization over high algebraic extensions of finite fields," in *Proc. 1997 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'97)*, 1997, pp. 184–188.
[45] K. S. Kedlaya and C. Umans, "Fast polynomial factorization and modular composition," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1767–1802, 2011.

PLACE PHOTO HERE

Anand Kumar Narayanan was born in Cuddalore, Tamil Nadu, India in 1985. He received the B.E in electronics and communication engineering from Madras Institute of Technology, Chennai, in 2006 and the Ph.D in computer science from the University of Southern California, Los Angeles in 2014. From 2014 to 2016, he was a postdoctoral researcher at the computing and mathematical sciences department at the California Institute of Technology. Since 2017, he is a research scientist at the Laboratoire d'Informatique de Paris 6 and Institut de Mathématiques de Jussieu–Paris Rive Gauche, Sorbonne Université, Paris.

PLACE PHOTO HERE

Matthew Weidner was born in Reading, PA, USA. He received the B.S. degree in mathematics from the California Institute of Technology, Pasadena, CA, USA in 2018 and the M.Phil. degree in advanced computer science from the University of Cambridge, Cambridge, UK in 2019. He is currently pursuing the Ph.D. degree in computer science at Carnegie-Mellon University, Pittsburgh, PA, USA.