Proximity Gaps for Reed–Solomon Codes

Eli Ben-Sasson*, Dan Carmon*, Yuval Ishai†, Swastik Kopparty‡ and Shubhangi Saraf‡

* *StarkWare Industries Ltd.*
Netanya, Israel.
{eli,dancar}@starkware.co
† *Computer Science Department*
Technion, Haifa, Israel
yuvali@cs.technion.ac.il
‡ *Department of Mathematics and Department of Computer Science*
Rutgers University, New Brunswick, NJ
{swastik.kopparty,shubhangi.saraf}@gmail.com

***Abstract*—A collection of sets displays a *proximity gap* with respect to some property if for every set in the collection, either (i) all members are δ-close to the property in relative Hamming distance or (ii) only a tiny fraction of members are δ-close to the property. In particular, no set in the collection has roughly half of its members δ-close to the property and the others δ-far from it.**

We show that the collection of affine spaces displays a proximity gap with respect to Reed–Solomon (RS) codes, even over small fields, of size polynomial in the dimension of the code, and the gap applies to any δ smaller than the Johnson/Guruswami–Sudan list-decoding bound of the RS code. We also show near-optimal gap results, over fields of (at least) *linear* size in the RS code dimension, for δ smaller than the unique decoding radius. Concretely, if δ is smaller than half the minimal distance of an RS code $V \subset \mathbb{F}_q^n$, every affine space is either entirely δ-close to the code, or alternatively at most an (n/q)-fraction of it is δ-close to the code. Finally, we discuss several applications of our proximity gap results to distributed storage, multi-party cryptographic protocols, and concretely efficient proof systems.

We prove the proximity gap results by analyzing the execution of classical algebraic decoding algorithms for Reed–Solomon codes (due to Berlekamp–Welch and Guruswami–Sudan) on a *formal element* of an affine space. This involves working with Reed–Solomon codes whose base field is an (infinite) rational function field. Our proofs are obtained by developing an extension (to function fields) of a strategy of Arora and Sudan for analyzing low-degree tests.

I. Introduction

A variety of protocols, arising in the contexts of interactive proofs, distributed storage and cryptography, give rise to the following problem regarding proximity to a linear code $V \subset \mathbb{F}_q^n$ over a finite field $\mathbb{F}_q$ of minimal relative distance δ_V. These myriad protocols assume oracle access to a batch of vectors $\mathbf{u} = \{u_0, \ldots, u_l\} \subset \mathbb{F}_q^n$ and their soundness requires that each and every vector u_i be close to V in relative Hamming distance. Furthermore, soundness deteriorates as a function of the largest distance between some vector u_i and the code V. Thus, we seek protocols that minimize the number of queries to the entries of the vectors in $\mathbf{u}$, while maximizing the probability of recognizing when some vector u_i is significantly far from V.

The linearity of V suggests a natural approach, first explored by Rothblum, Vadhan and Wigderson [1]: sample a uniformly random vector u' in the span of $\mathbf{u}$ (denoted $\mathrm{span}(\mathbf{u})$) and view the distance between u' and V, denoted $\Delta(u', V)$, as a proxy for the maximal distance between some member of $\mathbf{u}$ and V. To argue soundness, we would like to show that if even a single u_i is δ-far from (all members of) V, then a randomly chosen u' is also far from V. Indeed, the paper [1] that suggested this approach also showed for any V, that whenever a single u_i is δ-far from V, then nearly all samples u' are at least $\delta/2$-far from V. Here and henceforth, we use Δ to denote relative Hamming distance and say "u is δ-close to V", denoted $\Delta(u, V) \leq \delta$, when $\Delta(u, v) \leq \delta$ for some $v \in V$; otherwise we say "u is δ-far from V" (denoted $\Delta(u, V) > \delta$).

Note that the result above incurs a $2\times$ degradation in the proximity parameter δ: the worst-case assumption — that some u_i is δ-far from V — implies an average-case distance that is only $\delta/2$. Eliminating the proximity degradation is easy when the field size is exponential in the code length. More concretely, if $q \gg 2^{nH(\delta)}$, where H is the binary entropy function, then a union bound over agreement sets shows that for $\delta < \delta_V$, if u_i is δ-far from V then so are nearly all $u' \in \mathrm{span}(\mathbf{u})$. However, exponential field size is prohibitively large in the context of the motivating applications. Obtaining similar results over fields of sub-exponential size appears to be much more challenging.

A number of works looked at this question and were able to remove the degradation in δ with polynomial field size. Ames et al. [2] showed that for proximity parameters δ that are smaller than half of the unique-decoding radius of V (i.e., when $\delta < \delta_V/4$), nearly all $u' \in \mathrm{span}(\mathbf{u})$ are δ-far from V. The proximity bound was subsequently improved to $\delta < \delta_V/3$ by Roth and Zémor [3]. Ben-Sasson et al. [4] showed similar results for δ above the unique decoding radius, holding for

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00088

any $\delta < 1 - \sqrt[4]{1-\delta_V}$, and the state of the art[1] was given in [5], holding for any $\delta < 1 - \sqrt[3]{1-\delta_V}$. In fact, this latter result was shown to be tight for certain RS codes, in particular, of maximal blocklength $n = q$.

Ames et al., who were the first to show that in certain cases the average-case distance of $u' \in \mathrm{span}(\mathbf{u})$ from V is nearly-always equal to the worst-case distance of $u_i \in \mathbf{u}$ from V, also raised the following intriguing question, which is at the focus of our investigation here: For which codes and what range of δ does the following statement hold?

> *If some $u^* \in \mathrm{span}(\mathbf{u})$ is δ-far from V, then so are nearly all $u' \in \mathrm{span}(\mathbf{u})$.*

One implication of our main result is that when V is an RS code over a sufficiently large field — polynomially large in the code's blocklength — and when δ is smaller than the Johnson/Guruswami–Sudan list decoding bound, the above phenomenon holds. We refer to it as a *proximity gap*, as explained next.

A. Gaps and proximity gaps

When a "gap" in mentioned in theoretical computer science, it usually refers to a situation where all objects under consideration must fall into one of two categories, and these categories display a large gap according to some metric. Striking examples are given by PCP reductions whose outputs are constraint satisfaction problems that lie in one of two categories: satisfiable instances in which some assignment satisfies all constraints, and unsatisfiable instances in which all assignments fail to satisfy more than an ϵ fraction of constraints. Another gap example underlies randomized algorithms. For instance, the Miller–Rabin primality test relies on a gap between primes and composites: in the latter case (composites), at least three-quarters of the integers serve as composite witnesses whereas for primes none do, leading to a "gap" of measure $3/4$.

Our result can be phrased as a *proximity gap* according to the following definition.

Definition I.1 (Proximity gap)**.** *Let $\mathsf{P} \subset \Sigma^n$ be a property and $\mathsf{C} \subset 2^{\Sigma^n}$ be a collection of sets. Let Δ be a distance measure on Σ^n. We say that C displays a (δ, ϵ)-proximity gap with respect to P under Δ if every $\mathsf{S} \in \mathsf{C}$ satisfies exactly one of the following:*

1) $\Pr_{s \in \mathsf{S}}[\Delta(s, \mathsf{P}) \leq \delta] = 1$.
2) $\Pr_{s \in \mathsf{S}}[\Delta(s, \mathsf{P}) \leq \delta] \leq \epsilon$.

We call δ the proximity parameter *and ϵ is the* error parameter*. By default, Δ denotes the relative Hamming distance measure.*

Using this definition we can state our main result. Informally, it says that if $V \subset \mathbb{F}^n$ is an RS-code and $A \subset \mathbb{F}^n$ is an affine space, then either all elements of A are close to V, or otherwise, nearly all elements of A are far from V. In other words, there is no affine A in which roughly half of the elements are close to V while the other half are far from V.

Throughout this paper, $\mathbb{F}_q$ denotes the field of size q, and $\mathsf{RS}[\mathbb{F}_q, \mathcal{D}, k]$ is the RS code of *dimension* $k+1$ and *block-length* $n = |\mathcal{D}|$ containing as its codewords the polynomials of degree $\leq k$, evaluated on $\mathcal{D}$. We use ρ to denote the *rate* $\rho = \frac{k+1}{n}$ of the code. The letter δ will typically denote relative Hamming distance to the relevant RS code and ϵ will denote an error parameter, the probability that a "bad event" occurs (with varying definitions of the term "bad event").

The following result has two parts and each part has its own proof. The first part holds only below the unique decoding radius but has a smaller error parameter, denoted $\epsilon_{\cup}$; the second part holds for proximity parameters up to the Johnson/Guruswami–Sudan bound (which is greater than the unique decoding bound) but has a larger error bound $\epsilon_{\cap}$ (the proof of the second part is also significantly harder).

Theorem I.2 (Proximity Gap for RS codes)**.** *The collection $\mathsf{C}_{\mathsf{Affine}}$ of affine spaces in $\mathbb{F}_q^{\mathcal{D}}$ displays a (δ, ϵ)-proximity gap with respect to the RS-code $V := \mathsf{RS}[\mathbb{F}_q, \mathcal{D}, k]$ of blocklength n and rate $\rho = \frac{k+1}{n}$, for any $\delta \in (0, 1 - \sqrt{\rho})$, and $\epsilon = \epsilon(q, n, \rho, \delta)$ defined as the following piecewise function:*

- **Unique decoding bound:** *For $\delta \in \left(0, \frac{1-\rho}{2}\right]$, the error parameter ϵ is*

$$\epsilon = \epsilon_{\cup} = \epsilon_{\cup}(q, n) := \frac{n}{q}. \tag{I.1}$$

- **Johnson bound:** *For $\delta \in \left(\frac{1-\rho}{2}, 1 - \sqrt{\rho}\right)$, setting $\eta := 1 - \sqrt{\rho} - \delta$, the error parameter ϵ is*

$$\epsilon = \epsilon_{\cap} = \epsilon_{\cap}(q, n, \rho, \delta) := \frac{(k+1)^2}{\left(2 \min\left(\eta, \frac{\sqrt{\rho}}{20}\right)\right)^7 q}$$
$$= O\left(\frac{1}{(\eta\rho)^{O(1)}} \cdot \frac{n^2}{q}\right) \tag{I.2}$$

There are two striking aspects to this result. First, the proximity parameter δ can take any value smaller than the famous Johnson/Guruswami–Sudan bound, which is the largest distance for which we know of efficient (list) decoding algorithms. (Looking ahead, the Guruswami–Sudan algorithm will play a crucial, though non-algorithmic, role in our proofs.) Second, the size of the field needed to achieve this result is relatively small — linear in the blocklength when δ is below the unique decoding radius $\delta < (1-\rho)/2$ and, for fixed rate, quadratic in blocklength for larger δ up to the list decoding bound.

Remark I.1 (Tightness of results). The maximal proximity parameter δ for which Theorem I.2 applies happens to coincide with the Johnson/Guruswami–Sudan list-decoding bound $(1 - \sqrt{\rho})$. This evidently follows from the techniques we use here, that rely on list-decoding algorithms that reach that bound. However, we conjecture that Theorem I.2 holds even for larger proximity parameters, up to capacity $(1 - \rho)$. See Conjecture VII.3 in [6] and the discussion there.

Remark I.2 (Field size). The bound in Eq. (I.2) which reaches the Johnson bound becomes nontrivial only for fields of size

[1] We note that these improvements give a roughly $2\times$ improvement to the protocol of [1] in which this question was originally studied, when that protocol is instantiated with codes of sufficiently large relative distance (see Theorem 3.4 there).

q that are at least quadratically larger than the blocklength n. In contrast, the bound for smaller proximity parameters, below the unique decoding radius, works for $q = O(n)$ (see Eq. (I.1)). We point out that for certain combinations of fields and rate parameters one cannot hope to reach the Johnson bound with linear size fields, as this would contradict prior results from [4].

In the unique decoding regime, the result is sharp in the sense that affine spaces do not all display a proximity gap with $q \cdot \epsilon$ being sublinear in n, for fixed distance parameter δ. A simple example is of the affine line $\{u_0 + zu_1 : z \in \mathbb{F}_q\}$, where $u_0, u_1 : \mathcal{D} \to \mathbb{F}_q$ are such that on a set $\mathcal{D}' \subset \mathcal{D}$ of size $|\mathcal{D}'| = n(1-\delta) - 1$ we have $u_0|_{\mathcal{D}'} = u_1|_{\mathcal{D}'} = 0$, and on the complement we have that $u_1|_{\mathcal{D}\setminus\mathcal{D}'} = 1$, and u_0 takes $\delta n + 1$ pairwise different non-zero values. We then have $\mathrm{dist}(u_0 + zu_1, V) \leq \delta$ for each of the $\delta n + 1$ values of $z \in \mathbb{F}_q$ for which $-z$ is in the image of $u_0|_{\mathcal{D}\setminus\mathcal{D}'}$, but that $\mathrm{dist}(u_0, V) = \delta + \frac{1}{n} > \delta$, thus this line does not display a $(\delta, \frac{\delta n}{q})$ proximity gap with respect to the code.

B. Concentration bounds

Theorem I.2 implies the following concentration bound, saying that for any affine space in which the element farthest from the RS code is within the Johnson/Guruswami–Sudan radius, nearly all elements are at exactly the same distance from the code(!).

For two sets $U, V \subset \Sigma^n$ define the *divergence*[2] of U from V as $\mathsf{D}(U, V) := \max_{u \in U} \Delta(u, V)$.

Corollary I.3 (Concentration bounds)**.** *Let V, q, n, k and ρ be as defined in Theorem I.2. Let $U \subset \mathbb{F}_q^{\mathcal{D}}$ be an affine space over $\mathbb{F}_q$ and denote $\delta^* := \mathsf{D}(U, V)$. If δ^* is smaller than the Johnson/Guruswami–Sudan bound, then nearly all elements of U have distance* exactly *δ^* from the code. In other words, if $\delta^* \in (0, 1 - \sqrt{\rho})$, then*

$$\Pr_{u \in U}\left[\Delta(u, V) \neq \delta^*\right] \leq \epsilon,$$

where $\epsilon = \epsilon(q, n, \rho, \delta^*)$ is as defined in Theorem I.2.

When the divergence of U from the RS code V is greater than the Johnson/Guruswami–Sudan bound ($\delta^* > 1 - \sqrt{\rho}$) we may still use Theorem I.2 to conclude that nearly all elements of U are $\approx (1 - \sqrt{\rho})$-far from V, but what remains an interesting open problem is whether nearly all members of U are maximally far (δ^*-far) from V. An example from [5] show that this need not be the case for RS codes where $q = O(n)$.

C. Correlated agreement

Next, we state the main technical theorem proved in the paper. Consider two vectors $u_0, u_1 \in \mathbb{F}^{\mathcal{D}}$. The result says that if sufficiently many elements in the 1-dimensional affine space $A = \{u_0 + zu_1 : z \in \mathbb{F}\}$ are sufficiently close (δ-close) to the RS code V, then there must be a nontrivial subdomain $\mathcal{D}' \subset \mathcal{D}$ of density $1-\delta$ in $\mathcal{D}$, such that restricting u_0, u_1 to $\mathcal{D}'$ gives a valid RS codeword (evaluated over $\mathcal{D}'$). We refer to the property that such a $\mathcal{D}'$ exists as *correlated agreement*, in the sense that u_0, u_1 and the elements of A do not only have large agreement with the RS code individually, but also share a common large agreement set. The result has two ranges of parameters, as in prior statements in this paper. The proofs for both ranges are given in our full paper [6]: for proximity parameters in the unique decoding regime this is proved in Theorem IV.1, and for proximity parameters in the list decoding regime this is proved in Theorem V.1.

Theorem I.4 (Main Theorem — Correlated agreement over lines)**.** *Let V, q, n, k and ρ be as defined in Theorem I.2. For $u_0, u_1 \in \mathbb{F}_q^{\mathcal{D}}$, if $\delta \in (0, 1 - \sqrt{\rho})$ and*

$$\Pr_{z \in \mathbb{F}_q}\left[\Delta(u_0 + z \cdot u_1, V) \leq \delta\right] > \epsilon,$$

where ϵ is as defined in Theorem I.2, then there exists $\mathcal{D}' \subset \mathcal{D}$ and $v_0, v_1 \in V$ satisfying

- **Density:** *$|\mathcal{D}'|/|\mathcal{D}| \geq 1 - \delta$, and*
- **Agreement:** *v_0 agrees with u_0 and v_1 agrees with u_1 on all of $\mathcal{D}'$.*

Remark I.3 (Sampling from extension fields). One may sample z from a finite extension field $\mathbb{F}_{q'}$ of $\mathbb{F}_q$. In this case, the statement above holds with ϵ_{U} and ϵ_{J} modified by replacing q with q' in the denominators of Eqs. (I.1) and (I.2), respectively. Note that even in this setting, the vectors v_0, v_1 deduced to exist in Theorem I.2 belong to $\mathsf{RS}[\mathbb{F}_q, \mathcal{D}, k]$, not just in $\mathsf{RS}[\mathbb{F}_{q'}, \mathcal{D}, k]$, because v_0, v_1 have high agreement with $u_0, u_1 \in \mathbb{F}_q^{\mathcal{D}}$. The ability to sample from a larger field (and incur smaller error) applies to the other statements of this section but for simplicity we state all of them using a single field $\mathbb{F}_q$ to both define V and sample z from.

Motivated by applications (described later), we generalize the theorem above to two interesting cases: (i) low-degree parameterized curves, and (ii) higher-dimensional affine spaces; details follow.

Correlated agreement over parameterized curves: The first extension of Theorem I.4 extends it from the case of a "line" passing through u_0 and u_1 (the line being $\{u_0 + zu_1 : z \in \mathbb{F}\}$) to a "low-degree curve" with coefficients $u_0, u_1, \ldots, u_l$, as described below. This result is of particular importance for two reasons. First, it leads to derandomized testing of verifiable secret sharing and distributed storage protocols (cf. Section VII-A in [6]). Second, it improves the soundness analysis of the Fast RS IOPP (FRI) protocol [7], which is used in concretely efficient and transparent (public coin) proof systems [8–12]. We discuss this application in Sections III-B and VII-B in in [6].

Let $\mathbf{u} = \{u_0, \ldots, u_l\} \subset \mathbb{F}_q^{\mathcal{D}}$. The *parameterized curve* of degree l that is generated by $\mathbf{u}$ is the following collection of vectors in $\mathbb{F}_q^{\mathcal{D}}$:

$$\mathsf{curve}(\mathbf{u}) := \left\{ u_z := \sum_{i=0}^{l} z^i \cdot u_i \;\middle|\; z \in \mathbb{F}_q \right\}.$$

[2] Note that divergence is not symmetric as can be seen, e.g., when U is a strict subset of V.

Theorem I.5 (Correlated agreement for low-degree parameterized curves)**.** *Let V, q, n, k and ρ be as defined in Theorem I.2. Let $\mathbf{u} = \{u_0, \ldots, u_l\} \subset \mathbb{F}_q^{\mathcal{D}}$. If $\delta \in (0, 1 - \sqrt{\rho})$ and*

$$\Pr_{u \in \mathsf{curve}(\mathbf{u})} [\Delta(u, V) \leq \delta] > l \cdot \epsilon,$$

where ϵ is as defined in Theorem I.2, then there exists $\mathcal{D}' \subset \mathcal{D}$ and $v_0, \ldots, v_l \in V$ satisfying

- **Density:** *$|\mathcal{D}'|/|\mathcal{D}| \geq 1 - \delta$, and*
- **Agreement:** *for all $i \in \{0, \ldots, l\}$, the functions u_i and v_i agree on all of $\mathcal{D}'$.*

Correlated agreement for affine spaces: The second generalization of our Main Theorem I.4, extends it from the 1-dimensional case (affine line) to an affine space of arbitrary dimension. Theorem I.2 follows directly from the following statement. Note that Main Theorem I.4 is actually a case of the following result (for 1-dimensional spaces). However, we stated that special case separately because we prove it first, and from it deduce the more general case (see Section VI-C of [6] for details).

Theorem I.6 (Correlated agreement over affine spaces)**.** *Let V, q, n, k and ρ be as defined in Theorem I.2. For $u_0, u_1, \ldots u_l \in \mathbb{F}_q^{\mathcal{D}}$ let $U = u_0 + \mathrm{span}\{u_1, \ldots, u_l\} \subset \mathbb{F}_q^{\mathcal{D}}$ be an affine subspace. If $\delta \in (0, 1 - \sqrt{\rho})$ and*

$$\Pr_{u \in U} [\Delta(u, V) \leq \delta] > \epsilon,$$

where ϵ is as defined in Theorem I.2, then there exists $\mathcal{D}' \subset \mathcal{D}$ and $v_0, \ldots, v_l \in V$ satisfying

- **Density:** *$|\mathcal{D}'|/|\mathcal{D}| \geq 1 - \delta$, and*
- **Agreement:** *for all $i \in \{0, \ldots, l\}$, the functions u_i and v_i agree on all of $\mathcal{D}'$.*

Furthermore, in the unique decoding regime $\delta \in \left(0, \frac{1-\rho}{2}\right]$, there exists a unique maximal *$\mathcal{D}'$ satisfying the above, with unique v_i.*

Correlated agreement (Theorem I.6) is a sufficient condition for proximity gaps with the same error and proximity parameters (Theorem I.2). We leave as open problems (i) whether correlated agreement is also a necessary condition for a proximity gap. And, if the answer to this question is negative, an intriguing possibility arises: (ii) obtaining proximity gaps for $\delta > 1 - \sqrt{\rho}$ while bypassing the correlated agreement approach we took here.

Organization of the rest of the paper: We provide an overview of the proof of Main Theorem I.4 in Section II, and survey several applications of our results in Section III. The detailed proofs of our theorems, as well as further details on the applications of our results to Verifiable Secret Sharing (VSS) and Fast RS IOPs of Proximity (FRI), are omitted here and can be found in Sections IV–VII and Appendix A of the full paper [6].

II. Proof overview

In this section, we give an overview of our proof strategy of our main result, Theorem I.4.

Recall the setup. $V = \mathsf{RS}[\mathbb{F}_q, \mathcal{D}, k]$ of degree k polynomials evaluated at the points of $\mathcal{D} \subseteq \mathbb{F}_q$, where $|\mathcal{D}| = n$. We have functions $u_0, u_1 : \mathcal{D} \to \mathbb{F}_q$ such that for many $z \in \mathbb{F}_q$, the function $u_0 + zu_1$ is δ-close to V. We want to deduce that u_0 and u_1 are themselves close to V.

The main conceptual idea of our analysis is to work with the function field $\mathbb{K} = \mathbb{F}_q(Z)$ with a formal variable Z, and to study the various received words $u_0 + zu_1$ for the code V simultaneously by considering the *formal received word* $w = u_0 + Zu_1 : \mathcal{D} \to \mathbb{K}$ for the (big field) Reed–Solomon code $\mathsf{RS}[\mathbb{K}, \mathcal{D}, k]$. It turns out that showing that w is close to a (well-structured) codeword of this Reed–Solomon code is sufficient to show that u_0 and u_1 are both close to the original Reed–Solomon code V. With this viewpoint, our proof strategy is to *run a decoding algorithm for Reed–Solomon codes* on this received word $w = u_0 + Zu_1$. Our goal is to analyze the execution of this algorithm to show that it succeeds in finding a nearby Reed–Solomon codeword. We do such an analysis by relating it to the execution of that decoding algorithm on the various received words $u_0 + zu_1$ for the Reed–Solomon code V over the small field $\mathbb{F}_q$.

This strategy is instantiated with two different decoding algorithms for Reed–Solomon codes: the Berlekamp–Welch unique decoding algorithm, and the Guruswami–Sudan list decoding algorithm [13]. Both instantiations give rise to intriguing algebraic questions about polynomials, which we resolve using nontrivial tools from algebraic geometry and the theory of algebraic function fields.

Instantiation with the Berlekamp–Welch Algorithm: Over a field $\mathbb{F}$ and an evaluation domain $\mathcal{D}$, given a received word $r : \mathcal{D} \to \mathbb{F}$, the Berlekamp–Welch decoding algorithm for finding the (unique) nearby polynomial $P(X) \in \mathbb{F}[X]$ close to r works as follows. First it searches for low-degree polynomials $A(X), B(X) \in \mathbb{F}[X]$ such that for each $x \in \mathcal{D}$:

$$A(x)r(x) = B(x).$$

Then the nearby polynomial $P(X)$ is recovered as $B(X)/A(X)$ (which a priori may be a rational function).

In our setting, we first run the Berlekamp–Welch algorithm with received word $w = u_0 + Zu_1 : \mathcal{D} \to \mathbb{K}$ over the big field $\mathbb{K} = \mathbb{F}_q(Z)$ (we will sometimes view this as a function $w(x, z)$ with $w : \mathcal{D} \times \mathbb{F}_q \to \mathbb{F}_q$). Our goal is to find a nearby Reed–Solomon codeword (low-degree polynomial) $P(X) \in \mathbb{K}[X]$ which has the special form $P_0(X) + ZP_1(X)$, where each $P_i(X) \in \mathbb{F}_q[X]$. The first step of the Berlekamp–Welch algorithm gives us $A(X), B(X) \in \mathbb{K}[X] = \mathbb{F}_q(Z)[X]$. Making the Z dependence explicit, we write these as $A(X, Z), B(X, Z)$. This gives us a candidate, namely $A(X, Z)/B(X, Z)$, for being a Reed–Solomon codeword close to w. We will show two things: that $A(X, Z)/B(X, Z)$ is a polynomial in $\mathbb{F}_q(Z)[X]$ (a priori it is only a rational function), and that it is close to w.

The crucial step is to substitute $Z = z$ into $A(X, Z)$ and $B(X, Z)$ for various values of $z \in \mathbb{F}_q$. Letting $w_z = u_0 + zu_1 : \mathcal{D} \to \mathbb{K}$ (the result of substituting $Z = z$ into

w), it turns out that $A(X,z), B(X,z) \in \mathbb{F}_q[X]$ are what we would get if we run the Berlekamp–Welch algorithm (over the small field $\mathbb{F}_q$) on received word w_z. In particular, for many z we get that $B(X,z)$ is divisible by $A(X,z)$ in $\mathbb{F}_q[X]$, and $B(X,z)/A(X,z)$ equals the Reed–Solomon codeword close to w_z. This then allows us to use the Polishchuk–Spielman lemma (a strengthening of the classical Bezout theorem, which deduces divisibility of bivariate polynomials from divisibility of univariate restrictions) to conclude that $B(X,Z)/A(X,Z)$ is in fact a polynomial $P(X,Z)$ in $\mathbb{K}[X]$ of low degree in X.

The final step is to show that $P(X,Z)$, when viewed as a function from $\mathcal{D}$ to $\mathbb{K}$, is close to w, and that that the Z dependence of $P(X,Z)$ is simple (just linear in Z). This is again achieved by considering Z substitutions. We know that for many z, $P(X,z)$ is the degree at most k polynomial $P_z(X)$ that is close to w_z. This means that the X degree of $P(X,z)$ is at most k, and that for many $x \in \mathcal{D}$ and z there is agreement between $P(x,z)$ and $w_z(x) = w(x,z)$. On the other hand, for any $x \in \mathcal{D}$, $w(x,\cdot)$ is a linear function, and $P(x,\cdot)$ is a low degree rational function, and so they cannot agree on too many points unless the low degree rational function $P(x,\cdot)$ formally equals the linear function $w(x,\cdot)$. Therefore this formal equality must happen for many $x \in \mathcal{D}$, i.e., $P(\cdot,Z)$ is close to w. Finally, by simple linear algebra, if $P(x,Z)$ is linear in Z for many x, we conclude that $P(X,Z)$ is linear in Z. This gives us our desired conclusion.

Instantiation with the Guruswami–Sudan Algorithm: Over a field $\mathbb{F}$ and an evaluation domain $\mathcal{D}$, given a received word $r : \mathcal{D} \to \mathbb{F}$, the Sudan and Guruswami–Sudan decoding algorithms for finding all nearby polynomials $P(X) \in \mathbb{F}[X]$ close to r work as follows. First one searches for a low-degree polynomial $Q(X,Y) \in \mathbb{F}[X,Y]$ such that for each $x \in \mathcal{D}$,

$$Q(x, r(x)) = 0.$$

(This is the Sudan algorithm; for the Guruswami–Sudan algorithm we ask that Q vanishes at each $(x, r(x))$ with high multiplicity.) Then every nearby polynomial $P(X)$ turns out to have the property that $Y - P(X)$ divides $Q(X,Y)$ in the bivariate polynomial ring $\mathbb{F}_q[X,Y]$. This means that all such $P(X)$ can be found by factoring $Q(X,Y)$.

In our setting, we run the Guruswami–Sudan algorithm with received word $w = u_0 + Zu_1 : \mathcal{D} \to \mathbb{K}$ over the big field $\mathbb{K} = \mathbb{F}_q(Z)$. Our goal is to find a nearby low-degree polynomial $P(X) \in \mathbb{K}[X]$ which has the special form $P_0(X) + ZP_1(X)$, where each $P_i(X) \in \mathbb{F}_q[X]$. The first step of the Guruswami–Sudan algorithm gives us a bivariate polynomial $Q(X,Y) \in \mathbb{K}[X,Y]$ such that $Q(x, w(x)) = 0$ for each $x \in \mathcal{D}$. Again, we write $Q(X,Y)$ as $Q(X,Y,Z) \in \mathbb{F}_q(Z)[X,Y]$ to make the Z dependence explicit (and we can clear denominators in Z without affecting the vanishing property).

Substituting $Z = z$, we get that $Q(x, w_z(x), z) = 0$ for each $x \in \mathcal{D}$. This means that the polynomial $Q_z(X,Y) \in \mathbb{F}_q[X,Y]$ given by $Q_z(X,Y) = Q(X,Y,z) \in \mathbb{F}_q[X,Y]$ is the bivariate polynomial we would have found while running the Guruswami–Sudan algorithm with received word $w_z : \mathcal{D} \to \mathbb{F}_q$ over the small field $\mathbb{F}_q$. Since for many $z \in \mathbb{F}_q$ we have that w_z is close to some codeword $P_z(X) \in \mathbb{F}_q[X]$ of the Reed–Solomon code V, we get that $Y - P_z(X)$ divides $Q(X,Y,z)$ for many $z \in \mathbb{F}_q$. We would like to deduce from this that over the big field $\mathbb{K}$ there is a low-degree polynomial $P(X) \in \mathbb{K}[X]$ such that $Y - P(X)$ divides $Q(X,Y)$ in $\mathbb{K}[X,Y]$ (and furthermore, this $P(X)$ is close to w and has a simple Z dependence).

This is the most involved (and interesting) part of the analysis. We will factor $Q(X,Y,Z)$ completely into linear factors in Y.

$$Q(X,Y,Z) = C(X,Z)(Y - \gamma_1(X,Z))(Y - \gamma_2(X,Z)) \cdots (Y - \gamma_D(X,Z)). \tag{II.1}$$

This is natural to do, because we are searching for factors that are linear in Y. Then we substitute $Z = z$ into this, and we should see $P_z(X)$ as one of the factors.

However, getting such a factorization for $Q(X,Y,Z)$ may not be possible with polynomials $\gamma_i(X,Z)$, and we have to look (far) beyond. What kind of objects should we think of the γ_i as? After getting the $\gamma_i(X,Z)$, we would like to (a) argue about when $\gamma_i(X,Z)$ is a polynomial in X, and (b) substitute $Z = z$ into it and inspect the resulting object. To enable these, we will express $\gamma_i(X,Z)$ in the ring $R = \overline{\mathbb{K}}[[X]]$, the ring of power series in X, whose coefficients are in the algebraic closure of $\mathbb{K} = \mathbb{F}_q(Z)$. The power series in X representation allows us to see when γ_i is a polynomial in X, and the coefficients being simply algebraic functions in Z (such as $\sqrt{Z^3+Z+1}$) allows us to reason about substitutions $Z = z$. Having decided on R, it is a simple application of Hensel lifting (after possibly a random shift) to show that a factorization as in (II.1) is possible with the $\gamma_i \in R$.

Rather than describe what happens in full generality, we just sketch what would happen in a special case with most of the action. Suppose $\mathbb{F}_q$ is not of characteristic 2, and we have:

$$Q(X,Y,Z) = Y^2 - (Z^3 + Z + 1)(1 - ZX).$$

Going to the ring R, and letting $\alpha = \sqrt{Z^3+Z+1} \in \overline{\mathbb{K}}$, it turns out that $Q(X,Y,Z)$ factors as:

$$\begin{aligned} Q(X,Y,Z) &= \left(Y - \sqrt{Z^3+Z+1}\sqrt{1-ZX}\right) \\ &\quad \cdot \left(Y + \sqrt{Z^3+Z+1}\sqrt{1-ZX}\right) \\ &= \left(Y - \left(\alpha - \frac{\alpha \cdot Z}{2}X - \frac{\alpha \cdot Z^2}{16}X^2 + \dots\right)\right) \\ &\quad \cdot \left(Y + \left(\alpha - \frac{\alpha \cdot Z}{2}X - \frac{\alpha \cdot Z^2}{16}X^2 + \dots\right)\right) \end{aligned}$$

where we used the Taylor series expansion for $\sqrt{1-ZX}$. Now substitute $Z = z$ for $z \in \mathbb{F}_q$. Substituting values into algebraic functions like α is a slightly delicate operation (which square root do you choose? how do you make these choices consistent for different algebraic functions?), but it can be done using basic concepts from the theory of algebraic function fields. Another tool that we need from the theory of algebraic function fields is an analogue of the degree of a polynomial, to measure complexity of algebraic functions and

bound the number of their zeroes. In this sketch we avoid going into any such details.

Doing the substitution gives us:

$$Q_z(X,Y) = Q(X,Y,z)$$
$$= \left(Y - \left(\alpha(z) - \frac{\alpha(z)\cdot z}{2}X + \cdots + c_i\alpha(z)z^iX^i + \cdots\right)\right)$$
$$\times \left(Y + \left(\alpha(z) - \frac{\alpha(z)\cdot z}{2}X + \cdots + c_i\alpha(z)z^iX^i + \cdots\right)\right).$$

By properties of the Guruswami–Sudan decoding algorithm, we know for all "good" $z \in \mathbb{F}_q$ where w_z is close to some low degree polynomial P_z, we must have that $Y - P_z(X)$ divides $Q_z(X,Y)$. Given the factorization above, one of the following must occur:

1) $P_z(X) = \left(\alpha(z) - \frac{\alpha(z)\cdot z}{2}X - \frac{\alpha(z)\cdot z^2}{16}X^2 + \cdots\right)$,
2) $P_z(X) = -\left(\alpha(z) - \frac{\alpha(z)\cdot z}{2}X - \frac{\alpha(z)\cdot z^2}{16}X^2 + \cdots\right)$.

Whichever power series ends up equaling $P_z(X)$, the coefficient of X^{k+1} in that power series must equal 0. In our particular example, we deduce that $c_{k+1}\alpha(z)z^{k+1} = 0$ for some constant c_{k+1}. Assuming c_{k+1} is nonzero in $\mathbb{F}_q$, we get that $\alpha(z)z^{k+1} = 0$ for every good z. Finally we use the fact that a *nonzero* algebraic functions of low "degree" like $\alpha(Z)Z^{k+1} = \sqrt{Z^3+Z+1}\cdot Z^{k+1}$ cannot vanish at too many points z. This means that there cannot be too many good z, contradicting our hypothesis. We conclude that $Q(X,Y,Z)$ cannot equal $Y^2 - (Z^3+Z+1)(1-ZX)$!

A very similar argument derives a contradiction unless $Q(X,Y,Z)$ has a factor of the form $Y - P(X)$ for some $P(X) \in \overline{\mathbb{K}}[X]$ of degree at most k. The only twist is that we may have to focus on the coefficient of some different power X^{k+c} in the power series than the coefficient of X^{k+1} (in case the coefficient of X^{k+1} in the power series is identically 0). To make this argument work, we need to estimate the "degree" of the algebraic functions that appear as coefficients in these power series. This involves a careful study of the Hensel lifting process, especially its effect on the complexity of its coefficients.

The final part of the argument, showing that some $Y - P(X)$ factor of $Q(X,Y,Z)$ is such that $P(X)$ has high agreement with w and all the coefficients of $P(X)$ are linear polynomials in Z, is similar to what happened in the unique decoding case. Instead of using the fact that a low degree rational function and a linear function cannot have high agreement unless they are equal, we use the fact that a low degree algebraic function and a linear function cannot have high agreement unless they are equal. This completes our sketch of the proof.

Technical issues: When we actually implement the argument, there are some technical changes we make (both for simplicity and for optimizing parameters). First, we do not do the proof by contradiction, but instead show how to find the factor of the form $Y - P(X)$. Next, instead of directly doing Hensel lifting with Q, we factor Q into irreducible factors over $\mathbb{F}_q[X,Y,Z]$ and focus on a single irreducible factor that is "responsible" for many of the P_z. This helps in that we do not need to factor arbitrarily messy Q's completely into linear factors, but only those which have the property that $Q(X,Y,z)$ has a linear factor of the form $Y - P_z(X)$. Finally, instead of arguing over the algebraic closure $\overline{\mathbb{K}}$, we go to a small algebraic extension $\mathbb{L}$ of $\mathbb{K}$ which is rich enough to express all the coefficients of the relevant power series. These changes lead to some simplifications and quantitative improvements in our proofs.

Relationship with the Arora-Sudan low degree test [14]: A beautiful and fundamental paper of Arora and Sudan [14], analyzed the "line vs. line" low degree test for multivariate polynomials in the high error regime. The heart of their paper is a theorem that says that if a function $f : \mathbb{F}_q^2 \to \mathbb{F}_q$ is such that for most lines L given by $Y = aX+b$ in $\mathbb{F}_q^2$ the univariate function obtained from restricting f to L (denoted $f|_L$) is close to a low degree univariate polynomial, then f is itself close to a low degree bivariate polynomial. This is closely related to our theorem which deduces a similar conclusion about a received word $w : \mathcal{D} \times \mathbb{F}_q \to \mathbb{F}_q$, also based on restrictions to lines. Our proof is heavily influenced by the proof in [14] (which in turn builds on fundamental results on polynomial factorization and the Hilbert irreducibility theorem by Kaltofen [15, 16]). There is one crucial difference in our proof. Our approach is spearheaded by the idea of running all arguments over the big field $\mathbb{K} = \mathbb{F}_q(Z)$ (as opposed to treating Z as another variable over $\mathbb{F}_q$ just like X and Y, as is done in [14]). This difference affects our proofs in a tangible sense: our proofs are based on bivariate interpolation over the big field $\mathbb{K}$ rather than trivariate interpolation over the small field $\mathbb{F}_q$. Inside the analysis, our proofs use power series in one variable over function fields rather than power series in two variables over finite extensions of $\mathbb{F}_q$. This leads to more involved algebraic tools being needed for our proof (most seriously the use of algebraic function fields), but also yields three improvements. First, our result is about axis parallel restrictions $Z = z$ (for $z \in \mathbb{F}_q$) instead of more general linear restrictions $Z = aX + b$ (for $a, b \in \mathbb{F}_q$). This simpler form of restriction is important for our applications. Second, our result deduces structure all the way up to the Johnson radius, while the result in Arora-Sudan is to a smaller radius (polynomially worse in terms of agreement parameter). Third, our result works over fields that are quadratic in the degree of the polynomials involved whereas the Arora-Sudan result requires fields that are quartic (at least) in the degree.

III. Applications

Our proximity gap results are motivated by the following general setting. There are several purported codewords $\mathbf{u} = \{u_1, \ldots, u_l\} \subset \mathbb{F}_q^n$ of an RS code V. A verifier would like to be assured that they are all close to V. This is done by taking a random linear combination of the u_i and checking its proximity to V. The analysis of this simple test, which arises naturally in a variety of application scenarios, turns out to be surprisingly challenging. Indeed, it is closely related to the proximity gap problem we study in this work.

This batch verification problem arises in two kinds of

settings: a *distributed* setting, where entries of $\mathbf{u}$ are split between multiple servers and may not be known to any single entity, and a *centralized* setting, where $\mathbf{u}$ is entirely known to a prover and can be queried by a verifier. We briefly explain the role of proximity gaps in these two types of applications.

In the distributed setting, the coefficients of the random linear combination is either generated by a single verifier or jointly via a distributed coin tossing protocol. Each server then responds with its own share of the output. Verification succeeds if the joint output is a codeword, or alternatively it is close to the code. Examples for applications in the distributed setting include Verifiable Secret Sharing (see Section VII-A in [6]) and secure multiparty computation protocols, such as those from [17, 18]. These applications typically rely on unique decoding and can thus benefit from our near-optimal analysis for this regime. In this type of applications, the main challenge is protecting against an *adaptive* adversary who may choose which servers to corrupt after seeing the coefficients of the random linear combination. To defeat such an adversary, we need to ensure that if at least one of the u_i is far from the code, then (with high probability) so is their random linear combination. If this were not the case, an adaptive adversary could eliminate all inconsistencies by corrupting a small number of servers. Proximity gaps rule out this kind of attack.

In the centralized setting, $\mathbf{u}$ is known to a prover and can be queried by the verifier. A typical realization is using a tree-based succinct cryptographic commitment that binds the prover to a uniquely defined $\mathbf{u}$ and yet enables efficient local opening of symbols queried by the verifier. In this case, the verifier challenges the prover by choosing the coefficients r_i of the random linear combination. The prover, who claims that all u_i are codewords in V, must respond with a valid codeword $u \in V$. The verifier checks that u agrees with $u' = r_1 u_1 + \ldots + r_l u_l$ by querying a random entry of u and the corresponding entries of $\mathbf{u}$ and checking their consistency. (To amplify soundness, the verifier can query several random entries of u.) Here too, proximity gaps guarantee that if one of the u_i is far from V, then (with high probability) so is u'. This ensures that the verifier detects an inconsistency with high probability. Examples for applications in the centralized setting include communication-efficient proof systems [1, 2, 7], homomorphic commitment schemes [19], and secure two-party computation protocols [20, 21]. See more in Section III-B below.

An appealing feature of the simple "random linear combination" test is that it can be implemented with low communication and computation costs. In particular, in the distributed setting it suffices for each server to send a single field element to the verifier. In both settings, communicating the l random coefficients is typically not a bottleneck. This random challenge can be made shorter either by using a cryptographic pseudorandom generator or unconditionally by using simple derandomization techniques. In particular, one can generate all coefficients as distinct powers of a single random field elements and appeal to the parameterized curves variant of the proximity gap theorem (Theorem I.5).

Our new proximity gaps imply a tighter analysis of applications that test proximity to RS codes. Generally speaking, in the distributed setting the improved proximity gap bounds imply a constant-factor improvement in the *resilience threshold*, namely the number of corrupted parties that can be tolerated. In the centralized setting, one typically gets constant-factor savings in the overall communication and computation costs. While often ignored in theory-oriented research, the latter kind of improvements can be very significant in the context of practical succinct proof systems.

Why RS codes?: Reed–Solomon codes are commonly used in distributed storage, efficient proof systems, and cryptographic protocols. They are useful because of their MDS property, near-linear encoding, and efficient (list)-decoding algorithms. A more qualitative feature of RS codes, which is commonly used in proof systems and cryptography, is the following *multiplication-friendliness* property: when $n = |\mathcal{D}| > 2k$, the pointwise products of codewords in $V = \mathsf{RS}[\mathbb{F}_q, \mathcal{D}, k]$ span a linear code that has nontrivial minimal distance, namely the code $\mathsf{RS}[\mathbb{F}_q, \mathcal{D}, 2k]$.

We now give more concrete examples of applying proximity gaps to analyze batch-verification tasks that arise in different application scenarios.

A. Distributed storage and cryptography

Distributed storage.: Consider a scenario in which l users encode their inputs using a length-n RS code $V = \mathsf{RS}[\mathbb{F}_q, \mathcal{D}, k]$, where server i stores the i-th symbol of each of the l codewords. Suppose that some of the nl symbols were corrupted, say by a transient malware that overwrites a subset of the symbols before being discovered and eliminated. A verifier would like to get a quick estimate of the amount of damage caused by the malware. A natural idea is to have the servers communicate a random linear combination u' of the potentially corrupted codewords u_j. Using the basic proximity gap result (Theorem I.2), if at least one of u_j is δ-far from the code (for $\delta \leq \frac{1-\rho}{2}$ or $\delta < 1 - \sqrt{\rho}$), then u' is δ-far from the code except with small failure probability (at most n/q for $\delta \leq \frac{1-\rho}{2}$). Thus, for sufficiently large $\mathbb{F}_q$, the distance of u' from V provides a reliable upper bound on the maximal relative distance of a vector u_i from V within the proximity bounds of Theorem I.2. This estimate is not too pessimistic in the sense that if only a μ-fraction of the servers were affected, the upper bound obtained by the test is no bigger than μ.

Distributed proximity test for Interleaved RS codes.: The above analysis leaves something to be desired: if u' is within (sufficiently small) distance δ from V, the verifier is only assured that each u_j is *individually* within distance δ from V. In some applications, we would like to get the stronger guarantee that in such an event there is a δ-fraction of the coordinates whose removal makes *all* u_j consistent with V. Moreover, we would like to identify this set of coordinates, which is uniquely defined in the unique decoding regime. This is useful even in the above distributed storage scenario, but will be even more useful for the applications we discuss next.

The stronger feature can be conveniently captured using the notion of an *Interleaved Reed–Solomon* (IRS) code. In an $\mathrm{IRS}(V, l)$ code, the codewords are $l \times n$ matrices in which each row is a codeword in V. The symbols of such a codeword are the matrix columns. Namely, a codeword consists of n symbols in $\mathbb{F}_q^\ell$. The following theorem, which follows easily from Theorem I.6, phrases the stronger guarantee provided by the refined analysis in terms of proximity testing for IRS codes. We state it for the unique decoding regime, which suffices (and is sometimes required) for the applications we discuss next. For u within the unique decoding radius of V, we denote by $\Gamma(u, V)$ the set of coordinates on which u disagrees with the closest codeword from V.

Theorem III.1 (Distributed proximity test for Interleaved RS codes)**.** *Let $V = \mathsf{RS}[\mathbb{F}_q, \mathcal{D}, k]$ for $|\mathcal{D}| = n$ and $\mathbf{V} = \mathrm{IRS}(V, l)$. We view codewords in V and $\mathbf{V}$ as vectors in $\mathbb{F}_q^n$ and matrices in $\mathbb{F}_q^{l\times n}$ respectively. Let $\rho = \frac{k+1}{n}$ and $\delta \leq \frac{1-\rho}{2}$. Let $\mathbf{u} \in \mathbb{F}_q^{l\times n}$ and let $u' = r^T\mathbf{u}$ where $r \in_R \mathbb{F}_q^l$.*

- *Completeness: If $\Delta(\mathbf{u}, \mathbf{V}) \leq \delta$ then $\Pr[\Delta(u', V) \leq \delta] = 1$ and moreover $\Pr[\Gamma(u', V) \neq \Gamma(\mathbf{u}, \mathbf{V})] \leq n/q$.*
- *Soundness: If $\Delta(\mathbf{u}, \mathbf{V}) > \delta$ then $\Pr[\Delta(u', V) \leq \delta] \leq n/q$.*

We refer to the above test as distributed because it can be implemented with low communication complexity in the distributed setting, where each server holds a different column of $\mathbf{u}$. One can similarly obtain an affine version with the same guarantee, where $\mathbf{u}$ has an additional row u_0 that is always added to u' (i.e., with coefficient $r_0 = 1$), and the code $\mathbf{V}$ is extended to by $\mathrm{IRS}(V, l+1)$. This affine version is useful for zero-knowledge variants of the test, where a single random $u_0 \in V$ is used for blinding $u_1, \ldots, u_l$. This is used in the cryptographic applications we discuss next.

General cryptographic protocols.: Theorem III.1 serves as a useful tool for analyzing cryptographic protocols in the presence of an *adaptive* adversary who can dynamically choose the set of corrupted parties. For instance, it shows that secure multiparty computation protocols from [17, 18] are adaptively secure when the adversary can corrupt roughly $1/3$ of the parties. The best previous proximity gaps from [3–5] could only get up to 1/4 corruption threshold in the same setting. Adaptive security, in turn, is crucial for the general transformation from [20, 22] of these honest-majority protocols to two-party protocols and protocols for dishonest majority. Indeed, this is the context that gave rise to proximity gap in the analysis of the Ligero zero-knowledge proof system [2], which applies a variant of the transformation from [22] to a variant of the protocol from [17]. We give a detailed exposition of the application of proximity gaps to *verifiable secret sharing*, which serves as a basis for the above results on secure multiparty computation, in Section VII of [6].

B. Soundness of the Fast RS IOPP (FRI) protocol

FRI is an Interactive Oracle Proof of Proximity (IOP of Proximity, or IOPP) as defined in [23, 24]. An IOP is an interactive protocol in which the verifier has oracle access to messages sent by the prover, so she need not read and store those messages but may query random entries of them. FRI is one of a family of protocols for testing proximity to the RS code (an "RS proximity testing" (RPT) protocol). Its purpose is to check whether a received word $f : \mathcal{D} \to \mathbb{F}_q$ belongs to a pre-specified RS code $V := \mathsf{RS}[\mathbb{F}_q, \mathcal{D}, k]$ and to reject words that are δ-far from the code with high probability and low query complexity. Due to its efficiency it is used as a building block in several recent succinct zero knowledge protocols including scalable and transparent (public coins) arguments of knowledge (STARKs) [8, 9], Aurora [10] and its succinct version [11], and Fractal [12], to name a few. These systems have been shown by Chiesa et al. to be sound in the quantum random oracle model (hence are "plausibly post-quantum secure") [25]. Therefore, understanding the concrete soundness error of FRI, denoted ϵ_{FRI}, is of significant practical value, in addition to being a theoretically interesting question.

Consider the case of f that is maximally far from V, i.e., $\Delta(f, V) \approx 1 - \rho$ (this holds, e.g., for random f, with high probability). Fix a target soundness error bound $2^{-\lambda}$ (in concrete settings, λ is the "security parameter", often fixed to $\lambda = 128$). The communication complexity of FRI is dominated by the number t of iterations of the QUERY phase, so the question at hand is:

> *How many iterations t of the QUERY phase are needed to obtain $\epsilon_{\mathsf{FRI}} \leq 2^{-\lambda}$?*

The initial analysis of [7] required a number t that is quite large, and does not tend to 0 even for tiny rates ρ. This was improved by [4] to $t \approx 4\lambda \log \frac{1}{\rho}$, and then by [5] to $t \approx 3\lambda \log \frac{1}{\rho}$. Sadly, that paper also showed that this bound is tight, at least when the field size q equals the code's blocklength n. Our main result regarding FRI (Theorem VII.2 in [6]) shows that for $q \gg n^2$ we can reduce the number t of iterations by 33% to $t \approx 2\lambda \log \frac{1}{\rho}$, which leads to communication complexity that is at least 33% shorter, for provable soundness settings. The actual savings in the provable soundness case are likely larger, due to smaller field size and the ability of the improved analysis to operate with any sequence of oracle sizes in the FRI COMMIT phase (as discussed after the statement of Theorem VII.2).

Acknowledgments

Yuval Ishai was supported by ERC Project NTSC (742754), NSF-BSF grant 2015782, BSF grant 2018393, and a grant from the Ministry of Science and Technology, Israel and Department of Science and Technology, Government of India. Swastik Kopparty was supported in part by NSF grants CCF-1540634 and CCF-1814409 and BSF grant 2014359. Shubhangi Saraf was supported in part by NSF grants CCF-1540634 and CCF-1909683, BSF grant 2014359, a Sloan research fellowship and the Simons Collaboration on Algorithms and Geometry. Work done in part while Yuval Ishai and Swastik Kopparty were participating in the Simons Institute program on Proofs, Consensus, and Decentralizing Society.

References

[1] G. N. Rothblum, S. Vadhan, and A. Wigderson, "Interactive proofs of proximity: delegating computation in sublinear time," in *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. ACM, 2013, pp. 793–802.

[2] S. Ames, C. Hazay, Y. Ishai, and M. Venkitasubramaniam, "Ligero: Lightweight sublinear arguments without a trusted setup," in *Proceedings of the 24th ACM Conference on Computer and Communications Security*, October 2017.

[3] R. M. Roth and G. Zémor, "Personal communication," 2018.

[4] E. Ben-Sasson, S. Kopparty, and S. Saraf, "Worst-case to average case reductions for the distance to a code," in *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, 2018, pp. 24:1–24:23. [Online]. Available: https://doi.org/10.4230/LIPIcs.CCC.2018.24

[5] E. Ben-Sasson, L. Goldberg, S. Kopparty, and S. Saraf, "DEEP-FRI: sampling outside the box improves soundness," in *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, ser. LIPIcs, T. Vidick, Ed., vol. 151. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 5:1–5:32. [Online]. Available: https://doi.org/10.4230/LIPIcs.ITCS.2020.5

[6] E. Ben-Sasson, D. Carmon, Y. Ishai, S. Kopparty, and S. Saraf, "Proximity gaps for reed-solomon codes," *Electron. Colloquium Comput. Complex.*, vol. 27, p. 83, 2020. [Online]. Available: https://eccc.weizmann.ac.il/report/2020/083

[7] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Fast Reed-Solomon Interactive Oracle Proofs of Proximity," in *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP)*, 2018. [Online]. Available: https://eccc.weizmann.ac.il/report/2017/134

[8] ——, "Scalable, transparent, and post-quantum secure computational integrity," *IACR Cryptology ePrint Archive*, vol. 2018, p. 46, 2018. [Online]. Available: http://eprint.iacr.org/2018/046

[9] ——, "Scalable zero knowledge with no trusted setup," in *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III*, ser. Lecture Notes in Computer Science, A. Boldyreva and D. Micciancio, Eds., vol. 11694. Springer, 2019, pp. 701–732. [Online]. Available: https://doi.org/10.1007/978-3-030-26954-8_23

[10] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward, "Aurora: Transparent succinct arguments for R1CS," *IACR Cryptology ePrint Archive*, vol. 2018, p. 828, 2018. [Online]. Available: https://eprint.iacr.org/2018/828

[11] E. Ben-Sasson, A. Chiesa, L. Goldberg, T. Gur, M. Riabzev, and N. Spooner, "Linear-size constant-query iops for delegating computation," in *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II*, ser. Lecture Notes in Computer Science, D. Hofheinz and A. Rosen, Eds., vol. 11892. Springer, 2019, pp. 494–521. [Online]. Available: https://doi.org/10.1007/978-3-030-36033-7_19

[12] A. Chiesa, D. Ojha, and N. Spooner, "Fractal: Post-quantum and transparent recursive proofs from holography," *IACR Cryptology ePrint Archive*, vol. 2019, p. 1076, 2019, to appear in Eurocrypt 2020. [Online]. Available: https://eprint.iacr.org/2019/1076

[13] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon and algebraic-geometry codes," *IEEE Trans. Information Theory*, vol. 45, no. 6, pp. 1757–1767, 1999. [Online]. Available: http://dx.doi.org/10.1109/18.782097

[14] S. Arora and M. Sudan, "Improved low-degree testing and its applications," *Combinatorica*, vol. 23, no. 3, pp. 365–426, 2003, preliminary version appeared in STOC '97.

[15] E. Kaltofen, "Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization," *SIAM Journal on Computing*, vol. 14, no. 2, pp. 469–489, May 1985.

[16] ——, "Effective noether irreducibility forms and applications," *J. Comput. Syst. Sci*, vol. 50, no. 2, pp. 274–295, 1995.

[17] I. Damgård and Y. Ishai, "Scalable secure multiparty computation," in *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, ser. Lecture Notes in Computer Science, C. Dwork, Ed., vol. 4117. Springer, 2006, pp. 501–520. [Online]. Available: https://doi.org/10.1007/11818175_30

[18] Y. Ishai, M. Prabhakaran, and A. Sahai, "Secure arithmetic computation with no honest majority," in *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, ser. Lecture Notes in Computer Science, O. Reingold, Ed., vol. 5444. Springer, 2009, pp. 294–314. [Online]. Available: https://doi.org/10.1007/978-3-642-00457-5_18

[19] I. Cascudo, I. Damgård, B. David, N. Döttling, and J. B. Nielsen, "Rate-1, linear time and additively homomorphic UC commitments," in *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, 2016, pp. 179–207. [Online]. Available: https://doi.org/10.1007/978-3-662-53015-3_7

[20] Y. Ishai, M. Prabhakaran, and A. Sahai, "Founding cryptography on oblivious transfer - efficiently," in *Advances in Cryptology - CRYPTO 2008, 28th Annual*

International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings, ser. Lecture Notes in Computer Science, D. A. Wagner, Ed., vol. 5157. Springer, 2008, pp. 572–591. [Online]. Available: https://doi.org/10.1007/978-3-540-85174-5_32

[21] C. Hazay, Y. Ishai, A. Marcedone, and M. Venkitasubramaniam, "Leviosa: Lightweight secure arithmetic computation," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, L. Cavallaro, J. Kinder, X. Wang, and J. Katz, Eds. ACM, 2019, pp. 327–344. [Online]. Available: https://doi.org/10.1145/3319535.3354258

[22] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Zero-knowledge proofs from secure multiparty computation," *SIAM J. Comput.*, vol. 39, no. 3, pp. 1121–1152, 2009. [Online]. Available: https://doi.org/10.1137/080725398

[23] O. Reingold, G. N. Rothblum, and R. D. Rothblum, "Constant-round interactive proofs for delegating computation," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, 2016, pp. 49–62. [Online]. Available: http://doi.acm.org/10.1145/2897518.2897652

[24] E. Ben-Sasson, A. Chiesa, and N. Spooner, "Interactive oracle proofs," in *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, 2016, pp. 31–60. [Online]. Available: https://doi.org/10.1007/978-3-662-53644-5_2

[25] A. Chiesa, P. Manohar, and N. Spooner, "Succinct arguments in the quantum random oracle model," in *Theory of Cryptography*, D. Hofheinz and A. Rosen, Eds. Cham: Springer International Publishing, 2019, pp. 1–29.

[26] A. Polishchuk and D. A. Spielman, "Nearly-linear size holographic proofs," in *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, ser. STOC '94, 1994, pp. 194–203.

[27] D. A. Spielman, "Computationally efficient error-correcting codes and holographic proofs," Ph.D. dissertation, MIT, 1995.

[28] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract)," in *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, 1985, pp. 383–395. [Online]. Available: https://doi.org/10.1109/SFCS.1985.64

[29] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract)," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, ser. STOC '88, 1988, pp. 1–10.

[30] D. Chaum, C. Crépeau, and I. Damgård, "Multiparty unconditionally secure protocols (extended abstract)," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, 1988, pp. 11–19. [Online]. Available: https://doi.org/10.1145/62212.62214

[31] T. Rabin and M. Ben-Or, "Verifiable secret sharing and multiparty protocols with honest majority (extended abstract)," in *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, 1989, pp. 73–85. [Online]. Available: https://doi.org/10.1145/73007.73014

[32] M. K. Franklin and M. Yung, "Communication complexity of secure computation (extended abstract)," in *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, S. R. Kosaraju, M. Fellows, A. Wigderson, and J. A. Ellis, Eds. ACM, 1992, pp. 699–710. [Online]. Available: https://doi.org/10.1145/129712.129780

[33] R. Canetti, I. Damgård, S. Dziembowski, Y. Ishai, and T. Malkin, "Adaptive versus non-adaptive security of multi-party protocols," *J. Cryptology*, vol. 17, no. 3, pp. 153–207, 2004. [Online]. Available: https://doi.org/10.1007/s00145-004-0135-x