Repairing Reed-Solomon Codes over Prime Fields via Exponential Sums

Roni Con*, Noah Shutty†, Itzhak Tamo‡, and Mary Wootters§
*Blavatnik School of Computer Science, Tel Aviv University, roni.con93@gmail.com
†Stanford Institute for Theoretical Physics, Stanford University, noaj@alumni.stanford.edu
‡Department of Electrical Engineering-Systems, Tel Aviv University, zactamo@gmail.com
§Department of Computer Science and Electrical Engineering, Stanford University, marykw@stanford.edu

2023 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-7554-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISIT54713.2023.10206937

***Abstract*—This paper presents several repair schemes for low-rate Reed Solomon (RS) codes over prime fields that can repair any node by downloading a *constant* number of bits from each surviving node. The resulting total bandwidth is higher than the bandwidth incurred during the trivial repair; however, this is still interesting in the context of *leakage-resilient secret sharing*. In that language, our results give attacks that show that k-out-of-n Shamir's Secret Sharing over prime fields for small k is *not* leakage resilient, even if the parties only leak a *constant* number of bits. To the best of our knowledge, these are the first such attacks.**

As another application, we provide decoding schemes for RS codes over prime fields, where the entire RS codeword is recovered by transmitting a constant number of bits from each node.

Our results follow from a novel connection between exponential sums and repair of RS codes. In particular, we show that non-trivial bounds on certain exponential sums imply the existence of efficient nonlinear repair schemes for RS codes over prime fields.

I. Introduction

Reed-Solomon (RS) codes are a widely-used family of codes in both theory and practice. Among their many applications, RS codes are used in distributed storage systems (e.g., Facebook, IBM, Google, etc. see Table 1 in [1]) In such systems, a large file is encoded using an erasure-correcting code and then distributed over many nodes. When a node fails, we would like to be able to set up a replacement node efficiently using information from the remaining nodes. In our work, we focus on the *repair bandwidth*—that is, the total amount of information downloaded—as our metric of efficiency. The problem of recovering the failed node with low repair bandwidth was first considered in the seminal paper of Dimakis et al. [2] and has since been the topic of much research.

We begin by defining Reed-Solomon codes.

The work of Itzhak Tamo and Roni Con was partially supported by the European Research Council (ERC grant number 852953) and by the Israel Science Foundation (ISF grant number 1030/15). Noah Shutty was supported in part by NSF DGE-1656518. Mary Wootters was supported in part by NSF grants CCF-2133154 and CCF-1844628.

Definition 1. *Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be distinct points of the finite field $\mathbb{F}_q$ of order q. For $k < n$ the $[n,k]_q$* RS code *defined by the evaluation set $\{\alpha_1, \ldots, \alpha_n\}$ is the set of codewords*

$$\{(f(\alpha_1), \ldots, f(\alpha_n)) \mid f \in \mathbb{F}_q[x], \deg f < k\}\ .$$

When $n = q$, the resulting code is called a full-length RS code.

The repair problem of RS codes can be seen as a twist on the standard polynomial interpolation problem: Repairing, say, the i'th node is the same as recovering an evaluation $f(\alpha_i)$ using as little information as possible from the evaluations $f(\alpha_j)$ for $j \neq i$. Formally, a *repair scheme* for an $[n,k]_q$ RS code with evaluation points $\alpha_1, \ldots, \alpha_n$ consists of functions $\tau_j : \mathbb{F}_q \times [n] \to \{0,1\}^m$ for each $j \in [n]$; and a repair function $G : \{0,1\}^{m \times (n-1)} \times [n] \to \mathbb{F}_q$, for some parameter m. For any i (the index of a failed node), and any polynomial f of degree less than k (representing the stored data), each surviving node $j \neq i$ sends a message $\tau_j(f(\alpha_j), i)$. Then the reconstruction algorithm G takes in the messages $\tau_j(f(\alpha_j), i)$, as well as i, and outputs the missing information $f(\alpha_i)$. Our goal is to minimize m, the number of bits sent by each node.[1]

Via standard polynomial interpolation, it is clear that any k values of $f(\alpha_j)$ suffice to recover the polynomial f, and in particular, to recover $f(\alpha_i)$. This requires $k \log_2(q)$ bits of information; we refer to this as *trivial repair*. However, as was shown in a line of work including [3]–[6], it is possible to do better! That is, it is possible to recover $f(\alpha_i)$ using strictly less than $k \log(q)$ bits from the other nodes!

A. Repairing Reed-Solomon Codes over Prime Fields

Most of the RS repair schemes mentioned above (in fact, all but that in [6]) require that the underlying field be an extension field. In contrast, in our work, we focus on *prime fields*. Our main motivation comes from applications to *secret sharing*.

In secret sharing, RS codes are analogous to *Shamir's secret sharing scheme* (Shamir's SSS). Informally speaking (see Section I-C for more details), repair schemes for RS

[1]We note that traditionally the definition of a repair scheme allows for only a subset of the nodes to be contacted; and the nodes could all potentially send different amounts of information. In this work, we focus on the case where all surviving nodes send the same amount of information, so we specialize our definition to that case for simplicity.

	RS Code	Bandwidth per node (in bits)	Remarks
Theorem 3.3 in [6]	$[n,2]_p$	$\frac{\log(p)}{n-2}+O_n(1)$	Non-explicit construction.
Theorem 4.3 in [6]	$[n,k]_p$	$\frac{\log(p)}{n-k}+O_n(1)$	Repair only possible for a particular failed node (rather than any failed node)[2] using all the remaining nodes.
Theorem 5	$[p^\delta,3]_p$	3	$\frac{(\ln\ln p)^{\frac{1}{3}}}{(\ln p)^{\frac{2}{3}}} \le \delta \le \frac{1}{2}$
Theorem 6	$[p,\frac{\sqrt{p}}{2}]_p$	3	

TABLE I: Our result compared to [6]. If not stated in the Remarks column, all of the results presented are explicit constructions, and all the results give repair schemes that repair any single failed node using all the remaining nodes. For all results, p is a sufficiently large prime. For the results in [6], n and k are assumed to be constants relative to p.

codes provide attacks on Shamir's SSSs when the parties may each leak a small amount of adversarially selected information about their shares. That is, a repair scheme for RS codes shows that an instance of Shamir's SSS is not *leakage-resilient*.

Since the schemes of [3]–[5] and others require extension fields, a natural hope for constructing Shamir's SSSs that are leakage-resilient is to work over prime fields, and indeed several works [7]–[10] have shown that, when the dimension k of the code is large, $\Omega(n)$, Shamir's SSS over prime fields is leakage resilient. In particular, their results imply that for any constant m, there is some constant $\alpha \in (0,1)$ so that any $[n,k]_p$ RS code over a prime field with $k \ge \alpha n$ does *not* admit a repair scheme that downloads m bits from each surviving node (see Theorem 1).

Recently, [6] showed the existence of asymptotically optimal repair schemes for $[n,2]_p$ RS codes of dimension $k=2$ over sufficiently large prime fields. In their work, each party downloads a non-constant number of bits, specifically, $m=\frac{1}{n-2}\log(p)+O_n(1)$ bits (see Table I for their results for larger k). Our results continue the line of work of [6]. We make progress by giving schemes where each node transmits a *constant* number $m=O(1)$ bits. Our results have applications both to (attacks on) leakage-resilient secret sharing and to distributed storage.

B. Our Results

In this paper, we present repair schemes for RS codes over prime fields, in which every node transmits a *constant* number of bits. Specifically,

1) For $\exp\big((\ln p)^{2/3}(\ln\ln p)^{1/3}\big) \le n \le \sqrt{p}$, we present an $[n,3]_p$ RS code where any node can be repaired by downloading *three* bits from each of the remaining nodes.
2) For $k=\sqrt{p}/2$, we show that any node in a full-length, i.e., $[p,k]_p$, RS code can be repaired by downloading *three* bits from all the remaining $p-1$ nodes.

[2]This is achieved by puncturing the code in [6, Theorem 4.3].

We emphasize that the total bandwidth of the two repair schemes is much bigger than $k\cdot\log(p)$, the bandwidth of the trivial repair; thus our schemes are not competitive if one only cares about total bandwidth.

However, our schemes are useful in settings where each node can only transmit very few bits; in such a setting, the trivial repair is not an option. This could be the case in distributed storage models (e.g., in a model where each link can only transmit a few bits). It is also the case in the model of leakage-resilient secret sharing.

In particular, our results give attacks on Shamir's SSSs over prime fields, where each leaking party need only send a constant number of bits. To the best of our knowledge, these are the first such attacks with a constant number of bits per leaking party.

We also present *decoding schemes* of RS codes over prime fields that download only a constant number of bits from each node. That is, our schemes can recover the entire polynomial, and not just a single missing evaluation point. Specifically, our results are:

1) For any positive integers $T < k < n$ and a prime $p > n$ such that $k < O(n\log(T)/\log(p))$, there are n functions $\tau_1,\dots,\tau_n : \mathbb{F}_p \to [T]$ such that any polynomial f of degree at most $k-1$ can be recovered from the information $(\tau_1(f(\alpha_1)),\dots,\tau(f(\alpha_n)))$ where $\alpha_1,\dots,\alpha_n$ are any n distinct points in $\mathbb{F}_p$. We note that this result is similar to the result obtained in [8, Theorem 2] (see discussion in Section IV).
2) We construct an explicit decoding scheme for $[p-1,\sqrt{p}/2]_p$ RS code defined with the evaluation points $\mathbb{F}_p^*$. Namely, we show that any codeword in this code can be recovered by downloading *three* bits from each node.

Remark 1. *We have a repair scheme and a decoding scheme where both are applicable for $k=O(\sqrt{p})$ and require the same total bandwidth. The primary difference between the two schemes is that the first scheme (the repair scheme) only reveals the secret, while the second scheme (the decoding scheme) reveals the entire polynomial. One may wonder why we present the first scheme. We choose to present the first scheme because we hope that the techniques in it may be a first step towards a scheme that can recover a single failed node using at most $O(1)$ bits per party while still achieving significantly less bandwidth total than trivial repair. We leave this as an important open direction.*

C. Related Work

a) Low-bandwidth repair for distributed storage: As mentioned above, the low-bandwidth repair problem was introduced in [2], and since then there have been many code constructions and repair schemes aiming at optimal repair, for example [11]–[19]. The study of repairing Reed-Solomon codes was introduced in [3], and the works [4], [5], among others, have constructed repair schemes for RS codes over extension fields with total bandwidth that is much smaller than

the trivial bandwidth; in particular, the work [5] showed how to construct RS codes that achieve the *cut-set bound*.

Our work focuses on prime fields. To the best of our knowledge, the only work that gives positive results for repairing RS codes over prime fields is [6], discussed above and summarized in Table I.

b) Leakage-Resilient Secret Sharing: As mentioned above, prime fields are especially relevant for secret sharing. Shamir's SSS, which was first introduced in [20], is a fundamental cryptographic primitive that provides a secure method to distribute a secret among different parties, so that any k can recover the secret but no $k-1$ learn anything about it. Formally, given a secret $s \in \mathbb{F}$, a *reconstruction threshold* $k > 0$ and a number of parties n, Shamir's SSS works as follows. A dealer chooses a random polynomial f of degree at most $k-1$, so that $f(0) = s$. Then party i is given the share $f(\alpha_i)$, where $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$ are distinct, pre-selected points. It is easy to verify that any k parties can reconstruct the polynomial $f(x)$ and therefore recover the secret $f(0) = s$, while the shares of any $k-1$ parties reveal no information about the secret.

Benhamouda, Degwekar, Ishai, and Rabin [7] considered the question of *local leakage-resilience* of secret sharing schemes over prime fields and, in particular, Shamir's SSS over prime fields. In this setting, the model is that each party i may adversarially leak some function $\tau_i(f(\alpha_i)) \in \{0,1\}^m$ of their share, for some (small) m. A scheme is m-local leakage resilient if for any two secrets s, s', the total variation distance between the leaked messages under s and the leaked messages under s' is negligible. The work of Benhamouda et al. established the following.

Theorem 1. *[7, Corollary 4.12] Let m be a constant positive integer and let n go to infinity. There exists an $\alpha_m < 1$, for which Shamir's SSS with n players and threshold $k = \alpha_m n$ is m-local leakage resilient.*

Given Theorem 1, one cannot hope to repair RS codes over prime fields with rate arbitrarily close to 1 by downloading a constant number of bits from each node.

D. Organization

We present preliminaries in Section II. Our main theorems are presented in Sections III and IV. Due to space limitations, several proofs are omitted and will appear in the full version of this paper.

II. Preliminaries

We begin with some needed notations. Throughout, let p be a prime number. For integers $a < b$ let $[a,b] = \{a, a+1, \ldots, b\}$ and $[a] = \{1, 2, \ldots, a\}$. An arithmetic progression in some field $\mathbb{F}$ of length N and a step $s \in \mathbb{F}$ is a set of the form $\{a, a+s, \ldots, a+(N-1)s\}$ for some $a \in \mathbb{F}$. For two sets $A, B \subseteq \mathbb{F}_p$, define their *sumset* as $A + B := \{a + b \mid a \in A, b \in B\}$. For an element $\gamma \in \mathbb{F}_p$ we denote by $\gamma \cdot A := \{\gamma \cdot a : a \in A\}$ all the possible products of γ with elements in A. When we use the notation $\alpha \in \gamma \cdot [-t,t]$ for some $\alpha, \gamma \in \mathbb{F}_p$ and an integer $t < p$, we mean that there exists an integer $j \in [-t,t]$ for which $\alpha \equiv \gamma \cdot j \mod p$. We denote by $\boldsymbol{e}_p(x)$ the standard additive character that maps elements from $\mathbb{F}_p$ to $\mathbb{C}$, that is, $\boldsymbol{e}_p(x) := \exp\left(\frac{2\pi\sqrt{-1}}{p} \cdot x\right)$.

A. Exponential sum bounds

Exponential sums of various forms are studied extensively in number theory. They already have several applications in coding theory, and in this work we shall present another. That is, we will show how bounds like the Weil bound and bounds on Kloosterman sums imply repair schemes for RS codes over prime fields where every node transmits a constant number of bits. In this section, we state a few of the bounds we need, starting with the *Weil bound.*

Theorem 2. *[21, Theorem 5.38] Let $f \in \mathbb{F}_p[X]$ be a non-constant polynomial of degree at most k. It holds that*

$$\left| \sum_{\alpha \in \mathbb{F}_p} \boldsymbol{e}_p(f(\alpha)) \right| \leq (k-1) \cdot \sqrt{p} \, .$$

As one can see, the bound is non-trivial only when $\deg(f) \leq \sqrt{p}$. The Weil bound has applications throughout mathematics, theoretical computer science, and information theory. Motivated by these applications, there are extensions to the Weil bound that improve the bound in certain classes of polynomials, see e.g., [22], [23]. In [24], Bombieri extended Weil's bound and showed that it is true also for nontrivial rational functions defined over algebraic curves. We are interested in a particular case of this generalization; to simplify the notation, we present this particular case below. We mainly use the description from [25, pages 1-2, Equation 1.4] to present Bombieri's result.

Theorem 3. *[24, Theorem 5] Let $f = f_1/f_2$ be a rational function with $f_1, f_2 \in \mathbb{F}_p[x]$ and $\gcd(f_1, f_2) = 1$ in $\mathbb{Z}[x]$. Let $\deg(f) = \deg(f_1) + \deg(f_2)$ be the total degree of f. If $\deg(f) \geq 1$, then*

$$\left| \sum_{x=1}^{p-1} \boldsymbol{e}_p(f(x)) \right| \leq \begin{cases} (\deg(f)-1)\sqrt{p} + 1 & \deg(f_1) > \deg(f_2) \\ 2(\deg(f_2)-1)\sqrt{p} & \deg(f_1) \leq \deg(f_2) \end{cases} .$$

Another important family of exponential sums are *Kloosterman sums*, which first appeared in [26]. Again, bounds on these sums have broad applications, mainly in analytical number theory [27], [28], but also in coding theory [29]–[31].

Theorem 4. *[32, Theorems 2, 3] Let p be a sufficiently large prime and let n be an integer such that $e^{(\ln p)^{2/3} \cdot (\ln \ln p)^{1/3}} \leq n \leq \sqrt{p}$. Let $D = \frac{(\ln n)}{(\ln p)^{2/3} (\ln \ln p)^{1/3}}$. Then, it holds*

1) *For any a such that $\gcd(a,p) = 1$,*

$$\left| \sum_{1 \leq \nu \leq n} \boldsymbol{e}_p\left(\frac{a}{\nu}\right) \right| \leq n \cdot \frac{260 \ln D}{D} = o(n) \, .$$

2) *For any a, b such that* $\gcd(ab, p) = 1$

$$\left| \sum_{1 \le \nu \le n} \boldsymbol{e}_p\Big(\frac{a}{\nu} + b\nu\Big) \right| \le n \cdot \frac{222}{D^{3/4}} = o(n) \ .$$

B. Repair with arithmetic progressions

In this section, we recall the framework of [6] for using arithmetic progressions to define repair schemes. Recall the definition of a repair scheme for the nth node of a code $\mathcal{C}$. (For this discussion, we assume that the failed node is the nth one for notational simplicity, although we remove this assumption for our formal treatment). Such a repair scheme is comprised of a collection of $n-1$ functions $\tau_i : \mathbb{F}_p \to [s]$ and a function $G : [s]^{n-1} \to \mathbb{F}_p$ such that for any codeword $(c_1, \ldots, c_n) \in \mathcal{C}$

$$G(\tau_1(c_1), \ldots, \tau_{n-1}(c_{n-1})) = c_n. \tag{1}$$

Upon a failure of the nth node, the ith node, which holds the symbol c_i, computes $\tau_i(c_i)$ and transmits it using $\lceil \log s \rceil$ bits. Upon receiving the $n-1$ messages $\tau_i(c_i)$, the repair scheme is completed by calculating the nth symbol using (1). The bandwidth of the repair scheme, which is the total number of bits transmitted across the network during the repair, is equal to $(n-1)\lceil \log(s) \rceil$ bits.

Every function τ_i defines a partition $\{\tau_i^{-1}(a) : a \in [s]\}$ of $\mathbb{F}_p$. On the other hand, any partition of $\mathbb{F}_p$ into s sets defines a function whose value at the point $a \in \mathbb{F}_p$ is the index of the set that contains it. Hence, in the sequel, we will define the functions τ_i by partitions of $\mathbb{F}_p$ into s sets.

In [6], arithmetic progressions were used to construct partitions that give rise to efficient nonlinear repair schemes for RS codes over prime fields, as explained next.

Fix an integer $1 \le t \le p$, set $s = \lceil p/t \rceil$ and define $A_0, \ldots, A_{s-1}$ to be the partition of $\mathbb{F}_p$ into the following s arithmetic progressions of length t and step 1:

$$A_j = \begin{cases} \{jt, jt+1, \ldots, jt+t-1\} & 0 \le j \le s-2 \\ \{(s-1)t, \ldots, p-1\} & j = s-1. \end{cases} \tag{2}$$

For a nonzero $\gamma \in \mathbb{F}_p$, it is easy to verify that $\gamma \cdot A_0, \ldots, \gamma \cdot A_{s-1}$ is also a partition of $\mathbb{F}_p$ into arithmetic progressions of length t (except for the last set $\gamma \cdot A_{s-1}$) and step γ. Each function $\tau_i, i \in [n-1]$ of the repair scheme will be defined by a partition $\gamma_i \cdot A_0, \ldots, \gamma_i \cdot A_{s-1}$ for an appropriate selection of γ_i. Notice that the γ_i's will be distinct for distinct i's and therefore also the functions τ_i will be distinct for distinct i's. It was observed in [6] the partitions defined by the γ_i's extend to a *valid repair scheme* if (and only if) for any two codewords $c, c' \in \mathcal{C}$ that belong to the same set in all of the $n-1$ different partitions, i.e., $c_i, c_i' \in \gamma_i \cdot A_{j_i}$ for all $i \in [n-1]$, it holds that c, c' agree on their nth symbol, i.e., $c_n = c_n'$. In [6, Proposition 2.2], the authors provided a relatively simple sufficient condition for a linear code to have a valid repair scheme. We will rephrase their proposition to the specific case of RS codes, as that is the main focus of this paper.

Proposition 1 ([6])**.** *Consider an $[n, k]_p$ RS code defined with the evaluation points $\alpha_1, \ldots, \alpha_n$. Let $\ell \in [n]$ be the index of the failed node. Let $t < p$ be an integer and for $i \in [n] \setminus \{\ell\}$, let $\gamma_i \in \mathbb{F}_p^*$. If for any polynomial $f(x) \in \mathbb{F}_p[x]$ of degree less than k with $f(\alpha_i) \in \gamma_i \cdot [-t, t]$ for all $i \in [n] \setminus \{\ell\}$, it holds that $f(\alpha_\ell) = 0$, then, the γ_i's define a valid repair scheme for the ℓth node with a total bandwidth of $(n-1) \cdot \log\lceil p/t \rceil$ bits.*

In [6], the authors focused on the regime where $t = \Theta\Big(p^{1-\frac{1}{n-k+1}}\Big)$ and where n and k are constants compared to p. In particular, each surviving node sends $\Theta(\log(p))$ bits. As p is assumed to be growing for these results, this is not a constant.

In this work, we focus on the regime where $t = \Theta(p)$. Here, every node transmits a *constant* number ($\log\lceil p/t \rceil$) bits to the replacement node.

III. Repair using bounds on exponential sums

In this section, we present our main results for repairing single failed nodes for RS codes over prime fields. We shall extensively use the following simple lemma.

Lemma 1. *Let p be a prime number and let $t < p/4$. Let $a_1, \ldots, a_n \in \mathbb{F}_p$ such that $a_i \in [-t, t]$ for all $i \in [n]$. Then,* $\left|\sum_{i=1}^n \boldsymbol{e}_p(a_i)\right| \ge n \cdot \cos\Big(\frac{2\pi t}{p}\Big)$.

Our first repair scheme is capable of repairing every node in an $[n, 3]_p$ RS code for n that is small compared to p.

Theorem 5. *Let $B \ge 3$ be a positive integer. Let p be a large enough prime and n be an integer such that* $\exp\big((\ln p)^{2/3} \cdot (\ln \ln p)^{1/3}\big) \le n \le \sqrt{p}$*. The $[n+1, 3]_p$ RS code defined with the evaluation points $\alpha_i = i, i \in \{0\} \cup [n]$ admits repair of any node by downloading B bits from all the other nodes.*

Proof. Set $t := \lceil p/2^B \rceil$ and assume that we wish to repair the node ℓ. We will prove that the condition in Proposition 1 holds with $\gamma_i = i - \ell$ for every $i \in \{0\} \cup [n] \setminus \{\ell\}$. Assume that it does not hold. Thus, there exists a polynomial $f(x) = f_2(x-\ell)^2 + f_1(x-\ell) + f_0 \in \mathbb{F}_p[x]$ such that $f(i) \in (i-\ell) \cdot [-t, t]$ for all $i \in \{0\} \cup [n] \setminus \{\ell\}$ and $f(\ell) \ne 0$. Define

$$S := \left| \sum_{\substack{i=0 \\ i \ne \ell}}^{n} \boldsymbol{e}_p\Big(\frac{f(i)}{i-\ell}\Big) \right|$$

and note that by Lemma 1 and our choice of t, it holds that $S \ge n \cdot \cos\Big(\frac{2\pi t}{p}\Big) \ge n \cdot \cos\Big(\frac{\pi}{4} + \frac{2\pi}{p}\Big) > 0.7n$ for large enough p. On the other hand,

$$\begin{aligned} S &= \left| \sum_{\substack{i=0 \\ i \ne \ell}}^{n} \boldsymbol{e}_p\Big(f_2 \cdot (i-\ell) + f_1 + \frac{f_0}{i-\ell}\Big) \right| \\ &= \left| \sum_{\substack{i=0 \\ i \ne \ell}}^{n} \boldsymbol{e}_p\Big(f_2 \cdot (i-\ell) + \frac{f_0}{i-\ell}\Big) \right| \\ &\le \left| \sum_{i=1}^{\ell} \boldsymbol{e}_p\Big(f_2 \cdot i + \frac{f_0}{i}\Big) \right| + \left| \sum_{i=1}^{n-\ell} \boldsymbol{e}_p\Big(f_2 \cdot i + \frac{f_0}{i}\Big) \right| \end{aligned}$$

The second equality follows by extracting $|\boldsymbol{e}_p(f_1)|$ from all the summands and the inequality follows by the triangle inequality. Denote $\ell' = n - \ell$ and assume without loss of generality that $\ell \geq n/2 \geq \ell'$. Denote $S' = \left|\sum_{i=1}^{\ell} \boldsymbol{e}_p\left(f_2 \cdot i + \frac{f_0}{i}\right)\right|$.

Recall that our assumption states that $f(\ell) = f_0 \neq 0$ which implies that $\gcd(f_0, p) = 1$. Consider the following two options. First, if $f_2 \neq 0$ then $\gcd(f_0 f_2, p) = 1$ and by the second bound in Theorem 4, we have $S' = o(n)$. Second, if $f_2 = 0$ then $S' = \left|\sum_{i=1}^{\ell} \boldsymbol{e}_p\left(f_0 i^{-1}\right)\right|$ and by the first bound in Theorem 4, we have $S' = o(n)$. We conclude that $S \leq 0.5n + o(n)$ and arrive at a contradiction.

The claim about the bandwidth follows by noting that according to Proposition 1, each node sends $\lceil \log(\lceil \frac{p}{t} \rceil) \rceil \leq \lceil \log(2^B) \rceil = B$ bits. □

Our next construction presents a full-length RS code of dimension $O(\sqrt{p})$ where any node can be repaired by downloading a constant number of bits from all the remaining nodes. The proof of the next Theorem is similar to the proof of Theorem 5 but here we apply the bound given in Theorem 3.

Theorem 6. *Let $B \geq 3$ be a positive integer. Let k be an integer and p be a prime such that $k \leq \sqrt{p}/2$. The respective $[p, k]_p$ RS code admits repair of any node by downloading B bits from all the $p - 1$ other nodes.*

Remark 2. *The repair schemes described in this section download a constant number of bits from each node. While the total bandwidth of these schemes is larger than the amount of information required to determine the polynomial ($k \cdot \log(p)$), they do not reveal the entire polynomial. For example, consider the polynomials $f(x) = x$ and $g(x) = 2x$. In both schemes, each node $f(\alpha)$ and $g(\alpha)$ transmits the same value, since $f(\alpha), g(\alpha) \in \alpha \cdot [0, t-1]$.*

IV. Recovering the entire polynomial

In Section III, we constructed repair schemes for repairing a single code symbol. In this section, we show how one can download a constant number of bits from every node and recover the entire polynomial. Clearly, since the entire polynomial is recovered, we must download at least $k \log(p)$ bits from the nodes. Trivially, one can contact k nodes and download their symbol ($\log(p)$ bits), solve a system of linear equations, and get the polynomial. Here, we will download a constant number of bits from more than k nodes, and show that these bits suffice to learn the entire polynomial.

Let $\mathcal{C} \subseteq \mathbb{F}_p^n$ be a linear code over $\mathbb{F}_p$. A *decoding scheme* for $\mathcal{C}$ is a set of n functions $\tau_i : \mathbb{F}_p \to [s]$ and a function $G : [s]^n \to \mathbb{F}_p^n$ such that for any codeword $(c_1, \ldots, c_n) \in \mathcal{C}$

$$G(\tau_1(c_1), \ldots, \tau_n(c_n)) = (c_1, \ldots, c_n). \tag{3}$$

The trivial decoding scheme for an $[n, k]$ is just the scheme where we set k functions, say, $\tau_1, \ldots, \tau_k : \mathbb{F}_p \to \mathbb{F}_p$ to be the identity functions. Proposition 2 below shows that, in the regime where $k < O(n/\log(p))$, there are n functions τ_i, each of which output a constant number of bits from one of the n nodes, and whose output can be used to recover any polynomial of degree less than k. The proof of this Proposition is very similar to the proof of [8, Theorem 2]. It also randomly chooses the leakage functions and then uses the union bound. The difference is that we argue that we can learn the entire polynomial while in [8, Theorem 2] the authors aim for learning just the secret with high probability.

Proposition 2. *Let p be a prime. Let n, T, and k be integers such that $k < \frac{(n+1)\log(T)}{2\log(p)+\log(T)}$. Let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_p$ be distinct. There are n functions $\tau_1, \ldots, \tau_n : \mathbb{F}_p \to [T]$ such that for any $f \in \mathbb{F}_p[x]$ of degree at most $k-1$, the information $(\tau_1(f(\alpha_1)), \ldots, \tau_n(f(\alpha_n)))$ uniquely determines f among such polynomials.*

Before stating our decoding scheme, we rephrase the condition given in Proposition 1 to this case.

Proposition 3 (Follows from Proposition 1)**.** *Consider an $[n, k]_p$ RS code defined by the evaluation points $\alpha_1, \ldots, \alpha_n$. Let $t < p$ be an integer and for $i \in [n]$, let $\gamma_i \in \mathbb{F}_p^*$. If the only polynomial $f \in \mathbb{F}_p[x]$ of degree at most $k-1$ for which $f(\alpha_i) \in \gamma_i[-t, t]$ for all $i \in [n]$ is the zero polynomial, then the γ_i's define a decoding scheme with total bandwidth $n \cdot \log(\lceil p/t \rceil)$ bits.*

We now show that a simple application of the Weil bound together with a slight change in the repair scheme from Theorem 5 gives a decoding scheme for an $[\mathbb{F}_p^*, k]_p$ RS code with $k = O(\sqrt{p})$.

Theorem 7. *Let $B \geq 3$ be a positive integer. Let $k < \sqrt{p}/2$. The RS code $[\mathbb{F}_p^*, k]_p$ admits a decoding scheme by downloading B bits from all the nodes.*

Proof of Theorem 7. Set $t := \lceil p/2^B \rceil$. We will show that the condition in Proposition 3 holds with $\gamma_j = j^{-1}$ for all $j \in [p]$. Assume towards a contradiction that the condition does not hold. Namely, there is a nonzero polynomial $f(x)$ of degree $\leq k-1$ such that $f(j) \in j^{-1} \cdot [-t, t]$ for every $j \in [p]$. Define

$$S := \left|\sum_{i=1}^{p} \boldsymbol{e}_p(\alpha_i \cdot f(\alpha_i))\right|$$

and observe that by Lemma 1, $S > 0.7(p-1)$. On the other hand, as we assumed that $f \not\equiv 0$, the polynomial $x \cdot f(x)$ is not a constant and therefore, by the Weil bound given in Theorem 2,

$$S = \left|\sum_{i=1}^{p} \boldsymbol{e}_p(\alpha_i \cdot f(\alpha_i))\right| \leq k \cdot \sqrt{p} \leq \frac{p}{2}\,.$$

We arrive at a contradiction. Thus, f must be the zero polynomial. The claim about the bandwidth is identical to Theorem 5. □

V. Acknowledgement

We thank Maxim A. Korolev for helpful conversations about upper bounds on Kloosterman sums.

REFERENCES

[1] T. X. Dinh, L. Y. N. Nguyen, L. J. Mohan, S. Boztas, T. T. Luong, and S. H. Dau, "Practical considerations in repairing reed-solomon codes," *arXiv preprint arXiv:2205.11015*, 2022.

[2] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE transactions on information theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[3] K. Shanmugam, D. S. Papailiopoulos, A. G. Dimakis, and G. Caire, "A repair framework for scalar MDS codes," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 998–1007, 2014.

[4] V. Guruswami and M. Wootters, "Repairing Reed-Solomon codes," *IEEE transactions on Information Theory*, vol. 63, no. 9, pp. 5684–5698, 2017.

[5] I. Tamo, M. Ye, and A. Barg, "Optimal repair of Reed-Solomon codes: Achieving the cut-set bound," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2017, pp. 216–227.

[6] R. Con and I. Tamo, "Nonlinear repair of reed-solomon codes," *IEEE Transactions on Information Theory*, vol. 68, no. 8, pp. 5165–5177, 2022.

[7] F. Benhamouda, A. Degwekar, Y. Ishai, and T. Rabin, "On the local leakage resilience of linear secret sharing schemes," *Journal of Cryptology*, vol. 34, no. 2, pp. 1–65, 2021.

[8] J. B. Nielsen and M. Simkin, "Lower bounds for leakage-resilient secret sharing," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2020, pp. 556–577.

[9] H. K. Maji, H. H. Nguyen, A. Paskin-Cherniavsky, T. Suad, and M. Wang, "Leakage-resilience of the shamir secret-sharing scheme against physical-bit leakages," in *Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part II*. Springer, 2021, pp. 344–374.

[10] H. K. Maji, H. H. Nguyen, A. Paskin-Cherniavsky, and M. Wang, "Improved bound on the local leakage-resilience of shamir's secret sharing," in *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2022, pp. 2678–2683.

[11] S. El Rouayheb and K. Ramchandran, "Fractional repetition codes for repair in distributed storage systems," in *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2010, pp. 1510–1517.

[12] S. Goparaju, S. El Rouayheb, R. Calderbank, and H. V. Poor, "Data secrecy in distributed storage systems under exact repair," in *2013 International Symposium on Network Coding (NetCod)*. IEEE, 2013, pp. 1–6.

[13] D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Repair optimal erasure codes through Hadamard designs," *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 3021–3037, 2013.

[14] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Transactions on Information Theory*, vol. 59, no. 3, pp. 1597–1616, 2012.

[15] Z. Wang, I. Tamo, and J. Bruck, "Explicit minimum storage regenerating codes," *IEEE Transactions on Information Theory*, vol. 62, no. 8, pp. 4466–4480, 2016.

[16] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.

[17] M. Ye and A. Barg, "Explicit constructions of high-rate MDS array codes with optimal repair bandwidth," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2001–2014, 2017.

[18] ——, "Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization," *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6307–6317, 2017.

[19] S. Goparaju, A. Fazeli, and A. Vardy, "Minimum storage regenerating codes for all parameters," *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6318–6328, 2017.

[20] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[21] R. Lidl and H. Niederreiter, *Finite fields*. Cambridge university press, 1997, no. 20.

[22] T. Cochrane and C. Pinner, "An improved mordell type bound for exponential sums," *Proceedings of the American Mathematical Society*, vol. 133, no. 2, pp. 313–320, 2005.

[23] T. Kaufman and S. Lovett, "New extension of the weil bound for character sums with applications to coding," in *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*. IEEE, 2011, pp. 788–796.

[24] E. Bombieri, "On exponential sums in finite fields," *American Journal of Mathematics*, vol. 88, no. 1, pp. 71–105, 1966.

[25] T. Cochrane and Z. Zheng, "Exponential sums with rational function entries," *Acta Arithmetica*, vol. 95, no. 1, pp. 67–95, 2000.

[26] H. Kloosterman, "On the representation of numbers in the formax 2+ by 2+ cz 2+ dt 2," *Acta mathematica*, vol. 49, no. 3, pp. 407–464, 1927.

[27] D. Heath-Brown, "Arithmetic applications of Kloosterman sums," *Nieuw Archief voor Wiskunde*, vol. 1, pp. 380–384, 2000.

[28] H. Iwaniec and E. Kowalski, *Analytic number theory*. American Mathematical Soc., 2021, vol. 53.

[29] T. Helleseth and V. Zinoviev, "On z4-linear Goethals codes and Kloosterman sums," *Designs, Codes and Cryptography*, vol. 17, no. 1, pp. 269–288, 1999.

[30] M. J. Moisio, "The moments of a kloosterman sum and the weight distribution of a Zetterberg-type binary cyclic code," *IEEE transactions on information theory*, vol. 53, no. 2, pp. 843–847, 2007.

[31] V. A. Zinoviev, "On classical kloosterman sums," *Cryptography and Communications*, vol. 11, pp. 461–496, 2019.

[32] M. A. Korolev, "Karatsuba's method for estimating kloosterman sums," *Sbornik: Mathematics*, vol. 207, no. 8, p. 1142, 2016.