On the NP-Hardness of Bounded Distance Decoding of Reed-Solomon Codes

Venkata Gandikota
Purdue University
vgandiko@purdue.edu

Badih Ghazi
MIT
badih@mit.edu

Elena Grigorescu
Purdue University
elena-g@purdue.edu

***Abstract*—Guruswami and Vardy (IEEE Trans. Inf. Theory, 2005) show that given a Reed-Solomon code over a finite field $\mathbb{F}$, of length n and dimension k, and given a target vector $v \in \mathbb{F}^n$, it is NP-hard to decide if there is a codeword that disagrees with v on at most $n-k-1$ coordinates. Understanding the complexity of this Bounded Distance Decoding problem as the amount of error in the target decreases is an important open problem in the study of Reed-Solomon codes. In this work, we extend the result of Guruswami and Vardy by proving that it is NP-hard to decide the existence of a codeword that disagrees with v on $n-k-2$, and on $n-k-3$ coordinates. No other NP-hardness results were known before for an amount of error $< n-k-1$. The core of our proofs is showing the NP-hardness of a parameterized generalization of the Subset-Sum problem to higher degrees (called Moments Subset-Sum) that may be of independent interest.**

I. Introduction

A *linear error-correcting code* of length n and dimension k, over a finite field alphabet $\mathbb{F}$, is a vector space $\mathcal{C} \subseteq \mathbb{F}^n$ of dimension k. The Hamming distance between $x, y \in \mathbb{F}^n$ is $\delta(x,y) := |\{i \in [n] : x_i \neq y_i\}|$, and the *minimum distance* of $\mathcal{C}$ is $\Delta(\mathcal{C}) := \min_{x \neq y \in \mathcal{C}} \delta(x,y)$. In the Bounded Distance Decoding problem with parameter d, we are given a code $\mathcal{C} \subseteq \mathbb{F}^n$ and a target vector $v \in \mathbb{F}^n$, and we are asked to decide whether there exists a codeword $c \in \mathcal{C}$ such that the Hamming distance $\delta(v,c) \leq e(d, \Delta(\mathcal{C}))$, where $e(d,\cdot)$ is an error-weight function of interest. This is a fundamental question in the study of general error-correcting codes, and its complexity is not fully understood even for well-studied codes such as Reed-Solomon (RS) codes. More precisely, RS codes still exhibit a wide gap between the setting of small error-weight, where the problem is solvable in polynomial-time [Sud97], [GS99], and the setting of large error-weight, where the problem is known to be NP-hard [GV05]. In this work, we improve this gap by generalizing the decade-old NP-hardness results of [GV05] to smaller error parameters.

A Reed-Solomon code of length n, dimension k, defined over a finite field $\mathbb{F}$, and specified by an evaluation set $D = \{x_1, x_2, \ldots, x_n\} \subseteq \mathbb{F}$ is the set $RS_{D,k} = \{\langle f(x_1), \ldots, f(x_n)\rangle : x_1, \ldots, x_n \in D, f(x) \in \mathbb{F}[x], deg(f(x)) \leq k-1\}$. Its minimum distance is $\Delta = n-k+1$. We are interested in the Bounded Distance Decoding problem for RS codes with parameter d (denoted RS-BDD(d)) corresponding to the error-weight function $e(d,\Delta) = \Delta - d - 1 = n-k-d$.

When $e \leq n - \sqrt{nk}$ (i.e., the "Johnson radius"), the list-decoding algorithms of [Sud97], [GS99] can solve RS-BDD($\sqrt{nk}-k$) in polynomial time. At the other end of the spectrum, if $e = n-k$ one can easily interpolate a degree $k-1$ polynomial that agrees with the target on a set of k arbitrary entries. The famous result of [GV05] shows that right below this value, i.e., when $e = n-k-1$, RS-BDD(1) is NP-hard. Their proof is via a reduction from the 3D-matching problem, and it does not imply any hardness results for smaller values of the error-weight parameter. In the range where $e < n-k-1$, the only known hardness result is due to [CW10] who show a reduction from the Discrete Log problem, when the evaluation set is $D = \mathbb{F}_q^*$, for $e \geq \frac{2}{3}(n-k+1)$. Note that the hardness of Discrete Log is a stronger complexity-theoretic assumption than $P \neq NP$. In particular, the Discrete Log problem over finite fields is not believed to be NP-hard and [Sho97] gives a polynomial-time quantum algorithm solving it.

In this work, we prove that for every $d \in \{2,3\}$, RS-BDD(d) is NP-hard. We note that, as is the case of [GV05], our reductions require that n be polylogarithmic in $|\mathbb{F}|$. Our proof starts with the observation that an extension of the Subset-Sum problem up to d moments (called Moments Subset-Sum with parameter d) is polynomial-time reducible to RS-BDD(d). The crux of our proof is showing the NP-hardness of the Moments Subset-Sum problem, formally defined next.

Definition 1 (Moments Subset-Sum: MSS(d))**.** *Given a set $A = \{a_1, \ldots, a_n\}$, $a_i \in F$, integer t, and $m_1, \ldots, m_d \in \mathbb{F}$, decide if there exists a subset $S \subseteq A$ of size t, satisfying $\sum_{a \in S} a^i = m_i$ for all $i \in [d]$.*

Badih Ghazi is supported in part by NSF STC Award CCF 0939370 and NSF award number CCF-1217423.

Note that MSS(1) is the usual Subset-Sum problem, which is in fact used by [CM07] to show an alternate proof of [GV05]. The NP-hardness of Subset-Sum can be shown by a well-known reduction from 3-SAT. Surprisingly, it turns out to be much more difficult to prove NP-hardness for MSS(d) for values of $d \geq 2$. Intuitively, this is because the reduction from 3-SAT to Subset-Sum encodes satisfiability of a 3-SAT formula in the decimal representation of the integers of the Subset-Sum instance, and it becomes more difficult to control the decimal representations when constraints involving squares and higher powers are introduced.

In fact, the current barrier to extending our proof approach to show the NP-hardness of MSS(d) for larger values of d (and thereby obtain the NP-hardness of RS-BDD(d) for larger values of d) is the following intriguing algebraic question.

Question 1. *Given a finite field* $\mathbb{F}$*,* $a, b \in \mathbb{F}$*, and* $d \in \mathbb{N}$*, does there exist* $k = k(d)$ *and explicit* $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k \in \mathbb{F}$ *satisfying* $x_1 + x_2 + \ldots + x_k = y_1 + y_2 + \ldots + y_k$ *and* $a^i + \sum_{j=1}^{k} x_j^i = b^i + \sum_{j=1}^{k} y_j^i$ *for every* $i \in \{2, \ldots, d\}$ *?*

We note that the same question is also of interest when a, b, $x_1, \ldots, x_k$, $y_1, \ldots, y_k$ are rational numbers or integers. For $d \in \{2, 3\}$, we are able to answer Question 1 in the affirmative and to provide explicit constructions over domains that are either finite fields of large characteristic, the rational numbers or the ring of integers. These constructions form the starting point of our NP-hardness proofs.

II. Reduction from Moments Subset Sum

We start by reformulating the RS-BDD problem as a polynomial reconstruction question as follows.

Definition 2 (RS-BDD(d))**.** *Given a set of* n *distinct points in* $\mathbb{F}^2$*,* $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$*, and an integer* $k < n$*, decide if there exists a polynomial* $f(x) \in \mathbb{F}[x]$ *of degree at most* $k - 1$ *passing through at least* $k + d$ *points in* D*.*

Here, the set of evaluation points of the code is $D' = \{x_1, \ldots x_n\}$ and the target vector is $y = (y_1, \ldots, y_n) \in \mathbb{F}^n$. Note that if a polynomial $f(x)$ passes through at least $k+d$ points in D, then the corresponding codeword $\langle f(x_1), \ldots, f(x_n)\rangle$ is at a Hamming distance of at most $e = n-k-d$ from y. We will reduce from the following problem which can be easily seen to be equivalent to MSS(d) over large prime finite fields $\mathbb{F}$ using Newton's identities [Sta99]. We note that this connection has been previously made (e.g. [LW08]).

Definition 3 (Symmetric Subset-Sum (SSS(d)))**.** *Given a set of* n *distinct elements of* $\mathbb{F}$*,* $A = \{a_1, a_2, \ldots, a_n\}$*, integer* t*, and* $B_1, B_2, \ldots B_d \in \mathbb{F}$*, decide if there exists a subset* S *of* A *of size* t*, such that for every* $i \in [d]$ *the elementary symmetric sums of the elements of* $S = \{s_1, \ldots, s_t\}$ *satisfy* $E_i(S) = \sum_{1 \leq j_1 < j_2 < \cdots < j_i \leq t} s_{j_1} \ldots s_{j_i} = B_i$*.*

Lemma 1. *SSS(d) is polynomial-time reducible to RS-BDD(d).*

Proof Sketch: Given an instance $\langle A = \{a_1, \ldots, a_n\}, t, B_1, \ldots, B_d\rangle$ of SSS(d), let $k = t - d + 1$. To construct the set D of the RS-BDD(d) instance, first define the degree d polynomial $p(x) := x^d - B_1 x^{d-1} + \cdots + (-1)^{d-1} B_{d-1} x$. Let $D = \{(a_i^{-1}, -p(a_i)) : a_i \in A\} \cup \{(0, (-1)^d B_d)\}$. We then show that there is a polynomial $f(x) \in \mathbb{F}[X]$ of degree at most $k-1$ which agrees with at least $k+d$ points of D if and only if there is a solution to the given instance of SSS(d). The full proof appears in the full version. ■

III. NP-Hardness of MSS(2)

Theorem 4. *MSS(2) is NP-hard.*

Proof: We present the proof for the case where the domain is the ring of integers, and then observe that the proof extends to any sufficiently large prime field (see the end of the proof for the details). We give a polynomial-time reduction from the 1-in-3-SAT problem in which we are given a 3-SAT formula ϕ on n variables and m clauses and are asked to determine if there exists an assignment $x \in \{0, 1\}^n$ satisfying exactly one literal in each clause. It is known that this problem is NP-hard even for $m = O(n)$ [Sch78]. We start by recalling the reduction from 1-in-3-SAT to Subset-Sum which will be used in our reduction to MSS(2). In that reduction, each variable $(x_i, \overline{x_i})$ is mapped to 2 integers a_i (corresponding to x_i) and b_i (corresponding to $\overline{x_i}$). The integers a_i and b_i and the target B are defined in terms of their length-$(n + m)$ Σ-ary representation (for some sufficiently large even constant, say $\Sigma = 4$) as follows:

- The Σ-ary representatios of a_i and b_i consist of 2 parts: a variable region consisting of the leftmost n digits and a clause region consisting of the (remaining) rightmost m digits.
- In the variable region, a_i and b_i have a 1 at the ith digit and 0's at the other digits.
- In the clause region, for every $j \in [m]$, a_i (resp. b_i) has a 1 at the jth location if x_i (resp. $\overline{x_i}$) appears in clause j, and a 0 otherwise.
- The target B is set to the integer whose Σ-ary representation is the all 1's.

Those Σ-ary representations are illustrated in Figure 1.

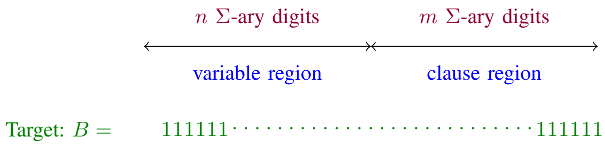

Fig. 1. Σ-ary representations in the original reduction from 1-in-3-SAT to Subset-Sum.

This reduction to Subset-Sum can be seen to be complete and sound. We now use it to give a reduction to $\mathrm{MSS}(2)$. In this reduction, each variable $(x_i, \overline{x_i})$ is mapped to 6 integers $a_{i,1}, a_{i,2}, a_{i,3}$ (corresponding to x_i) and $b_{i,1}, b_{i,2}, b_{i,3}$ (corresponding to $\overline{x}_i$). Let $\{a_i, b_i : i \in [n]\}$ be the integers produced by the above reduction to Subset-Sum. We denote by a_i^v (resp. a_i^c) the Σ-ary representation of the variable (resp. clause) region of a_i. For any Σ-ary representation x of a natural number, let $(x)_{\Sigma'}$ be the natural number whose Σ'-representation is x. For example, if x is the Σ-ary number (10011), then $(x)_{\Sigma'} = \Sigma'^4 + \Sigma' + 1$. Denote by 1^l the concatenation of l ones. Let ν and h be natural numbers to be specified later on. The integers $a_{i,1}, a_{i,2}, a_{i,3}, b_{i,1}, b_{i,2}, b_{i,3}$ and the targets B_1 and B_2 are defined as follows:

- $a_{i,1} = \Sigma^{\nu+m+h-1}(a_i^v)_{\Sigma^h} + \Sigma^\nu (a_i^c)_\Sigma$
- $b_{i,1} = \Sigma^{\nu+m+h-1}(b_i^v)_{\Sigma^h} + \Sigma^\nu (b_i^c)_\Sigma$
- $a_{i,2} = \frac{a_{i,1}^2 - b_{i,1}^2}{4\Sigma^i} - \frac{\Sigma^i}{2}$ and $a_{i,3} = -a_{i,2}$.
- $b_{i,2} = \frac{a_{i,1}^2 - b_{i,1}^2}{4\Sigma^i} + \frac{\Sigma^i}{2}$ and $b_{i,3} = -b_{i,2}$.
- $B_1 = \Sigma^{\nu+m+h-1}(1^n)_{\Sigma^h} + \Sigma^\nu (1^m)_\Sigma$
- $B_2 = \displaystyle\sum_{i=1}^{n} a_{i,1}^2 + a_{i,2}^2 + a_{i,3}^2$.

Note that $a_{i,1}$ (resp. $a_{i,2}$) defined above can be seen as obtained by inserting $h-1$ consecutive 0's in the variable region of the Σ-ary representation of a_i (resp. b_i), and then inserting ν zeros at the right. Those Σ-ary representations are illustrated in Figure 2. Moreover, observe that $a_{i,2}, a_{i,3}, b_{i,2}, b_{i,3}$ are defined above in such a way that $a_{i,2} + a_{i,3} = b_{i,2} + b_{i,3} = 0$ and $a_{i,1}^2 + a_{i,2}^2 + a_{i,3}^2 = b_{i,1}^2 + b_{i,2}^2 + b_{i,3}^2$. Note that this reduction runs in time polynomial in n, m, h and ν. We next prove the correctness of the reduction.

a) Completeness: Assume that the starting 3-SAT formula has a satisfying 1-in-3-SAT assignment x. Consider the subset $S = \cup_{i\in[n]:x_i=1}\{a_{i,1}, a_{i,2}, a_{i,3}\} \cup \cup_{i\in[n]:x_i=0}\{b_{i,1}, b_{i,2}, b_{i,3}\}$ of size $k = 3n$. We now check that the first 2 moments of S equal B_1 and B_2.

$$\sum_{y\in S} y = \sum_{\substack{i\in[n]:\\ x_i=1}} a_{i,1} + a_{i,2} + a_{i,3} + \sum_{\substack{i\in[n]:\\ x_i=0}} b_{i,1} + b_{i,2} + b_{i,3}$$

$$\begin{aligned}
&= \sum_{i\in[n]:x_i=1} a_{i,1} + \sum_{i\in[n]:x_i=0} b_{i,1} \\
&= \sum_{i\in[n]:x_i=1} \Sigma^{\nu+m+h-1}(a_i^v)_{\Sigma^h} + \Sigma^\nu (a_i^c)_\Sigma \\
&+ \sum_{i\in[n]:x_i=0} \Sigma^{\nu+m+h-1}(b_i^v)_{\Sigma^h} + \Sigma^\nu (b_i^c)_\Sigma \\
&= \Sigma^{\nu+m+h-1}(\sum_{i\in[n]:x_i=1} a_i^v + \sum_{i\in[n]:x_i=0} b_i^v)_{\Sigma^h} \\
&+ \Sigma^\nu(\sum_{i\in[n]:x_i=1} a_i^c + \sum_{i\in[n]:x_i=0} b_i^c)_\Sigma \\
&= \Sigma^{\nu+m+h-1}(1^n)_{\Sigma^h} + \Sigma^\nu(1^m)_\Sigma \\
&= B_1
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\sum_{y\in S} y^2 &= \sum_{i\in[n]:x_i=1} a_{i,1}^2 + a_{i,2}^2 + a_{i,3}^2 \\
&+ \sum_{i\in[n]:x_i=0} b_{i,1}^2 + b_{i,2}^2 + b_{i,3}^2 \\
&= \sum_{i\in[n]} a_{i,1}^2 + a_{i,2}^2 + a_{i,3}^2 \\
&= B_2
\end{aligned}$$

where we used the fact that $a_{i,1}^2 + a_{i,2}^2 + a_{i,3}^2 = b_{i,1}^2 + b_{i,2}^2 + b_{i,3}^2$ for every $i \in [n]$.

b) Soundness: Assume that S is a satisfying subset of the $\mathrm{MSS}(2)$ instance. We will use S in order to construct a 1-in-3-SAT satisfying assignment to the starting 3-SAT instance ϕ. Let's first sketch the high-level intuition of the proof. First, we argue that S should contain the same number of positive and negative integers from the set $\{a_{1,2}, b_{1,2}, a_{1,3}, b_{1,3}\}$. This is because these four integers have roughly the same absolute value, which turns out to be much larger than B_1 and than any other possible integer in S. Thus, unless S contains the same number of positive and negative elements from $\{a_{1,2}, b_{1,2}, a_{1,3}, b_{1,3}\}$, the integers in S cannot add up to B_1. Then, we show using an inductive argument that for each $i \in [n]$, S contains the same number of positive and negative elements from $\{a_{i,2}, b_{i,2}, a_{i,3}, b_{i,3}\}$ (i.e., Lemma 2). We next prove that, in fact, the elements of S from the set $\cup_{i=1}^n\{a_{i,2}, b_{i,2}, a_{i,3}, b_{i,3}\}$ *exactly* cancel each other out, i.e., they add up to 0. Hence, we obtain that the integers of S from $\cup_{i=1}^n\{a_{i,1}, b_{i,1}\}$ add up to B_1, at which point we can run the soundness analysis of the original reduction from 1-in-3-SAT to Subset Sum in order to construct an assignment satisfying exactly one litteral in each clause. We now give the formal proof, which starts with the following lemma whose proof is deferred to the full version. We assume henceforth that $\nu \geq 10(m + nh)$ and $h \geq 10m$.

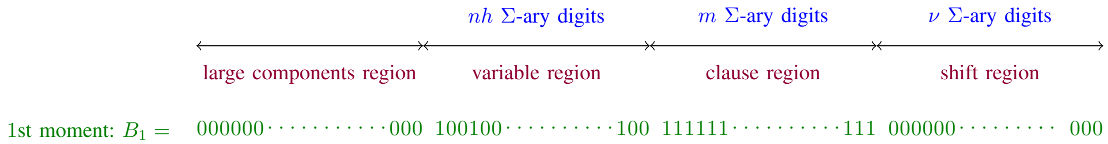

Fig. 2. Σ-ary representations in the reduction from 1-in-3-SAT to MSS(2). In this figure, $h = 3$. The "large components region" only contains zeros in $\{a_{i,1}, b_{i,1} : \ i \in [n]\}$ but contains non-zeros in $\{|a_{i,2}|, |b_{i,2}| : \ i \in [n]\}$.

Lemma 2. *For every $i \in [n]$, $|S \cap \{a_{i,2}, b_{i,2}\}| = |S \cap \{a_{i,3}, b_{i,3}\}|$.*

Lemma 2 implies that for every $i \in [n]$, $\displaystyle\sum_{y \in S \cap \{a_{i,2},b_{i,2},a_{i,3},b_{i,3}\}} y = r_i \times \Sigma^i$ for some $r_i \in \{-1, 0, 1\}$. The next lemma, whose proof is deferred to the full version, argues that in fact each r_i equals 0.

Lemma 3. *For every $i \in [n]$, $r_i = 0$.*

Lemma 3 implies that $\displaystyle\sum_{y \in S \cap \cup_{i=1}^{n} \{a_{i,2},b_{i,2},a_{i,3},b_{i,3}\}} y = 0$ and hence that $\displaystyle\sum_{y \in S \cap \cup_{i=1}^{n} \{a_{i,1},b_{i,1}\}} y = B_1$. We now carry out the soundness analysis of the original reduction from 1-in-3-SAT to Subset Sum in order to construct an assignmnent that satisfies exactly one litteral in each clause of the given 3-SAT formula. This concludes the soundness analysis of our reduction. To sum up, setting $h = 10m$, $\nu = 10(m + nh)$ and Σ to a sufficiently large even constant (say 4) yields a polynomial-time reduction from 1-in-3-SAT to MSS(2). Thus, MSS(2) is NP-hard. We now extend the proof to any sufficiently large prime field. Let $p \geq 2^{cn^2}$ be a prime number with c some absolute constant. This ensures that the magnitudes of all integers appearing in the above proof are $< p/2$. Performing the above computations modulo p, we can carry out all the arguments as in the case of the integers, and thus the proof holds over $\mathbb{F}_p$. ■

IV. NP-Hardness of MSS(3)

Theorem 5. *MSS(3) is NP-hard.*

Proof: We present the proof over the field of rational numbers, and then observe that the proof extends to any sufficiently large prime field (see the end of the proof for the details). We give a polynomial-time reduction from the 1-in-3-SAT problem in which we are given a 3-SAT formula ϕ on n variables and m clauses and are asked to determine whether there exists an assignment $x \in \{0, 1\}^n$ that satisfies exactly one literal in each clause. It is known that this problem is NP-hard even for $m = O(n)$ [Sch78]. Recall the known reduction from 1-in-3-SAT to Subset-Sum that is described at the beginning of the proof of Theorem 4. Let a_i and b_i be the positive integers corresponding to variable $(x_i, \overline{x_i})$ and let B be the target positive integer in that reduction. As before the Σ-ary representations of the a_i's, the b_i's and B are illustrated in Figure 1. We now use this reduction to reduce 1-in-3-SAT to MSS(3). In the new reduction, each variable $(x_i, \overline{x_i})$ is mapped to 6 rational numbers ${a'}_{i,1}, {a'}_{i,2}, {a'}_{i,3}$ (corresponding to x_i) and ${b'}_{i,1}, {b'}_{i,2}, {b'}_{i,3}$ (corresponding to $\overline{x}_i$). The symbols a_i^v, a_i^c and $(x)_{\Sigma'}$ are defined as in the proof of Theroem 4. Let ν and h be natural numbers to be specified later on. The rational numbers ${a'}_{i,1}, {a'}_{i,2}, {a'}_{i,3}, {b'}_{i,1}, {b'}_{i,2}, {b'}_{i,3}$ and the targets B_1, B_2 and B_3 are defined as follows:

- ${a'}_{i,1} = \Sigma^{\nu+m+h-1}(a_i^v)_{\Sigma^h} + \Sigma^{\nu}(a_i^c)_{\Sigma}$
- ${b'}_{i,1} = \Sigma^{\nu+m+h-1}(b_i^v)_{\Sigma^h} + \Sigma^{\nu}(b_i^c)_{\Sigma}$
- Define $\beta_i := \Sigma^i$, $\gamma_i := \frac{{a'}_{i,1}^2 - {b'}_{i,1}^2}{2\beta_i}$ and $\alpha_i := \frac{{a'}_{i,1}^3 - {b'}_{i,1}^3}{6\beta_i\gamma_i}$.
- ${a'}_{i,2} = \alpha_i + \frac{\gamma_i}{2} - \frac{\beta_i}{2}$.
- ${a'}_{i,3} = \alpha_i - \frac{\gamma_i}{2} + \frac{\beta_i}{2}$.
- ${b'}_{i,2} = \alpha_i + \frac{\gamma_i}{2} + \frac{\beta_i}{2}$.
- ${b'}_{i,3} = \alpha_i - \frac{\gamma_i}{2} - \frac{\beta_i}{2}$.
- $B_1 = \Sigma^{\nu+m+h-1}(1^n)_{\Sigma^h} + \Sigma^{\nu}(1^m)_{\Sigma} + \displaystyle\sum_{i=1}^{n} {a'}_{i,2} + {a'}_{i,3}$.
- $B_2 = \displaystyle\sum_{i=1}^{n} {a'}_{i,1}^2 + {a'}_{i,2}^2 + {a'}_{i,3}^2$.
- $B_3 = \displaystyle\sum_{i=1}^{n} {a'}_{i,1}^3 + {a'}_{i,2}^3 + {a'}_{i,3}^3$.

Note that ${a'}_{i,1}$ and ${b'}_{i,1}$ are positive integers defined as in the proof of Theroem 4 and their Σ-ary representations are identical to those illustrated in Figure 2. This reduction runs in time $\mathrm{poly}(n, m, h, \nu)$.

c) Completeness: The completeness analysis uses the next proposition whose proof appears in the full version.

Proposition 1. *For every $i \in [n]$, the quantities ${a'}_{i,2}, {a'}_{i,3}, {b'}_{i,2}, {b'}_{i,3}$ defined above satisfy*

$$ {a'}_{i,2} + {a'}_{i,3} = {b'}_{i,2} + {b'}_{i,3} $$

$$ {a'}_{i,1}^2 + {a'}_{i,2}^2 + {a'}_{i,3}^2 = {b'}_{i,1}^2 + {b'}_{i,2}^2 + {b'}_{i,3}^2 $$

$$ {a'}_{i,1}^3 + {a'}_{i,2}^3 + {a'}_{i,3}^3 = {b'}_{i,1}^3 + {b'}_{i,2}^3 + {b'}_{i,3}^3 $$

Assume that the starting 3−SAT formula has a satisfying 1-in-3-SAT assignment x. Consider the subset $S = \cup_{i\in[n]:x_i=1}\{a'_{i,1}, a'_{i,2}, a'_{i,3}\} \cup \cup_{i\in[n]:x_i=0}\{b'_{i,1}, b'_{i,2}, b'_{i,3}\}$ of size $k = 3n$. In the full version, we use Proposition 1 to check that the first 3 moments of S equal B_1, B_2 and B_3 respectively.

d) Soundness: Assume that S is a satisfying subset of the MSS(3) instance. We will use S in order to construct a 1-in-3-SAT satisfying assignment to the starting 3-SAT formula ϕ. We first sketch the high-level intuition of the proof. First, we prove using an inductive argument (i.e., Lemma 4) that, for each $i \in [n]$, S should contain the same number of positive and negative rational numbers from the set $\{a'_{i,2}, b'_{i,2}, a'_{i,3}, b'_{i,3}\}$. This is because these four numbers have roughly the same absolute value, which turns out to be much larger than B_1 and than any other possible number in S. Thus, unless S contains the same number of positive and negative elements from $\{a'_{i,2}, b'_{i,2}, a'_{i,3}, b'_{i,3}\}$, the elements of S cannot add up to B_1. Next, we show using another inductive argument (i.e., Lemma 5) that, for every $i \in [n]$, S should contain exactly two elements from the set $\{a'_{i,2}, b'_{i,2}, a'_{i,3}, b'_{i,3}\}$. This is because otherwise, the sum of squares of the elements in $S \cap \{a'_{i,2}, b'_{i,2}, a'_{i,3}, b'_{i,3}\}$ would be very far from $a'^2_{i,2} + a'^2_{i,3}$, which would imply that the sum of squares of the elements in S is very far from B_2. We then show (in Lemma 6) that, in fact, the elements of $S \cap \{a'_{i,2}, b'_{i,2}, a'_{i,3}, b'_{i,3}\}$ *exactly* add up to $2\alpha_i$; namely, the β_i and γ_i terms cancel each other out. Hence, we obtain that the elements of S from $\cup_{i=1}^{n}\{a'_{i,1}, b'_{i,1}\}$ add up to $B_1 - 2\sum_{i\in[n]}\alpha_i = \Sigma^{\nu+m+h-1}(1^n)_{\Sigma^h} + \Sigma^{\nu}(1^m)_{\Sigma}$, at which point we run the soundness analysis of the original reduction from 1-in-3-SAT to Subset Sum to construct an assignment satisfying exactly one literal in each clause. We now give the formal proof which uses the following lemmas whose proofs appear in the full version. We set $\nu = 10(m + nh)$ and $h = 10m$.

Lemma 4. $\forall i \in [n]$, $|S \cap \{a'_{i,2}, b'_{i,2}\}| = |S \cap \{a'_{i,3}, b'_{i,3}\}|$.

Lemma 5. $\forall i \in [n]$, $|S \cap \{a'_{i,2}, b'_{i,2}, a'_{i,3}, b'_{i,3}\}| = 2$.

Lemma 6.
$$\sum_{y\in S\cap\cup_{i=1}^{n}\{a'_{i,2},b'_{i,2},a'_{i,3},b'_{i,3}\}} y = 2\sum_{i=1}^{n}\alpha_i.$$

Lemmas 4, 5 and 6 imply that $\sum_{y\in S\cap\cup_{i=1}^{n}\{a'_{i,1},b'_{i,1}\}} y = \Sigma^{\nu+m+h-1}(1^n)_{\Sigma^h} + \Sigma^{\nu}(1^m)_{\Sigma}$. We now carry out the soundness analysis of the original reduction from 1-in-3-SAT to Subset Sum to construct an assignmnent that satisfies exactly one literal in each clause of the given 3-SAT formula. This concludes our soundness analysis. Thus, setting Σ to be a sufficiently large even constant (say 4) yields a polynomial-time reduction from 1-in-3-SAT to MSS(3). Hence, MSS(3) is NP-hard. The proof also holds over the ring of integers by first multiplying the numbers $\{a'_{i,j}, b'_{i,j}\}$, B_1, B_2, B_3 by the least common multiple of the denominators of $\{\alpha_i, \gamma_i/2, \alpha_i/2\}$. This preserves the moment constraints since they are homogeneous equations. Let $c = O(1)$ and $p > 2^{cn^3}$ be a prime number such that all integers appearing in this proof are $< p/2$. The proof extends to $\mathbb{F}_p$ by performing all computations modulo p. ■

V. Conclusion

In this work, we proved that RS-BDD(2) and RS-BDD(3) are NP-hard. The main open problem raised by our work is to understand the limits of our approach. In particular, can one design explicit constructions satisfying the properties in Question 1, and use them to prove the NP-hardness of RS-BDD(d) for larger values of d? Finally, we remark that a positive answer to Question 1 could yield to a better understanding of the structure of BCH codes over large fields, and to their the local testability properties (e.g., [GK12].)

Acknowledgment

The authors would like to thank Amir Abboud, Swastik Kopparty, Madhu Sudan and Andrew Sutherland for very helpful discussions and pointers.

References

[CM07] Q. Cheng and E. Murray. On deciding deep holes of Reed-Solomon codes. In *TAMC 2007*, pages 296–305, 2007.

[CW10] Q. Cheng and D. Wan. Complexity of decoding positive-rate primitive Reed-Solomon codes. *IEEE Trans. Inf. Theory*, 56(10):5217–5222, 2010.

[GK12] E. Grigorescu and T. Kaufman. Explicit low-weight bases for BCH codes. *IEEE Trans. Inf. Theory*, 58(1):78–81, 2012.

[GS99] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inf. Theory*, 45(6):1757–1767, 1999.

[GV05] V. Guruswami and A. Vardy. Maximum-likelihood decoding of Reed-Solomon codes is NP-hard. *IEEE Trans. Inf. Theory*, 51(7):2249–2256, 2005.

[LW08] J. Li and D. Wan. On the subset sum problem over finite fields. *Finite Fields and Their Applications*, 14(4):911–929, 2008.

[Sch78] T. J. Schaefer. The complexity of satisfiability problems. In *STOC*, pages 216–226. ACM, 1978.

[Sho97] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[Sta99] R P Stanley. *Enumerative Combinatorics, vol.2*. Cambridge University Press, 1999.

[Sud97] M. Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *J. Complexity*, 13(1):180–193, 1997.