NP-Hardness of Reed-Solomon Decoding, and the Prouhet-Tarry-Escott Problem

Venkata Gandikota [*]   Badih Ghazi [†]   Elena Grigorescu [‡]

Abstract

Establishing the complexity of *Bounded Distance Decoding* for Reed-Solomon codes is a fundamental open problem in coding theory, explicitly asked by Guruswami and Vardy (IEEE Trans. Inf. Theory, 2005). The problem is motivated by the large current gap between the regime when it is NP-hard, and the regime when it is efficiently solvable (i.e., the Johnson radius).

We show the first NP-hardness results for asymptotically smaller decoding radii than the maximum likelihood decoding radius of Guruswami and Vardy. Specifically, for Reed-Solomon codes of length N and dimension $K = O(N)$, we show that it is NP-hard to decode more than $N - K - c\frac{\log N}{\log \log N}$ errors (with $c > 0$ an absolute constant). Moreover, we show that the problem is NP-hard under quasipolynomial-time reductions for an error amount $> N - K - c \log N$ (with $c > 0$ an absolute constant).

An alternative natural reformulation of the Bounded Distance Decoding problem for Reed-Solomon codes is as a *Polynomial Reconstruction* problem. In this view, our results show that it is NP-hard to decide whether there exists a degree K polynomial passing through $K + c\frac{\log N}{\log \log N}$ points from a given set of points $(a_1, b_1), (a_2, b_2) \ldots, (a_N, b_N)$. Furthermore, it is NP-hard under quasipolynomial-time reductions to decide whether there is a degree K polynomial passing through $K + c \log N$ many points.

These results follow from the NP-hardness of a generalization of the classical Subset Sum problem to higher moments, called *Moments Subset Sum*, which has been a known open problem, and which may be of independent interest.

We further reveal a strong connection with the well-studied Prouhet-Tarry-Escott problem in Number Theory, which turns out to capture a main barrier in extending our techniques. We believe the Prouhet-Tarry-Escott problem deserves further study in the theoretical computer science community.

Contents

1 Introduction

Despite being a classical problem in the study of error-correcting codes, the computational complexity of decoding Reed-Solomon codes [RS60] in the presence of large amounts of error is not fully understood. In the Bounded Distance Decoding problem, the goal is to recover a message corrupted by a bounded amount of error. Motivated by the large gap between the current efficient decoding regime, and the NP-hard regime for Reed-Solomon codes, we study the NP-hardness of Bounded Distance Decoding for asymptotically smaller error radii than previously known. In this process, we unravel a strong connection with the Prouhet-Tarry-Escott, a famous problem from number theory that has been studied for more than two centuries.

A Reed-Solomon (RS) code of length N, dimension K, defined over a finite field $\mathbb{F}$, is the set of vectors (called *codewords*) corresponding to evaluations of low-degree univariate polynomials on a given set of evaluation points $\mathcal{D} = \{\alpha_1, \alpha_2, \ldots, \alpha_N\} \subseteq \mathbb{F}$. Formally, $RS_{\mathcal{D},K} = \{\langle p(\alpha_1), \ldots, p(\alpha_N)\rangle :$ $p \in \mathbb{F}[x]$ is a univariate polynomial of degree $< K\}$. The Hamming distance between $x, y \in \mathbb{F}^N$ is $\Delta(x,y) := |\{i \in [N] : x_i \neq y_i\}|$. In the *Bounded Distance Decoding (BDD) problem*, given a target vector $y \in \mathbb{F}^N$ and a distance parameter λ, the goal is to output $c \in \mathcal{C}$ such that $\Delta(c,y) \leq \lambda$.

It is well-known that if the number of errors is $\lambda \leq (N-K)/2$, there is a unique codeword within distance λ from the message, which can be found efficiently [Pet60, BW86]. Further, Sudan [Sud97] and Guruswami and Sudan [GS99] show efficient decoding up to $\lambda = N - \sqrt{NK}$ errors (the "Johnson radius"), a setting in which the algorithm may output a small list of possible candidate messages. At the other extreme, if the number of errors is at least $N - K$ (the covering radius), finding one close codeword becomes trivial, amounting to interpolating a degree $K - 1$ polynomial through $\leq K$ points. However, just below that radius, namely at $N - K - 1$ errors, the problem becomes NP-hard, a celebrated result of Guruswami and Vardy [GV05]. The proof approach of [GV05] is only applicable to the Maximum Likelihood Decoding setting of $N - K - 1$ errors, prompting the fundamental problem of understanding the complexity of BDD in the wide remaining range between $N - \sqrt{NK}$ and $N - K - 1$:

> [GV05] *"It is an extremely interesting problem to show hardness of bounded distance decoding of Reed-Solomon codes for smaller decoding radius."*

Some partial progress on improving the NP-hardness regime was shown in a recent result by the same authors [GGG15] for $N - K - 2$ and $N - K - 3$ errors. The only other work addressing the hardness of decoding RS codes are due to Cheng and Wan [CW07, CW10] who show randomized reductions from the Discrete Log problem over finite fields, which is not believed to be NP-hard.

In this work, we study the complexity of the decision version of BDD, where the number of errors is parametrized by $d \geq 0$, as formalized next:

> **Problem** *Bounded Distance Decoding of Reed-Solomon codes with parameter d (*RS-BDD(d)*)*
>
> **Input** $\mathcal{D} = \{\alpha_1, \alpha_2, \ldots, \alpha_N\} \subseteq \mathbb{F}$, where $\alpha_i \neq \alpha_j$ for all $i \neq j$, target $y = (y_1, y_2, \ldots, y_N)$, and integer $K < N$
>
> **Goal** Decide if there exists $p \in RS_{\mathcal{D},K}$ such that $\Delta(y,p) \leq (N - K) - d$

We emphasize that the BDD problem above is in fact the basic and natural Polynomial Reconstruction problem, where the input is a set of points $\mathcal{D} = \{(\alpha_1, y_1), (\alpha_2, y_2), \ldots, (\alpha_N, y_N)\} \subseteq \mathbb{F} \times \mathbb{F}$,

and the goal is to decide if there exists a polynomial p of degree $< K$ that passes through at least $K + d$ points in $\mathcal{D}$.

We state our main result in both forms.

1.1 Contributions

Our main technical contribution is the first NP-hardness result for BDD of RS codes, for a number of errors that is asymptotically smaller than $N - K$, and its alternative view in terms of polynomial reconstruction.

Theorem 1.1. *There exists $c > 0$, such that for every $1 \leq d \leq c \cdot \frac{\log N}{\log \log N}$, the RS-BDD$(d)$ problem for Reed-Solomon codes of length N, dimension $K = N/2 - d + 1$ and field size $|\mathbb{F}| = 2^{\mathsf{poly}(N)}$ is NP-hard. Furthermore, there exists $c > 0$, such that for every $1 \leq d \leq c \cdot \log N$, RS-BDD$(d)$ over fields of size $|\mathbb{F}| = 2^{N^{O(\log N)}}$ does not have $N^{O(\log N)}$-time algorithms unless NP has quasi-polynomial time algorithms.*

Equivalently, there exists $c > 0$, such that for every $1 \leq d \leq c \cdot \frac{\log N}{\log \log N}$, it is NP-hard to decide whether there exists a polynomial of degree $< K = N/2 - d + 1$ passing through $K + d$ many points from a given set $\mathcal{D} = \{(\alpha_1, y_1), (\alpha_2, y_2), \ldots, (\alpha_N, y_N)\} \subseteq \mathbb{F} \times \mathbb{F}$, with $|\mathbb{F}| = 2^{\mathsf{poly}(N)}$. Furthermore, there exists $c > 0$, such that for every $1 \leq d \leq c \cdot \log N$, the same interpolation problem over fields of size $|\mathbb{F}| = 2^{N^{O(\log N)}}$ does not have $N^{O(\log N)}$-time algorithms unless NP has quasi-polynomial time algorithms.

Our results significantly extend [GV05, GGG15], which only show NP-hardness for $d \in \{1, 2, 3\}$. As in [GV05, GGG15], we require the field size to be exponential in N.

The bulk of the proof of Theorem 1.1 is showing the NP-hardness of a natural generalization of the classic Subset Sum problem to higher moments, that may be of independent interest.

> **Problem** *Moments Subset Sum with parameter d, over a field $\mathbb{F}$ (MSS(d))*
>
> **Input** Set $A \subseteq \mathbb{F}$ of size $|A| = N$, integer k, elements $m_1, m_2, \ldots, m_d \in \mathbb{F}$
>
> **Goal** Decide if there exists $S \subseteq A$ such that $\sum_{s \in S} s^\ell = m_\ell$, for all $\ell \in [d]$, and $|S| = k$.

We note that the reduction from MSS(d) to RS-BDD(d) uses the equivalence between elementary symmetric polynomials and moments polynomials, when the field is of characteristic larger than $\Omega(d!)$ (see Lemma 2.2 for a formal reduction.)

We point out that the Moments Subset Sum problem has natural analogs over continuous domains in the form of generalized moment problems and truncated moments problems, which arise frequently in economics, operations research, statistics and probability [Las09].

In this work, we prove NP-hardness of the Moments Subset Sum problem for large degrees.

Theorem 1.2. *There exists $c > 0$, such that for every $1 \leq d \leq c \cdot \frac{\log N}{\log \log N}$, the Moments Subset Sum problem MSS(d) over prime fields of size $|\mathbb{F}| = 2^{\mathsf{poly}(N)}$ is NP-hard. Furthermore, there exists $c > 0$, such that for every $1 \leq d \leq c \cdot \log N$, the Moments Subset Sum problem MSS(d) over fields of size $|\mathbb{F}| = 2^{N^{O(\log N)}}$ does not have $N^{O(\log N)}$-time algorithms unless NP has quasi-polynomial time algorithms.*

Furthermore, we reveal a connection with the famous Prouhet-Tarry-Escott (PTE) problem in Diophantine Analysis, which is the main barrier for extending Theorem 1.2 and Theorem 1.1 to $d = \omega(\log N)$, as explained shortly.

The PTE problem [Pro51, Dic13, Wri59] first appeared in letters between Euler and Goldbach in 1750-1751, and it is an important topic of study in classical number theory (see, e.g., the textbooks of Hardy and Wright [HW79] and Hua [Hua82]). It is also related to other classical problems in number theory, such as variants of the Waring problem and problems about minimizing the norm of cyclotomic polynomials, considered by Erdös and Szekeres [ES59, BI94].

In the Prouhet-Tarry-Escott problem, given $k \geq 1$, the goal is to find disjoint sets of integers $\{x_1, x_2, \ldots, x_t\}$ and $\{y_1, y_2, \ldots, y_t\}$ satisfying the system:

$$
\begin{aligned}
x_1 + x_2 + \cdots + x_t &= y_1 + y_2 + \cdots + y_t \\
x_1^2 + x_2^2 + \cdots + x_t^2 &= y_1^2 + y_2^2 + \cdots + y_t^2 \\
&\cdots \\
x_1^k + x_2^k + \cdots + x_t^k &= y_1^k + y_2^k + \cdots + y_t^k.
\end{aligned}
$$

We call t the size of the PTE solution. It turns out that the completeness proof of our reduction in Theorem 1.2 relies on *explicit* solutions to this system for degree $k = d$ and of size $t = 2^k$. As explained next, despite significant efforts that have been devoted to constructing PTE solutions during the last 100 years, no explicit solutions of size $t = o(2^k)$ are known. This constitutes the main barrier to extending our Theorem 1.2 and Theorem 1.1 to $d = \omega(\log N)$.

The main open problem that has been tackled in the PTE literature is constructing solutions of small size t compared to the degree k. It is relatively easy to show that $t \geq k+1$, and straightforward (yet non-constructive!) pigeon-hole counting arguments show the existence of solutions with $t = O(k^2)$. If we further impose the constraint that the system is not satisfied for degree $k + 1$ (which is a necessary constraint for our purposes), then solutions of size $t = O(k^2 \log k)$ are known to exist [Hua82]. However, these results are non-constructive, and the only general explicit solutions have size $t = O(2^k)$ (e.g., [Wri59, BI94]). A special class of solutions studied in the literature is for $t = k+1$ (of minimum possible size). Currently there are known explicit parametric constructions of infinitely many minimum-size solutions for $k \leq 12$ (e.g., [BI94, BLP03]), and finding such solutions often involves numerical simulations and extensive computer-aided searches [BLP03].

From a computational point of view, an important open problem is to understand whether PTE solutions of size $O(k^2)$ (which are known to exist) can be *efficiently constructed*, i.e., in time $\mathsf{poly}(k)$.

We identify the following generalization of the PTE problem as a current barrier to extending our results:

Problem 1.3. *Given a field $\mathbb{F}$, integer d, and $a, b \in \mathbb{F}$, efficiently construct $x_1, \ldots, x_t, y_1, \ldots, y_t \in \mathbb{F}$, with $t = o(2^d)$, satisfying:*

$$
x_1 + x_2 + \cdots + x_t = y_1 + y_2 + \cdots + y_t
$$

$$
a^i + \sum_{j=1}^{t} x_j^i = b^i + \sum_{j=1}^{t} y_j^i \quad \forall i \in \{2, \ldots, d\}
$$

We believe that this question is worth further study in the theoretical computer science community. In this work, we prove the following theorem, which is at the core of the completeness of our reduction.

Theorem 1.4. *There is an explicit construction of solutions for Problem 1.3 with $t = O(2^d)$, and which can be computed in time* $\mathsf{poly}(t)$.

In the next section, we outline the proof of Theorem 1.2, and in the process, we explain how PTE solutions of degree d naturally arise when studying the computational complexity of $\mathrm{MSS}(d)$.

1.2 Proof Overview

To prove Theorem 1.2, we begin with the classical reduction from 1-in-3-SAT to Subset-Sum, in which one needs to construct a set of integers such that there is a subset whose sum equals a given target m_1, if and only if there is an assignment that satisfies exactly one literal of each clause of the 3-SAT formula (we refer the reader to Section 3 for more details about this standard reduction). Extending this reduction so that the 2nd moment also hits target m_2 raises immediate technical hurdles, since we have very little handle on the extra moment. In [GGG15], the authors manage to handle a reduction for 2nd and 3rd moments via ad-hoc arguments and identities tailored to the degree-2 and degree-3 cases. The problem becomes much more complex as we need to ensure both completeness and soundness for a large number of moments. In this work, we achieve such a reduction where the completeness will rely on explicit solutions to "inhomogeneous PTE instances" and the soundness will rely on a delicate balancing of the magnitudes of these explicit solutions. We now describe the details of this reduction.

For each 1-in-3-SAT variable, we create a collection of *explicit* auxiliary numbers which "stabilize" the contribution of this variable to all i-th moment equations with $2 \leq i \leq d$, while having no net effect on the 1st moment equation. Concretely, if a and b are the numbers corresponding to the two literals of the given variable, then we need to find numbers $x_1, \ldots, x_t, y_1, \ldots, y_t$ satisfying:

$$x_1 + x_2 + \cdots + x_t = y_1 + y_2 + \cdots + y_t$$

$$a^i + \sum_{j=1}^{t} x_j^i = b^i + \sum_{j=1}^{t} y_j^i \quad \forall i \in \{2, \ldots, d\} \tag{†}$$

Note that in order for the overall reduction to run in polynomial-time, the above auxiliary variables should be *efficiently constructible*. Moreover, we observe that (†) is an inhomogeneous PTE instance: for $a = b$, it reduces to a PTE instance of degree d. Of course, in our case a and b will not be equal, and (†) is a more general system (and is hence harder to solve) than PTE instances. Nevertheless, as we will see shortly, solving (†) can be essentially reduced to finding explicit PTE solutions of degrees $k \leq d$.

In addition, we need to ensure that the added auxiliary numbers satisfy some "bimodality" property regarding their magnitudes, which would allow the recovery of a satisfying 1-in-3-SAT assignment from any solution to the $\mathrm{MSS}(d)$ instance:

Property 1.5 (Bimodality (informal))**.** *Every subset S of the auxiliary variables is such that either $|\sum_{s \in S} s|$ is tiny, or $|\sum_{s \in S} s|$ is huge.*

We note that the existence of explicit and efficiently constructible solutions of small size $t = O(d)$ to system (†) (and hence to a PTE system too) would at least ensure the completeness of a reduction with $d = O(N)$. If soundness can also be ensured for such solutions, then our techniques would extend to radii closer to the Johnson Bound radius.

4

Overview of procedure for solving system (†) We build the variables x_i and y_i recursively, by reducing the construction for degree i to a solution to degree $i - 1$. Towards this goal, we design a sub-procedure, called ATOMICSOLVER, that takes as inputs an integer $i \in \{2, 3, \ldots, d\}$, and a number R_i, and outputs 2^i rational[1] numbers $\{x_{i,j}, y_{i,j}\}_{j \in [2^{i-1}]}$ that satisfy a PTE system of degree $i - 1$, along with a non-homogeneous equation of degree i:

$$\sum_{\ell=1}^{2^{i-1}} (x_{i,\ell}^j - y_{i,\ell}^j) = 0 \quad \forall \, 2 \le j < i, \tag{2a}$$

$$\sum_{\ell=1}^{2^{i-1}} (x_{i,\ell}^i - y_{i,\ell}^i) = R_i. \tag{2b}$$

We can then run ATOMICSOLVER sequentially on inputs $i \in \{2, \ldots, d\}$ with the R_i input corresponding to a "residual" term that accounts for the contributions to the degree-i equation of the outputs of ATOMICSOLVER(j, R_j) for all $2 \le j < i$, namely,

$$R_i = b^i - a^i + \sum_{2 \le j < i} \sum_{\ell=1}^{2^{j-1}} (y_{j,\ell}^i - x_{j,\ell}^i). \tag{3}$$

Note that the aim of the ATOMICSOLVER(i, R_i) procedure is to satisfy the degree-i equation (2b) without affecting the lower-degree equations (2a).

We then argue that the union $\cup_{2 \le i \le d} \{x_{i,j}, y_{i,j}\}_{j \in [2^{i-1}]}$ of all output variables satisfies the polynomial constraints in (†) with $t = \exp(d)$.

Specifics of the ATOMICSOLVER We next illustrate the ATOMICSOLVER procedure by describing its operation in the particular case where $i = d = 4$. In what follows, we drop "$i = 4$ subscripts" and denote $R = R_4$, $x_\ell = x_{4,\ell}$ and $y_\ell = y_{4,\ell}$ for all $1 \le \ell \le 8$. Then, Equation (2b) above that we need to satisfy becomes

$$\sum_{\ell=1}^{8} (x_\ell^4 - y_\ell^4) = R. \tag{4}$$

First, we let α be a constant parameter (to be specified later on) and we set

$$x_1 - y_1 = \alpha \tag{5a}$$

$$y_2 - x_2 = \alpha \tag{5b}$$

Namely, in Equations (5a) and (5b), we "*couple*" the ordered pairs (x_1, y_1) and (y_2, x_2) in the same way. Then, using Equations (5a) and (5b), we substitute $y_1 = x_1 - \alpha$ and $x_2 = y_2 - \alpha$, and the sum of the $\ell = 1$ and $\ell = 2$ terms in Equation (4) can be written as

$$(x_1^4 - y_1^4) - (y_2^4 - x_2^4) = p_\alpha(x_1) - p_\alpha(y_2) \tag{6}$$

where p_α is a *cubic* polynomial. If we set $x_1 - y_2 = \beta$, then (6) further simplifies to

$$p_\alpha(x_1) - p_\alpha(y_2) = q_{\alpha,\beta}(x_1) \tag{7}$$

---

[1]In our case, we can afford having *rational* solutions to Equations (2a) and (2b). Note that this system is still a generalization of the PTE problem since we can always scale the rational solutions by their least common denominator to get a PTE solution of degree $i - 1$.

where $q_{\alpha,\beta}$ is a *quadratic* polynomial[2].

In the next step, we couple the ordered tuple (y_3, x_3, y_4, x_4) in the same way that we have so far coupled the tuple (x_1, y_1, x_2, y_2). The sum of the first four terms in the LHS of (4) then becomes

$$\sum_{\ell=1}^{4}(x_\ell^4 - y_\ell^4) = (x_1^4 - y_1^4 + x_2^4 - y_2^4) - (y_3^4 - x_3^4 + y_4^4 - x_4^4) \tag{8}$$
$$= q_{\alpha,\beta}(x_1) - q_{\alpha,\beta}(y_3).$$

As before, we set $x_1 - y_3 = \gamma$ and (8) further simplifies to

$$q_{\alpha,\beta}(x_1) - q_{\alpha,\beta}(y_3) = w_{\alpha,\beta,\gamma}(x_1) \tag{9}$$

where $w_{\alpha,\beta,\gamma}(x_1)$ is a *linear* polynomial in x_1. Finally, we couple the ordered tuple $(y_5, x_5, y_6, x_6, y_7, x_7, y_8, x_8)$ in the same way that we have so far coupled the tuple $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, and we obtain that the following equation is equivalent to Equation (4) above:

$$w_{\alpha,\beta,\gamma}(x_1) - w_{\alpha,\beta,\gamma}(y_5) = R. \tag{10}$$

Setting $x_1 - y_5 = \theta$, Equation (10) further simplifies to

$$\theta \cdot h_{\alpha,\beta,\gamma} = R, \tag{11}$$

where $h_{\alpha,\beta,\gamma}$ is the coefficient of x_1 in the linear polynomial $w_{\alpha,\beta,\gamma}(x_1)$. We conclude that to satisfy (4), it suffices to choose α, β, γ such that $h_{\alpha,\beta,\gamma} \neq 0$, and to then set $\theta = R/h_{\alpha,\beta,\gamma}$.

It is easy to see that there exist α, β, γ such that $h_{\gamma,\beta,\alpha} \neq 0$, and that the above recursive coupling of the variables guarantees that (2a) is satisfied. The more difficult part will be to choose α, β, γ in a way that ensures the soundness of the reduction. This is briefly described next.

Bimodality of solutions In the above description of the particular case where $i = d = 4$, it can be seen that the produced solutions are $\{0, \pm 1\}$-linear combinations of $\{\alpha, \beta, \gamma, \theta\}$, which are required to satisfy (11). It turns out that in this case $h_{\alpha,\beta,\gamma} = 24 \cdot \alpha \cdot \beta \cdot \gamma$, and so (11) becomes

$$\theta \cdot \alpha \cdot \beta \cdot \gamma = \frac{R}{24}. \tag{12}$$

So assuming we can upper bound $|R|$,[3] we would be able to set θ to a sufficiently large power of 10 while letting α, β and γ to have tiny absolute values and satisfy (12). Using the fact that the auxiliary x_i and y_i variables are set to $\{0, \pm 1\}$-linear combinations of $\{\alpha, \beta, \gamma, \theta\}$, this implies that the bimodality property is satisfied. In Section 3, we show that the bimodality property ensures that in any feasible solution to MSS(d), the auxiliary variables should have no net contribution to the degree-1 moment equation (Proposition 3.4), which then implies the soundness of the reduction.

---

[2]Intuitively, we can think the LHS of (7) (along with the setting $x_1 - y_2 = \beta$) as being a *"derivative operator"*. This explains the fact that we are starting from a cubic polynomial $p_\alpha(\cdot)$ and getting a quadratic polynomial $q_{\alpha,\beta}(\cdot)$. This intuition was also used (twice) in (6), and will be again used in (9) and (10) in order to reduce the degree further.

[3]which we will do by inductively upper bounding $|R_i|$.

General finite fields We remark that as described above, our solution works over the rational numbers, and, by scaling appropriately, over the integers. By taking the integer solution modulo a large prime p (i.e., $p = 2^{\mathsf{poly}(N)}$) the same arguments extend to $\mathbb{F}_p$. Moving to general finite fields $\mathbb{F} = \mathbb{F}_{p^\ell}$, we first observe that system (†) (and thus a PTE system too) has non-constructive solutions of size $O(d)$, which follows from the Weil bound (see Section 6). Our reduction in the proof of Theorem 1.2 also extends to general fields $\mathbb{F} = \mathbb{F}_{p^\ell}$, where p is a prime $p = \Omega(d!)$, and $\ell = \mathsf{poly}(N, d!)$. The reduction now uses a representation of field elements in a polynomial basis $\{1, \gamma, \gamma^2, \ldots, \gamma^{\ell-1}\} \subseteq \mathbb{F}$, instead of decimal representations. See Section 7 for the changes that need to be made to the proof over the integers.

1.3 Related Work

A number of fundamental works address the polynomial reconstruction problem in various settings. In particular, Goldreich et al. [GRS00] show that that the polynomial reconstruction problem is NP-complete for univariate polynomials p over large fields. Håstad's celebrated results [Hås01] imply NP-hardness for linear multivariate polynomials over finite fields. Gopalan et al. [GKS10] show NP-hardness for multivariate polynomials of larger degree, over the field $\mathbb{F}_2$.

We note that in general, the polynomial reconstruction problem does not require that the evaluation points are all distinct (i.e., $x_i \neq x_j$ whenever $i \neq j$). This distinction is crucial to the previous results on polynomial reconstruction (eg. [GRS00, GKS10]). It is this distinction that prevents those results from extending to the setting of Reed-Solomon codes, and to their multivariate generalization, Reed-Muller codes.

On the algorithmic side, efficient algorithms for decoding of Reed-Solomon codes and their variants are well-studied. As previously mentioned, [Sud97, GS99] gave the first efficient algorithms in the list-decoding regime. Parvaresh and Vardy [PV05] and Guruswami and Rudra [GR08] construct capacity achieving codes based on variants of RS codes. Koetter and Vardy [KV03] propose soft decision decoders for RS codes. More recently, Rudra and Wooters [RW14] prove polynomial list-bounds for random RS codes.

A related line of work is the study of BDD and of Maximum Likelihood Decoding in general codes, possibly under randomized reductions, and when an unlimited amount of preprocessing of the code is allowed. These problems have been extensively studied under diverse settings, e.g., [Var97, ABSS97, DKRS03, DMS03, FM04, Reg04, GV05, Che08].

2 Preliminaries

We start by recalling the formal definition of the MSS(d) problem.

Definition 2.1 (Moments Subset-Sum: MSS(d))**.** *Given a set $A = \{a_1, \ldots, a_N\}$, $a_i \in \mathbb{F}$, integer k, and $m_1, \ldots, m_d \in \mathbb{F}$, decide if there exists a subset $S \subseteq A$ of size k, satisfying $\sum_{a \in S} a^i = m_i$ for all $i \in [d]$. We call k the size of the MSS(d) instance.*

We next recall the reduction from MSS(d) to RS-BDD(d).

Lemma 2.2 ([GGG15])**.** MSS(d) *is polynomial-time reducible to RS-BDD(d). Moreover, the reduction maps instances of MSS(d) on N numbers and of size k to Reed-Solomon codes of block length $N + 1$ and of dimension $k - d + 1$. The reduction holds over finite fields $\mathbb{F}$ of large characteristic.*

The reduction proceeds via SSS(d), a problem which is equivalent to MSS(d) over large fields.

Definition 2.3 (Symmetric Subset-Sum (SSS(d))). *Given a set of N distinct elements of $\mathbb{F}$, $A = \{a_1, a_2, \ldots, a_N\}$, integer k, and $E_1, E_2, \ldots E_d \in \mathbb{F}$, decide if there exists a subset $S \subseteq A$ of size k, such that for every $i \in [d]$ the elementary symmetric sums of the elements of $S = \{s_1, \ldots, s_k\}$ satisfy $E_i(S) = \sum_{1 \le j_1 < j_2 < \cdots < j_i \le k} s_{j_1} \ldots s_{j_i} = E_i$.*

Given an instance $\langle A, k, E_1, E_2, \ldots, E_d \rangle$ of SSS(d), we construct an instance $\langle \mathcal{D}, y, K \rangle$ of RS-BDD(d) such that there exists a Reed-Solomon codeword $p \in RS_{\mathcal{D},K}$ with $\Delta(y, p) \le (N - K) - d$ if and only if there is a solution to the given instance of SSS(d). SSS(d) can be easily seen to be equivalent to MSS(d) over large prime finite fields $\mathbb{F}$ using Newton's identities [Sta99], which will complete the proof of Lemma 2.2. We note that this connection has been previously made (e.g. [LW08]).

Lemma 2.4. SSS(d) *is polynomial-time reducible to* RS-BDD(d).

Proof. Given an instance $\langle A, k, E_1, E_2, \ldots, E_d \rangle$ of SSS(d), we construct an instance $\langle \mathcal{D}, y, K \rangle$ of RS-BDD(d) such that there exists a Reed-Solomon codeword $p \in RS_{\mathcal{D},K}$ with $\Delta(y, p) \le (N - K) - d$ if and only if there is a solution to the given instance of SSS(d). Here, $A = \{a_1, a_2, \ldots, a_N\}$ is a set of distinct, non-zero elements of $\mathbb{F}$, $E_1, E_2, \ldots, E_d \in \mathbb{F}$ and $k \in \mathbb{Z}$.

Let $K = k - d + 1$. Define the degree d polynomial $f(x) := x^d - E_1 x^{d-1} + \cdots + (-1)^{d-1} E_{d-1} x$. For each a_i of A, define an element $y_i \in \mathbb{F}$ as $y_i = -f(a_i)$. Define the target vector $y = (y_1, \cdots, y_N, (-1)^d E_d)$. The set $\mathcal{D}$ is then given by $\mathcal{D} = \{a_1^{-1}, \cdots, a_N^{-1}, 0\}$ Note that $\langle \mathcal{D}, y, K \rangle$ is an instance of RS-BDD(d) which can be constructed in polynomial time given the instance $\langle A, k, E_1, \ldots, E_d \rangle$ of SSS(d). Let $D = \{(a_i^{-1}, y_i)$ for all $a_i \in A\} \cup \{(0, (-1)^d E_d)\}$. Note that a Reed-Solomon codeword $p \in RS_{\mathcal{D},K}$ at a distance $(N - K) - d$ from y corresponds to a univariate polynomial $p(x)$ of degree at most $K - 1$ which agrees with D in $K + d$ points.

Let S be the solution to SSS(d). We now show that there exists a polynomial of degree at most $k - d$ $(= K - 1)$ which agrees with D in at least $k + 1$ $(= K + d)$ points. Define the following degree k polynomial,

$$g(x) := \prod_{a_i \in S} (x - a_i) = c_0 + c_1 x + \cdots + c_{k-1} x^{k-1} + x^k$$

The coefficients of this polynomial are the symmetric sums of the roots of $g(x)$. Therefore, $c_{k-d} = (-1)^d E_d, \ldots, c_{k-2} = E_2$, and $c_{k-1} = -E_1$. Now define,

$$
\begin{aligned}
p(x) &= (x^k g(1/x) - x^d f(1/x))/x^d \\
&= c_0 x^{k-d} + c_1 x^{k-d-1} + \cdots + c_{k-d}
\end{aligned}
$$

and note that $p(x)$ has degree $k - d$. We point out that $g(1/x)$ refers to the rational function obtained by replacing x by $1/x$ in the polynomial $g(x)$. Also, the constant term of this polynomial is $c_{k-d} = (-1)^d E_d$. Hence, $p(0) = (-1)^d E_d$ and since $g(a_i) = 0$, for all $a_i \in S$, it follows that $p(a_i^{-1}) = -f(a_i) = y_i$ for all $a_i \in S$. Therefore, $p(x)$ agrees with $k + 1$ points in D.

Conversely, we now show that if there is a polynomial $p(x)$ of degree at most $K - 1$ $(= k - d)$ which agrees with $K + d$ $(= k + 1)$ points in D, then there is a solution to SSS(d). We first observe

that if a degree $k - d$ polynomial passes through $k + 1$ points of D, then it has to pass through $(0, (-1)^d E_d)$. To show this, assume $p(x)$ agrees with $k + 1$ points of the form $(a_i^{-1}, y_i) \in D$. Let $g(x)$ be a degree k polynomial defined as,

$$g(x) = x^{k-d}(p(1/x) + f(x))$$

Therefore, if $p(x) = c_0 + c_1 x + \cdots + c_{k-d} x^{k-d}$, $g(x)$ can be written as

$$g(x) = x^k + E_1 x^{k-1} + \cdots + (-1)^{d-1} E_{d-1} x^{k-d+1} + c_0 x^{k-d} + c_1 x^{k-d-1} + \cdots + c_{k-d}$$

If $p(a_i^{-1}) = y_i = -f(a_i)$ for $k + 1$ points, we have by definition that $g(a_i) = 0$ for these $k + 1$ $a_i's$. This is a contradiction since $g(x)$ has degree at most k and it cannot have $k + 1$ roots. Therefore, $p(0) = c_0 = (-1)^d E_d$. Also, $g(x)$ has k roots which have their first d symmetric sums equal to $E_1, E_2, \ldots, E_d$ respectively. Hence, there exists a solution to the given instance of SSS(d). $\square$

Given an instance $\langle A, k, B_1, \cdots, B_d \rangle$ of MSS(d), we can construct an instance $\langle A, k, E_1, \cdots, E_d \rangle$ of SSS(d) by setting

$$E_j = \frac{1}{j!} \begin{vmatrix} B_1 & 1 & 0 & \cdots & \\ B_2 & B_1 & 2 & 0 & \cdots \\ \vdots & & \ddots & \ddots & \\ B_{j-1} & B_{j-2} & \cdots & B_1 & j-1 \\ B_j & B_{j-1} & \cdots & B_2 & B_1 \end{vmatrix} \quad \text{for every } j \in [d].$$

The reduction from MSS(d) to SSS(d) then follows from Newton's identities. Note that this reduction from from MSS(d) to SSS(d) holds over finite fields $\mathbb{F}$ if $(j!)^{-1} \in \mathbb{F}$.

We will use 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 $z \in \{0, 1\}^n$ satisfying exactly one literal in each clause. It is known that this problem is NP-hard even for $m = O(n)$ [Sch78]. We will use $[n]$ to denote the set $\{1, 2, \ldots, n\}$.

3   Reduction from 1-in-3-SAT to MSS(d)

We start proving Theorem 1.2 by describing the reduction from from 1-in-3-SAT to MSS(d) and its properties. Henceforth, we denote by 1^ℓ the concatenation of ℓ ones, and we let $(1^\ell)_{10}$ denote the positive integer whose decimal representation is 1^ℓ.

Subset Sum Reduction   We start by recalling the reduction from 1-in-3-SAT to Subset-Sum which will be used in our reduction to MSS(d). In that reduction, each variable $(z_t, \overline{z_t})$, $t \in [n]$ is mapped to 2 integers a_t' (corresponding to z_t) and b_t' (corresponding to $\overline{z_t}$). The integers a_t' and b_t' and the target B have the following decimal representation of length-$(n + m)$:

- The decimal representations of a_t' and b_t' consist of two 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_t' and b_t' have a 1 at the t-th digit and 0's at the other digits. Denote that by $(a_t)^{'v}$.

- In the clause region, for every $j \in [m]$, a'_t (resp. b'_t) has a 1 at the jth location if z_t (resp. $\overline{z_t}$) appears in clause j, and a 0 otherwise. We denote the clause part of a'_t by $(a_t)'^c$.

- We define $a'_t = 10^m a'^v_t + a'^c_t$. We define b'_t similarly.

- The target B is set to the integer whose decimal representation is the all 1's, i.e., we set $B = 10^m (1^n)_{10} + (1^m)_{10}$.

See Figure 1 for an illustration of the decimal representations. This reduction to Subset-Sum is complete and sound. Indeed given a satisfying assignment to the 3-SAT formula $\phi(z)$, the subset $S = \{a'_t \mid t \in [n], z_t = 1\} \cup \{b'_t \mid t \in [n], z_t = 0\}$ is seen to satisfy that $\sum_{s \in S} s = \sum_{\substack{t \in [n] \\ z_t = 1}} a'_t + \sum_{\substack{t \in [n] \\ z_t = 0}} b'_t = B$.

Conversely, given a subset $S \subseteq \{a'_t, b'_t \mid t \in [n]\}$ such that $\sum_{s \in S} s = B$, a satisfying assignment to $\phi(z)$ is constructed from it by setting $z_i = 1$ if $a'_t \in S$ and 0 otherwise.

Figure 1: Decimal representations in the original reduction from 1-in-3-SAT to Subset-Sum.

Our Reduction from 1-in-3-SAT to $\mathrm{MSS}(d)$ An instance of $\mathrm{MSS}(d)$ consists of a tuple $\langle A, k, B_1, \ldots, B_d \rangle$. In this reduction, each variable $(z_t, \overline{z_t})$ is mapped to $2^{d+1} - 2$ distinct rationals: $\{a_t\} \cup \{x_{t,i} \mid i \in [2^d - 2]\}$ (corresponding to z_t) and $\{b_t\} \cup \{y_{t,i} \mid i \in [2^d - 2]\}$ (corresponding to $\overline{z_t}$). Let $\{a'_t, b'_t : t \in [n]\}$ be the integers produced by the above reduction to Subset-Sum. We denote by a'^v_t (resp. a'^c_t) the variable (resp. clause) region of a'_t. Let ν be a natural number to be specified later on. Define:

$$
\begin{aligned}
a_t &:= 10^\nu (10^m a'^v_t + a'^c_t) \quad \text{and,} \\
b_t &:= 10^\nu (10^m b'^v_t + b'^c_t).
\end{aligned}
\tag{13}
$$

For each $t \in [n]$, we will explicitly construct two sets of $2^d - 2$ *auxiliary variables*, $X_t = \{x_{t,i} \mid i \in [2^d - 2]\}$ and $Y_t = \{y_{t,i} \mid i \in [2^d - 2]\}$ which satisfy the following properties:

Property (1): $\displaystyle\sum_{x \in X_t} x = \sum_{y \in Y_t} y = 0$.

Property (2): $\displaystyle\sum_{x \in X_t} x^k - \sum_{y \in Y_t} y^k = b_t^k - a_t^k$ for every $k \in \{2, \ldots, d\}$.

Property (3): For any subset $S \subseteq \bigcup_{t \in [n]} (X_t \cup Y_t)$, either $\left| \sum_{s \in S} s \right| > 10^{m+2n+\nu}$ or $\left| \sum_{s \in S} s \right| < 10^\nu$.

10

Property (4): Every rational number of $\bigcup_{t\in[n]}(X_t\cup Y_t)$ can be written as a fraction whose numerator and denominator are integers of magnitudes at most $10^{\mathsf{poly}(n,d!)}$. Moreover,

$$\left|\bigcup_{t\in[n]}(X_t\cup Y_t)\right| = n\cdot(2^{d+1}-4).$$

Properties (1) and (2) will be used to ensure completeness, Property (3) will be used to ensure soundness, and Property (4) will guarantee the polynomial running-time. Constructing such auxiliary variables forms the crux of the reduction.

Define the set $A = \bigcup_{t\in[n]}(\{a_t\}\cup\{b_t\}\cup X_t\cup Y_t)$. We will observe that $|A| = n(2^{d+1}-2)$ by showing that all the variables $\{a_t\},\{b_t\}$ and those in X_t and Y_t for $t\in[n]$ are distinct.

Let $N = |A| = n(2^{d+1}-2), k = \frac{N}{2}$. The targets $B_1,\ldots,B_d$ are defined as follows:

$$B_1 := 10^\nu(10^m(1^n)_{10} + (1^m)_{10}),$$

$$B_j := \sum_{t=1}^n a_t^j + \sum_{t=1}^n \sum_{x\in X_t} x^j \text{ for every } j\in\{2,\ldots,d\}. \tag{14}$$

Note that a_t (and b_t and B_1, respectively) defined above are obtained by inserting ν zeros to the right of the decimal representation of a_t' (resp. b_t' and B). Therefore, $a_t = 10^\nu\cdot a_t'$. Similarly, $b_t = 10^\nu\cdot b_t'$ and $B_1 = 10^\nu\cdot B$ (see Figure 2 for a pictorial illustration). The following fact is immediate from the definitions,

Fact 3.1. *For any $x\in\{a_t,b_t \mid t\in[n]\}\cup B_1$, we have*

$$10^\nu < |x| < 10^{m+n+\nu+1}$$
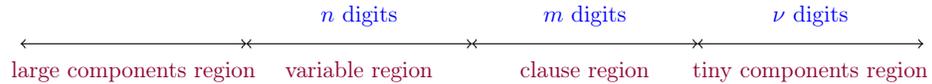
Figure 2: Decimal representations in the reduction from 1-in-3-SAT to MSS(d). The "large components region" only contains zeros in $\{a_t,b_t : t\in[n]\}$ but contains non-zeros in $\{|x_{t,i}|,|y_{t,i}| : t\in[n], i\in[2^d-2]\}$.

The following lemma is proved using Property (4) (and its proof appears in Section 4).

Lemma 3.2. *For any integer d, the total number of variables in the instance of MSS(d) is $N = n\cdot(2^{d+1}-2)$ and every variable has a $\mathsf{poly}(n,d!)$ digit representation in base 10.*

In Section 3.1, we will show how to construct variables satisfying Properties (1), (2), (3) and (4). The proof of Theorem 1.2 will follow from the next lemma and Lemma 3.2. The proof of Theorem 1.1 will then follow from Theorem 1.2 and Lemma 2.2.

Lemma 3.3. *(Main) There exists a satisfying assignment to a 3-SAT instance $\phi(z_1,\ldots,z_n)$ if and only if there exists a subset $S\subseteq A$ of size $|S| = n(2^d-1)$ such that for every $k\in[d]$,*

$$\sum_{s\in S}s^k = B_k.$$

11

Proof of Theorem 1.2. Recall that $N = n(2^{d+1} - 2)$, and so $|S| = |A|/2 = N/2$. From Lemma 3.2 above, we know that every element constructed in the instance of $\mathrm{MSS}(d)$ has $\mathsf{poly}(n, d!)$ digit representation. Therefore, for $d = O(\log n / \log \log n)$, the reduction runs in $\mathsf{poly}(n)$ time.

Let $c > 0$ be a sufficiently small absolute constant. The NP-hardness of $\mathrm{MSS}(d)$ for $d < c \log N / \log \log N$ (under polynomial-time reductions) and for $d < c \log N$ (under quasipolynomial time reductions) over the field of rationals then follows from Lemma 3.3.

By Lemma 3.2 above, we deduce the same hardness results for $\mathrm{MSS}(d)$ over prime fields of size $2^{\mathsf{poly}(N)}$. $\qquad\square$

We now prove Lemma 3.3.

Proof of Lemma 3.3. We start by proving the completeness of our reduction. We show that given a satisfying assignment z to the 3-SAT instance $\phi(z_1, \ldots, z_n)$, there exists a subset $S \subseteq A$ such that for every $k \in [d]$,

$$\sum_{s \in S} s^k = B_k.$$

Consider the following subset S of variables:

$$S \triangleq \bigcup_{t \in [n], z_t = 1} \{a_t\} \bigcup_{t \in [n], z_t = 1} X_t \bigcup_{t \in [n], z_t = 0} \{b_t\} \bigcup_{t \in [n], z_t = 0} Y_t.$$

Note that $|S| = n(2^d - 1) = \frac{N}{2}$ since the number of auxiliary variables included in S corresponding to each $t \in [n]$ is exactly $2^d - 2$.

For every $k \in [d]$, we have that

$$\sum_{s \in S} s^k = \sum_{\substack{t \in [n] \\ z_t = 1}} \left(a_t^k + \sum_{x \in X_t} x^k \right) + \sum_{\substack{t \in [n] \\ z_t = 0}} \left(b_t^k + \sum_{y \in Y_t} y^k \right) \tag{15}$$

By Property (2) of the auxiliary variables, we have that for any $t \in [n]$ and any $k \in \{2, 3, \ldots, d\}$,

$$\sum_{x \in X_t} x^k - \sum_{y \in Y_t} y^k = b_t^k - a_t^k.$$

Summing this equation over all $t \in [n]$, such that $z_t = 0$, we get

$$\sum_{\substack{t \in [n] \\ z_t = 0}} \left(b_t^k + \sum_{y \in Y_t} y^k \right) = \sum_{\substack{t \in [n] \\ z_t = 0}} \left(a_t^k + \sum_{x \in X_t} x^k \right) \tag{16}$$

From 15 and 16, we conclude that for every $k \in \{2, 3, \ldots, d\}$,

$$\sum_{s \in S} s^k = \sum_{t=1}^{n} \left(a_t^k + \sum_{x \in X_t} x^k \right) = B_k$$

For $k = 1$, Property (1) implies that for every $t \in [n]$, $\sum_{x \in X_t} x = 0$ and $\sum_{y \in Y_t} y = 0$. Therefore,

$$\sum_{s \in S} s = \sum_{\substack{t \in [n] \\ z_t = 1}} a_t + \sum_{\substack{t \in [n] \\ z_t = 0}} b_t \tag{17}$$

Recall the variables a'_t, b'_t and B from the Subset Sum reduction defined at the beginning of the proof. Note that $(\sum_{\substack{t \in [n] \\ z_t = 1}} a'_t + \sum_{\substack{t \in [n] \\ z_t = 0}} b'_t) = B$. Therefore, we can rewrite Equation (17) as:

$$\sum_{s \in S} s = 10^\nu \cdot \left(\sum_{\substack{t \in [n] \\ z_t = 1}} a'_t + \sum_{\substack{t \in [n] \\ z_t = 0}} b'_t \right) = 10^\nu \cdot B = B_1.$$

We now prove the soundness of our reduction. Let S be a solution to the MSS(d) instance. That is, $S \subseteq A$ is such that $\sum_{s \in S} s^k = B_k$ for every $k \in [d]$. Proposition 3.4 – which is stated below – shows that the auxiliary variables in S should sum to 0. Therefore, there exists a subset $S' \subseteq \{a_t, b_t \mid t \in [n]\}$ such that $\sum_{s \in S'} s = B_1$. By definition of a_t, b_t and B_1, it follows that there exists a subset of $\{a'_t, b'_t \mid t \in [n]\}$ which sums to B, and the soundness of our reduction then follows from the soundness of the Subset Sum reduction.

Proposition 3.4. *Let $S \subseteq A$ be such that $\sum_{s \in S} s = B_1$. Let $D = \bigcup_{t \in [n]} (X_t \cup Y_t)$ be the set of all the auxiliary variables. Then,*

$$\sum_{y \in S \cap D} y = 0.$$

Proof of Proposition 3.4. Since $\sum_{s \in S} s = B_1$, we have that

$$\sum_{y \in S \cap D} y + \sum_{s \in S \setminus D} s = B_1.$$

Note that $S \setminus D \subseteq \{a_t, b_t \mid t \in [n]\}$. Since the ν least significant digits of B_1 and those of each element of $S \setminus D$ are all equal to 0, either $\left| B_1 - \sum_{s \in S \setminus D} s \right| = 0$ or $\left| B_1 - \sum_{s \in S \setminus D} s \right| > 10^\nu$. If $\left| B_1 - \sum_{s \in S \setminus D} s \right| = 0$, then we are done. Henceforth, we assume that $\left| B_1 - \sum_{s \in S \setminus D} s \right| > 10^\nu$. By Fact 3.1, the elements of $S \setminus D$ as well as B_1 all have magnitudes at most $10^{m+n+\nu+1}$. Therefore, $\left| B_1 - \sum_{s \in S \setminus D} s \right| \leq (2n+1) \cdot 10^{m+n+\nu+1} < 10^{m+2n+\nu}$. On the other hand, by Property (3) of the auxiliary variables,

13

we know that either $\left| \sum_{y \in S \cap D} y \right| > 10^{m+2n+\nu}$ or $\left| \sum_{y \in S \cap D} y \right| < 10^{\nu}$. Since $\left| \sum_{y \in S \cap D} y \right| = \left| B_1 - \sum_{s \in S \backslash D} s \right|$,

we get a contradiction. Therefore, $\sum_{y \in S \cap D} y = 0$.

$\square$

$\square$

3.1   Constructing the auxiliary variables X_t, Y_t

We now show how to construct the auxiliary variables, starting from the a_t, b_t variables described before, for every $t \in [n]$. We do so in Algorithm 1, the AUXILIARYVARIABLEGENERATOR. For every $t \in [n]$, we construct $2(2^d - 2)$ distinct auxiliary variables which satisfy the Properties 1, 2, 3 and 4 stated above. The AUXILIARYVARIABLEGENERATOR outputs the union of the variables generated in Algorithm 2, the ATOMICSOLVER, using the recursive coupling idea described in Section 1.2. We use $\mathbf{1}^{\ell}$ (and $\mathbf{0}^{\ell}$) to denote a column vector of ℓ 1's (0's) respectively. For any vector v, let v^T denote its transpose.

Algorithm 1: AUXILIARYVARIABLEGENERATOR:

Input: $\bigcup_{t \in [n]} \{a_t, b_t\}$

Output: Sets of auxiliary variables X_t, Y_t for every $t \in [n]$.

1: **for** $t \in [n]$ **do**
2:     $X_t = \emptyset$
3:     $Y_t = \emptyset$
4:     **for** $i \in \{2, \ldots, d\}$ **do**
5:         $R_{t,i} = (b_t^i - a_t^i) + \sum_{y \in Y_t} y^i - \sum_{x \in X_t} x^i$
6:         Let $\left\{ x_{t,i,j} \mid j \in [2^{i-1}] \right\} \bigcup \left\{ y_{t,i,j} \mid j \in [2^{i-1}] \right\} = $ ATOMICSOLVER$(t, i, R_{t,i})$
7:         Let $X_t = X_t \bigcup \{x_{t,i,j} \mid j \in [2^{i-1}]\}$ and $Y_t = Y_t \bigcup \{y_{t,i,j} \mid j \in [2^{i-1}]\}$
8:     **end for**
9: **end for**

We now give the details of ATOMICSOLVER$(t, i, R_{t,i})$ for any $t \in [n]$ and $i \in \{2, 3, \ldots, d\}$. Let $\nu = n^2$, and $M = m + \nu + n + 1$. For every $t \in [n], i \in \{2, 3, \ldots, d\}$ and $r \in [i]$, we define the functions $f(t,i) := (i-1)! \cdot \nu_t$ and $g(t,i,r) := (t-1)d^2 + (i-1)i + r$, where ν_t is the tth prime integer greater than n^4. Note that $M = O(n^3)$ and $10^M > B_1$, by Fact 3.1. We will use the fact that ν_t is much larger than M later. Using the Prime Number Theorem [Sho09], it follows that the number of primes in the interval $[n^4, n^5]$ is larger than n, and thus $\nu_n < n^5$. Moreover, these n primes can be found in deterministic polynomial time [AKS04].

We will implement the recursive coupling idea of the ATOMICSOLVER described in Section 1.2, in terms of matrix algebra. For example, recall that in the first step of the variable coupling, we set $x_1 - y_1 = \alpha$, $y_2 - x_2 = \alpha$ and $x_1 - y_2 = \beta$. We can then express x_1, x_2, y_1, y_2 as a linear combination of α, β, where we use the extra degree of freedom to choose $x_1 = -x_2$, as follows:

14

$(x_1, x_2)^T = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \cdot (\alpha, \beta)^T$, and $(y_1, y_2)^T = \frac{1}{2}\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \cdot (\alpha, \beta)^T$. In general, the polynomial equations give rise to $2^i - 1$ linear constraints on 2^i unknowns $(x_1, \cdots, x_{2^{i-1}}, y_1, \cdots, y_{2^{i-1}})$. The extra degree of freedom allows us to preserve the symmetry of the solution, which enables us to describe the algorithm and its analysis in a clean form.

Algorithm 2: ATOMICSOLVER(t,i, $R_{t,i}$):

Input: $i, t, R_{t,i}$
Output: Set of auxiliary variables, $\{x_{t,i,j} \mid j \in [2^{i-1}]\} \bigcup \{y_{t,i,j} \mid j \in [2^{i-1}]\}$

1: Let ν_t be the tth prime integer greater than n^4
2: Let $f(t,i) = (i-1)! \cdot \nu_t$
3: Let $g(t,i,r) = (t-1)d^2 + (i-1)i + r$ for all $1 < r < i$
4: $\alpha_{t,i,1} = 10^{f(t,i)}$
5: $\alpha_{t,i,r} = 10^{g(t,i,r)}$ for all $1 < r < i$
6: $\alpha_{t,i,i} = R_{t,i}/(i! \prod_{r \in [i-1]} \alpha_{t,i,r})$
7: $\alpha_{t,i} = [\alpha_{t,i,1}, \ldots, \alpha_{t,i,i}]^T$
8: **if** $i = 2$ **then**
9: $\quad A_2 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$ and $B_2 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$
10: **else**
11: $\quad A_i = \begin{bmatrix} A_{i-1} & \mathbf{1}^{2^{i-2}} \\ B_{i-1} & -\mathbf{1}^{2^{i-2}} \end{bmatrix}$ and $B_i = \begin{bmatrix} B_{i-1} & \mathbf{1}^{2^{i-2}} \\ A_{i-1} & -\mathbf{1}^{2^{i-2}} \end{bmatrix}$
12: **end if**
13: $[x_{t,i,1}, \ldots, x_{t,i,2^{i-1}}]^T = \frac{1}{2} \cdot A_i \cdot \alpha_{t,i}$
14: $[y_{t,i,1}, \ldots, y_{t,i,2^{i-1}}]^T = \frac{1}{2} \cdot B_i \cdot \alpha_{t,i}$
15: Return $\{x_{t,i,j} \mid j \in [2^{i-1}]\} \bigcup \{y_{t,i,j} \mid j \in [2^{i-1}]\}$

Figure 3: Relative distribution of $\alpha_{t,i,r}$ for any $i \in \{2, \cdots, d\}$ with respect to a_t, b_t and B_1.

4 Verifying Properties 1, 2, 3, 4

In this section, we prove that the variables generated by the AUXILIARYVARIABLEGENERATOR satisfy Properties 1, 2, 3, 4. This is done via the following lemmas.

Lemma 4.1. *For every $t \in [n]$, the auxiliary variables satisfy the following conditions*

$$\sum_{x \in X_t} x = \sum_{y \in Y_t} y = 0$$

$$\sum_{x \in X_t} x^k - \sum_{y \in Y_t} y^k = b_t^k - a_t^k \text{ for every } k \in \{2, \ldots, d\}.$$

Lemma 4.2. *For any subset $S \subseteq \bigcup_{t \in [n]} X_t \cup Y_t$ of the auxiliary variables, either*

$$|\sum_{y \in S} y| > 10^{m+2n+\nu} \quad or \quad |\sum_{y \in S} y| < 10^\nu.$$

We restate the following lemma from Section 3.

Lemma 3.2. *For any integer d, the total number of variables in the instance of $\mathrm{MSS}(d)$ is $N = n \cdot (2^{d+1} - 2)$ and every variable has a $\mathsf{poly}(n, d!)$ digit representation in base 10.*

In order to prove Lemma 4.1, Lemma 4.2 and Lemma 3.2 we first state some properties of the auxiliary variables generated by the $\textsc{AtomicSolver}(t, i, R_{t,i})$ and prove them in Section 5.

Proposition 4.3. *For any $(t, i) \in [n] \times \{2, \ldots, d\}$, $\textsc{AtomicSolver}(t, i, R_{t,i})$ on input a rational $R_{t,i}$, returns two sets of auxiliary variables $\{x_{t,i,j} \mid j \in [2^{i-1}]\}$ and $\{y_{t,i,j} \mid j \in [2^{i-1}]\}$ which satisfy:*

$$\sum_{j=1}^{2^{i-1}} (x_{t,i,j}^i - y_{t,i,j}^i) = R_{t,i},$$

$$\sum_{j=1}^{2^{i-1}} (x_{t,i,j}^k - y_{t,i,j}^k) = 0 \text{ for every } k \in \{1, \ldots, i-1\}.$$

Proposition 4.4. *For any $t \in [n]$, and $i \in \{2, 3, \ldots, d\}$,*

$$\sum_{j=1}^{2^{i-1}} x_{t,i,j} = \sum_{j=1}^{2^{i-1}} y_{t,i,j} = 0.$$

Proposition 4.5. *For any $t \in [n], i \in \{2, \ldots, d\}$, we have*

(a) $i! \prod_{r=1}^{i} \alpha_{t,i,r} = R_{t,i}$

(b) $10^{n^4} < \alpha_{t,i,1} < 10^{d!n^5}$

(c) $\alpha_{t,i,r} < 10^{nd^2} \text{ for } 1 < r < i - 1$

(d) $|\alpha_{t,i,i}| < 2$

(e) $\sum_{r=2}^{i} |\alpha_{t,i,r}| < 10^{\nu - nd}.$

16

Proposition 4.6. *For any $t \in [n]$, $i \in \{2, \ldots, d\}$ and $j \in [2^{i-1}]$, we have that*

$$10^{(i-1)!\nu_t} - 10^{\nu - nd} \leq 2 \cdot |x_{t,i,j}| \leq 10^{(i-1)!\nu_t} + 10^{\nu - nd}.$$

The analogous statement also holds for $y_{t,i,j}$.

Proposition 4.7. *We have that:*

1. *For every $(t_1, i_1, j_1) \neq (t_2, i_2, j_2)$, we have that $x_{t_1, i_1, j_1} \neq x_{t_2, i_2, j_2}$.*

2. *For every $(t_1, i_1, j_1) \neq (t_2, i_2, j_2)$, we have that $y_{t_1, i_1, j_1} \neq y_{t_2, i_2, j_2}$.*

3. *For every $(t_1, i_1, j_1), (t_2, i_2, j_2)$, we have that $x_{t_1, i_1, j_1} \neq y_{t_2, i_2, j_2}$.*

4.1 Proof of Lemma 4.1

We now prove Lemma 4.1 which implies Properties 1 and 2 of the auxiliary variables.

Proof of Lemma 4.1. From Proposition 4.4, we have that for any $t \in [n]$, and $i \in \{2, 3, \ldots, d\}$, $\sum_{j=1}^{2^{i-1}} x_{t,i,j} = \sum_{j=1}^{2^{i-1}} y_{t,i,j} = 0$. Summing the variables over all $i \in \{2, 3, \ldots, d\}$, we get

$$\sum_{x \in X_t} x = \sum_{y \in Y_t} y = 0.$$

For the second part of the lemma, for any $k \in \{2, \ldots, d\}$

$$\sum_{x \in X_t} x^k - \sum_{y \in Y_t} y^k = \sum_{i=2}^{d} \sum_{j=1}^{2^{i-1}} (x_{t,i,j}^k - y_{t,i,j}^k)$$

$$= \sum_{i=2}^{k-1} \sum_{j=1}^{2^{i-1}} (x_{t,i,j}^k - y_{t,i,j}^k) + \sum_{j=1}^{2^{k-1}} (x_{t,k,j}^k - y_{t,k,j}^k) + \sum_{i=k+1}^{d} \sum_{j=1}^{2^{i-1}} (x_{t,i,j}^k - y_{t,i,j}^k)$$

From the definition of the residual, $R_{t,k}$, the first term, $\sum_{i=2}^{k-1} \sum_{j=1}^{2^{i-1}} (x_{t,i,j}^k - y_{t,i,j}^k) = b_t^k - a_t^k - R_{t,k}$.

Also, from Proposition 4.3 it follows that $\sum_{j=1}^{2^{k-1}} (x_{t,k,j}^k - y_{t,k,j}^k) = R_{t,k}$ and $\sum_{i=k+1}^{d} \sum_{j=1}^{2^{i-1}} (x_{t,i,j}^k - y_{t,i,j}^k) = 0$.

Substituting these values in the above equation, we get,

$$\sum_{x \in X_t} x^k - \sum_{y \in Y_t} y^k = b_t^k - a_t^k$$

$\square$

4.2 Proof of Lemma 4.2

Before we prove Lemma 4.2, we note that each auxiliary variable, $x_{t,i,j}$ and $y_{t,i,j}$ is a $(\pm\frac{1}{2})$-linear combination of the $\alpha_{t,i,r}$ variables. From Proposition 4.5 (b), (c) we note that each variable $\alpha_{t,i,r}$ is either of small magnitude, i.e. $|\alpha_{t,i,r}| < 10^{nd^2}$ or of fairly large magnitude, i.e. $|\alpha_{t,i,r}| > 10^{n^4}$. Also, we note that there is only one large magnitude term, i.e., $\alpha_{t,i,1}$, for every pair $(t,i) \in [n] \times \{2,\cdots,d\}$.

Recall that D is the set of all the auxiliary variables

$$D = \{x_{t,i,j}, y_{t,i,j} \mid t \in [n], i \in \{2,3,\ldots,d\}, j \in [2^{i-1}]\}.$$

For any auxiliary variable $z \in D$, we can split z into terms of the form $\pm\frac{1}{2}\alpha_{t,i,r}$ with large magnitudes and terms with small magnitudes.

$$z = z_U + z_L,$$

where z_U is the term with large magnitude and z_L is the linear combinations of terms with small magnitudes. We now state and prove two properties of the small magnitude sum and the large magnitude sum which will imply the proof of Lemma 4.2.

Claim 4.8. *For any subset $S \subseteq D$, $\displaystyle\sum_{z \in S} z_L < 10^\nu$.*

Proof. For any subset $S \subseteq D$,

$$\sum_{z \in S} z_L \leq \frac{1}{2} \sum_{t=1}^{n} \sum_{i=2}^{d} \sum_{r=2}^{i} |\alpha_{t,i,r}|.$$

From Proposition 4.5(e), we know that for any $(t,i) \in [n] \times \{2,\ldots,d\}$, $\displaystyle\sum_{r=2}^{i} |\alpha_{t,i,r}| \leq 10^{\nu-nd}$. Summing over all (t,i), we upper bound the sum of small magnitude terms as follows:

$$\sum_{z \in S} z_L \leq \frac{1}{2} \sum_{t=1}^{n} \sum_{i=2}^{d} \sum_{r=2}^{i} |\alpha_{t,i,r}|$$
$$\leq nd \cdot 10^{\nu-nd}$$
$$< 10^\nu \qquad \qquad \square$$

Claim 4.9. *Let $S \subseteq D$ such that $\displaystyle\sum_{z \in S} z_U \neq 0$, then $\left|\displaystyle\sum_{z \in S} z_U\right| \geq \frac{1}{2} \cdot 10^{n^4}$.*

Proof. We show that for any subset of the auxiliary variables, the contribution of the large magnitudes is either 0, or larger than $\frac{1}{2} \cdot 10^{n^4}$. Note that all the large magnitude terms, i.e., $\alpha_{t,i,1}$ for any (t,i), are powers of 10 larger than n^4 and therefore, each z_U, being a $\pm\frac{1}{2}$ multiple of the large term, is divisible by $\frac{1}{2} \cdot 10^{n^4}$. Thus, the sum $\left|\displaystyle\sum_{z \in S} z_U\right|$ is divisible by $\frac{1}{2} \cdot 10^{n^4}$. If the sum is non-zero, then it is a non-zero multiple of $\frac{1}{2} \cdot 10^{n^4}$ and hence is larger than $\frac{1}{2} \cdot 10^{n^4}$.

$$\square$$

The proof of Lemma 4.2 now follows by combining Claim 4.8 and Claim 4.9.

18

Proof of Lemma 4.2. For any subset $S \subseteq D$, we can split the sum of the variables as:

$$\sum_{z \in S} z = \sum_{z \in S} z_U + \sum_{z \in S} z_L.$$

If $\sum_{z \in S} z_U \neq 0$, then from Claim 4.8 and Claim 4.9 we have,

$$\left| \sum_{z \in S} z \right| \geq \left| \sum_{z \in S} z_U \right| - \left| \sum_{z \in S} z_L \right|$$

$$\geq \frac{1}{2} \cdot 10^{n^4} - 10^{\nu} = \Omega(10^{n^4}) > 10^{m+2n+\nu} \quad \text{[using the fact that } \nu = n^2\text{]}.$$

On the other hand, if $\sum_{z \in S} z_U = 0$, then from Claim 4.8,

$$\left| \sum_{z \in S} z \right| = \left| \sum_{z \in S} z_L \right| \leq 10^{\nu}.$$

$\square$

4.3   Proof of Lemma 3.2

Proof of Lemma 3.2. In the construction of the instance of MSS(d), we create 2 variables, i.e., a_t, b_t and $2^{d+1} - 4$ auxiliary variables $X_t \cup Y_t$ corresponding to each of the n literals in the 1-in-3 SAT instance. From Claim 4.13 below, we know that all variables in the set A are distinct. Therefore, the size of the set A in the instance of MSS(d), $N = n(2^{d+1} - 2)$. Now we show that every element constructed in the instance of MSS(d) has $\mathsf{poly}(n, d!)$ digit representation.

From Fact 3.1, Proposition 4.6 and Claim 4.10 below, we know that the magnitudes of all the numbers generated by the reduction are bounded by $10^{\mathsf{poly}(n,d!)}$. Therefore to complete the proof, it remains to show that the denominators of all the rational numbers in the instance of MSS(d) are also bounded by $10^{\mathsf{poly}(n,d!)}$

Observe from Definition 13 that a_t and b_t for every $t \in [n]$ are integers. Also, for any $t \in [n]$ and $i \in \{2, \cdots, d\}$ each $\alpha_{t,i,r}$ for $1 \leq r \leq i-1$ constructed by ATOMICSOLVER(t,i, $R_{t,i}$) is a unique power of 10, and hence an integer, but $\alpha_{t,i,i}$ is a rational number. Each auxiliary variable generated by ATOMICSOLVER(t,i, $R_{t,i}$) is therefore a rational number due to the contribution from $\alpha_{t,i,i}$. From Claim 4.12 below, it follows that every rational number in the instance of MSS(d) has magnitude at most $10^{\mathsf{poly}(n,d!)}$ and therefore a $\mathsf{poly}(n, d!)$ digit representation. $\square$

The following claim bounds the magnitudes of the targets in the MSS(d) instance.

Claim 4.10. *For every $k \in \{2, \cdots, d\}$,*

$$|B_k| \leq 10^{k(d!)n^6}.$$

Proof. Recall from Definition 14,

$$B_k = \sum_{t=1}^{n} a_t^k + \sum_{t=1}^{n} \sum_{x \in X_t} x^k \text{ for every } k \in \{2, \ldots, d\}.$$

Using bounds on the magnitudes of a_t and $x \in X_t$ from Fact 3.1 and Proposition 4.6 we get

$$|B_k| \le n(10^{k(m+n+\nu+1)}) + n2^d((10^{(d-1)!\nu_n} + 10^{\nu-nd})^k)$$
$$\le 10^{k(d!)n^6}. \qquad \square$$

We now bound the magnitude of the denominators of $\alpha_{t,i,i}$ for every $(t,i) \in [n] \times \{2, \cdots, d\}$. This bound will be used in Claim 4.12 to bound the denominators of all the rational numbers in the instance of MSS(d). Let $D(x)$ denote the irreducible denominator of a rational number x.

Claim 4.11. *For any $(t,i) \in [n] \times \{2, \cdots, d\}$,*

$$D(\alpha_{t,i,i}) \le 10^{(i!)^2 \cdot n^6}$$

Proof. The proof proceeds by first obtaining a recursive expression for $D(\alpha_{t,i,i})$, and we then use induction on i to show the bound. Recall the definition of $\alpha_{t,i,i}$ from Algorithm 2,

$$\alpha_{t,i,i} = \frac{R_{t,i}}{i! \prod_{r \in [i-1]} \alpha_{t,i,r}},$$

where $R_{t,i}$ is defined as

$$R_{t,i} = b_t^i - a_t^i + \sum_{u=2}^{i-1} \sum_{v=1}^{2^{u-1}} (y_{t,u,v}^i - x_{t,u,v}^i).$$

Therefore, it follows that the denominator of $\alpha_{t,i,i}$ is bounded by the product of the denominator of $R_{t,i}$ and $i! \cdot \prod_{r=1}^{i-1} \alpha_{t,i,r}$. i.e.,

$$D(\alpha_{t,i,i}) \le D(R_{t,i}) \cdot (i! \cdot \prod_{r=1}^{i-1} \alpha_{t,i,r})$$

$$= D(R_{t,i}) \cdot (i! \cdot 10^{(i-1)!\nu_t + \sum_{r=2}^{i-1} g(t,i,r)})$$

$$\le D(R_{t,i}) \cdot (i! \cdot 10^{(i-1)!n^5 + nd^3})$$

The last inequality follows from the fact that $\sum_{r=2}^{i-1} g(t,i,r) = \sum_{r=2}^{i-1}(t-1)d^2 + (i-1)i + r \le td^3$ for all $2 \le i \le d$ and $\nu_t < n^5$ for any $t \in [n]$. We now obtain an expression for $D(R_{t,i})$. Since b_t and a_t are both integers, note that $D(R_{t,i}) = D\left(\sum_{u=2}^{i-1} \sum_{v=1}^{2^{u-1}} (y_{t,u,v}^i - x_{t,u,v}^i)\right)$. Also, recall that all the auxiliary variables obtained from a given ATOMICSOLVER(t,u, $R_{t,u}$), described in Algorithm 2, have the same denominator to which $D(\alpha_{t,u,u})$ contributes, i.e., $D(x_{t,u,v}) = D(y_{t,u,v}) = 2 \cdot D(\alpha_{t,u,u})$, for all $v \in [2^{u-1}]$. Therefore, $D(y_{t,u,v}^i - x_{t,u,v}^i) = 2^i \cdot D(\alpha_{t,u,u}^i)$. From this observation, it follows that $D(\sum_{v=1}^{2^{u-1}} y_{t,u,v}^i - x_{t,u,v}^i) = 2^i \cdot D(\alpha_{t,u,u}^i)$, and we get an expression for $D(R_{t,i})$ as follows:

$$D(R_{t,i}) = LCM(\{2^i \cdot D(\alpha_{t,u,u}^i) \mid u \in \{2, \cdots, i-1\}\}) \le 2^{i^2} \cdot \prod_{u=2}^{i-1} D(\alpha_{t,u,u}^i).$$

20

Substituting the above expression for $D(R_{t,i})$ back in the expression obtained for $D(\alpha_{t,i,i})$, we get

$$D(\alpha_{t,i,i}) \leq \left(\prod_{u=2}^{i-1} D(\alpha_{t,u,u}^i)\right) \cdot (2^{i^2} \cdot i! \cdot 10^{(i-1)!n^5+nd^3}) \tag{18}$$

We now use induction on i to show that that $D(\alpha_{t,i,i}) \leq 10^{(i!)^2 \cdot n^6}$ for every $i \in \{2, \cdots, d\}$. For the base case, $i = 2$, from definitions we know that

$$D(\alpha_{t,2,2}) = 2 \cdot 10^{\nu_t} < 10^{n^6}$$

Let us assume the induction hypothesis that for all $i < \ell \leq d$,

$$D(\alpha_{t,i,i}) \leq 10^{(i!)^2 \cdot n^6}.$$

From Equation 18, we know that

$$\begin{aligned}
D(\alpha_{t,\ell,\ell}) &\leq \left(\prod_{u=2}^{\ell-1} D(\alpha_{t,u,u}^\ell)\right) \cdot (2^{\ell^2} \cdot \ell! \cdot 10^{(\ell-1)!n^5+nd^3}) \\
&\leq \left(\prod_{u=2}^{\ell-1} (10^{(u!)^2 \cdot n^6})^\ell\right) \cdot (10^{(\ell-1)!n^5+nd^3+2\ell^2}) \\
&\leq 10^{\ell \sum_{u=2}^{\ell-1}((u!)^2 \cdot n^6)+(\ell)!n^5+nd^3+2\ell^2} \\
&\leq 10^{\ell \cdot (\ell-1) \cdot (\ell-1)!^2 \cdot n^6+(\ell)!n^5+nd^3+2\ell^2} \\
&\leq 10^{(\ell!)^2 \cdot n^6},
\end{aligned}$$

where the last inequality follows from the fact that $\ell(\ell-1)!^2 n^6 > (\ell)!n^5+nd^3+2\ell^2$ for any $\ell \leq d$. $\qquad \square$

Claim 4.12. *For any $x \in A \bigcup \{B_1, \cdots, B_d\}$,*

$$D(x) < 10^{\mathsf{poly}(n,d!)}.$$

Proof of Claim 4.12. We first observe that the elements constructed from the 3-SAT clauses and variables are all integers. So, $D(a_t) = D(b_t) = 1$ for all $t \in [n]$. Next, we argue about the denominators of the auxiliary variables and show that they are all bounded by $2 \cdot 10^{(d!)^2 \cdot n^6}$. Consider the set of auxiliary variables generated by $\textsc{AtomicSolver}(t,i, R_{t,i})$ for some $t \in [n]$ and $i \in \{2, 3, \cdots, d\}$. Each $x_{t,i,j}$ (or $y_{t,i,j}$) is a $\pm\frac{1}{2}$-linear combination of the $\{\alpha_{t,i,r} \mid r \in [i]\}$ variables. From the definitions in Algorithm 2, we note that all $\alpha_{t,i,r}$ variables constructed by the $\textsc{AtomicSolver}$ are integers except for $\alpha_{t,i,i}$. Therefore, each $x_{t,i,j}$ and $y_{t,i,j}$ have the same denominator as $\alpha_{t,i,i}/2$. Using Claim 4.11, we get that for every (t,i), $D(\alpha_{t,i,i}) \leq 10^{(i!)^2 \cdot n^6}$. Therefore, for any $j \in [2^{i-1}]$, $D(x_{t,i,j}) < 2 \cdot 10^{(i!)^2 \cdot n^6}$. A similar argument applies to $y_{t,i,j}$.

We now bound the magnitudes of the denominators of the target, $B_1, \cdots, B_d$ defined in the $\mathrm{MSS}(d)$ instance. Recall from Definition 14 that B_1 is an integer. Therefore, $D(B_1) = 1$. All other targets are rational numbers defined as

$$B_k = \sum_{t=1}^n a_t^k + \sum_{t=1}^n \sum_{x \in X_t} x^k \text{ for every } k \in \{2, \ldots, d\}.$$

21

The denominator of B_k is defined by the denominator of the sum, $\sum_{t=1}^{n} \sum_{x \in X_t} x^k$. This sum can be

expanded as $\sum_{t=1}^{n} \sum_{i=2}^{d} \sum_{j=1}^{2^{i-1}} x^k$. From the fact that $D(\sum_{j=1}^{2^{i-1}} x^k) = D(\alpha_{t,i,i}^k)$ and Claim 4.11, we get,

$$
\begin{aligned}
D(B_k) &\leq \prod_{\substack{t \in [n] \\ i \in \{2...,d\}}} D(\alpha_{t,i,i}^k) \\
&\leq \prod_{\substack{t \in [n] \\ i \in \{2...,d\}}} 10^{k(i!)^2 \cdot n^6} \\
&\leq (10^{k(d!)^2 \cdot n^6})^{nd} = 10^{kd(d!)^2 \cdot n^7}
\end{aligned}
$$

Therefore, we conclude that every element of the instance of $\mathrm{MSS}(d)$ constructed by the reduction has a denominator of magnitude at most $10^{\mathsf{poly}(n,d!)}$. $\qquad \square$

Claim 4.13. *All variables in the set A are distinct.*

Proof of Claim 4.13. From Proposition 4.7, we know that all auxiliary variables are distinct. Also, the distinctness of the variables $\{a_t, b_t \mid t \in [n]\}$ follows from the construction. The only thing that remains to show is that all the auxiliary variables are different from $\{a_t, b_t \mid t \in [n]\}$.

We show this fact by comparing the magnitudes of the two sets of variables. From Fact 3.1, we know that $|v| < 10^{m+n+\nu+1}$ for every $v \in \{a_t, b_t \mid t \in [n]\}$, and from Proposition 4.6, we know that all auxiliary variables are larger than $10^{\nu_1} - 10^{\nu-nd} > 10^{m+n+\nu+1}$. Therefore the two sets of variables are disjoint. $\qquad \square$

5 Proofs of the Helper Propositions 4.3, 4.4, 4.5, 4.6, 4.7

In this section, we prove the helper claims stated in the previous section.

Proof of Proposition 4.3. We first show a structural property of the auxiliary variables generated by any ATOMICSOLVER. The proof of Proposition 4.3 follows from it.

Claim 5.1. *For any $i \in \{2, \cdots, d\}$, Let A_i, B_i are matrices defined in the ATOMICSOLVER and let $\{\alpha_r \mid r \in [i]\}$ be some rational numbers. If*

$$
\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{2^{i-1}} \end{bmatrix} = \frac{1}{2} \cdot A_i \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_i \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{2^{i-1}} \end{bmatrix} = \frac{1}{2} \cdot B_i \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_i \end{bmatrix},
$$

22

then $\{x_j \mid j \in [2^{i-1}]\}$ *and* $\{y_j \mid j \in [2^{i-1}]\}$ *satisfy:*

$$\sum_{j=1}^{2^{i-1}} (x_j^k - y_j^k) = 0 \ for\ every\ k \in \{1, \ldots, i-1\}$$

$$\sum_{j=1}^{2^{i-1}} (x_j^i - y_j^i) = i! \prod_{r=1}^{i} \alpha_r$$

Proof of Claim 5.1. We use induction on i. For the base case, consider $i = 2$. From the definition of A_2 and B_2 we get,

$$x_1 = \frac{\alpha_1}{2} + \frac{\alpha_2}{2}$$
$$x_2 = -\frac{\alpha_1}{2} - \frac{\alpha_2}{2}$$
$$y_1 = \frac{\alpha_1}{2} - \frac{\alpha_2}{2}$$
$$y_2 = -\frac{\alpha_1}{2} + \frac{\alpha_2}{2}$$

Therefore,

$$x_1 + x_2 - y_1 - y_2 = 0$$
$$x_1^2 + x_2^2 - y_1^2 - y_2^2 = 2 \cdot \alpha_1 \cdot \alpha_2$$

and the claim holds for $i = 2$.

Let us assume the induction hypothesis for all $i < \ell \le d$. For $i = \ell$, we have,

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{2^{\ell-1}} \end{bmatrix} = \frac{1}{2} \cdot A_\ell \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_\ell \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{2^{\ell-1}} \end{bmatrix} = \frac{1}{2} \cdot B_\ell \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_\ell \end{bmatrix}$$

From the recursive definitions of the matrices A_ℓ, B_ℓ in Algorithm 2, we can split the above

equations as

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{2^{\ell-2}} \end{bmatrix} = \frac{1}{2} \cdot A_{\ell-1} \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{\ell-1} \end{bmatrix} + \frac{1}{2} \cdot \begin{bmatrix} \alpha_\ell \\ \alpha_\ell \\ \vdots \\ \alpha_\ell \end{bmatrix}$$

$$\begin{bmatrix} x_{2^{\ell-2}+1} \\ x_{2^{\ell-2}+2} \\ \vdots \\ x_{2^{\ell-1}} \end{bmatrix} = \frac{1}{2} \cdot B_{\ell-1} \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{\ell-1} \end{bmatrix} - \frac{1}{2} \cdot \begin{bmatrix} \alpha_\ell \\ \alpha_\ell \\ \vdots \\ \alpha_\ell \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{2^{\ell-2}} \end{bmatrix} = \frac{1}{2} \cdot B_{\ell-1} \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{\ell-1} \end{bmatrix} + \frac{1}{2} \cdot \begin{bmatrix} \alpha_\ell \\ \alpha_\ell \\ \vdots \\ \alpha_\ell \end{bmatrix}$$

$$\begin{bmatrix} y_{2^{\ell-2}+1} \\ y_{2^{\ell-2}+2} \\ \vdots \\ y_{2^{\ell-1}} \end{bmatrix} = \frac{1}{2} \cdot A_{\ell-1} \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{\ell-1} \end{bmatrix} - \frac{1}{2} \cdot \begin{bmatrix} \alpha_\ell \\ \alpha_\ell \\ \vdots \\ \alpha_\ell \end{bmatrix}$$

Equivalently, they can be rewritten as

$$x_j = \begin{cases} x'_j + \frac{1}{2} \cdot \alpha_\ell \text{ if } j \le 2^{\ell-2} \\ y'_{j-2^{\ell-2}} - \frac{1}{2} \cdot \alpha_\ell \text{ if } j > 2^{\ell-2} \end{cases}$$

Similarly,

$$y_j = \begin{cases} y'_j + \frac{1}{2} \cdot \alpha_\ell \text{ if } j \le 2^{\ell-2} \\ x'_{j-2^{\ell-2}} - \frac{1}{2} \cdot \alpha_\ell \text{ if } j > 2^{\ell-2} \end{cases}$$

where, the $\{x'_j, y'_j \mid j \in [2^{\ell-2}]\}$ by induction hypothesis satisfy

$$\sum_{j=1}^{2^{\ell-2}} (x_j'^{\ell-1} - y_j'^{\ell-1}) = (\ell-1)! \prod_{r=1}^{\ell-1} \alpha_r$$

$$\sum_{j=1}^{2^{\ell-2}} (x_j'^k - y_j'^k) = 0 \text{ for every } k \in \{1, \dots, \ell-2\}.$$

Therefore, for any $k \in \mathbb{N}$, we get that

$$\sum_{j=1}^{2^{\ell-1}} (x_j^k - y_j^k) = \sum_{j=1}^{2^{\ell-2}} (x_j' + \frac{1}{2} \cdot \alpha_\ell)^k - (y_j' + \frac{1}{2} \cdot \alpha_\ell)^k$$

$$+ \sum_{j=2^{\ell-2}+1}^{2^{\ell-1}} (y_{j-2^{\ell-2}}' - \frac{1}{2} \cdot \alpha_\ell)^k - (x_{j-2^{\ell-2}}' - \frac{1}{2} \cdot \alpha_\ell)^k$$

$$= \sum_{j=1}^{2^{\ell-2}} (x_j' + \frac{1}{2} \cdot \alpha_\ell)^k - (x_j' - \frac{1}{2} \cdot \alpha_\ell)^k$$

$$- \sum_{j=1}^{2^{\ell-2}} (y_j' + \frac{1}{2} \cdot \alpha_\ell)^k - (y_j' - \frac{1}{2} \cdot \alpha_\ell)^k$$

$$= \sum_{j=1}^{2^{\ell-2}} \left(2 \sum_{\substack{r=0 \\ r \equiv 1 \text{ mod } 2}}^{k} \frac{1}{2^r} \cdot \binom{k}{r} x_j'^{k-r} \alpha_\ell^r \right)$$

$$- \sum_{j=1}^{2^{\ell-2}} \left(2 \sum_{\substack{r=0 \\ r \equiv 1 \text{ mod } 2}}^{k} \frac{1}{2^r} \cdot \binom{k}{r} y_j'^{k-r} \alpha_\ell^r \right)$$

$$= \sum_{j=1}^{2^{\ell-2}} \left(2 \sum_{\substack{r=0 \\ r \equiv 1 \text{ mod } 2}}^{k} \frac{1}{2^r} \cdot \binom{k}{r} (x_j'^{k-r} - y_j'^{k-r}) \alpha_\ell^r \right)$$

Observe that, for all $k \leq \ell - 1$, $k - r \leq \ell - 2$, since $r \equiv 1$ mod 2. Therefore, for all $k \leq \ell - 1$, from induction hypothesis, we have, $(x_j'^{k-r} - y_j'^{k-r}) = 0$. And,

$$\sum_{j=1}^{2^{\ell-1}} (x_j^k - y_j^k) = 0.$$

For $k = \ell$,

$$\sum_{j=1}^{2^{\ell-1}} (x_j^\ell - y_j^\ell) = \sum_{j=1}^{2^{\ell-2}} \left(2 \sum_{\substack{r=0 \\ r \equiv 1 \text{ mod } 2}}^{\ell} \frac{1}{2^r} \cdot \binom{\ell}{r} (x_j'^{\ell-r} - y_j'^{\ell-r}) \alpha_\ell^r \right)$$

$$= 2 \cdot \frac{1}{2} \cdot \binom{\ell}{1} \cdot \alpha_\ell \sum_{j=1}^{2^{\ell-2}} (x_j'^{\ell-1} - y_j'^{\ell-1})$$

$$= \ell \cdot (\ell - 1)! \cdot \prod_{r=1}^{\ell-1} \alpha_r \cdot \alpha_\ell$$

$$= \ell! \cdot \prod_{r=1}^{\ell} \alpha_r.$$

$\square$

25

Note that Claim 5.1 is independent of t and the choice of the α variables. Recall the construction of AtomicSolver$(t,i, R_{t,i})$ for any $(t,i) \in [n] \times \{2, \cdots, d\}$. It returns two sets of auxiliary variables $\{x_{t,i,j} \mid j \in [2^{i-1}]\}$ and $\{y_{t,i,j} \mid j \in [2^{i-1}]\}$ which are constructed using matrices A_i and B_i. From Claim 5.1, it then follows that these auxiliary variables satisfy:

$$\sum_{j=1}^{2^{i-1}} (x^i_{tij} - y^i_{tij}) = i! \prod_{r=1}^{i} \alpha_{t,i,r}$$

$$\sum_{j=1}^{2^{i-1}} (x^k_{tij} - y^k_{tij}) = 0 \text{ for every } k \in \{1, \ldots, i-1\}$$

Using Proposition 4.5 (a), we get,

$$\sum_{j=1}^{2^{i-1}} (x^i_{tij} - y^i_{tij}) = b^i_t - a^i_t + R_{t,i}$$

$\square$

Proof of Proposition 4.4. The proof uses the recursive structure of the matrices A_i and B_i. Recall that $\mathbf{1}^\ell$ denotes a vector of ℓ ones, and $\mathbf{0}^\ell$ denotes a vector of ℓ zeros. Note that for any $(t,i) \in [n] \times \{2, \cdots, d\}$,

$$\sum_{j=1}^{2^{i-1}} x_{t,i,j} = \frac{1}{2} \cdot (\mathbf{1}^{2^{i-1}})^T \cdot A_i \cdot \begin{bmatrix} \alpha_{t,i,1} & \cdots & \alpha_{t,i,i} \end{bmatrix}^T.$$

Similarly, the sum of all the $\{y_{t,i,j} \mid j \in [2^{i-1}]\}$ can be written as

$$\sum_{j=1}^{2^{i-1}} y_{t,i,j} = \frac{1}{2} \cdot (\mathbf{1}^{2^{i-1}})^T \cdot B_i \cdot \begin{bmatrix} \alpha_{t,i,1} & \cdots & \alpha_{t,i,i} \end{bmatrix}^T.$$

We show by induction on $i \geq 2$ that

$$(\mathbf{1}^{2^{i-1}})^T \cdot A_i = (\mathbf{0}^i)^T \text{ and } (\mathbf{1}^{2^{i-1}})^T \cdot B_i = (\mathbf{0}^i)^T.$$

For the base case, $i = 2$, it can be verified that

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \cdot A_2 = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix} \text{ and}$$

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \cdot B_2 = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Let us assume the induction hypothesis for all $i < \ell \leq d$. For $i = \ell$, observe that

$$(\mathbf{1}^{2^{\ell-1}})^T \cdot A_i = \begin{bmatrix} (\mathbf{1}^{2^{\ell-2}})^T & (\mathbf{1}^{2^{\ell-2}})^T \end{bmatrix} \cdot \begin{bmatrix} A_{\ell-1} & \mathbf{1}^{2^{\ell-2}} \\ B_{i-1} & -\mathbf{1}^{2^{\ell-2}} \end{bmatrix}$$

$$= \begin{bmatrix} (\mathbf{1}^{2^{\ell-2}})^T \cdot A_{\ell-1} + (\mathbf{1}^{2^{\ell-2}})^T \cdot B_{\ell-1} & 0 \end{bmatrix}$$

By the induction hypothesis, we know that $(\mathbf{1}^{2^{\ell-2}})^T \cdot A_{\ell-1} + (\mathbf{1}^{2^{\ell-2}})^T \cdot B_{\ell-1} = (\mathbf{0}^{\ell-1})^T$, Therefore,

$$(\mathbf{1}^{2^{\ell-1}})^T \cdot A_\ell = \begin{bmatrix} \mathbf{0}^{\ell-1} & 0 \end{bmatrix}$$

Similarly,

$$(\mathbf{1}^{2^{\ell-1}})^T \cdot B_\ell = \left[(\mathbf{1}^{2^{\ell-2}})^T \cdot B_{\ell-1} + (\mathbf{1}^{2^{\ell-2}})^T \cdot A_{\ell-1} \quad 0 \right] = \begin{bmatrix} \mathbf{0}^{\ell-1} & 0 \end{bmatrix}$$

$\square$

We now show certain bounds on the magnitudes of $\alpha_{t,i,r}$ and hence on the auxiliary variables $x_{t,i,j}, y_{t,i,j}$. For any two tuples of same dimensions, we say that $(p_1, p_2, \cdots, p_d) > (q_1, q_2, \cdots, q_d)$ if there is an $i \in [d]$ such that $p_i > q_i$ and $p_j = q_j$ for all $j < i$. In order to prove Proposition 4.5, we will need the following claim.

Claim 5.2. *For any $t \in [n]$, $i \in \{2, \ldots, d\}$ and any $j \in [2^{i-1}]$,*

$$|x_{t,i,j} - y_{t,i,j}| = \alpha_{t,i,2}$$

Proof of Claim 5.2. We use the recursive matrix definitions to show that for every $t, i, j \in [n] \times \{2, \cdots, d\} \times [2^{i-1}]$,

$$|x_{t,i,j} - y_{t,i,j}| = \alpha_{t,i,2}$$

We use induction on i.

For the base case, $i = 2$,

$$x_{t,2,1} - y_{t,2,1} = \frac{\alpha_{t,2,1}}{2} + \frac{\alpha_{t,2,2}}{2} - \frac{\alpha_{t,2,1}}{2} + \frac{\alpha_{t,2,2}}{2} = \alpha_{t,2,2}$$

$$x_{t,2,2} - y_{t,2,2} = -\frac{\alpha_{t,2,1}}{2} - \frac{\alpha_{t,2,2}}{2} + \frac{\alpha_{t,2,1}}{2} - \frac{\alpha_{t,2,2}}{2} = -\alpha_{t,2,2}$$

Let us assume the induction hypothesis for all $i < \ell \le d$.
From the definition of the ATOMICSOLVER we know that,

$$\begin{bmatrix} x_{t,\ell,1} \\ x_{t,\ell,2} \\ \vdots \\ x_{t,\ell,2^{\ell-1}} \end{bmatrix} = \frac{1}{2} \cdot A_\ell \cdot \begin{bmatrix} \alpha_{t,\ell,1} \\ \alpha_{t,\ell,2} \\ \vdots \\ \alpha_{t,\ell,\ell} \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} y_{t,\ell,1} \\ y_{t,\ell,2} \\ \vdots \\ y_{t,\ell,2^{\ell-1}} \end{bmatrix} = \frac{1}{2} \cdot B_\ell \cdot \begin{bmatrix} \alpha_{t,\ell,1} \\ \alpha_{t,\ell,2} \\ \vdots \\ \alpha_{t,\ell,\ell} \end{bmatrix}.$$

Therefore,

$$\begin{bmatrix} x_{t,\ell,1} - y_{t,\ell,1} \\ x_{t,\ell,2} - y_{t,\ell,2} \\ \vdots \\ x_{t,\ell,2^{\ell-1}} - y_{t,\ell,2^{\ell-1}} \end{bmatrix} = \frac{1}{2} \cdot (A_\ell - B_\ell) \cdot \begin{bmatrix} \alpha_{t,\ell,1} \\ \alpha_{t,\ell,2} \\ \vdots \\ \alpha_{t,\ell,\ell} \end{bmatrix}.$$

From the recursive definition of the matrices A_ℓ and B_ℓ, we get that $A_\ell - B_\ell = \begin{bmatrix} A_{\ell-1} - B_{\ell-1} & \mathbf{0}^{2^{\ell-2}} \\ B_{\ell-1} - A_{\ell-1} & \mathbf{0}^{2^{\ell-2}} \end{bmatrix}$.

From the induction hypothesis, we get,

$$A_\ell - B_\ell = \begin{bmatrix} A_2 - B_2 & 0 & \cdots & 0 \\ B_2 - A_2 & 0 & \cdots & 0 \\ & & & \vdots \\ A_2 - B_2 & 0 & \cdots & 0 \\ B_2 - A_2 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 0 & \cdots & 0 \\ 0 & -2 & 0 & \cdots & 0 \\ & & & & \vdots \\ 0 & 2 & 0 & \cdots & 0 \\ 0 & -2 & 0 & \cdots & 0 \end{bmatrix}$$

27

and therefore for every $j \in [2^{\ell-1}]$,

$$|x_{t,\ell,j} - y_{t,\ell,j}| = \alpha_{t,\ell,2}.$$

$\square$

We are now ready to prove Proposition 4.5.

Proof of Proposition 4.5.

(a) Follows from the definition of $\alpha_{t,i,i}$ in Algorithm 2.

(b) $\alpha_{t,i,1} = 10^{f(t,i)} = 10^{(i-1)!\nu_t}$, where ν_t is the tth prime greater than n^4. Since ν_t is increasing in t, and for a fixed t, $\alpha_{t,i,1}$ is increasing in i, $\max_{t,i}\{\alpha_{t,i,1}\} = \alpha_{n,d,1}$ and $\min_{t,i}\{\alpha_{t,i,1}\} = \alpha_{1,2,1}$. We had noted earlier that from Prime Number Theorem, the nth prime greater than n^4 is at most n^5. Therefore,

$$10^{n^4} < 10^{\nu_1} = \alpha_{1,2,1} \leq \alpha_{t,i,1} \leq \alpha_{n,d,1} = 10^{(d-1)!\nu_n} < 10^{d!n^5}.$$

(c) From the definitions in Algorithm 2, for every $1 < r < i - 1$, $\alpha_{t,i,r} = 10^{g(t,i,r)}$. Note that $\max_{t,i,r}\{g(t,i,r)\} = g(n,d,d-1) \leq nd^2$ and therefore,

$$\alpha_{t,i,r} \leq \alpha_{n,d,d-1} = 10^{g(n,d,d-1)} \leq 10^{nd^2}.$$

(d and e) Fix an arbitrary $t \in [n]$. We prove by induction on $i \in \{2, \cdots, d\}$ that,

$$|\alpha_{t,i,i}| < 2 \text{ and } \sum_{r=2}^{i}|\alpha_{t,i,r}| \leq 10^{\nu-nd}.$$

For the base case, $i = 2$, $\alpha_{t,2,2} = \frac{b_t^2 - a_t^2}{2\alpha_{t,2,1}}$. Recall from Definition 13 that the variable part of a_t and b_t is the same. Therefore $|b_t - a_t| \leq 10^{m+\nu}$. From Fact 3.1, we know that $|a_t|$ and $|b_t|$ are at most $10^{m+\nu+n+1} = 10^M$, so we get,

$$|b_t^2 - a_t^2| = |(b_t - a_t)(b_t + a_t)| \leq 10^{m+\nu} \cdot 2\max\{a_t, b_t\} < 10^{m+\nu} \cdot 2 \cdot 10^M.$$

Since $m + \nu < M$, $|\alpha_{t,2,2}| < \frac{10^{m+\nu} \cdot 2 \cdot 10^M}{2 \cdot 10^{f(t,2)}} < 10^{2M-f(t,2)}$. By definitions in Algorithm 2, $f(t,2) = \nu_t$ and ν_t is a prime larger than n^4. Also, $M = O(n^3)$ and $f(t,2) > 2M$ therefore, it follows that, $|\alpha_{t,2,2}| < 1 < 10^{\nu-nd}$ and the claim holds for $i = 2$.

Let us assume the induction hypothesis for all $i < \ell \leq d$, and we now prove the claim for $i = \ell$. We need to show that

$$|\alpha_{t,\ell,\ell}| < 2 \text{ and } \sum_{r=2}^{\ell}|\alpha_{t,\ell,r}| \leq 10^{\nu-nd}$$

We first bound the magnitude of $\alpha_{t,\ell,\ell}$ for any $t \in [n]$. Recall from the definitions in Algorithm 2,

$$|\alpha_{t,\ell,\ell}| = \frac{|R_{t,\ell}|}{i! \cdot \prod_{r \in [\ell-1]} \alpha_{t,\ell,r}}, \quad \text{where,} \quad R_{t,\ell} = b_t^\ell - a_t^\ell + \sum_{u=2}^{\ell-1} \sum_{v=1}^{2^{u-1}} y_{t,u,v}^\ell - x_{t,u,v}^\ell.$$

We will bound each individual term in the definition of $\alpha_{t,\ell,\ell}$ separately.

The term $|b_t^\ell - a_t^\ell|$ in $R_{t,\ell}$ can be factorized as $|b_t^\ell - a_t^\ell| = |(b_t - a_t)(\sum_{k=0}^{\ell-1} b_t^k a_t^{\ell-1-k})|$. We had seen earlier that $|(b_t - a_t)| < 10^{m+\nu} < 10^M$ and from Fact 3.1, $\max\{a_t, b_t\} < 10^M$. Using these observations, we get

$$\begin{aligned}
|b_t^\ell - a_t^\ell| &= |(b_t - a_t)(\sum_{k=0}^{\ell-1} b_t^k a_t^{\ell-1-k})| \\
&< 10^M \cdot \ell \cdot \max\{a_t^{\ell-1}, b_t^{\ell-1}\} \\
&\leq 10^M \cdot \ell \cdot 10^{M(\ell-1)} = \ell \cdot 10^{M\ell}.
\end{aligned} \tag{19}$$

Using the definitions of $\alpha_{t,\ell,r}$, the denominator in the expression for $\alpha_{t,\ell,\ell}$ can be written as

$$\begin{aligned}
\ell! \prod_{r=1}^{\ell-1} \alpha_{t,\ell,r} &= \ell! \cdot 10^{f(t,\ell) + \sum_{r=2}^{\ell-1} g(t,\ell,r)} \\
&\geq \ell! \cdot 10^{f(t,\ell) + g(t,\ell,2)}
\end{aligned} \tag{20}$$

Now to bound the magnitude of $\left| \sum_{u=2}^{\ell-1} \sum_{v=1}^{2^{u-1}} y_{t,u,v}^\ell - x_{t,u,v}^\ell \right|$, we have

$$\begin{aligned}
\left| \sum_{u=2}^{\ell-1} \sum_{v=1}^{2^{u-1}} y_{t,u,v}^\ell - x_{t,u,v}^\ell \right| &\leq \sum_{u=2}^{\ell-1} \sum_{v=1}^{2^{u-1}} |y_{t,u,v}^\ell - x_{t,u,v}^\ell| \\
&= \sum_{u=2}^{\ell-1} \sum_{v=1}^{2^{u-1}} |(y_{t,u,v} - x_{t,u,v})(\sum_{k=0}^{\ell-1} y_{t,u,v}^k x_{t,u,v}^{\ell-1-k})| \\
&\leq \sum_{u=2}^{\ell-1} \sum_{v=1}^{2^{u-1}} |(y_{t,u,v} - x_{t,u,v})| \cdot \ell \cdot \max\{|x_{t,u,v}|^{\ell-1}, |y_{t,u,v}|^{\ell-1}\}
\end{aligned}$$

Using Claim 5.2, we know that for any $(t, u, v) \in [n] \times \{2, \cdots, \ell-1\} \times [2^{u-1}]$,

$$|x_{t,u,v} - y_{t,u,v}| = \alpha_{t,u,2} = 10^{g(t,u,2)}.$$

Also, from definition of the auxiliary variables in Algorithm 2, each $x_{t,u,v}$ and $y_{t,u,v}$ for any $t \in [n]$, is a $(\pm\frac{1}{2})$-linear combinations of $\{\alpha_{t,u,r} \mid r \in [u]\}$. Therefore,

$$\max\{|x_{t,u,v}|, |y_{t,u,v}|\} \leq \frac{1}{2} \sum_{r=1}^{u} |\alpha_{t,u,r}|.$$

29

Since $u < \ell$, using the induction hypothesis, we know that $\sum_{r=2}^{u} |\alpha_{t,u,r}| < 10^{\nu-nd}$. So, we get

$$\max\{|x_{t,u,v}|, |y_{t,u,v}|\} \leq |\alpha_{t,u,1}| + \sum_{r=2}^{u} |\alpha_{t,u,r}| < \frac{1}{2}(10^{f(t,u)} + 10^{\nu-nd}) < 10^{f(t,u)}.$$

From these observations, we get

$$\left| \sum_{u=2}^{\ell-1} \sum_{v=1}^{2^{u-1}} y_{t,u,v}^{\ell} - x_{t,u,v}^{\ell} \right| \leq \sum_{u=2}^{\ell-1} \sum_{v=1}^{2^{u-1}} 10^{g(t,u,2)} \cdot \ell \cdot (10^{f(t,u)})^{\ell-1}$$

Note that $\max_{u}\{g(t,u,2)\} = g(t,\ell-1,2)$ and for a fixed $t, f(t,i)$ is increasing in i, therefore, $f(t,u) \leq f(t,\ell-1)$ for all $u \leq \ell-1$. Therefore,

$$\left| \sum_{u=2}^{\ell-1} \sum_{v=1}^{2^{u-1}} y_{t,u,v}^{\ell} - x_{t,u,v}^{\ell} \right| \leq \ell \cdot 2^{\ell} \cdot 10^{g(t,\ell-1,2)} \cdot 10^{(\ell-1)f(t,\ell-1)} \tag{21}$$

Combining Equations 19, 20, 21, we get an upper bound on the magnitude $\alpha_{t,\ell,\ell}$ as

$$|\alpha_{t,\ell,\ell}| = \frac{|R_{t,\ell}|}{\ell! \displaystyle\prod_{r=1}^{\ell-1} \alpha_{t,\ell,r}}$$

$$\leq \frac{|b_t^{\ell} - a_t^{\ell}|}{\ell! \displaystyle\prod_{r=1}^{\ell-1} \alpha_{t,\ell,r}} + \frac{\left| \displaystyle\sum_{u=2}^{\ell-1} \sum_{v=1}^{2^{u-1}} y_{t,u,v}^{\ell} - x_{t,u,v}^{\ell} \right|}{\ell! \displaystyle\prod_{r=1}^{\ell-1} \alpha_{t,\ell,r}}$$

$$\leq \frac{\ell \cdot 10^{M\ell}}{\ell! \cdot 10^{f(t,\ell)+g(t,\ell,2)}} + \frac{\ell \cdot 2^{\ell} \cdot 10^{g(t,\ell-1,2)+(\ell-1)f(t,\ell-1)}}{\ell! \cdot 10^{f(t,\ell)+g(t,\ell,2)}}$$

We now show that each individual term is at most 1, and therefore, $|\alpha_{t,\ell,\ell}| < 2$.

The first term can be simplified by plugging in the definition of $f(t,\ell)$ and using the fact that $g(t,\ell,r) > 2$.

$$\frac{\ell \cdot 10^{M\ell}}{\ell! \cdot 10^{f(t,\ell)+g(t,\ell,2)}} < \frac{1}{(\ell-1)!} \cdot 10^{M\cdot\ell-(\ell-1)!\nu_t-2}$$

Since $\ell \cdot M < (\ell-1)!\nu_t$, it follows that

$$\frac{\ell \cdot 10^{M\ell}}{\ell! \cdot 10^{f(t,\ell)+g(t,\ell,2)}} < 1.$$

For the second term, note that $f(t,\ell) = (\ell-1)!\nu_t = (\ell-1) \cdot (\ell-2)!\nu_t = (\ell-1)f(t,\ell-1)$ and $g(t,\ell,2) - g(t,\ell-1,2) = 2\ell - 2 \geq 2$ for $\ell \geq 2$. Also, for $\ell \geq 2$, we have $\frac{2^{\ell}}{(\ell-1)!} \leq 4$. Therefore,

$$\frac{\ell \cdot 2^{\ell} \cdot 10^{g(t,\ell-1,2)+(\ell-1)f(t,\ell-1)}}{\ell! \cdot 10^{f(t,\ell)+g(t,\ell,2)}} = \frac{2^{\ell}}{(\ell-1)!} \cdot 10^{g(t,\ell-1,2)-g(t,\ell,2)} \cdot 10^{(\ell-1)f(t,\ell-1)-f(t,\ell)}$$

$$\leq 4 \cdot 10^{-1} < 1$$

Now that we have established $|\alpha_{t,\ell,\ell}| < 2$, we show that $\sum_{r=2}^{\ell} |\alpha_{t,\ell,r}| < 10^{\nu-nd}$. We split this summation into two terms as

$$\sum_{r=2}^{\ell} |\alpha_{t,\ell,r}| = \sum_{r=2}^{\ell-1} |\alpha_{t,\ell,r}| + |\alpha_{t,\ell,\ell}|.$$

From the definition of $\alpha_{t,\ell,r}$ for $1 < r < \ell$, we have

$$\sum_{r=2}^{\ell-1} |\alpha_{t,\ell,r}| = \sum_{r=2}^{\ell-1} 10^{g(t,\ell,r)} < 10^{g(t,\ell,\ell-1)+1}.$$

Since $g(t,i,r)$ is increasing in t,i,r, $g(t,\ell,\ell-1)+1 \leq g(n,d,d-1)+1 = nd^2$. Recall that $\nu = n^2$, and therefore, for any $d = o(\sqrt{n})$, we have,

$$10^{g(t,\ell,\ell-1)+1} \leq 10^{nd^2} \leq 10^{\nu-nd-1}.$$

Therefore, it follows that

$$\sum_{r=2}^{\ell} |\alpha_{t,\ell,r}| \leq \sum_{r=2}^{\ell-1} |\alpha_{t,\ell,r}| + |\alpha_{t,\ell,\ell}| < 10^{\nu-nd-1} + 2 < 10^{\nu-nd}.$$

$\square$

Proof of Proposition 4.6. From the definition of $\alpha_{t,i,r}$ in Algorithm 2, we know that each auxiliary variable is a $(\pm\frac{1}{2})$-linear combination of $\{\alpha_{t,i,r} \mid r \in [i]\}$. i.e

$$x_{t,i,j} = \sum_{r=1}^{i} u_r \alpha_{t,i,r} \quad \text{for some } u_r \in \{\pm\frac{1}{2}\}.$$

Therefore,

$$\frac{1}{2} \cdot |\alpha_{t,i,1}| - \frac{1}{2} \cdot \sum_{r=2}^{i} |\alpha_{t,i,r}| \leq \left| \sum_{r=1}^{i} u_r \alpha_{t,i,r} \right| \leq \frac{1}{2} \cdot |\alpha_{t,i,1}| + \frac{1}{2} \cdot \sum_{r=2}^{i} |\alpha_{t,i,r}|$$

Using Proposition 4.5 (e), we know that $\sum_{r=2}^{i} |\alpha_{t,i,r}| \leq 10^{\nu-nd}$ and from definitions, $\alpha_{t,i,1} = 10^{(i-1)!\nu_t}$.
Therefore,

$$\frac{1}{2} \cdot (10^{(i-1)!\nu_t} - 10^{\nu-nd}) \leq |x_{t,i,j}| \leq \frac{1}{2} \cdot (10^{(i-1)!\nu_t} + 10^{\nu-nd})$$

$\square$

Proof of Proposition 4.7. Let $t_1, t_2 \in [n], i_1, i_2 \in \{2, \cdots, d\}, j_1 \in [2^{i_1} - 1]$ and $j_2 \in [2^{i_2} - 1]$. If $(t_1, i_1, j_1) = (t_2, i_2, j_2)$, then from Claim 5.2, we know $|x_{t_1,i_1,j_1} - y_{t_1,i_1,j_1}| = \alpha_{t,i,2} \neq 0$ and it follows that $x_{t_1,i_1,j_1} \neq y_{t_1,i_1,j_1}$. Now we show that if $(t_1, i_1, j_1) \neq (t_2, i_2, j_2)$, then $x_{t_1,i_1,j_1} \neq x_{t_2,i_2,j_2}$. The proof holds if either or both the $x_{t,i,j}$'s replaced with $y_{t,i,j}$. Let,

$$x_{t_1,i_1,j_1} = u_1 \cdot 10^{(i_1-1)!\nu_{t_1}} + \sum_{r=2}^{i_1} u_r \cdot \alpha_{t_1,i_1,r} \quad \text{for some } u_r \in \{\pm\tfrac{1}{2}\}$$

and,

$$x_{t_2,i_2,j_2} = v_1 \cdot 10^{(i_2-1)!\nu_{t_2}} + \sum_{r=2}^{i_2} v_r \cdot \alpha_{t_2,i_2,r} \quad \text{for some } v_r \in \{\pm\tfrac{1}{2}\}.$$

If $x_{t_1,i_1,j_1} = x_{t_2,i_2,j_2}$, then on reordering the terms we get,

$$\left| u_1 \cdot 10^{(i_1-1)!\nu_{t_1}} - v_1 \cdot 10^{(i_2-1)!\nu_{t_2}} \right| = \left| \sum_{r=2}^{i_2} v_r \cdot \alpha_{t_2,i_2,r} - \sum_{r=2}^{i_1} u_r \cdot \alpha_{t_1,i_1,r} \right|$$

Note that if $\left| u_1 \cdot 10^{(i_1-1)!\nu_{t_1}} - v_1 \cdot 10^{(i_2-1)!\nu_{t_2}} \right|$ is non-zero, then using the fact that ν_{t_1} and ν_{t_2} are prime integers larger than n^4 we have,

$$\left| u_1 \cdot 10^{(i_1-1)!\nu_{t_1}} - v_1 \cdot 10^{(i_2-1)!\nu_{t_2}} \right| \geq 10^{n^4}$$

But from Proposition 4.5, (e),

$$\left| \sum_{r=2}^{i_2} v_r \cdot \alpha_{t_2,i_2,r} - \sum_{r=2}^{i_1} u_r \cdot \alpha_{t_1,i_1,r} \right| \leq \frac{1}{2} \sum_{r=2}^{i_2} |\alpha_{t_2,i_2,r}| + \frac{1}{2} \sum_{r=2}^{i_1} |\alpha_{t_1,i_1,r}| \leq 10^{\nu-nd}$$

which is a contradiction. Therefore, $t_1 = t_2, i_1 = i_2$ and $u_1 = v_1$.

Let us assume $t_1 = t_2 = t, i_1 = i_2 = i$ and $j_1 > j_2$. If $x_{t,i,j_1} = x_{t,i,j_2}$, then,

$$\sum_{r=2}^{i}(v_r - u_r) \cdot \alpha_{t,i,r} = 0$$

We know that $(v_r - u_r) \in \{0, \pm1\}$, so there exists a $\{0, \pm1\}$- linear combination of $\alpha_{t,i,r}$ equal to 0. If $u_r = v_r$ for every $r \in \{2, \cdots, i\}$, then $j_1 = j_2$ since each auxiliary variable is a distinct linear combination of the $\alpha_{t,i,r}$'s. So, there exists at least one $r \in \{2, \cdots, i\}$ such that $u_r \neq v_r$. Let r^* be the largest such r. We know that

$$0 = \left| \sum_{r=2}^{i}(v_r - u_r) \cdot \alpha_{t,i,r} \right| \geq \left| |\alpha_{t,i,r^*}| - |\sum_{r=2}^{r^*-1}(v_r - u_r) \cdot \alpha_{t,i,r}| \right|$$

But each $\alpha_{t,i,r} = 10^{g(t,i,r)}$ for $r \in \{2, \cdots, i-1\}$ is a distinct power of 10 and $|\alpha_{t,i,i}| < 2$. So, $\left| |\alpha_{t,i,r^*}| - \left| \sum_{r=2}^{r^*-1}(v_r - u_r) \cdot \alpha_{t,i,r} \right| \right| \neq 0$, which is a contradiction. Therefore, $j_1 = j_2$. $\qquad\square$

6 Existence of (Inhomogeneous) PTE Solutions over General Finite Fields

Recall that a solution to a PTE system of size s and degree d satisfies

$$
\begin{aligned}
x_1 + x_2 + \cdots + x_s &= y_1 + y_2 + \cdots + y_s \\
x_1^2 + x_2^2 + \cdots + x_s^2 &= y_1^2 + y_2^2 + \cdots + y_s^2 \\
&\cdots \\
x_1^d + x_2^d + \cdots + x_s^d &= y_1^d + y_2^d + \cdots + y_s^d.
\end{aligned}
$$

We will show that such a system always has a solution over a field $\mathbb{F} = \mathbb{F}_{p^\ell}$, for $d < |\mathbb{F}|^{1/2-\delta}$, for $\delta > 0$. In fact, the proof will also hold for inhomogeneous PTE systems such as (†).

Theorem 6.1. *Let $\mathbb{F}$ be a finite field, and let $r_1, r_2, \ldots, r_d \in \mathbb{F}$. Let d be a positive integer such that $d \le |\mathbb{F}|^{1/2-\delta}$. Then, there exists a solution in $\mathbb{F}$ to the system $\sum_{i=1}^s x_i^j - \sum_{i=1}^s y_i^j = r_j$, for $j \in [d]$, with $s = 3d/\delta$.*

Moreover, if $|\mathbb{F}|$ is a sufficiently large function of δ, then we can ensure that the x_i's and the y_i's are all distinct.

Let G be a group. An additive character of G is a a function $\chi : G \to \mathbb{C}$ such that $\chi(x + y) = \chi(x)\chi(y)$ for all $x, y \in G$. We will now define characters over groups of the form $\mathbb{F}^n$, where $\mathbb{F} = \mathbb{F}_{p^\ell}$ and p is a prime.

Let $\omega = e^{2\pi i/p}$ be a primitive pth root of unity, and let $Tr : \mathbb{F}_{p^\ell} \to \mathbb{F}_p$ be the Trace operator $Tr(x) = \sum_{i=0}^{\ell-1} x^{p^i}$. Then, an additive character of $\mathbb{F}^n = (\mathbb{F}_{p^\ell})^n$ is $\chi_a(x) = \omega^{Tr(a \cdot x)}$, where $a, x \in \mathbb{F}^n$, and $a \cdot x$ denotes the inner product over $\mathbb{F}^n$.

We will use of some results of [KS13]. Let μ be a distribution over vectors in $\mathbb{F}^n$, and denote by $\mu^{(s)}$ the distribution of $x_1 + x_2 + \ldots + x_s$, where the x_i's are picked independently from μ.

Theorem 6.2. *([KS13], Appendix B) Suppose that for some β, any non-trivial character χ of $\mathbb{F}^s$ satisfies*

$$
|\mathbb{E}_{x \sim \mu} \chi(x)| \le \beta.
$$

Then

$$
\sum_{x \in \mathbb{F}^n} \left| \mu^{(s)}(x) - \frac{1}{|\mathbb{F}|^n} \right| \le \beta^s |\mathbb{F}|^n,
$$

and so $\mu^{(s)}$ is $\beta^s |\mathbb{F}|^n$- close to the uniform distribution over $\mathbb{F}^n$ in statistical distance.

Recall the Weil/Deligne bound.

Theorem 6.3. *(Weil [Wei48], Deligne [Del78]) Let $f(x_1, x_2, \ldots, x_t)$ be a t-variate polynomial over $\mathbb{F}$ of degree at most $|\mathbb{F}|^{1/2-\delta}$, for some $\delta > 0$. Then, either $\chi(f(x))$ is constant for all $x \in \mathbb{F}$, or χ satisfies $|\mathbb{E}_{x \in \mathbb{F}} \chi(f(x))| \le |\mathbb{F}|^{-\delta}$.*

Proof of Theorem 6.1. For $x, y \in \mathbb{F}$, let $v_{x,y} = (x - y, x^2 - y^2, \ldots, x^d - y^d) \in \mathbb{F}^d$. Let μ be the distribution of $v_{x,y}$ when x, y are distributed independently and uniformly in $\mathbb{F}$. Note that for a nontrivial character χ_a with $a \in (\mathbb{F}^*)^d$, we have

$$
\mathbb{E}_{v_{x,y} \sim \mu} [\chi_a(v_{x,y})] = \mathbb{E}[\omega^{a \cdot v_{x,y}}] = \mathbb{E}_{x,y}[\omega^{g(x,y)}] = \mathbb{E}_{x,y}[\chi_a(g(x,y))]
$$

for the polynomial $g(x,y) = \sum_{i=1}^{d} a_i (x^i - y^i)$ of degree $d \leq |\mathbb{F}|^{1/2-\delta}$.

By Deligne's Theorem 6.3, we have that

$$|\mathbb{E}_{v_{x,y} \sim \mu} [\chi_a(v_{x,y})]| = |\mathbb{E}[\chi_a(g(x,y))]| \leq |\mathbb{F}|^{-\delta}. \tag{22}$$

Let $\mu^{(s)}$ be the distribution of $S = \sum_{i=1}^{s} v_{x_i,y_i}$ when we pick s vectors $v_{x_1,y_1}, v_{x_2,y_2}, \ldots, v_{x_s,y_s} \in \mathbb{F}^d$ independently, according to μ. Note that $\mu^{(s)}$ is precisely the distribution of $(\sum x_i - \sum y_i, \sum x_i^2 - \sum y_i^2, \ldots, \sum x_i^d - \sum y_i^d)$, when we pick the x_i's and y_i's independently and uniformly in $\mathbb{F}$.

By Theorem 6.2 and Equation (22), it follows that

$$\sum_{v \in \mathbb{F}^d} \left| \mu^{(s)}(v) - \frac{1}{|\mathbb{F}|^d} \right| \leq (|\mathbb{F}|^{-\delta})^s |\mathbb{F}|^d = |\mathbb{F}|^{-\delta s + d}.$$

Picking $s = 3d/\delta$, we get that $\mu^{(s)}((r_1, r_2, \ldots, r_d)) \geq |\mathbb{F}|^{-d} - |\mathbb{F}|^{-2d} > 0$.

We can also ensure that all x_i's and y_i's are distinct, by noticing that the $\Pr[|\{x_1, x_2, \ldots, x_s, y_1, \ldots, y_s\}| = 2s] = \prod_{i=0}^{2s-1} \frac{1}{|\mathbb{F}|-i} < (|\mathbb{F}| - 2s)^{-2s} < |\mathbb{F}|^{-2d} < |\mathbb{F}|^{-d} - |\mathbb{F}|^{-2d}$ for $|\mathbb{F}|$ being sufficiently large as a function of δ.

$\square$

7 Reduction from 1-in-3 SAT to $\mathrm{MSS}(d)$ over $\mathbb{F}_{p^\ell}$

We will choose prime $p = O(d!)$ and $\ell = \mathsf{poly}(n)$ for this reduction. To generate the field $\mathbb{F}_q = \mathbb{F}_{p^\ell}$, we consider an irreducible polynomial over $\mathbb{F}_p$ of degree ℓ. Let γ be a root of this polynomial in the algebraic closure of $\mathbb{F}_p$. Every element of $\mathbb{F}_q$ can then be generated as a linear combination of $1, \gamma, \cdots, \gamma^{\ell-2}, \gamma^{\ell-1}$ over $\mathbb{F}_p$ (We refer to [LN97] for a general treatment of finite fields.). Then, for $v = \sum v_i \gamma^i \in \mathbb{F}_q$, we will abuse notation and view v as the vector $(v_1, v_2, \ldots, v_{\ell-1})$. We define an analogue of the notion of "magnitude" used in the previous sections. For $v \in \mathbb{F}_q$, define $|v|$ to be the largest non-zero index $i \in [\ell]$ in the vector representation of v. Note that this definition of magnitude satisfies the property that $|u + v| \leq \max(|u|, |v|)$ for every $u, v \in \mathbb{F}_q$, and thus also satisfies that the triangle inequality.

We now sketch a proof of the reduction, which follows analogously to the proof over the rational field, with some small modifications, as described next.

An instance of $\mathrm{MSS}(d)$ consists of a tuple $\langle A, k, B_1, \ldots, B_d \rangle$. Similar to the rational field reduction, each variable $(z_t, \overline{z_t})$ is mapped to $2^{d+1} - 2$ distinct elements $\{a_t\} \cup \{x_{t,i} \mid i \in [2^d - 2]\}$ (corresponding to z_t) and $\{b_t\} \cup \{y_{t,i} \mid i \in [2^d - 2]\}$ (corresponding to $\overline{z_t}$). Let $\{a'_t, b'_t : t \in [n]\}$ be the elements of $\mathbb{F}_q$ produced by the reduction of 1-in-3 SAT to Subset-Sum defined as follows:

- The vector representations of a'_t and b'_t consist of two parts: a clause region consisting of the leftmost m coordinates and a variable region consisting of the next n indices.

- In the variable region, a'_t and b'_t have a 1 at the t-th index and 0's at the other indices. Denote that by $(a_t)'^v$.

- In the clause region, for every $j \in [m]$, a'_t (resp. b'_t) has a 1 at the jth location if z_t (resp. $\overline{z_t}$) appears in clause j, and a 0 otherwise. We denote the clause part of a'_t by $(a_t)'^c$.

- $a'_t = (a_t'^c, a_t'^v, 0^{\ell-m-n})$. Similarly for b'_t.

34

- The target B is set to the element whose field representation is the vector which takes 1's in the first $m + n$ indices and 0 everywhere else. i.e. $B = (1^m, 1^n, 0^{\ell-m-n})$.

Define,
$$a_t = (0^\nu, a_t'^c, a_t'^v, 0^{\ell-\nu-m-n}) \quad \text{and,} \quad b_t = (0^\nu, b_t'^c, b_t'^v, 0^{\ell-\nu-m-n}).$$

For each $t \in [n]$, we will explicitly construct two sets of $2^d - 2$ *auxiliary variables*, $X_t = \{x_{t,i} \mid i \in [2^d - 2]\}$ and $Y_t = \{y_{t,i} \mid i \in [2^d - 2]\}$ which satisfy the following properties:

Property (1): $\displaystyle\sum_{x \in X_t} x = \sum_{y \in Y_t} y = 0$

Property (2): $\displaystyle\sum_{x \in X_t} x^k - \sum_{y \in Y_t} y^k = b_t^k - a_t^k$ for every $k \in \{2, \ldots, d\}$.

Property (3): Additionally, an appropriately scaled set of auxiliary variables, can be shown to satisfy the bimodal property. Namely, for any subset $S \subseteq \bigcup_{t \in [n]} X_t \cup Y_t$, and a scaling factor

$K = \gamma^h$, where $h = \mathsf{poly}(n, d!)$, either

$$\left| \sum_{s \in S} \gamma^h s \right| > h + n^4 \quad \text{or} \quad \left| \sum_{s \in S} \gamma^h s \right| < h + \nu.$$

Define the set $A = \bigcup_{t \in [n]} \{a_t\} \cup \{b_t\} \cup X_t \cup Y_t$. The targets $B_1, \ldots, B_d$ are defined as follows:

$$B_1 = (0^\nu, 1^m, 1^n, 0^{\ell-\nu-m-n}),$$

$$B_k = \sum_{t=1}^n a_t^k + \sum_{t=1}^n \sum_{x \in X_t} x^k \quad \text{for every } k \in \{2, \ldots, d\}$$

Note that $a_t = \gamma^\nu \cdot a_t'$. Similarly, $b_t = \gamma^\nu \cdot b_t'$ and $B_1 = \gamma^\nu \cdot B$.

We now define a scaled version of the subset sum instance over the finite fields. Let $h = \mathsf{poly}(n, d!)$ and let $K = \gamma^h$, be the scaling factor . Scaling of all the elements of the instance of $\mathrm{MSS}(d)$ is roughly equivalent to scaling the rational solutions by a large power of 10. The scaling of all the variables maintains Properties 1, and 2 of the auxiliary variables and also satisfies the solution to achieve Property 3.

Let $A_h = \{\gamma^h a \mid a \in A\}$. $B_{k,h} = \gamma^{kh} B_k$ for every $k \in \{2, \ldots, d\}$. The following lemma shows that the $\mathrm{MSS}(d)$ instance and its scaled version as defined above are equivalent.

Lemma 7.1. *Let $h = \mathsf{poly}(n, d!)$ and $K = \gamma^h$ be the scaling factor. There exists a subset $S \subseteq A$ such that for every $k \in [d]$*
$$\sum_{s \in S} s^k = B_k.$$

if and only if there exists a subset $S_h \subseteq A_h$ such that for every $k \in [d]$
$$\sum_{s \in S_h} s^k = B_{k,h}.$$

The proof of Lemmas 7.1 is straightforward and follows from the fact that the moment equations in MSS(d) are homogeneous and therefore scaling all the variables and the targets does not change the problem.

We can then state the analogous statement of Lemma 3.3, which implies the NP-hardness of MSS(d) over $\mathbb{F}_q$.

Lemma 7.2. *There exists a satisfying assignment to a 3-SAT instance $\phi(z_1, \ldots, z_n)$ if and only if there exists a subset $S \subseteq A_h$ such that for every $k \in [d]$,*

$$\sum_{s \in S} s^k = B_{k,h}.$$

The proof of Lemma 7.2 follows from the properties of the auxiliary variables stated above and all the steps of the proof over the rationals can be carried over here, because we chose to scale the instance by a large enough power of γ, and we chose p and ℓ large enough, in order to ensure that there is no wrapping around when we add terms with large magnitudes.

8   Conclusion

The main open question that comes up from this work is to explicitly and efficiently construct degree-d PTE solutions of size subexponential in d (Problem 1.3). It would also be very interesting to prove analogous NP-hadness results for Bounded Distance Decoding of Reed-Solomon codes in the case where preprocessing is allowed. Finally, our NP-hardness results for Reed-Solomon codes apply to the case where the field size is exponential in the block length N; it would be very interesting to prove analogous NP-hardness results for smaller fields.

Acknowledgements

References

[ABSS97] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. 7

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of mathematics*, pages 781–793, 2004. 14

[BI94] P. Borwein and C. Ingalls. The Prouhet-Tarry-Escott problem revisited. *Enseign. Math*, 40:3–27, 1994. 3

[BLP03] Peter Borwein, Petr Lisonek, and Colin Percival. Computational investigations of the Prouhet-Tarry-Escott problem. *Math. Comput.*, 72(244):2063–2070, 2003. 3

[BW86] E.R. Berlekamp and L.R. Welch. Error correction for algebraic block codes, 1986. US Patent 4,633,470. 1

[Che08] Qi Cheng. Hard problems of algebraic geometry codes. *IEEE Trans. Information Theory*, 54(1):402–406, 2008. 7

[CW07] Qi Cheng and Daqing Wan. On the list and bounded distance decodability of Reed-Solomon codes. *SIAM J. Comput.*, 37(1):195–209, 2007. 1

[CW10] Q. Cheng and D. Wan. Complexity of decoding positive-rate primitive Reed-Solomon codes. *IEEE Trans. Inf. Theory*, 56(10):5217–5222, 2010. 1

[Del78] P. Deligne. Aplications de la formule des traces aux sommes trigonometriques. *SGA $4\frac{1}{2}$ Springer Lecture Notes in Math*, 569, 1978. 33

[Dic13] Leonard Eugene Dickson. *History of the Theory of Numbers, Volume II: Diophantine Analysis*, volume 2. Courier Corporation, 2013. 3

[DKRS03] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003. 7

[DMS03] Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Information Theory*, 49(1):22–37, 2003. 7

[ES59] P Erdos and G Szekeres. On the product $\prod$ n k= 1 (1- zak), acad. *Serbe Sci. Publ. Inst. Math*, 13:29–34, 1959. 3

[FM04] Uriel Feige and Daniele Micciancio. The inapproximability of lattice and coding problems with preprocessing. *J. Comput. Syst. Sci.*, 69(1):45–67, 2004. 7

[GGG15] Venkata Gandikota, Badih Ghazi, and Elena Grigorescu. On the NP-hardness of bounded distance decoding of Reed-Solomon codes. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, pages 2904–2908. IEEE, 2015. 1, 2, 4, 7

[GKS10] Parikshit Gopalan, Subhash Khot, and Rishi Saket. Hardness of reconstructing multivariate polynomials over finite fields. *SIAM J. Comput.*, 39(6):2598–2621, 2010. 7

[GR08] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on Information Theory*, 54(1):135–150, 2008. 7

[GRS00] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. Learning polynomials with queries: The highly noisy case. *SIAM J. Discrete Math.*, 13(4):535–570, 2000. 7

[GS99] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inf. Theory*, 45(6):1757–1767, 1999. 1, 7

[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. 1, 2, 7

[Hås01]   Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. 7

[Hua82]   Loo Keng Hua. *Introduction to number theory*. Springer, 1982. 3

[HW79]   Godfrey Harold Hardy and Edward Maitland Wright. *An introduction to the theory of numbers*. Oxford Science Publications. Clarendon Press, Oxford, 1979. 3

[KS13]   Swastik Kopparty and Shubhangi Saraf. Local list-decoding and testing of random linear codes from high error. *SIAM J. Comput.*, 42(3):1302–1326, 2013. 33

[KV03]   Ralf Koetter and Alexander Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. *IEEE Transactions on Information Theory*, 49(11):2809–2825, 2003. 7

[Las09]   Jean Bernard Lasserre. *Moments, positive polynomials and their applications*, volume 1. World Scientific, 2009. 2

[LN97]   Rudolf Lidl and Harald Niederreiter. *Finite fields*, volume 20. Cambridge university press, 1997. 34

[LW08]   J. Li and D. Wan. On the subset sum problem over finite fields. *Finite Fields and Their Applications*, 14(4):911–929, 2008. 8

[Pet60]   W. Wesley Peterson. Encoding and error-correction procedures for the Bose-Chaudhuri codes. *IRE Transactions on Information Theory*, 6(4):459–470, 1960. 1

[Pro51]   Eugéne Prouhet. Mémoire sur quelques relations entre les puissances des nombres. *CR Acad. Sci. Paris*, 33(225):1851, 1851. 3

[PV05]   Farzad Parvaresh and Alexander Vardy. Correcting errors beyond the guruswami-sudan radius in polynomial time. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 285–294, 2005. 7

[Reg04]   Oded Regev. Improved inapproximability of lattice and coding problems with preprocessing. *IEEE Trans. Information Theory*, 50(9):2031–2037, 2004. 7

[RS60]   Irving S Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, 8(2):300–304, 1960. 1

[RW14]   Atri Rudra and Mary Wootters. Every list-decodable code for high noise has abundant near-optimal rate puncturings. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 764–773, 2014. 7

[Sch78]   T. J. Schaefer. The complexity of satisfiability problems. In *STOC*, pages 216–226. ACM, 1978. 9

[Sho09]   Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge university press, 2009. 14

[Sta99]   R P Stanley. *Enumerative Combinatorics, vol.2*. Cambridge University Press, 1999. 8

[Sud97]   M. Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *J. Complexity*, 13(1):180–193, 1997. 1, 7

[Var97]   Alexander Vardy. Algorithmic complexity in coding theory and the minimum distance problem. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 92–109, 1997. 7

[Wei48]   A. Weil. Sur les courbes algebriques et les varietes qui s'en deduisent. *Actualities Sci. et Ind.*, (1041), 1948. 33

[Wri59]   E. M. Wright. Prouhet's 1851 solution of the Tarry-Escott problem of 1910. *The American Mathematical Monthly*, 66(3):199–201, 1959. 3