Regular Syndrome Decoding Is Easier Than Previously Estimated

Jiseung Kim[1], Minkyu Kim[2], Changmin Lee[3], and Hochang Lee[2]

[1] Jeonbuk National University
jiseungkim@jbnu.ac.kr
[2] The Affiliated Institute of ETRI
mkkim@nsr.re.kr, lhc254@nsr.re.kr
[3] Korea University
changminlee@korea.ac.kr

Abstract. The Regular Syndrome Decoding (RSD) problem is a structured variant of classical Syndrome Decoding and underlies a growing number of efficient cryptographic constructions, including zero-knowledge protocols, pseudorandom correlation generators, and MPC-in-the-Head signatures. Although RSD is generally believed to be computationally hard, recent works have shown that algebraic attacks can outperform information-set decoding on important RSD parameter regimes.

Building on this line of work, we revisit the concrete hardness of RSD and show that many widely used parameter sets are substantially easier to solve than previously estimated. This improvement stems from a refined hybrid-algebraic attack that exposes and removes a previously overlooked source of concrete complexity: the joint blockwise placement of zero-coordinate guesses and syndrome-based variable eliminations. Rather than treating the two guessing stages and linear elimination as largely separate choices, our method coordinates them across all regular blocks so that the guessing penalty is balanced while sufficient equations are retained for linearization. This gives a systematic block-balancing principle for the entire guess–reduce–solve pipeline.

We reassess several RSD-based applications, including Ferret (CCS 2020), Wolverine (IEEE S&P 2021), and committed VOLE (CCS 2025). Our estimates place at least one parameter set per application below its 128-bit target, with losses approaching 67 bits. On PCG benchmarks, we improve previous best attack costs by up to 29.77 and 25.91 bits over non-binary and binary fields, respectively. Moreover, for the revised 128-bit PCG parameter sets of Liu et al. (EUROCRYPT 2024) evaluated here, estimated attack costs are all below 2^{111} field operations. These results substantially revise the concrete-security landscape of RSD and indicate that several existing parameter choices overestimate the hardness of their underlying RSD instances.

Keywords: regular syndrome decoding · algebraic attack · linearization.

1 Introduction

The regular syndrome decoding (RSD) problem is a structured variant of the classical syndrome decoding (SD) problem and has emerged as an increasingly important hardness assumption in modern cryptography. Its appeal comes from the additional structure imposed on the error vector: rather than allowing the nonzero coordinates to occur at arbitrary positions, RSD requires them to be distributed regularly across prescribed blocks. This structure can be exploited by protocol designers to obtain substantially more efficient constructions than those based on an unstructured decoding problem, while retaining a decoding-based security assumption.

As a result, RSD and closely related regular-noise assumptions now appear in a broad range of cryptographic primitives and protocols. They underlie pseudorandom correlation generators (PCGs), including constructions for efficiently generating correlated randomness and multiparty Beaver triples [7, 18, 22]; correlated oblivious transfer and VOLE-based protocols, including Ferret and committed VOLE [26, 31]; zero-knowledge protocols such as Wolverine [30]; and secure multiparty computation frameworks such as TinyKeys [14, 15]. RSD has also become particularly relevant to post-quantum signatures, where it is used in several constructions based on the MPC-in-the-Head paradigm [1,9–11,23].Thus, the concrete hardness of RSD is no longer relevant to a single cryptographic construction: security estimates for RSD directly affect the parameter selection and security claims of a diverse collection of cryptographic systems.

This growing use of RSD makes its concrete security particularly important. A moderate improvement in the best known algorithm for solving RSD can translate directly into a loss of security for several seemingly unrelated protocols that instantiate the same underlying hardness assumption. Consequently, understanding the precise cost of solving concrete RSD instances, rather than considering only its asymptotic hardness, is essential for assessing the security margins of existing constructions.

In the standard syndrome decoding problem, one is given a linear code defined by a parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ and a syndrome vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$. The goal is to find an error vector $\mathbf{x} \in \mathbb{F}_q^n$ such that $\mathbf{Hx} = \mathbf{s}$ and $\mathbf{x}$ has Hamming weight h, i.e., exactly h of its coordinates are nonzero. The RSD problem imposes an additional regularity constraint on the support of $\mathbf{x}$. The coordinate set $1, \ldots, n$ is partitioned into h disjoint blocks of size $\beta := n/h \in \mathbb{Z}$, and exactly one coordinate in each block is required to be nonzero. Thus, whereas standard syndrome decoding allows the support to be chosen arbitrarily among all weight-h vectors, RSD restricts the support to one position per block. This additional combinatorial structure is the defining feature that distinguishes RSD from ordinary syndrome decoding and is the main source of the specialized algebraic techniques considered in this work.

1.1 Algebraic attacks on the RSD problem

Information-set decoding (ISD) and its variants are the standard combinatorial tools to solve syndrome decoding problems [4, 21, 24, 25]. For RSD, however, algebraic attacks can become competitive and can even outperform ISD in some low code rate parameter regimes because the regularity constraints admit a compact polynomial representation. In particular, if two coordinates belong to the same block, regularity implies quadratic relations of the form $x_i x_j = 0$ [8]. Together with the linear syndrome equation, these relations define a multivariate polynomial system whose solution reveals the regular error vector.

The direct algebraic attack quickly becomes expensive as the number of variables grows. Consequently, the most effective attacks use a hybrid strategy: they first guess part of the support information, reduce the number of variables with the syndrome equations, and then solve the resulting polynomial system by algebraic techniques such as linearization or Gröbner-basis methods. The central concrete trade-off is therefore not only how many coordinates are guessed, but also where those guesses and eliminations occur across the regular blocks.

Wang et al. [28] recently gave the strongest hybrid algebraic attack for many RSD parameters considered in this work. Their method follows a guess–reduce–solve paradigm: it first guesses error-free positions, uses Gaussian elimination on the syndrome equation to eliminate variables, and then applies a second hybrid algebraic step to the reduced quadratic system. The second guessing stage, however, is concentrated in a restricted set of blocks, while the variables eliminated by the syndrome equations are also allocated unevenly. As we show below, this blockwise allocation creates an avoidable loss in guessing probability and leaves substantial room to reduce the concrete attack complexity.

1.2 Our contribution

Our main contribution is a stronger hybrid-algebraic algorithm for concrete RSD instances. The key novelty is a new way to *coordinate the existing primitives*: zero guessing, syndrome-based elimination, preprocessing, and the final algebraic solve are treated as coupled blockwise operations rather than optimized in isolation. This viewpoint reveals that a substantial part of the complexity of prior attacks comes from an avoidable imbalance in where these operations are placed across the regular blocks.

The source of the previous complexity loss A hybrid algebraic attack benefits from guessing error-free coordinates because the resulting polynomial system contains fewer variables. In RSD, however, each zero guess also removes quadratic equations of the form $x_i x_j = 0$ from the same block. Consequently, the cost of the attack is governed by a trade-off between the probability of a correct guess and the number of equations that remain available after the guess.

This trade-off depends strongly on how the guesses are distributed among the regular blocks. For example, guessing two zero coordinates in one block succeeds

with probability $1-2/\beta$ and removes $2\beta-3$ quadratic equations, whereas guessing one zero coordinate in each of two distinct blocks succeeds with probability $(1-1/\beta)^2 > 1-2/\beta$ while removing $2(\beta-1)$ equations. Thus, concentrating guesses may preserve slightly more equations, but it can incur a substantially larger guessing penalty.

Re-examining the attack of Wang et al. [28],we identify exactly this effect at scale: the second zero-guessing stage and the syndrome-based eliminations occupy an unbalanced subset of blocks. Because these operations compete for the same blockwise coordinates, their interaction amplifies the guessing penalty. This is a structural inefficiency of the attack schedule, rather than a limitation of linearization itself.

A globally balanced guess–reduce–solve technique Our main algorithmic contribution is a globally balanced organization of the entire guess–reduce–solve pipeline. Previous approaches optimize or place the individual stages largely in isolation, which can lead to a substantial loss in the overall guessing probability. In contrast, our algorithm jointly coordinates all operations that determine the final polynomial system: the first zero guesses, the variables eliminated using the syndrome equations, the variables involved in preprocessing, and the second zero guesses. These operations are distributed across the regular blocks as evenly as the discrete parameters permit.

The key point is that the gain does not come from simply increasing or decreasing the number of guessed coordinates, but from changing their interaction with syndrome elimination at the block level. By balancing the syndrome eliminations, our method preserves more uniformly sized candidate sets for the subsequent zero-guessing stage. The AM–GM principle underlying our probability analysis then shows that this joint balancing improves the overall success probability while retaining sufficiently many equations for linearization. Thus, the proposed strategy removes a source of concrete complexity that remains even when the individual ingredients of the hybrid attack are already well optimized.

We complement this algorithmic improvement with two refinements that further sharpen the concrete complexity. First, we revisit the preprocessing cost and identify operations within the preprocessing procedure that are essentially free. Explicitly accounting for these operations yields a tighter estimate of the actual preprocessing cost, without changing the underlying preprocessing algorithm. Second, we exploit the overdetermined structure of the final random linear system more precisely, allowing incorrect guesses to be rejected with fewer excess equations.

These contributions improve the attack at complementary levels. The global blockwise balancing substantially increases the probability of reaching the correct final system, while the refined complexity analysis reduces the estimated algebraic cost of processing each resulting system. Their combination leads to the significant concrete complexity reductions reported in Table 1 and Section 5.

The source of these improvements is already visible in a representative benchmark instance. Consider the RSD instance with $n = 2^{20}$, $k = 32771$, and

$h = 2467$ over $\mathbb{F}_{2^{128}}$. As reported in Table 2, the previous best hybrid algebraic attack has an estimated complexity of $2^{114.27}$ field operations, whereas our attack requires only $2^{84.50}$. Notably, the two attacks guess almost the same number of zero coordinates: the previous attack uses 11,819 zero guesses, as reported in Table 5 of [28], while our optimized attack uses 11,826, as reported in Table 1. Thus, the improvement comes primarily from how these guesses are distributed across the regular blocks, rather than from the number of guesses itself. For the corresponding zero-guessing distributions, the probability that all guesses are correct is approximately $2^{-62.18}$ for the previous attack and $2^{-40.37}$ for our balanced strategy; the general success-probability analysis is given in Section 3.3. This difference alone accounts for about 21.8 of the 29.8 bits of complexity reduction. The remaining gain comes from the reduced algebraic cost resulting from balanced variable elimination, refined preprocessing, and a tighter treatment of the final overdetermined system.

Concrete security consequences We reassess RSD instances used as concrete hardness assumptions in seven cryptographic applications. In every application, either all or a subset of the parameter sets targeting 128-bit security fall below their claimed level under our estimates, with security losses reaching up to 66 bits. For the refined 128-bit parameter sets of [20], none achieves the claimed security level: the estimated losses range from 27 to 47 bits over non-binary fields and from 17 to 48 bits over binary fields.

The impact of our refinements is also evident on benchmark RSD instances. For parameter sets from [7, 20] that have repeatedly served as benchmarks in subsequent cryptanalytic work [8, 12, 17, 28], our estimates lower the previous best attack complexity by up to 29 bits over non-binary fields and 25 bits over binary fields. The refined attack gives the lowest complexity for all listed benchmark instances except two binary cases. In particular, although R-ISD previously outperformed algebraic attacks on 10 of the 14 binary benchmark parameters, the refined algebraic attack is best on 12 of the 14 parameters considered here.

Table 1 summarizes these concrete complexity estimates together with the corresponding optimized algorithmic parameters. Section 5 gives the full comparison and discusses the implications for the affected constructions.

These results shift the emphasis from proposing another RSD solver to revising the current understanding of the concrete hardness of RSD. In the parameter regimes considered here, the gap between previously reported attacks and the refined analysis is large enough to invalidate all or part of the claimed secure parameter sets in each application we examine.

1.3 Related work

Briaud and Øygarden [8] initiated the modern study of algebraic attacks on RSD. Their analysis characterizes the witness degree of the corresponding polynomial systems and shows that algebraic techniques can outperform previously known approaches on several concrete parameters. Their attack already uses blockwise

Table 1. Summary for complexity estimates for the various parameter sets and corresponding algorithmic parameters. The algorithmic parameters are specified in Section 3, while the symbols ≀, †, and ‡ are defined in Section 5.

Source	n	k	h	q	our	$(\eta_1, \eta, \eta_{\text{pre}})$
Stationary SD [18]	2^{16}	2^{15}	69	2^{128}	97.90	(11011,28235,4477)
	2^{14}	2^{13}	74		95.57	(3948,7092,1069)
	2^{12}	2^{11}	90		101.90	(1301,1768,233)
	2^{10}	2^{9}	107		104.13	(339,404,46)
	2^{16}	2^{15}	69	2	97.58	(13687,28156,4487)
	2^{14}	2^{13}	74		94.52	(4398,7007,1080)
	2^{12}	2^{11}	90		96.46	(1277,1667,244)
	2^{10}	2^{9}	107		76.40†	(229,286,57)
PCGs Beaver [22]	2^{20}	2^{18}	47	2	73.97	(262086,262086,0)
	2^{21}	2^{18}	31		64.01	(262107,262107,0)
	2^{21}	2^{19}	45		76.07	(524221,524221,0)
	2^{22}	2^{19}	30		66.76	(524250,524250,0)
	2^{22}	2^{20}	44		78.58	(1048523,1048523,0)
	2^{23}	2^{21}	42		80.67	(2097073,2097073,0)
	2^{24}	2^{22}	41		83.05	(4194243,4194243,0)
	2^{25}	2^{23}	39		85.02	(8388563,8388563,0)
	2^{26}	2^{24}	37		86.99	(16777173,16777173,0)
Committed VOLE [26, 27]	1456	760	91	2	90.89†	(484,585,83)
	1472	762	92		90.70†	(483,585,84)
	1488	765	93		90.66†	(483,586,85)
	1272	433	159		61.90†	(134,198,75)
	1296	434	162		61.02†	(128,194,77)
Ferret [31]	609728	36288	1269	2	98.38	(5296,17832,16578)
	10805248	589760	1319		110.23	(16035,299181,289259)
Wolverine [30]	642048	19870	2508	$2^{61}-1$	82.78	(230,7221,12648)
	10805248	589760	1319		110.23	(6958,300538,289221)
TinyKeys [15]	1280	860	80	2	118.60‡	(664,726,53)
MPCitH [9]	1736	776	217	2	125.63†	(434,484,74)

Source	n	k	h	$q = 2^{128}$		$q = 2$	
				our	$(\eta_1, \eta, \eta_{\text{pre}})$	our	$(\eta_1, \eta, \eta_{\text{pre}})$
PCGs [7, 20]	2^{22}	64770	4788	86.70	(0,6904,57849)	85.07	(0,20803,39162)
	2^{20}	32771	2467	84.50	(0,11826,20843)	79.53	(0,9307,20895)
	2^{18}	15336	1312	93.45	(2211,7293,6986)	83.73	(1220,5944,7023)
	2^{16}	7391	667	110.45	(2752,4894,2326)	99.78	(2278,4200,2353)
	2^{14}	3482	338	115.96	(1746,2592,729)	103.64	(1510,2234,749)
	2^{12}	1589	172	120.21	(984,1251,197)	104.25	(836,1065,211)
	2^{10}	652	106	150.32≀	(509,550,31)	113.86‡	(396,434,41)
	2^{22}	64770	2735	86.51	(0,0,0)	83.10	(0,35404,25081)
	2^{20}	32771	1419	80.86	(0,5403,26014)	77.74	(0,15701,14296)
	2^{18}	15336	760	74.68	(0,5359,9272)	71.64	(0,6600,7271)
	2^{16}	7391	389	73.97	(154,4114,3092)	70.35	(221,3704,3113)
	2^{14}	3482	198	77.83	(790,2357,976)	73.56	(811,2143,992)
	2^{12}	1589	98	80.06	(702,1237,273)	74.75	(680,1127,285)
	2^{10}	652	57	89.31	(450,547,49)	78.66†	(400,482,57)

Source	n	h	$q = 2^{128}$			$q = 2$		
			k	our	$(\eta_1, \eta, \eta_{\text{pre}})$	k	our	$(\eta_1, \eta, \eta_{\text{pre}})$
PCGs [20]	2^{22}	4788	75396	87.93	(0,15018,60361)	80074	87.45	(0,21484,53785)
	2^{20}	2467	30978	80.90	(0,9996,20880)	33354	79.88	(0,9902,20883)
	2^{18}	1312	14796	89.32	(1643,6738,7001)	15996	88.75	(1910,6622,7005)
	2^{16}	667	6091	89.28	(1346,3542,2378)	7560	102.54	(2461,4376,2346)
	2^{14}	338	2909	94.93	(1096,1986,762)	3655	109.99	(1704,2417,739)
	2^{12}	172	1377	100.36	(726,1022,214)	1657	110.53	(916,1138,206)

balancing for an initial round of zero guesses, but does not combine this balancing with syndrome-based variable elimination.

Subsequent work further developed hybrid algebraic techniques. AGB2.0 [19] employs two rounds of guessing and distributes zero guesses across blocks, while Kim and Lee [17] combine two guessing stages with algebraic reduction. These approaches improve the concrete cost in different parameter regimes, but either retain the full syndrome system or separate the two guessing stages in a way that limits the reuse of preprocessing work.

Wang et al. [28] provide the strongest previous algebraic estimates for many of the instances studied here by combining zero guessing, syndrome-based variable elimination, preprocessing, and a final hybrid algebraic solve. Our analysis starts from this framework and isolates a remaining source of concrete inefficiency: the relevant operations are not jointly balanced across the regular blocks. Correcting this imbalance, together with two smaller refinements in preprocessing and analysis of the final overdetermined system, yields the complexity reductions reported in this paper.

Combinatorial attacks remain an important point of comparison. Among the ISD-style algorithms relevant to our parameter regime, the regular-decoding techniques of Esser and Santini [12] provide the strongest competing estimates and are therefore included in our concrete comparisons.

2 Preliminaries

2.1 Notation

For prime power $q \in \mathbb{N}$, the finite field with q elements is denoted by $\mathbb{F}_q$. For $a \in \mathbb{N}$, let us define $[a]$ as the set $\{1, \ldots, a\}$. For $i \in \mathbb{Z}$ and $h \in \mathbb{N}$, $[i]_h$ (resp. $i \bmod h$) denote the unique integers in $[h]$ (resp. $\{0, 1, \ldots, h-1\}$) which is congruent to i modulo h.

Matrices and vectors are denoted by boldface uppercase and lowercase letters, respectively. All vectors are represented as column vectors. Let $\mathbf{A}^\mathbf{t}$ denote the transpose of $\mathbf{A}$. In particular, $\mathbf{a}^\mathbf{t}$ is a row vector. We will often write $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_l)$ to indicate that $\mathbf{x}$ is a vertical concatenation of l column vectors $\mathbf{x}_i$. We also write $\mathbf{A} = [\mathbf{A}_1 | \cdots | \mathbf{A}_l]$ to denote that $\mathbf{A}$ is a horizontal concatenation of sub-matrices $\mathbf{A}_i \in \mathbb{F}_q^{r \times n_i}$ where $n = \sum_{i \in [l]} n_i$. In particular, $\mathbf{A} = [\mathbf{a}_1 | \cdots | \mathbf{a}_n] \in \mathbb{F}_q^{r \times n}$, which means that $\mathbf{A}$ is a horizontal concatenation of column vectors $\mathbf{a}_i \in \mathbb{F}_q^r$. For a vector $\mathbf{x} = (x_1, \ldots, x_n)$, a matrix $\mathbf{A} = [\mathbf{a}_1 | \cdots | \mathbf{a}_n]$, and a set $I = \{i_1 < \cdots < i_t\} \subseteq [n]$, we define $\mathbf{x}_I = (x_{i_1}, \ldots, x_{i_t})$ and $\mathbf{A}_I = [\mathbf{a}_{i_1} | \cdots | \mathbf{a}_{i_t}]$. Let $|\mathbf{x}|$ denote the Hamming weight of $\mathbf{x}$, i.e., the number of nonzero coordinates of $\mathbf{x}$.

2.2 The regular syndrome decoding problem

The main interest of this paper is to solve a variant of the syndrome decoding problem defined as follows.

Definition 1 (Computational RSD problem). *Given* $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}, \mathbf{s} \in \mathbb{F}_q^{n-k}$*, the regular syndrome decoding (RSD) problem, denoted as* $\mathrm{RSD}(n,k,h)$*, asks to find* $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_h) \in \mathbb{F}_q^n$ *such that* $\mathbf{Hx} = \mathbf{s}$ *and* $|\mathbf{x}_i| = 1$ *for* $i \in [h]$ *where* $\mathbf{x}_i \in \mathbb{F}_q^{n/h}$*.*

The case where $q = 2$ is referred to as binary $\mathrm{RSD}(n,k,h)$ to distinguish it from other cases.

The plain syndrome decoding (SD) problem, denoted $\mathrm{SD}(n,k,h)$, asks to find $\mathbf{x} \in \mathbb{F}_q^n$ such that $\mathbf{Hx} = \mathbf{s}$ and $|\mathbf{x}| = h$. The SD problem is known as NP-complete [5]. There is a provable security reduction from the SD problem to the RSD problem [13, Theorem 1]: given an SD instance $(\mathbf{H}, \mathbf{t})$, applying a random permutation in the columns of $\mathbf{H}$, one can obtain a RSD instance with probability $\binom{n/h}{1}^h / \binom{n}{h}$. Therefore, if the error rate $rho := h/n$ is sufficiently small, the difficulty of the RSD problem can be guaranteed by the difficulty of the SD problem. However, if the code rate is also small at the same time, it becomes vulnerable to attacks specific to the RSD problem, and this is the parameter regime that we have in mind.

2.3 Linear algebra over finite fields

Gaussian elimination Gaussian elimination is the primary operation used in the algebraic attacks against the RSD problem, and the complexity of algebraic attacks varies significantly depending on the algorithm used to perform Gaussian elimination.

A $m \times N$ matrix reduction takes $\mathcal{O}(m \cdot N \cdot \min\{m, N\})$ operations using the standard Gaussian elimination to reach reduced row echelon form. For more advanced methods like Strassen's algorithm, the complexity becomes $\mathcal{O}(m^{c_{\mathrm{mm}}})$ for square matrices of size m, and when applied to non-square matrices, it becomes $\mathcal{O}(\max\{m, N\}^{c_{\mathrm{mm}}})$, where $c_{\mathrm{mm}} \approx 2.8$. We do not consider algorithms with a smaller exponent (e.g. $c_{\mathrm{mm}} = 2.3713$ in [2]) since they perform less well due to their huge hidden constants. This is consistent with the choices made in previous RSD-related works [7, 17, 28].

If m is sufficiently larger than the size of the finite field q, then the striped matrix elimination [3] can be a better choice, and it has a complexity of $\mathcal{O}(m^2/\log_q m)$ row operations, and $\mathcal{O}(m^2 N/\log_q m)$ in total, which is asymptotically close to being optimal. In particular, the striped matrix elimination is useful for performing Gaussian elimination over the binary field $\mathbb{F}_2$.

We shall use the notation $\mathrm{C}_{\mathrm{ge}}(n, M)$ to denote the minimum complexity of performing Gaussian elimination over $m \times N$ matrix over $\mathbb{F}_q$, $q \neq 2$, i.e.

$$\mathrm{C}_{\mathrm{ge}}(m, N) = \min\left\{mN \min(m, N),\ \max(m, N)^{c_{\mathrm{mm}}}.\right\}$$

For the binary field case, it is represented by

$$\mathrm{C}_{\mathrm{ge}}^{\mathrm{bin}}(m, N) = \min\left\{mN \min(m, N),\ \max(m, N)^{c_{\mathrm{mm}}},\ m^2 N/\log_2 m\right\}.$$

Invertibility of random matrices Let $\mathrm{GL}(m,q)$ be a set of invertible $m \times m$ matrices over $\mathbb{F}_q$. Then the number of elements in $\mathrm{GL}(m,q)$ is computed as $|\mathrm{GL}(m,q)| = \prod_{i\in[m]} \left(q^m - q^{i-1}\right)$. Thus, the probability that a random $m \times m$ matrix is invertible is

$$p(m,q) = |\mathrm{GL}(m,q)| \,/ q^{m^2} = \prod_{i\in[m]} \left(1 - q^{i-m-1}\right).$$

For a fixed q and increasing m, the probability $p(m,q)$ converges to $p(\infty,q) := \prod_{i=1}^{\infty} \left(1 - q^{-i}\right)$ and it can be expanded as

$$p(\infty,q) = \textstyle\sum_{i=-\infty}^{\infty} (-1)^i q^{-\frac{i(3i-1)}{2}}$$

by the Euler's Pentagonal theorem. As q increases, $p(\infty,q)$ will increase monotonically to 1. For example, $p(\infty,2) \approx 0.2889$, $p(\infty,3) \approx 0.5605$, and $p(\infty,13) \approx 0.9170$.

Overdetermined linear system We examine how much overdetermined a random system of linear equations would be to have no solution. Note that the following is a more refined analysis of the result presented in [6] over general finite fields.

Assume that $m \geq N$. The image of a matrix $\mathbf{M} \in \mathbb{F}_q^{m\times N}$ contains a *random* vector $\mathbf{d} \in \mathbb{F}_q^m$ with probability $q^{\mathrm{rank}(\mathbf{M})-m}$. Since there are $q^{N-\mathrm{rank}(\mathbf{M})}$ solutions when the linear system $\mathbf{Mz} = \mathbf{d}$ is consistent, the expected number of solutions α is bounded as

$$\alpha \leq q^{\mathrm{rank}(\mathbf{M})-m} \cdot q^{N-\mathrm{rank}(\mathbf{M})} = q^{N-m}.$$

On the other hand, the number of $m \times N$ matrices over $\mathbb{F}_q$ of rank N is $\prod_{i\in[N]} \left(q^m - q^{i-1}\right)$, and so the probability that a random $m \times N$ matrix is full-rank ϵ is

$$\epsilon = \prod_{i\in[N]} \left(q^m - q^{i-1}\right) / q^{mN} = \prod_{i\in[N]} \left(1 - q^{i-m-1}\right) \geq (1 - q^{N-m})^N$$

which converges to $1 - Nq^{-(m-N)} + \mathcal{O}(N^2 q^{-2(m-N)})$ as $N \to \infty$. Note that $Nq^{-(m-N)} \ll 1$ in our parameter regime.

To summarize, we will set $m - N \geq \xi(N)$ so that a random matrix $\mathbf{M}$ is full-rank with overwhelming probability and the expected number of solutions of the linear system $\mathbf{Mz} = \mathbf{d}$ for a random $\mathbf{b}$ is upper-bounded by $q^{-\xi(N)}$. In particular, if the matrix is full-rank, then there is at most one solution. For example, when q is exponentially large such as $q = 2^{128}$, we take $\xi(N) = c_{\mathrm{excess}}$ for some constant c_{excess}.

3 Refined hybrid algebraic attack over non-binary fields

We first describe the refinements at a high level and explain why they reduce the concrete cost of the existing hybrid algebraic framework.

3.1 Overview

The refined attack addresses the regular syndrome decoding problem by transforming it into a quadratic multivariate system and subsequently applying a linearization technique. For the parameter regimes of interest, the resulting system is typically underdetermined for linearization: the number of monomials substantially exceeds the number of available equations. The hybrid strategy addresses this gap by guessing selected coordinates to be zero, thereby reducing the number of variables before linearization.

Let $\beta = \frac{n}{h} \in \mathbb{Z}$ and $\omega = \frac{n-k}{h} \in \mathbb{Z}$. Let $(\mathbf{H}, \mathbf{s})$ be an instance of the $\mathrm{RSD}(n, k, h)$ problem. It can be translated into a quadratic multivariate system by exploiting the regularity condition. This initial quadratic system $\mathcal{F}$ consists of n variables and $n - k + h\binom{\beta}{2}$ equations as

$$\mathbf{Hx} = \mathbf{s}, \qquad x_{i_1} x_{i_2} = 0$$

where $i_1 \neq i_2$ belong to the same block.

The refined attack starts with a first zero-guessing step, in which μ_1 variables in each block are guessed to be zero. The syndrome equation $\mathbf{Hx} = \mathbf{s}$ is used to eliminate ω variables from each block, for a total of $n - k$ variables. Next, ν variables in each block are selected as irrelevant variables for the second zero-guessing step. We then compute linear combinations of given $h\binom{\beta-\mu_1}{2}$ polynomials in which all monomials contain at most one irrelevant variable; it can be done by reducing the original polynomials in reduced row echelon form with an appropriate monomial ordering. Note that this type of preprocessing also has been used in previous algebraic attacks [6, 16, 19, 28]. The resulting system has $h\binom{\beta-\mu_1}{2} - hv(hv+1)/2$ equations of the form

$$\textstyle\sum_{i \in S_1} a_i\, x_i + \sum_{i \in S_1, j \in S_2} b_{i,j}\, x_i x_j + \sum_{j \in S_2} c_j\, x_j + \sum_{j_1, j_2 \in S_2} d_{j_1, j_2}\, x_{j_1} x_{j_2} = e$$

where S_1 is the set of positions of irrelevant variables, and S_2 is the remainder.

Finally, the attack performs a second zero-guessing step, in which additional μ_2 variables in each block are guessed to be zero. This produces a final system consisting of $k - h(\mu_1 + \mu_2)$ variables and $h\binom{\beta-\mu_1-\mu_2}{2} - hv(hv+1)/2$ equations of the form

$$\textstyle\sum_{i \in S_1} a_i\, x_i + \sum_{i \in S_1, j \in S_2'} b_{i,j}\, x_i x_j + \sum_{j \in S_2'} c_j\, x_j + \sum_{j_1, j_2 \in S_2'} d_{j_1, j_2}\, x_{j_1} x_{j_2} = e$$

where S_2' is a subset of S_2 of size $|S_2| - h\mu_2$. If the final system has at least as many equations as monomials, then it can be solved by linearization. Any candidate solution obtained in this way is then verified against the original RSD instance.

The reduction in concrete complexity comes from two key observations. On the one hand, after the first zero-guessing step, the variables eliminated by the linear equations are distributed as evenly as possible across the blocks, rather than being concentrated in only a few of them. This balanced allocation increases the success probability of the second zero-guessing step and, by the AM–GM inequality, is favorable for minimizing the overall guessing cost. Figure 1 provides

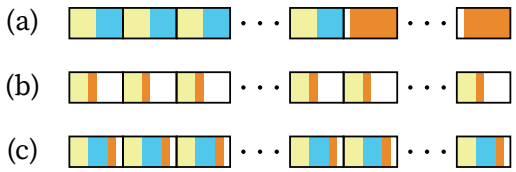


Fig. 1. Visualization of the blockwise distribution of the first zero guesses (yellow), the variables eliminated by the linear equations (blue), and the second zero guesses (orange) in [28], [19], and our balanced refinement. In (a), the two zero-guessing stages are separated and the linear eliminations are then filled from the front. In (b), the zero guesses are distributed almost evenly without accounting for the linear eliminations. In (c), all components are balanced across the blocks.

a visual comparison of the differences from the previous methods. On the other hand, we refine the complexity accounting of the preprocessing stage. In particular, we identify operations within the preprocessing procedure that can be performed at essentially negligible cost and explicitly account for this fact in our complexity estimate. This does not modify the underlying preprocessing algorithm, but yields a tighter estimate of its actual algebraic cost. Together with the balanced blockwise allocation, this refined accounting contributes to the improved concrete complexity obtained by our attack.

We next extend the balanced construction to the case where the number of components is not a multiple of the number of blocks. This general case is discussed in the following section.

3.2 Balanced strategy for guessing and eliminating variables

At a high level, we distribute the operations that remove variables as evenly as possible across the regular blocks. In particular, the zero guesses and the variables eliminated through the syndrome equations are assigned so that the number of removed variables differs by at most one between blocks. As a result, the remaining candidate sets stay nearly balanced across blocks, which improves the aggregate guessing probability without sacrificing the number of equations available for the subsequent algebraic solve.

Let $\beta = n/h \in \mathbb{Z}$. We use the following parameters to describe the blockwise allocation of guessed, eliminated, and retained variables. Let η be the number of variables that are guessed to be 0 in the first and second guessing stage. Let η_1 be the number of variables that are guessed to be 0 in the first guessing stage. Let $\eta_2 = \eta - \eta_1$ be the number of variables that are guessed to be 0 in the second guessing stage. Let η_{pre} be the number of variables that are free from being guessed as 0 in the first and second guessing stage.

Let $\mathbf{x} = (x_1, \ldots, x_n)$. We then call x_i the variable at position i. The set $[n]$ is partitioned as $[n] = \dot{\bigcup}_{i\in[h]} \mathsf{B}_i$ where $\mathsf{B}_i = \{(i-1)\beta + 1, \ldots, i\beta\}$, and each block B_i is also partitioned as $\mathsf{B}_i = \dot{\bigcup}_{j\in[5]} \mathsf{B}_{i,j}$ where

- $\mathsf{B}_{i,1}$: set of positions for the variables that are eliminated in the first guessing stage
- $\mathsf{B}_{i,2}$: set of positions for the variables that are eliminated by the given linear equations
- $\mathsf{B}_{i,3}$: set of positions for the variables that are not the subject of the second guessing stage
- $\mathsf{B}_{i,4}$: set of positions for the variables that are eliminated in the second guessing stage
- $\mathsf{B}_{i,5}$: set of positions for the remaining variables

Let us define $\bar{\mathsf{B}}_j = \cup_{i\in[h]}\mathsf{B}_{i,j}$ for $j \in [5]$.

Distributing linear equations. We describe a way to distribute variables removed by the linear system to each block. When the underlying field is non-binary, we have $n-k$ linear equations such as $\mathbf{Hx} = \mathbf{t}$. Let us define $\omega = \lfloor (n-k)/h \rfloor$ and $f_{\mathrm{lin}} = n-k \bmod h$. Then each of the first f_{lin} blocks has $\omega+1$ variables to be eliminated by the linear equations, and each of the remaining $h - f_{\mathrm{lin}}$ blocks has ω variables to be eliminated by the linear equations. Thus, we have

$$|\mathsf{B}_{i,2}| = \begin{cases} \omega+1 & \text{for} \quad i \in I_{\mathrm{lin}} := \{1,\ldots,f_{\mathrm{lin}}\} \\ \omega & \text{for} \quad i \in [h] \setminus I_{\mathrm{lin}} \end{cases}.$$

Guessing strategy We describe a way to distribute variables guessed as 0 to each block. Let $\mu := \lfloor \eta/h \rfloor$, and $f := \eta \bmod h$.

First stage of zero guessing. Let us define $\mu_1 = \lfloor \eta_1/h \rfloor$, $f_1 = \eta_1 \bmod h$, and $I_1 = \{[f_{\mathrm{lin}}+1]_h, \ldots, [f_{\mathrm{lin}}+f_1]_h\}$. The block B_i for $i \in I_1$ has μ_1+1 positions indicating variables guessed as 0, and each of the remaining $h-f_1$ blocks has μ_1 positions indicating variables guessed as 0, i.e.,

$$|\mathsf{B}_{i,1}| = \begin{cases} \mu_1+1 & \text{for} \quad i \in I_1 \\ \mu_1 & \text{for} \quad i \in [h] \setminus I_1 \end{cases}.$$

Second stage of zero guessing. Let us define $\mu_2 = \lfloor \eta_2/h \rfloor$, and $f_2 = \eta_2 \bmod h$. Each of the blocks indexed by numbers in $I_2 := \{[f_{\mathrm{lin}}+f_1+1]_h, \ldots, [f_{\mathrm{lin}}+f_1+f_2]_h\}$ has μ_2+1 variables guessed as 0, and each of the remaining $h-f_2$ blocks has μ_2 variables guessed as 0, i.e.,

$$|\mathsf{B}_{i,4}| = \begin{cases} \mu_2+1 & \text{for} \quad i \in I_2 \\ \mu_2 & \text{for} \quad i \in [h] \setminus I_2 \end{cases}.$$

Zero guessing in total. We claim that

$$|\mathsf{B}_{i,1}| + |\mathsf{B}_{i,4}| = \begin{cases} \mu+1 & \text{for} \quad i \in I := \{[f_{\mathrm{lin}}+1]_h, \ldots, [f_{\mathrm{lin}}+f]_h\} \\ \mu & \text{for} \quad [h] \setminus I \end{cases}.$$

If $0 \le f_1 + f_2 \le h$, then $I_1 \cap I_2 = \emptyset$ and

$$\left(|\mathsf{B}_{i,1}|, |\mathsf{B}_{i,4}| \right) = \begin{cases} (\mu_1 + 1, \mu_2) & \text{for} \quad i \in I_1 \\ (\mu_1, \mu_2 + 1) & \text{for} \quad i \in I_2 \\ (\mu_1, \mu_2) & \text{for} \quad i \in [h] \setminus (I_1 \cup I_2) \end{cases}.$$

In case of $f_1 + f_2 = 0$, $|\mathsf{B}_{i,1}| = \mu_1$, $|\mathsf{B}_{i,4}| = \mu_2$ for all $i \in [h]$, $f = 0$, $I = \emptyset$, and $|\mathsf{B}_{i,1}| + |\mathsf{B}_{i,4}| = \mu = \mu_1 + \mu_2 \quad \text{for} \quad i \in [h]$. In case of $0 < f_1 + f_2 < h$, the claim holds for $\mu = \mu_1 + \mu_2$, $f = f_1 + f_2$. In case of $f_1 + f_2 = h$, $f = 0$, $I = \emptyset$ and $|\mathsf{B}_{i,1}| + |\mathsf{B}_{i,4}| = \mu = \mu_1 + \mu_2 + 1 \quad \text{for} \quad i \in [h]$. If $f_1 + f_2 > h$, then $I_1 \cap I_2 \neq \emptyset$, $I_1 \cup I_2 = [h]$, and

$$\left(|\mathsf{B}_{i,1}|, |\mathsf{B}_{i,4}| \right) = \begin{cases} (\mu_1 + 1, \mu_2 + 1) & \text{for} \quad i \in I_1 \cap I_2 \\ (\mu_1 + 1, \mu_2) & \text{for} \quad i \in I_1 \setminus I_2 \\ (\mu_1, \mu_2 + 1) & \text{for} \quad i \in I_2 \setminus I_1 \end{cases}.$$

In this case, the claim holds for $\mu = \mu_1 + \mu_2 + 1$, $f = f_1 + f_2 - h$, $I = I_1 \cap I_2$.

Distributing preprocessed variables Let us define $\nu = \lfloor \eta_{\mathrm{pre}}/h \rfloor$ and $f_{\mathrm{pre}} = \eta_{\mathrm{pre}} \bmod h$. Let $I_{\mathrm{pre}} = \left\{ [f_{\mathrm{lin}} + f + 1]_h, \ldots, [f_{\mathrm{lin}} + f + f_{\mathrm{pre}}]_h \right\}$. Each of the blocks whose position belongs to the set I_{pre} has $\nu + 1$ variables free from being guessed as 0, and each of the remaining $h - f_{\mathrm{pre}}$ blocks has ν variables free from being guessed as 0, i.e.,

$$|\mathsf{B}_{i,3}| = \begin{cases} \nu + 1 & \text{for} \quad i \in I_{\mathrm{pre}} \\ \nu & \text{for} \quad i \in [h] \setminus I_{\mathrm{pre}} \end{cases}.$$

Requirement for block size Clearly, there is one condition that the block size must satisfy: the sum of the cardinality of the first four sub-blocks must not exceed β, i.e., $\sum_{j=1}^{4} |\mathsf{B}_{i,j}| \le \beta$ for all $i \in [h]$.

Let $a_1 := \lfloor (n - k + \eta)/h \rfloor$ and $b_1 := (n - k) + \eta \bmod h$. By using similar arguments used above, we can show that

$$|\mathsf{B}_{i,2}| + (|\mathsf{B}_{i,1}| + |\mathsf{B}_{i,4}|) = \begin{cases} a_1 + 1 & \text{for} \quad i \in J_1 := \left\{ 1, \ldots, b_1 \right\} \\ a_1 & \text{for} \quad i \in [h] \setminus J_1 \end{cases}$$

where $a_1 \in \{\omega + \mu, \omega + \mu + 1\}$. Moreover, letting $a_2 := \lfloor (n - k + \eta + \eta_{\mathrm{pre}})/h \rfloor$, $b_2 := b_1 + f_{\mathrm{pre}} \bmod h$, we also can show that

$$|\mathsf{B}_{i,1}| + |\mathsf{B}_{i,2}| + |\mathsf{B}_{i,3}| + |\mathsf{B}_{i,4}| = \begin{cases} a_2 + 1 & \text{for} \quad i \in J_2 := \left\{ 1, \ldots, b_2 \right\} \\ a_2 & \text{for} \quad i \in [h] \setminus J_2 \end{cases}$$

where $a_2 \in \{a_1 + \nu, a_1 + \nu + 1\}$. By combining these, we can reach at the inequality $\sum_{j=1}^{4} |\mathsf{B}_{i,j}| \le a_1 + \nu + 2 \le \omega + \mu + \nu + 3$. More precisely, we have $\max_{i \in [h]} \sum_{j=1}^{4} |\mathsf{B}_{i,j}| = \omega + \mu + \nu + \left\lceil \frac{f + f_{\mathrm{lin}} + f_{\mathrm{pre}}}{h} \right\rceil = \left\lceil \frac{\eta + (n-k) + \eta_{\mathrm{pre}}}{h} \right\rceil$.

Related numbers We use the following quantities to summarize the block partitions and the resulting equation counts.

- $\Delta_1 := \sum_{i\in[h]} \binom{|\mathsf{B}_i \setminus (\mathsf{B}_{i,1}\cup\mathsf{B}_{i,2})|}{2}$
$$= \begin{cases} h\binom{\beta-\omega-\mu_1-1}{2} + (h-f_{\mathrm{lin}}-f_1)(\beta-\omega-\mu_1-1) & \text{for} \quad f_{\mathrm{lin}}+f_1 \le h \\ h\binom{\beta-\omega-\mu_1-1}{2} - (f_{\mathrm{lin}}+f_1-h)(\beta-\omega-\mu_1-2) & \text{for} \quad f_{\mathrm{lin}}+f_1 > h \end{cases}.$$
- $\Delta_2 := \sum_{i\in[h]} \binom{|\mathsf{B}_i \setminus (\mathsf{B}_{i,1}\cup\mathsf{B}_{i,2}\cup\mathsf{B}_{i,4})|}{2}$
$$= \begin{cases} h\binom{\beta-\omega-\mu-1}{2} + (h-f_{\mathrm{lin}}-f)(\beta-\omega-\mu-1) & \text{for} \quad f_{\mathrm{lin}}+f \le h \\ h\binom{\beta-\omega-\mu-1}{2} - (f_{\mathrm{lin}}+f-h)(\beta-\omega-\mu-2) & \text{for} \quad f_{\mathrm{lin}}+f > h \end{cases}.$$
- $\zeta := \sum_{i\in[h]} \binom{|\mathsf{B}_{i,3}|}{2} = f_{\mathrm{pre}}\binom{\nu+1}{2} + (h-f_{\mathrm{pre}})\binom{\nu}{2}$.
- $\delta_1 := \sum_{i\in[h]} \binom{|\mathsf{B}_i \setminus (\mathsf{B}_{i,1}\cup\mathsf{B}_{i,2})|}{2} - \sum_{i\in[h]} \binom{|\mathsf{B}_{i,3}|}{2} = \Delta_1 - \zeta$.
- $\delta_2 := \sum_{i\in[h]} \binom{|\mathsf{B}_i \setminus (\mathsf{B}_{i,1}\cup\mathsf{B}_{i,2}\cup\mathsf{B}_{i,4})|}{2} - \sum_{i\in[h]} \binom{|\mathsf{B}_{i,3}|}{2} = \Delta_2 - \zeta$.
- $\tau := \sum_{i\in[h]} |\mathsf{B}_{i,2}| \cdot |\mathsf{B}_{i,4}|$.

3.3 Refined attack procedure

We start with the system $\mathcal{F}$ of $m := n - k + h\binom{\beta}{2}$ equations and n variables, derived from the given RSD problem. Concretely, the system $\mathcal{F}$ consists of the following $n-k$ linear equations and $h\binom{\beta}{2}$ quadratic equations:

$$\begin{aligned} \mathbf{H}\mathbf{x} &= \mathbf{s}, \\ x_{i_1}x_{i_2} &= 0 \quad \text{for } i_1 \neq i_2 \in \mathsf{B}_i,\ i \in [h]. \end{aligned} \tag{1}$$

where $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ and $\mathbf{x} = (x_1, \ldots, x_n)$.

By eliminating $n-k$ variables using the given linear equations, we can obtain a system $\mathcal{F}'$ of $h\binom{\beta}{2}$ quadratic equations and k variables. If $h\binom{\beta}{2} < \binom{k}{2} + 2k$, then $\mathcal{F}'$ is underdetermined from the perspective of linearization attack. In this case, we can make the system $\mathcal{F}'$ solvable by guessing some variables as 0. Otherwise, we can solve the system using the linearization technique. Even in this case, if the loss resulting from the probability of guessing is acceptable, the RSD problem can be solved with less complexity (which rarely happens).

Step 1. First stage of zero guessing The following steps are executed as a nested guessing procedure. At the outer level, we repeatedly perform the first stage of zero guessing. In each outer trial, we randomly select a new set $\bar{\mathsf{B}}_1 \subseteq [n]$ of η_1 positions as described in Section 3.2, and set $\mathbf{x}_{\bar{\mathsf{B}}_1} = \mathbf{0}$. For this first-stage guess, we construct a quadratic system $\mathcal{F}_1$ by eliminating $n-k$ variables using the given linear equations. More precisely, for each choice of $\bar{\mathsf{B}}_1$, we obtain the linear system

$$\mathbf{H}_{[n]\setminus\bar{\mathsf{B}}_1}\mathbf{x}_{[n]\setminus\bar{\mathsf{B}}_1} = \mathbf{s}. \tag{2}$$

Next, we randomly select a set $\bar{\mathsf{B}}_2 \subseteq [n] \setminus \bar{\mathsf{B}}_1$ of $n-k$ positions as described in Sec 3.2. By performing Gaussian elimination, we transform $\mathbf{H}_{[n]\setminus\bar{\mathsf{B}}_1}$ into $\mathbf{A}$ such that the sub-matrix $\mathbf{H}_{\bar{\mathsf{B}}_2}$ of $\mathbf{H}_{[n]\setminus\bar{\mathsf{B}}_1}$ is reduced to $\mathbf{I}_{n-k}$, i.e., $\mathbf{Q}\mathbf{H}_{\bar{\mathsf{B}}_2} = \mathbf{I}_{n-k}$ where $\mathbf{Q}$ denote an invertible matrix satisfying $\mathbf{Q}\mathbf{H}_{[n]\setminus\bar{\mathsf{B}}_1} = \mathbf{A}$. Then, (2) is transformed into

$$\mathbf{A}\mathbf{x}_{[n]\setminus\bar{\mathsf{B}}_1} = \mathbf{Q}\mathbf{s}. \tag{3}$$

Thus, writing

$$\mathbf{Q}\mathbf{H}_{[n]\setminus(\bar{\mathsf{B}}_1\cup\bar{\mathsf{B}}_2)} = \big[\mathbf{a}_1|\cdots|\mathbf{a}_{n-k}\big]^{\mathsf{t}} \text{ for some } \mathbf{a}_i \in \mathbb{F}_q^{k-\eta_1} \text{ and } \mathbf{Q}\mathbf{s} = (t_1,\ldots,t_{n-k}),$$

from (3), we have

$$x_i = t_i - \mathbf{a}_i^{\mathsf{t}}\mathbf{x}_{[n]\setminus(\bar{\mathsf{B}}_1\cup\bar{\mathsf{B}}_2)} \quad \text{for} \quad i \in \bar{\mathsf{B}}_2. \tag{4}$$

Substituting these $n-k$ linear equations into (1), we can obtain a system $\mathcal{F}_1$ of m_1 quadratic equations and $n_1 := k-\eta_1$ variables as follows:

$$\begin{aligned} x_{i_1}x_{i_2} &= 0 \text{ for } i_1 \neq i_2 \in \mathsf{B}_i\setminus(\mathsf{B}_{i,1}\cup\mathsf{B}_{i,2}),\\ x_{i_1}\big(\mathbf{a}_{j_2}^{\mathsf{t}}\mathbf{x}_{[n]\setminus(\bar{\mathsf{B}}_1\cup\bar{\mathsf{B}}_2)} - t_{j_2}\big) &= 0 \text{ for } i_1 \in \mathsf{B}_i\setminus(\mathsf{B}_{i,1}\cup\mathsf{B}_{i,2}), j_2 \in \mathsf{B}_{i,2},\\ \big(\mathbf{a}_{j_1}^{\mathsf{t}}\mathbf{x}_{[n]\setminus(\bar{\mathsf{B}}_1\cup\bar{\mathsf{B}}_2)} - t_{j_1}\big)\big(\mathbf{a}_{j_2}^{\mathsf{t}}\mathbf{x}_{[n]\setminus(\bar{\mathsf{B}}_1\cup\bar{\mathsf{B}}_2)} - t_{j_2}\big) &= 0 \text{ for } j_1 \neq j_2 \in \mathsf{B}_{i,2} \end{aligned} \tag{5}$$

where i runs through the set $[h]$. Then, the number of equations is

$$m_1 = \textstyle\sum_{i\in[h]} \binom{|\mathsf{B}_i\setminus\mathsf{B}_{i,1}|}{2} = f_1\binom{\beta-\mu_1-1}{2} + (h-f_1)\binom{\beta-\mu_1}{2}.$$

We call $x_{i_1}x_{i_2}$ (resp. $x_{i_1}x_{i_2}=0$) *free monomial* (resp. *free equation*) since it can be eliminated from the system $\mathcal{F}_1$ for almost free. Note that there are $\Delta_1 := \sum_{i\in[h]}\binom{|\mathsf{B}_i\setminus(\mathsf{B}_{i,1}\cup\mathsf{B}_{i,2})|}{2}$ free monomials.

Let $\mathbf{z}_1$ be a vector of length $N_1 := \binom{n_1}{2} + 2n_1$ such that each coordinate of $\mathbf{z}$ corresponds to the monomials of the variables of $\mathbf{x}_{[n]\setminus(\bar{\mathsf{B}}_1\cup\bar{\mathsf{B}}_2)}$ with degree ≤ 2. Then, under some appropriate ordering of monomials, we can write the system $\mathcal{F}_1$ as $\mathbf{M}_1\mathbf{z}_1 = \mathbf{d}_1$ where $\mathbf{M}_1 = \begin{bmatrix}\mathbf{I}_{\Delta_1} & \mathbf{O}\\ \mathbf{M}_1' & \mathbf{M}_1''\end{bmatrix} \in \mathbb{F}_q^{m_1\times N_1}$, $\mathbf{d}_1 \in \mathbb{F}_q^{m_1}$. After reducing all free monomials, the system $\mathcal{F}_1$ can be represented as

$$\hat{\mathbf{M}}_1\mathbf{z}_1 = \hat{\mathbf{d}}_1, \quad \hat{\mathbf{M}}_1 = \begin{bmatrix}\mathbf{I}_{\Delta_1} & \mathbf{O}\\ \mathbf{O} & \hat{\mathbf{M}}_1''\end{bmatrix}.$$

The system $\mathcal{F}_1$ constructed above is associated with the current first-stage guess $\bar{\mathsf{B}}_1$. For this fixed first-stage guess, we next perform the preprocessing in Step 2 and then try multiple second-stage guesses in Step 3. If these trials do not yield a valid solution, we return to Step 1, choose a new set $\bar{\mathsf{B}}_1$, and repeat the procedure. Consequently, Step 1 forms the outer level of the guessing procedure, whereas Step 3 forms an inner level whose trials reuse the preprocessing obtained in Step 2.

Remark 1. In Step 1, we assume that $\mathbf{H}_{\bar{\mathsf{B}}_2}$ is invertible. This assumption does not pose a significant issue, since a random square matrix is invertible with constant probability. More precisely, as discussed in Section 2.3, the probability that a uniformly random $(n-k)\times(n-k)$ matrix over $\mathbb{F}_q$ is invertible is $p(n-k,q)=\prod_{i\in[n-k]}\left(1-q^{i-n-k-1}\right)$. This probability is close to 1 when q is sufficiently large; in particular, it is essentially 1 for $q=2^{128}$. For $q=2$, it approaches approximately 0.2889, which is also consistent with the experiments reported in Supplementary A.1. Similar experimental observations are given in [29, Section C.1]. Since the invertibility probability is bounded below by a positive constant, conditioning on this event changes the expected running time by at most a constant factor. Accordingly, as is standard in ISD-type algorithms [12, 20] and previous algebraic attacks [17, 28], we omit this constant factor from our complexity estimates.

Step 2. Preprocessing via Gaussian elimination For each first-stage trial in Step 1, we perform preprocessing once on the resulting system $\mathcal{F}_1$ before trying the corresponding second-stage zero guesses. This preprocessing is specific to the current first-stage guess $\bar{\mathsf{B}}_1$, but its result is shared by all second-stage guesses considered under this first-stage trial. To this end, we randomly select a set $\bar{\mathsf{B}}_3\subseteq[n]\setminus(\bar{\mathsf{B}}_1\cup\bar{\mathsf{B}}_2)$ of η_{pre} positions as described in Section 3.2. These positions correspond to variables that are excluded from the variables considered for zero guessing in the second stage.

Let $\mathbf{x}_{\mathrm{pre}}:=\mathbf{x}_{\bar{\mathsf{B}}_3}\in\mathbb{F}_q^{\eta_{\mathrm{pre}}}$ and $\mathbf{x}_{\mathrm{var}}:=\mathbf{x}_{[n]\setminus(\bar{\mathsf{B}}_1\cup\bar{\mathsf{B}}_2\cup\bar{\mathsf{B}}_3)}\in\mathbb{F}_q^{k-\eta_1-\eta_{\mathrm{pre}}}$. Then, after performing Gaussian elimination to $\mathbf{M}_1$, the system $\mathcal{F}_1$ can be represented by the following equations:

$$
\begin{aligned}
x_{i_1}x_{i_2}&=0 && \text{for } i_1\neq i_2\in\mathsf{B}_i\setminus(\mathsf{B}_{i,1}\cup\mathsf{B}_{i,2}),\\
b_{i_1,i_2}\,x_{i_1}x_{i_2}&+\mathbf{c}_{i_1,i_2}^{\mathsf{t}}\mathbf{x}_{\mathrm{pre}}+\mathbf{x}_{\mathrm{pre}}^{\mathsf{t}}\mathbf{D}_{i_1,i_2}\mathbf{x}_{\mathrm{var}}\\
&\quad+\mathbf{x}_{\mathrm{var}}^{\mathsf{t}}\mathbf{E}_{i_1,i_2}\mathbf{x}_{\mathrm{var}}+\mathbf{e}_{i_1,i_2}^{\mathsf{t}}\mathbf{x}_{\mathrm{var}}=d'_{i_1,i_2} && \text{for } i_1\in\mathsf{B}_{i,3},i_2\in\mathsf{B}_{j,3},i\neq j,\\
b_{i_1,i_1}\,x_{i_1}^2&+\mathbf{c}_{i_1,i_1}^{\mathsf{t}}\mathbf{x}_{\mathrm{pre}}+\mathbf{x}_{\mathrm{pre}}^{\mathsf{t}}\mathbf{D}_{i_1,i_1}\mathbf{x}_{\mathrm{var}}\\
&\quad+\mathbf{x}_{\mathrm{var}}^{\mathsf{t}}\mathbf{E}_{i_1,i_1}\mathbf{x}_{\mathrm{var}}+\mathbf{e}_{i_1,i_1}^{\mathsf{t}}\mathbf{x}_{\mathrm{var}}=d'_{i_1} && \text{for } i_1\in\mathsf{B}_{i,3},\\
\mathbf{c}_l^{\mathsf{t}}\mathbf{x}_{\mathrm{pre}}&+\mathbf{x}_{\mathrm{pre}}^{\mathsf{t}}\mathbf{D}_l\mathbf{x}_{\mathrm{var}}+\mathbf{x}_{\mathrm{var}}^{\mathsf{t}}\mathbf{E}_l\mathbf{x}_{\mathrm{var}}+\mathbf{e}_l^{\mathsf{t}}\mathbf{x}_{\mathrm{var}}=d'_l && \text{for } l\in[m'_1]
\end{aligned}
$$

where $b_{i_1,i_2},b_{i_1,i_2}\in\{0,1\}$, $\mathbf{E}_{i_1,i_2},\mathbf{E}_{i_1,i_1},\mathbf{E}_l$ are upper triangular, and i,j runs through $[h]$. Here m'_1 denote the number of equations that are not free and contain no quadratic monomials of variables of $\mathbf{x}_{\mathrm{pre}}$. Since there are at most $\gamma:=\binom{\eta_{\mathrm{pre}}}{2}+\eta_{\mathrm{pre}}$ equations that contain quadratic terms of variables of $\mathbf{x}_{\mathrm{pre}}$, letting δ_1 be the number of equations that are free but are not consists only of the variables of $\mathbf{x}_{\mathrm{pre}}$, we have

$$m'_1\geq m_1-\gamma-\delta_1$$

where $\delta_1:=\left|\left\{x_{i_1}x_{i_2}:i_1\neq i_2\in\mathsf{B}_i\setminus(\mathsf{B}_{i,1}\cup\mathsf{B}_{i,2})\text{ for }i\in[h]\right\}\setminus\left\{x_{i_1}x_{i_2}:i_1\neq i_2\in\mathsf{B}_{i,3}\text{ for }i\in[h]\right\}\right|$ and the equality holds when $b_{i_1,i_2}=b_{i_1,i_1}=1$ for all i_1,i_2.

Once this preprocessing has been completed, the resulting system is reused for multiple trials of the second-stage zero guessing in Step 3. Thus, for a fixed first-stage guess $\bar{\mathsf{B}}_1$, Step 2 is performed once, while Step 3 may be performed multiple times with different choices of $\bar{\mathsf{B}}_4$. If none of these second-stage trials produces a valid solution, the algorithm returns to Step 1 and starts a new outer trial with a different choice of $\bar{\mathsf{B}}_1$. The preprocessing in Step 2 is then recomputed for the new system $\mathcal{F}_1$.

Step 3. Second stage of zero guessing For the fixed first-stage guess and the preprocessing obtained in Steps 1 and 2, respectively, we repeatedly try second-stage zero guesses. In each inner trial, we select a new admissible set $\bar{\mathsf{B}}_4$ of η_2 positions and set $\mathbf{x}_{\bar{\mathsf{B}}_4} = \mathbf{0}$. The preprocessed system obtained in Step 2 is reused across these second-stage trials. We focus on solving the subsystem $\mathcal{F}_1'$ of $\mathcal{F}_1$ consisting only of the fourth type of equations:

$$\mathbf{c}_l^{\mathsf{t}}\,\mathbf{x}_{\mathrm{pre}} + \mathbf{x}_{\mathrm{pre}}^{\mathsf{t}}\mathbf{D}_l\mathbf{x}_{\mathrm{var}} + \mathbf{x}_{\mathrm{var}}^{\mathsf{t}}\mathbf{E}_l\mathbf{x}_{\mathrm{var}} + \mathbf{e}_l^{\mathsf{t}}\,\mathbf{x}_{\mathrm{var}} = d_l' \quad \text{for } l \in [m_1']. \tag{6}$$

If m_1' is not sufficiently large, we additionally guess some remaining variables as 0 so that the resulting subsystem is solvable via linearization technique.

In each second-stage trial, we randomly select a new set $\bar{\mathsf{B}}_4$ of η_2 positions as described in Section 3.2, and set $\mathbf{x}_{\bar{\mathsf{B}}_4} = \mathbf{0}$. Let $\mathbf{x}'_{\mathrm{var}} = \mathbf{x}_{\bar{\mathsf{B}}_5}$ where $\bar{\mathsf{B}}_5 = [n] \setminus (\cup_{i=1}^{4} \bar{\mathsf{B}}_i)$. Then the new system $\mathcal{F}_2$ obtained as a result of applying $\mathbf{x}_{\bar{\mathsf{B}}_4} = \mathbf{0}$ to $\mathcal{F}_1$ consists of the following equations:

$$\begin{aligned}
x_{i_1}x_{i_2} &= 0 \qquad \text{for } i_1 \neq i_2 \in \mathsf{B}_i \setminus (\mathsf{B}_{i,1} \cup \mathsf{B}_{i,2} \cup \mathsf{B}_{i,4}),\\
b_{i_1,i_2}x_{i_1}x_{i_2} + \mathbf{c}_{i_1,i_2}^{\mathsf{t}}\mathbf{x}_{\mathrm{pre}} + \mathbf{x}_{\mathrm{pre}}^{\mathsf{t}}\mathbf{D}'_{i_1,i_2}\mathbf{x}'_{\mathrm{var}}&\\
+ (\mathbf{x}'_{\mathrm{var}})^{\mathsf{t}}\mathbf{E}'_{i_1,i_2}\mathbf{x}'_{\mathrm{var}} + (\mathbf{e}'_{i_1,i_2})^{\mathsf{t}}\,\mathbf{x}'_{\mathrm{var}} &= d'_{i_1,i_2} \quad \text{for } i_1 \in \mathsf{B}_{i,3}, i_2 \in \mathsf{B}_{j,3}, i \neq j,\\
b_{i_1,i_1}x_{i_1}^2 + \mathbf{c}_{i_1,i_1}^{\mathsf{t}}\mathbf{x}_{\mathrm{pre}} + \mathbf{x}_{\mathrm{pre}}^{\mathsf{t}}\mathbf{D}'_{i_1,i_1}\mathbf{x}'_{\mathrm{var}}&\\
+ (\mathbf{x}'_{\mathrm{var}})^{\mathsf{t}}\mathbf{E}'_{i_1,i_1}\mathbf{x}'_{\mathrm{var}} + (\mathbf{e}'_{i_1,i_1})^{\mathsf{t}}\,\mathbf{x}'_{\mathrm{var}} &= d'_{i_1} \qquad \text{for } i_1 \in \mathsf{B}_{i,3},\\
\mathbf{c}_l^{\mathsf{t}}\,\mathbf{x}_{\mathrm{pre}} + \mathbf{x}_{\mathrm{pre}}^{\mathsf{t}}\mathbf{D}'_l\mathbf{x}'_{\mathrm{var}}&\\
+(\mathbf{x}'_{\mathrm{var}})^{\mathsf{t}}\mathbf{E}'_l\mathbf{x}'_{\mathrm{var}} + (\mathbf{e}'_l)^{\mathsf{t}}\,\mathbf{x}'_{\mathrm{var}} &= d'_l \qquad \text{for } l \in [m_2']
\end{aligned}$$

where $b_{i_1,i_2}, b_{i_1,i_1} \in \{0,1\}$, $\mathbf{E}'_{i_1,i_2}, \mathbf{E}'_{i_1,i_1}, \mathbf{E}'_l$ are upper triangular, and i, j runs through $[h]$. Then the number of equations in $\mathcal{F}_2$ is

$$m_2 = \textstyle\sum_{i\in[h]} \binom{|\mathsf{B}_i \setminus (\mathsf{B}_{i,1} \cup \mathsf{B}_{i,4})|}{2} = f\binom{\beta-\mu-1}{2} + (h-f)\binom{\beta-\mu}{2}.$$

and the number of free equations is $\Delta_2 = \sum_{i\in[h]} \binom{|\mathsf{B}_i \setminus (\mathsf{B}_{i,1} \cup \mathsf{B}_{i,2} \cup \mathsf{B}_{i,4})|}{2}$. For the same reason as before, we have

$$m_2' \geq m_2 - \gamma - \delta_2$$

where $\delta_2 := \left|\left\{x_{i_1}x_{i_2} : i_1 \neq i_2 \in \mathsf{B}_i \setminus (\mathsf{B}_{i,1} \cup \mathsf{B}_{i,2} \cup \mathsf{B}_{i,4}) \text{ for } i \in [h]\right\} \setminus \left\{x_{i_1}x_{i_2} : i_1 \neq i_2 \in \mathsf{B}_{i,3} \text{ for } i \in [h]\right\}\right|$ and the equality holds when $b_{i_1,i_2} = b_{i_1,i_1} = 1$ for all i_1, i_2.

One thing to note is that $m_2' \le m_1'$. This is because each equation $\mathbf{c}_l^{\mathsf{t}}\mathbf{x}_{\mathrm{pre}} + \mathbf{x}_{\mathrm{pre}}^{\mathsf{t}}\mathbf{D}_l'\mathbf{x}_{\mathrm{var}}' + (\mathbf{x}_{\mathrm{var}}')^{\mathsf{t}}\mathbf{E}_l'\mathbf{x}_{\mathrm{var}}' + (\mathbf{e}_l')^{\mathsf{t}}\mathbf{x}_{\mathrm{var}}' = d_l'$ in $\mathcal{F}_2$ is the result of substituting $\mathbf{x}_{\bar{\mathsf{B}}_4} = \mathbf{0}$ into (6) which a linear combination of equations in (5), where $x_{i_1}\left(\mathbf{a}_{j_2}^{\mathsf{t}}\mathbf{x}_{[n]\setminus(\bar{\mathsf{B}}_1\cup\bar{\mathsf{B}}_2)} - t_{i_2}\right) = 0$ becomes zero equation when 0 is assigned to x_{i_1} for $i_1 \in \bar{\mathsf{B}}_4$. Thus, after assigning $\mathbf{0}$ to $\mathbf{x}_{\bar{\mathsf{B}}_4}$, there are $m_2' = m_1' - \tau$ non-zero equations where $\tau := \sum_{i\in[h]} |\mathsf{B}_{i,2}| \cdot |\mathsf{B}_{i,4}|$. It is also verified from the fact that $(m_1 - \gamma - \delta_1) - (m_2 - \gamma - \delta_2) = (m_1 - \Delta_1) - (m_2 - \Delta_2) = \sum_{i\in[h]} |\mathsf{B}_{i,2}| \cdot |\mathsf{B}_{i,4}|$.

Step 4. Solve a system of equations via linearization We now try to solve the subsystem $\mathcal{F}_2'$ of $\mathcal{F}_2$ consists of the following equations:

$$\mathbf{c}_l^{\mathsf{t}}\mathbf{x}_{\mathrm{pre}} + \mathbf{x}_{\mathrm{pre}}^{\mathsf{t}}\mathbf{D}_l'\mathbf{x}_{\mathrm{var}}' + (\mathbf{x}_{\mathrm{var}}')^{\mathsf{t}}\mathbf{E}_l'\mathbf{x}_{\mathrm{var}}' + (\mathbf{e}_l')^{\mathsf{t}}\mathbf{x}_{\mathrm{var}}' = d_l' \quad \text{for } l \in [m_2'].$$

Since the coefficient $\mathbf{c}_l$ have already been reduced while performing Gaussian elimination in Step 2, the system $\mathcal{F}_2'$ can be represented by the following equations:

$$\begin{aligned} b_l x_l + \mathbf{x}_{\mathrm{pre}}^{\mathsf{t}}\mathbf{D}_l'\mathbf{x}_{\mathrm{var}}' + (\mathbf{x}_{\mathrm{var}}')^{\mathsf{t}}\mathbf{E}_l'\mathbf{x}_{\mathrm{var}}' + (\mathbf{e}_l')^{\mathsf{t}}\mathbf{x}_{\mathrm{var}}' = 0 \quad & \text{for } l \in \bar{\mathsf{B}}_3 \\ \mathbf{x}_{\mathrm{pre}}^{\mathsf{t}}\mathbf{D}_l'\mathbf{x}_{\mathrm{var}}' + (\mathbf{x}_{\mathrm{var}}')^{\mathsf{t}}\mathbf{E}_l'\mathbf{x}_{\mathrm{var}}' + (\mathbf{e}_l')^{\mathsf{t}}\mathbf{x}_{\mathrm{var}}' = d_l' \quad & \text{for } l \in \left[m_2''\right] \end{aligned}$$

where $b_l \in \{0,1\}$ and $m_2'' \ge m_2' - \left|\bar{\mathsf{B}}_3\right| = m_2' - \eta_{\mathrm{pre}}$.

We first solve the subsystem $\mathcal{F}_2''$ of $\mathcal{F}_2'$ defined by the second type of equations via the linearization technique. That is, we write the system $\mathcal{F}_2''$ as

$$\mathbf{M}_2''\mathbf{z}_2'' = \mathbf{d}_2'', \quad \mathbf{M}_2'' \in \mathbb{F}_q^{m_2''\times N_2''}$$

where $\mathbf{z}_2'' = \left((x_i x_j)_{i\in\bar{\mathsf{B}}_3, j\in\bar{\mathsf{B}}_5}, (x_{j_1}x_{j_2})_{j_1\le j_2\in\bar{\mathsf{B}}_5}, (x_j)_{j\in\bar{\mathsf{B}}_5}\right)$ is a vector of length $N_2'' := N_2' - \eta_{\mathrm{pre}}$. To filter out incorrect zero guesses, we require that

$$m_2'' - N_2'' \ge \xi(N_2'')$$

so that $\mathcal{F}_2''$ has no solution with overwhelming probability when the guessing $\mathbf{x}_{\bar{\mathsf{B}}_1\cup\bar{\mathsf{B}}_4} = \mathbf{0}$ is incorrect. Once we find a solution $\mathbf{z}$ which includes $\mathbf{x}_{\mathrm{var}}'$, we compute $\mathbf{x}_{\mathrm{pre}}$ by substituting $\mathbf{z}$ into the remaining l equations. In turn, by substituting $\mathbf{x}_{\mathrm{pre}}$, $\mathbf{x}_{\mathrm{var}}'$, and $\mathbf{x}_{\bar{\mathsf{B}}_1\cup\bar{\mathsf{B}}_4} = \mathbf{0}$ into (4), we find $\mathbf{x}_{\bar{\mathsf{B}}_2}$.

The obtained candidate is verified against the original RSD instance. If it is valid, the algorithm terminates. Otherwise, we return to Step 3 and try another second-stage zero guess while reusing the preprocessing computed in Step 2. After the prescribed second-stage trials associated with the current first-stage guess have been exhausted without finding a valid solution, we return to Step 1 and start a new outer trial.

Remark 2. In Step 4, we assume that an incorrect zero guess leads to an inconsistent system $\mathcal{F}_2''$ with overwhelming probability. The underlying intuition is that an incorrect guess removes the structure associated with the true error support, so that the resulting system behaves essentially like a random overdetermined system. As discussed in Section 2.3, such a system is expected to have no solution except with negligible probability. We further support this assumption with experiments, whose results are reported in Supplementary A.1.

3.4 Complexity analysis

The transformation

$$\mathcal{F} \longrightarrow \mathcal{F}_1 \longrightarrow \mathcal{F}_2 \longrightarrow \mathcal{F}_2' \longrightarrow \mathcal{F}_2''$$

describes a single path through the attack. The complete attack repeatedly executes this procedure using a nested guessing strategy.

At the outer level, Step 1 is repeated with different first-stage zero guesses $\bar{\mathsf{B}}_1$. For each such first-stage trial, Step 2 constructs preprocessing specific to the resulting system $\mathcal{F}_1$. At the inner level, the preprocessing is then reused while Steps 3 and 4 are repeatedly executed with different second-stage zero guesses $\bar{\mathsf{B}}_4$. If no valid solution is obtained from the inner trials associated with the current first-stage guess, the algorithm returns to Step 1 and starts another outer trial.

This nested structure is important for the complexity analysis. The first-stage guess is correct with probability p_{η_1}, so the costs of Steps 1 and 2 are incurred an expected $\mathrm{p}_{\eta_1}^{-1}$ times. On the other hand, a complete first- and second-stage zero guess is correct with probability p_{η}, so the cost associated with each second-stage trial is incurred an expected p_{η}^{-1} times. This gives rise to the two different repetition factors appearing in the total complexity below.

Recall that each system $\mathcal{F}$ (resp. $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}_2'$, $\mathcal{F}_2''$) has m (resp. m_1, m_2, m_2', m_2'') equations, n (resp. n_1, n_2, n_2', n_2'') variables, and N (resp. N_1, N_2, N_2', N_2'') monomials where

$$m_1 = f_1\tbinom{\beta-\mu_1-1}{2} + (h-f_1)\tbinom{\beta-\mu_1}{2}, \quad n_1 = k-\eta_1, \quad N_1 = \tfrac{n_1(n_1+3)}{2},$$
$$m_2 = f\tbinom{\beta-\mu-1}{2} + (h-f)\tbinom{\beta-\mu}{2}, \quad n_2 = n_2' = n_2'' = k-\eta, \quad N_2 = \tfrac{n_2(n_2+3)}{2},$$
$$m_2' = m_2 - \gamma - \delta_2, \quad N_2' = N_2 - \gamma - \delta_2, \qquad m_2'' = m_2' - \eta_{\mathrm{pre}}, \quad N_2'' = N_2' - \eta_{\mathrm{pre}},$$

and $\gamma = \frac{\eta_{\mathrm{pre}}(\eta_{\mathrm{pre}}+1)}{2}$.

We can now state the resulting concrete complexity bound.

Theorem 1. *The refined attack with input parameters* $(\eta_1, \eta, \eta_{\mathrm{pre}})$ *satisfying*

$$m_2'' - N_2'' \geq \xi(N_2''), \quad \eta + \eta_{\mathrm{pre}} \leq k, \quad \lceil (n-k+\eta+\eta_{\mathrm{pre}})/h \rceil \leq \beta$$

can solve the $\mathrm{RSD}(n,k,h)$ *problem in time complexity* [4]

$$\mathrm{p}_{\eta_1}^{-1} \cdot (\mathrm{T}_1 + \mathrm{T}_2) + \mathrm{p}_{\eta}^{-1} \cdot \mathrm{T}_3$$

where $\mathrm{p}_x := \left(1 - \frac{\lfloor x/h \rfloor + 1}{\beta}\right)^{x \bmod h} \left(1 - \frac{\lfloor x/h \rfloor}{\beta}\right)^{h-(x \bmod h)}$ *and*

$$\mathrm{T}_1 = \mathcal{O}\big(\mathrm{C}_{\mathrm{ge}}(n-k, n-\eta_1) + \Delta_1 + (m_1-\Delta_1)(n_1+1)^2\big),$$
$$\mathrm{T}_2 = \mathcal{O}\big(\mathrm{C}_{\mathrm{ge}}(m_1-\Delta_1, N_1-\Delta_1)\big),$$
$$\mathrm{T}_3 = \mathcal{O}\big(m_2'(k-\eta_1+1)\eta_2 + \mathrm{C}_{\mathrm{ge}}(m_2'', N_2'')\big).$$

[4] It is measured in terms of the number of $\mathbb{F}_q$-operations.

Proof. We will calculate the complexity of each step in order.

In Step 1, the probability that all η_1 zero-guessing are correct is $\left(1-\frac{\mu_1+1}{\beta}\right)^{f_1}\left(1-\frac{\mu_1}{\beta}\right)^{h-f_1}$ where $\mu_1 = \lfloor \eta_1/h \rfloor$ and $f_1 = \eta_1 \bmod h$. The cost of performing Gaussian elimination on $\mathbf{H}_{[n]\setminus\bar{\mathsf{B}}_1} \in \mathbb{F}_q^{(n-k)\times(n-\eta_1)}$ is $\mathcal{O}(\mathrm{C_{ge}}(n-k, n-\eta_1))$. Since each quadratic equation $x_{i_1}x_{i_2} = 0$ for $i_1 \neq i_2 \in \mathsf{B}_i \setminus (\mathsf{B}_{i,1} \cup \mathsf{B}_{i,2})$ can be constructed in a constant time and the others can be constructed in time $\mathcal{O}(n_1^2)$, the time for constructing m_1 equations in $\mathcal{F}_1$ is $\mathcal{O}(\Delta_1 + (m_1-\Delta_1)(n_1+1)^2)$. Next, since the computation of the matrix $\hat{\mathbf{M}}_1$, i.e., removing all free monomials from equations in $\mathcal{F}_1$, can be done while constructing equations, no additional cost is required. These explain the term $\mathrm{p}_{\eta_1}^{-1} \cdot \mathrm{T}_1$ in total complexity.

In Step 2, since only Gaussian elimination is performed to eliminate quadratic monomials $x_{i_1}x_{i_2}$ for $i_1, i_2 \in \bar{\mathsf{B}}_3$ and all free monomials already have been reduced, the complexity of this step becomes $\mathcal{O}(\mathrm{C_{ge}}(m_1-\Delta_1, N_1-\Delta_1))$, which explains the term $\mathrm{p}_{\eta_1}^{-1}\mathrm{T}_2$ in total complexity.

In Step 3, we first compute the probability P_2 that all η_2 zero-guessing are correct given that all η_1 zero-guessing are correct. Let E_i be the event that all $|\mathsf{B}_{i,4}|$ zero-guessings are correct given that all $|\mathsf{B}_{i,1}|$ zero-guessings are correct over $\mathbf{x}_{\mathsf{B}_i}$. Let A_i be the event that $\mathbf{x}_{\mathsf{B}_{i,2}\cup\mathsf{B}_{i,3}}$ contains 1 given that all $|\mathsf{B}_{i,1}|$ zero-guessings are correct over $\mathbf{x}_{\mathsf{B}_i}$. Then

$$\begin{aligned}\Pr[E_i] &= \Pr[A_i]\Pr[E_i|A_i] + \Pr[A_i^c]\Pr[E_i|A_i^c] \\ &= \tfrac{|\mathsf{B}_{i,2}|+|\mathsf{B}_{i,3}|}{\beta-|\mathsf{B}_{i,1}|}\cdot 1 + \left(1-\tfrac{|\mathsf{B}_{i,2}|+|\mathsf{B}_{i,3}|}{\beta-|\mathsf{B}_{i,1}|}\right)\left(1-\tfrac{|\mathsf{B}_{i,4}|}{\beta-|\mathsf{B}_{i,1}|-|\mathsf{B}_{i,2}|-|\mathsf{B}_{i,3}|}\right) \\ &= 1-\tfrac{|\mathsf{B}_{i,4}|}{\beta-|\mathsf{B}_{i,1}|}.\end{aligned}$$

Thus, we have

$$\begin{aligned}\mathrm{P}_2 &= \textstyle\prod_{i\in[h]}\Pr[E_i] = \prod_{i\in[h]}\left(1-\frac{|\mathsf{B}_{i,4}|}{\beta-|\mathsf{B}_{i,1}|}\right) \\ &= \textstyle\prod_{i\in[h]}\left(1-\frac{|\mathsf{B}_{i,1}|+|\mathsf{B}_{i,4}|}{\beta}\right)/\prod_{i\in[h]}\left(1-\frac{|\mathsf{B}_{i,1}|}{\beta}\right) = \mathrm{p}_\eta/\mathrm{p}_{\eta_1}.\end{aligned}$$

The time for constructing m_2' equations in $\mathcal{F}_2'$ is $\mathcal{O}(m_2'(\eta_{\mathrm{pre}}\eta_2 + (k-\eta_1-\eta_{\mathrm{pre}})\eta_2) + \eta_2) = \mathcal{O}(m_2'(k-\eta_1+1)\eta_2)$ since it consists only of operations to erase monomial terms related to η_2 guessed variables and it requires constant-time to remove all zero or linear dependent equations, which can be done by tracing positions that equations of the form $x_{i_1}(\mathbf{a}_{j_2}^{\mathsf{t}}\mathbf{x}_{[n]\setminus\bar{\mathsf{B}}_1\cup\bar{\mathsf{B}}_2} - t_{i_2})$ are placed and removing all equations at these positions. Finally, the complexity of solving the linear system $\mathbf{M}_2''\mathbf{z}_2'' = \mathbf{d}_2''$ where $\mathbf{M}_2'' \in \mathbb{F}_q^{m_2''\times N_2''}$ is $\mathcal{O}(\mathrm{C_{ge}}(m_2'', N_2''))$. Thus, these explain the term $\mathrm{p}_\eta^{-1}\mathrm{T}_3 = \mathrm{p}_{\eta_1}^{-1}\cdot(p_\eta/p_{\eta_1})^{-1}\cdot\mathrm{T}_3$ in the total complexity. □

Corollary 1. *Take $\xi(N_2'') = c_{\mathrm{excess}} \in \mathbb{N}$ in Theorem 1. Then, the refined attack with input parameters $(\eta_1, \eta, \eta_{\mathrm{pre}})$ satisfying*

$$m_2'' \geq N_2'' + c_{\mathrm{excess}}, \quad \eta + \eta_{\mathrm{pre}} \leq k, \quad \lceil (n-k+\eta+\eta_{\mathrm{pre}})/h \rceil \leq \beta$$

can solve the $\mathrm{RSD}(n,k,h)$ *problem in time complexity*

$$\mathrm{p}_{\eta_1}^{-1}\cdot(\check{\mathrm{T}}_1+\check{\mathrm{T}}_2)+\mathrm{p}_{\eta}^{-1}\cdot\check{\mathrm{T}}_3$$

where

$$\begin{aligned}
\check{\mathrm{T}}_1 &= \mathcal{O}\big(\mathrm{C}_{\mathrm{ge}}(n-k,n-\eta_1)+\Delta_1+(N_2+c_{\mathrm{excess}}+\tau-\Delta_2)(n_1+1)^2\big),\\
\check{\mathrm{T}}_2 &= \mathcal{O}\big(\mathrm{C}_{\mathrm{ge}}(N_2+c_{\mathrm{excess}}+\tau-\Delta_2,N_1-\Delta_1)\big),\\
\check{\mathrm{T}}_3 &= \mathcal{O}\big((N_2'+c_{\mathrm{excess}})(k-\eta_1+1)\eta_2+\mathrm{C}_{\mathrm{ge}}(N_2''+c_{\mathrm{excess}},N_2'')\big).
\end{aligned}$$

Proof. The refinement is effective when the final system contains enough equations to solve $\mathcal{F}_2''$, i.e., $m_2''\geq N_2''+c_{\mathrm{excess}}$. To obtain $\check{m}_2''\coloneqq N_2''+c_{\mathrm{excess}}$ equations at Step 4, we use only as many equations as necessary from the previous steps. Thus, it is sufficient to use $\check{m}_1,\check{m}_2'$ equations in $\mathcal{F}_1,\mathcal{F}_2'$, respectively, where

$$\begin{aligned}
&\check{m}_1=\check{m}_1'+\gamma+\delta_1=N_2+c_{\mathrm{excess}}+\tau+\Delta_1-\Delta_2,\\
&\check{m}_1'=\check{m}_2'+\tau=N_2'+c_{\mathrm{excess}}+\tau,\quad \check{m}_2'=\check{m}_2''+\eta_{\mathrm{pre}}=N_2'+c_{\mathrm{excess}}.
\end{aligned}$$

Thus, with these choices, the claimed complexity can be obtained. □

3.5 Asymptotic consequences

We briefly state the two main asymptotic consequences of the refined non-binary attack; the complete derivation, including the rounded blockwise guessing exponent, is given in Supplementary B.1. Let $R=k/n$, $\rho=h/n$, and $\beta=1/\rho$. First, without zero guessing, the refined attack is polynomial whenever

$$R^2\rho n+3R\rho\leq 1-\rho. \tag{7}$$

Indeed, after eliminating the $n-k$ syndrome variables, the system contains $m_2=\frac{n}{2\rho}(1-\rho)$ quadratic equations and $N_2=\frac{R^2n^2+3Rn}{2}$ linearized monomials; the preprocessing removes the same number of equations and monomials, so $m_2\geq N_2$ suffices. Thus, up to lower-order terms, the polynomial-time region is $R^2\rho\leq(1-\rho-o(1))/n$, or $R^2\rho\leq n^{-1}$ in the usual coarse convention.

Second, if $k\leq n-h$, taking $\eta_1=0$, $\eta=k$, and $\eta_{\mathrm{pre}}=0$ removes all variables after the balanced zero-guessing step. Each trial is then polynomial, and the total cost is

$$\mathrm{poly}(n)\cdot 2^{\rho n\Lambda_\beta(R)+O(1)}, \tag{8}$$

where Λ_β is the exact rounded blockwise exponent defined in Supplementary B.1. Under the continuous approximation, this becomes $\mathrm{poly}(n)\cdot 2^{\rho n\log_2(1/(1-R))}$.

4 Extension of the refined attack to binary RSD

The binary case follows the same refined framework as the non-binary case, with changes induced by the additional block-sum equations and the binary monomial structure.

4.1 Differences from the non-binary case

For binary RSD over $\mathbb{F}_2$, regularity implies an additional linear relation in every block: the coordinates in each block sum to one. Thus, in addition to the syndrome equations, we have h linear equations of the form $\sum_{j \in \mathsf{B}_i} x_j = 1$ for $i \in [h]$. These additional equations change the number of variables that can be eliminated before the algebraic stage, but otherwise preserve the blockwise balancing strategy.

Distributing linear equations We describe how the variables eliminated by the linear equations are distributed across the blocks. In the binary setting, besides the $n-k$ linear equations arising from $\mathbf{Hx} = \mathbf{t}$, there are h additional block-wise linear equations $\sum_{j \in \mathsf{B}_i} x_j = 1$ where $i \in [h]$. Thus, in contrast to the non-binary setting, the total number of linear equations is $n-k+h$. Accordingly, the parameters used to distribute these equations are given by

$$\omega = \left\lfloor \tfrac{n-k+h}{h} \right\rfloor, \qquad f_{\mathrm{lin}} = (n-k+h) \bmod h.$$

Apart from this change in the values of ω and f_{lin}, the blockwise construction is identical to that in the non-binary case. In particular, $\mathsf{B}_{i,2}$ is defined in the same way, and its cardinality is computed from the corresponding values of ω and f_{lin} by the same rule.

Refined attack procedure The algorithm for solving the binary RSD problem follows the same overall procedure as in the non-binary case. Once ω and f_{lin} are fixed as above, the partitions $\mathsf{B}_{i,1},\ \mathsf{B}_{i,2},\ \mathsf{B}_{i,3},\ \mathsf{B}_{i,4},\ \mathsf{B}_{i,5}$ are determined exactly as before, and the same argument gives the condition on the block size.

There are, however, two binary-specific differences. First, the system contains the additional h block-wise linear equations described above. Second, the number of quadratic monomials is smaller, since over the binary field any quadratic monomial involving a single variable reduces to a linear monomial. These changes affect the final constraint used to filter out incorrect zero guesses, and hence the function $\xi(N)$ over the binary field differs from its non-binary counterpart. Further details are given in Corollary 3.4. Since all other steps are unchanged from the non-binary algorithm, we now proceed to the complexity analysis.

4.2 Complexity analysis

Because the attack structure is unchanged, the complexity analysis mirrors the non-binary case; only the variable counts, monomial counts, and Gaussian-elimination costs differ. As in the non-binary case, the attack successively transforms the target system $\mathcal{F}$ into smaller systems:

$$\mathcal{F} \longrightarrow \mathcal{F}_1 \longrightarrow \mathcal{F}_2 \longrightarrow \mathcal{F}_2' \longrightarrow \mathcal{F}_2''$$

We recall that each system $\mathcal{F}$ (resp. $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}'_2$, $\mathcal{F}''_2$) consists of m (resp. m_1, m_2, m'_2, m''_2) equations, n (resp. n_1, n_2, n'_2, n''_2) variables, and N (resp. N_1, N_2, N'_2, N''_2) monomials. In the binary case,

$$n_1 = k-h-\eta_1, \quad n_2 = n'_2 = n''_2 = k-h-\eta, \quad N_1 = \tfrac{n_1(n_1+1)}{2}, \quad N_2 = \tfrac{n_2(n_2+1)}{2},$$

and $\gamma = \binom{\eta_{\mathrm{pre}}}{2}$, whereas $m_1, m_2, m'_2, m''_2, N'_2, N''_2$ are defined exactly as in the non-binary setting.

The following theorem gives the binary counterpart of Theorem 1.

Theorem 2. *The refined attack with input parameters $(\eta_1, \eta, \eta_{\mathrm{pre}})$ satisfying*

$$m''_2 - N''_2 \geq \xi(N''_2), \quad \eta + \eta_{\mathrm{pre}} \leq k-h, \quad \lceil (n-k+h+\eta+\eta_{\mathrm{pre}})/h \rceil \leq \beta$$

can solve the binary $\mathrm{RSD}(n,k,h)$ *problem in time complexity*

$$\mathrm{p}_{\eta_1}^{-1} \cdot (\mathrm{T}_1 + \mathrm{T}_2) + \mathrm{p}_{\eta}^{-1} \cdot \mathrm{T}_3$$

where $\mathrm{p}_x := \left(1 - \frac{\lfloor x/h \rfloor + 1}{\beta}\right)^{x \bmod h} \left(1 - \frac{\lfloor x/h \rfloor}{\beta}\right)^{h - (x \bmod h)}$ *and*

$$\begin{aligned}
\mathrm{T}_1 &= \mathcal{O}\big((n-k)(n-\eta_1) + \mathrm{C}_{\mathrm{ge}}^{\mathrm{bin}}(n-k, n-\eta_1-h) + \Delta_1 + (m_1-\Delta_1)(n_1+1)^2\big),\\
\mathrm{T}_2 &= \mathcal{O}\big(\mathrm{C}_{\mathrm{ge}}^{\mathrm{bin}}(m_1-\Delta_1, N_1-\Delta_1)\big),\\
\mathrm{T}_3 &= \mathcal{O}\big(m'_2(k-h-\eta_1+1)\eta_2 + \mathrm{C}_{\mathrm{ge}}^{\mathrm{bin}}(m''_2, N''_2)\big).
\end{aligned}$$

Proof. The refined attack proceeds identically to the non-binary case, except for the elimination arising from the additional h linear equations associated with the blocks $\sum_{j \in \mathsf{B}_i \setminus \mathsf{B}_{i,1}} x_j = 1$ where $i \in [h]$. After the first zero-guessing step with η_1 variables, we obtain $\mathbf{H}_{[n]\setminus\bar{\mathsf{B}}_1} \in \mathbb{F}_2^{(n-k)\times(n-\eta_1)}$. This matrix can be further reduced using the above h linear equations, and the number of operations required for this step is bounded by $(n-k)(\beta-\mu_1)h$. Hence, the cost of obtaining $\mathbf{H}'_{[n]\setminus\bar{\mathsf{B}}_1} \in \mathbb{F}_2^{(n-k)\times(n-\eta_1-h)}$ is $\mathcal{O}\big((n-k)(n-\eta_1)\big)$. Consequently, the cost of Gaussian elimination for $\mathbf{H}'_{[n]\setminus\bar{\mathsf{B}}_1}$ is $\mathcal{O}\big(\mathrm{C}_{\mathrm{ge}}^{\mathrm{bin}}(n-k,\, n-\eta_1-h)\big)$. □

Corollary 2. *Take $\xi(N''_2) = aN''_2$ in Theorem 2. Then, the refined attack with input parameters $(\eta_1, \eta, \eta_{\mathrm{pre}})$ satisfying*

$$m''_2 \geq (a+1)N''_2, \quad \eta + \eta_{\mathrm{pre}} \leq k-h, \quad \lceil (n-k+h+\eta+\eta_{\mathrm{pre}})/h \rceil \leq \beta$$

can solve the binary $\mathrm{RSD}(n,k,h)$ *problem in time complexity*

$$\mathrm{p}_{\eta_1}^{-1} \cdot (\check{\mathrm{T}}_1 + \check{\mathrm{T}}_2) + \mathrm{p}_{\eta}^{-1} \cdot \check{\mathrm{T}}_3$$

where

$$\begin{aligned}
\check{\mathrm{T}}_1 &= \mathcal{O}\big((n-k)(n-\eta_1) + \mathrm{C}_{\mathrm{ge}}^{\mathrm{bin}}(n-k, n-\eta_1-h) + \Delta_1 + (N_2 + aN''_2 + \tau - \Delta_2)(n_1+1)^2\big),\\
\check{\mathrm{T}}_2 &= \mathcal{O}\big(\mathrm{C}_{\mathrm{ge}}^{\mathrm{bin}}(N_2 + aN''_2 + \tau - \Delta_2, N_1 - \Delta_1)\big),\\
\check{\mathrm{T}}_3 &= \mathcal{O}\big((N'_2 + aN''_2)(k-\eta_1+1)\eta_2 + \mathrm{C}_{\mathrm{ge}}^{\mathrm{bin}}((a+1)N''_2, N''_2)\big).
\end{aligned}$$

Proof. To obtain $\check{m}_2'' := (a+1)N_2''$ equations we use only as many equations as necessary from the previous steps. Thus, it is sufficient to use $\check{m}_1, \check{m}_2'$ equations in $\mathcal{F}_1, \mathcal{F}_2'$, respectively, where

$$\check{m}_1 = \check{m}_1' + \gamma + \delta_1 = N_2 + aN_2'' + \tau + \Delta_1 - \Delta_2,$$
$$\check{m}_1' = \check{m}_2' + \tau = N_2' + aN_2'' + \tau, \quad \check{m}_2' = \check{m}_2'' + \eta_{\mathrm{pre}} = N_2' + aN_2''.$$

Thus, with these choices, the claimed complexity can be obtained. □

4.3 Asymptotic consequences

The binary case admits the analogous asymptotic picture; full proofs and the rounded guessing analysis are deferred to Supplementary B.2. Let $R = k/n$ and $\rho = h/n$. If $R \leq \rho$, the syndrome equations together with the h block-sum equations already remove all asymptotic degrees of freedom. For $R > \rho$, no zero guessing is needed in the polynomial-time region, and the sufficient condition is

$$(R-\rho)^2\rho n + (R-\rho)\rho \leq 1 - \rho. \tag{9}$$

Here the remaining number of variables is $(R-\rho)n$, the number of linearized monomials is $\frac{(R-\rho)^2n^2+(R-\rho)n}{2}$, and the number of quadratic equations is $\frac{n}{2\rho}(1-\rho)$. Consequently, the coarse polynomial-time condition is $(R-\rho)^2\rho \leq n^{-1}$.

For $R > \rho$, choosing $\eta_1 = 0$, $\eta = k - h$, and $\eta_{\mathrm{pre}} = 0$ removes all variables after the balanced zero-guessing step and gives

$$\mathrm{poly}(n) \cdot 2^{\rho n \Lambda_\beta(R-\rho)+O(1)} \tag{10}$$

operations over $\mathbb{F}_2$. Under the continuous approximation, the dominant exponential term simplifies to $2^{\rho n \log_2(1/(1-R+\rho))}$. These statements are sufficient for comparing the asymptotic regimes; the concrete estimates in the applications use the exact finite-parameter formulas rather than these approximations.

For completeness, the continuous bound immediately yields the following useful regimes. If R and ρ are constants, the exponent is $\rho n \log_2(1/(1-R+\rho))$. If R is constant and $\rho = o(1)$, it becomes $\rho n \log_2(1/(1-R)) + o(\rho n)$. If $R = o(1)$ and $\rho = o(R)$, then

$$\log_2 \mathrm{T}_{\mathrm{total}} \leq \frac{\rho n R}{\ln 2} + o(\rho n R) + \mathrm{polylog}(n),$$

whereas for $\rho = \lambda R$ with a fixed $0 < \lambda < 1$,

$$\log_2 \mathrm{T}_{\mathrm{total}} \leq \frac{\rho n(1-\lambda) R}{\ln 2} + o(\rho n R) + \mathrm{polylog}(n).$$

These expansions are obtained from $-\log_2(1-x) = x/\ln 2 + O(x^2)$ and make explicit how the exponent decreases when the regular weight scales below the code rate.

5 Concrete security reassessment of existing RSD parameters

We now turn to the main concrete consequence of the refined attack: a reassessment of RSD instances used in existing cryptographic constructions [7, 9, 14, 18, 22, 26, 30, 31]. Rather than viewing the following results merely as a benchmark of a new solver, we use the complexity bounds from Sections 3 and 4 to determine whether previously selected RSD parameters still meet their intended security targets.

For each RSD instance, an exhaustive search is carried out to determine $(\eta_1, \eta, \eta_{\mathrm{pre}})$ attaining the minimum complexity. Throughout this search, condition $N_2'' q^{-\xi(N_2'')} \ll p_\eta$ is imposed so that, when zero guessing is incorrect, the resulting system is inconsistent with high probability. For most instances over non-binary fields, this condition is satisfied by taking $\xi(N_2'') = c_{\mathrm{excess}} = 1$, while for most binary instances it is satisfied by taking $\xi(N_2'') = aN_2''$, $a = \frac{1}{2}$. However, some parameter sets require a different choice of $\xi(N)$ in order to satisfy this condition. In Table 1, the symbol $\wr$ denotes complexities computed over non-binary fields with $c_{\mathrm{excess}} = 2$, whereas the symbols † and ‡ denote complexities computed over binary fields with $a = \frac{3}{2}$ and $a = \frac{5}{2}$, respectively. It should also be noted that some RSD instances do not satisfy $n/h \in \mathbb{Z}$. In such cases, we assume that the last $n - h\lfloor n/h \rfloor$ coordinates are zero, as done in several previous works [8, 12, 17, 20, 28]. The complexity estimates and corresponding parameters are summarized in Table 1. All values reported in Table 1 can be reproduced using the supplementary materials [5].

PCG/COT/VOLE Pseudorandom Correlation Generators provide a general framework for generating large amounts of correlated randomness from short seeds, and have found numerous applications in cryptography. Several PCG-based constructions are designed based on the hardness of the regular-noise Learning Parity with Noise problem, which naturally gives rise to RSD.

The first table in Table 1 shows parameter sets, proposed in [18], intended to provide 128-bit security for stationary syndrome decoding problem that consists of multiple RSD instances sharing a supporter. The analysis shows that none of these instances achieves the claimed security. More precisely, four non-binary parameter sets exhibit a security loss of 23–32 bits, while four binary parameter sets exhibit a security loss of 30–51 bits.

[22] presents PCGs for multiparty Beaver triples based on both the primal and dual variants of LPN problem with regular noise, which give rise to the RSD problem. All of these RSD instances are evaluated under the proposed attack, and the resulting complexity estimates show that none of the proposed parameter sets achieves the claimed 128 bits of security, with security losses of 41–63 bits.

Sun et al. [26] introduce a committed VOLE protocol whose security is based on two dual LPN problems or on both dual and primal LPN problems. Three

[5] `https://github.com/ho-chang/RSD_ReEstimate`

pairs of RSD instances are identified in the conference version [26] and the full version [27], respectively. Since it is sufficient to analyze only one of the two RSD parameters constituting each pair, a total of five RSD parameters is subject to analysis; three from [26] and two from [27]. The analysis shows that none of these instances achieves the claimed 128-bit security level against the proposed attack. More precisely, all five instances exhibit a security loss of 37–66 bits.

Ferret [31] is a correlated OT extension protocol and its security is based on the LPN problem with regular noise, which naturally gives rise to the RSD problem. The results show that none of these parameter sets achieves the claimed 128-bit security against the proposed attack. It should be noted that one of the two instances had already been shown by previous attacks to fall below the claimed security level; for this instance, the proposed attack further improves the best previously known complexity by 28 bits (see Table 3). For the other instance, the proposed attack also reduces the complexity below the claimed 128-bit security level, corresponding to a security loss of 17 bits.

Moreover, the second and third tables in Table 1, introduced in [7] and later revisited in [20], have been widely adopted as benchmark instances in subsequent cryptanalytic studies [8, 12, 17, 28]. The result shows that the proposed attack improves upon the best previously known attacks by up to 29 bits in the non-binary case and up to 25 bits in the binary case (see Table 2). For the refined 128-bit parameter sets from [20], none of these instances achieves the claimed security level against the proposed attack, with security losses of 27–47 bits over non-binary fields and 17–48 bits over the binary field (third table in Table 1).

ZKP Wolverine [30] is an zero-knowledge proof protocol for Boolean and arithmetic circuits. Its offline phase is based on an sVOLE extension protocol whose security relies on the LPN problem with regular noise. Our results show that none of parameter sets proposed in Wolverine attains the claimed 128 bits of security. It is worth noting that one of the two instances had already been shown by previous attacks to fall below the claimed security; for this instance, the proposed attack further improves the best previously known complexity by 28 bits (see Table 3). For the other instance, the proposed attack also reduces the claimed 128-bit security by 17 bits.

MPC TinyKeys [14] is an MPC framework for the dishonest-majority setting that improves the efficiency of GMW and BMR style protocols, whose security relies on the RSD problem. Among the parameter sets proposed in the full version [15] of TinyKeys, one RSD instance is estimated to fall short of the claimed 128-bit security level by 9 bits under the proposed attack.

Signature from MPC-in-the-Head Bui et al. [9] introduce a post-quantum digital signature scheme based on the MPC-in-the-head paradigm. The scheme relies on the regular syndrome decoding assumption, and its zero-knowledge protocol is designed to verify the correctness of an RSD instance within the

Table 2. Complexity estimates for the parameter sets from [7, 20].

(n, k, h)	q	AGB [8]	ISD [20]	R-ISD [12]	RP [17]	HAGB [28]	Ours
$(2^{22}, 64770, 4788)$	2^{128}	110	152	173	151	89.27	**86.70**
$(2^{20}, 32771, 2467)$		131	155	173	148	114.27	**84.50**
$(2^{18}, 15336, 1312)$		133	153	160	147	119.03	**93.45**
$(2^{16}, 7391, 667)$		150	151	160	142	131.94	**110.45**
$(2^{14}, 3482, 338)$		150	150	153	139	133.15	**115.96**
$(2^{12}, 1589, 172)$		150	155	148	140	132.60	**120.21**
$(2^{10}, 652, 106)$		179	194	173	159	158.21	**150.32**
$(2^{22}, 64770, 2735)$	2^{128}	104	108	126	105	86.51	**86.51**
$(2^{20}, 32771, 1419)$		102	107	122	103	83.91	**80.86**
$(2^{18}, 15336, 760)$		107	105	117	100	90.95	**74.68**
$(2^{16}, 7391, 389)$		111	103	113	97	96.39	**73.97**
$(2^{14}, 3482, 198)$		110	101	107	94	95.62	**77.83**
$(2^{12}, 1589, 98)$		107	100	101	89	93.08	**80.06**
$(2^{10}, 652, 57)$		111	111	106	98	97.61	**89.31**
$(2^{22}, 64770, 4788)$	2	103	135	140	143	87.04	**85.07**
$(2^{20}, 32771, 2467)$		125	135	138	139	105.44	**79.53**
$(2^{18}, 15336, 1312)$		122	129	126	137	108.27	**83.73**
$(2^{16}, 7391, 667)$		139	125	126	130	119.67	**99.78**
$(2^{14}, 3482, 338)$		138	122	118	125	118.53	**103.64**
$(2^{12}, 1589, 172)$		135	121	109	120	113.69	**104.25**
$(2^{10}, 652, 106)$		145	159	**113**	117	115.58	113.86
$(2^{22}, 64770, 2735)$	2	103	99	103	102	86.92	**83.10**
$(2^{20}, 32771, 1419)$		98	97	98	100	81.09	**77.74**
$(2^{18}, 15336, 760)$		104	92	93	97	86.74	**71.64**
$(2^{16}, 7391, 389)$		108	87	90	92	90.90	**70.35**
$(2^{14}, 3482, 198)$		106	83	84	88	88.97	**73.56**
$(2^{12}, 1589, 98)$		103	80	78	84	82.16	**74.75**
$(2^{10}, 652, 57)$		101	90	**76**	84	78.44	78.66

MPC-in-the-head framework. We evaluate the parameter set proposed in [9], which was intended to achieve the 128-bit security level. According to Table 1, this parameter set is estimated to provide approximately 2 bits less security against the proposed attack than claimed.

6 Conclusion

We revisited the concrete hardness of RSD and showed that substantial improvements are possible without introducing a fundamentally new solving paradigm. By jointly balancing zero guesses and syndrome-based eliminations across the regular blocks, refining the complexity accounting of the shared preprocessing, and sharpening the analysis of the final overdetermined system, we obtain significantly lower hybrid-algebraic complexities for many parameters used in the literature.

The concrete impact is substantial. On standard benchmark instances, the previous best attack complexity drops by up to 29 bits over non-binary fields and 25 bits over binary fields. Across the seven cryptographic applications analyzed here, at least one parameter set in every application falls below its claimed 128-bit target, with estimated losses of up to 66 bits. These results show that

Table 3. Complexity estimates for the parameter sets from Wolverine [30] and Ferret [31] that were proposed to provide 128-bit security.

Source	(n, k, h)	q	AGB [8]	ISD [20]	R-ISD [12]	RP [17]	HAGB [28]	Ours
[31]	$(609728, 36288, 1269)$	2	143	138	140	144	126.59	**98.38**
	$(10805248, 589760, 1319)$		164	152	157	155	149.80	**110.23**
[30]	$(642048, 19870, 2508)$	$2^{61}-1$	126	154	172	147	110.78	**82.78**
	$(10805248, 589760, 1319)$		165	160	177	155	149.48	**110.23**

parameter selection for RSD-based constructions should account for the refined hybrid-algebraic complexity rather than relying on earlier estimates.

References

1. Aguilar Melchor, C., Bettaieb, S., Bidoux, L., Feneuil, T., Gaborit, P., Gama, N., Gueron, S., Howe, J., Hülsing, A., Joseph, D., Joux, A., Kulkarni, M., Persichetti, E., Randrianarisoa, T.H., Rivain, M., Yue, D.: SDitH — Syndrome Decoding in the Head. Tech. rep., National Institute of Standards and Technology (2024), available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures
2. Alman, J., Duan, R., Williams, V.V., Xu, Y., Xu, Z., Zhou, R.: More asymmetry yields faster matrix multiplication. In: Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). pp. 2005–2039 (2025). https://doi.org/10.1137/1.9781611978322.63, https://epubs.siam.org/doi/abs/10.1137/1.9781611978322.63
3. Andrén, D., Hellström, L., Markström, K.: On the complexity of matrix reduction over finite fields. Advances in Applied Mathematics **39**(4), 428–452 (2007). https://doi.org/https://doi.org/10.1016/j.aam.2006.08.008, https://www.sciencedirect.com/science/article/pii/S0196885807000711
4. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 520–536. Springer, Berlin, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_31
5. Berlekamp, E., McEliece, R., van Tilborg, H.: On the inherent intractability of certain coding problems (corresp.). IEEE Transactions on Information Theory **24**(3), 384–386 (1978). https://doi.org/10.1109/TIT.1978.1055873
6. Bouillaguet, C., Delaplace, C., Trimoska, M.: A simple deterministic algorithm for systems of quadratic polynomials over $\mathbb{F}_2$. In: Symposium on Simplicity in Algorithms (SOSA). pp. 285–296. SIAM (2022). https://doi.org/10.1137/1.9781611977066.22, https://hal.science/hal-03363031
7. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y.: Compressing vector OLE. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018. pp. 896–912. ACM Press (Oct 2018). https://doi.org/10.1145/3243734.3243868
8. Briaud, P., Øygarden, M.: A new algebraic approach to the regular syndrome decoding problem and implications for PCG constructions. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 391–422. Springer, Cham (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_14

9. Bui, D., Carozza, E., Couteau, G., Goudarzi, D., Joux, A.: Faster signatures from MPC-in-the-head. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024, Part I. LNCS, vol. 15484, pp. 396–428. Springer, Singapore (Dec 2024). https://doi.org/10.1007/978-981-96-0875-1_13
10. Carozza, E., Couteau, G., Joux, A.: Short signatures from regular syndrome decoding in the head. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 532–563. Springer, Cham (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_19
11. Cui, H., Liu, H., Yan, D., Yang, K., Yu, Y., Zhang, K.: ReSolveD: Shorter signatures from regular syndrome decoding and VOLE-in-the-head. In: Tang, Q., Teague, V. (eds.) PKC 2024, Part I. LNCS, vol. 14601, pp. 229–258. Springer, Cham (Apr 2024). https://doi.org/10.1007/978-3-031-57718-5_8
12. Esser, A., Santini, P.: Not just regular decoding: Asymptotics and improvements of regular syndrome decoding attacks. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part VI. LNCS, vol. 14925, pp. 183–217. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68391-6_6
13. Feneuil, T., Joux, A., Rivain, M.: Syndrome decoding in the head: Shorter signatures from zero-knowledge proofs. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 541–572. Springer, Cham (Aug 2022). https://doi.org/10.1007/978-3-031-15979-4_19
14. Hazay, C., Orsini, E., Scholl, P., Soria-Vazquez, E.: TinyKeys: A new approach to efficient multi-party computation. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part III. LNCS, vol. 10993, pp. 3–33. Springer, Cham (Aug 2018). https://doi.org/10.1007/978-3-319-96878-0_1
15. Hazay, C., Orsini, E., Scholl, P., Soria-Vazquez, E.: TinyKeys: A new approach to efficient multi-party computation. Journal of Cryptology **35**(2), 13 (Apr 2022). https://doi.org/10.1007/s00145-022-09423-5
16. Joux, A., Vitse, V.: A crossbred algorithm for solving boolean polynomial systems. In: Kaczorowski, J., Pieprzyk, J., Pomykała, J. (eds.) Number-Theoretic Methods in Cryptology. pp. 3–21. Springer International Publishing, Cham (2018). https://doi.org/https://doi.org/10.1007/978-3-319-76620-1_1
17. Kim, J., Lee, C.: Reduce and prange: Revisiting prange's information set decoding for LPN and RSD. Cryptology ePrint Archive, Report 2024/276 (2024), https://eprint.iacr.org/2024/276
18. Kolesnikov, V., Peceny, S., Raghuraman, S., Rindal, P.: Stationary syndrome decoding for improved PCGs. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part I. LNCS, vol. 16000, pp. 284–317. Springer, Cham (Aug 2025). https://doi.org/10.1007/978-3-032-01855-7_10
19. Liu, H., Wang, X., Yang, K., Yin, L., Yu, Y.: AGB 2.0: Refined algebraic attack against regular syndrome decoding for PCG applications. Cryptology ePrint Archive, Report 2025/1476 (2025), https://eprint.iacr.org/2025/1476
20. Liu, H., Wang, X., Yang, K., Yu, Y.: The hardness of LPN over any integer ring and field for PCG applications. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part VI. LNCS, vol. 14656, pp. 149–179. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58751-1_6
21. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 107–124. Springer, Berlin, Heidelberg (Dec 2011). https://doi.org/10.1007/978-3-642-25385-0_6

22. Miao, P., Murphy, A., Srinivasan, A., Tromanhauser, M.: Pseudorandom correlation generators for multiparty beaver triples over $\mathbb{F}_2$. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025, Part VII. LNCS, vol. 16251, pp. 474–504. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5122-4_15
23. Ouyang, Y., Tang, D., Xu, Y.: Code-based zero-knowledge from VOLE-in-the-head and their applications: Simpler, faster, and smaller. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024, Part V. LNCS, vol. 15488, pp. 436–470. Springer, Singapore (Dec 2024). https://doi.org/10.1007/978-981-96-0935-2_14
24. Prange, E.: The use of information sets in decoding cyclic codes. IRE Transactions on Information Theory **8**(5), 5–9 (1962). https://doi.org/10.1109/TIT.1962.1057777
25. Stern, J.: A method for finding codewords of small weight. In: Cohen, G., Wolfmann, J. (eds.) Coding Theory and Applications. pp. 106–113. Springer Berlin Heidelberg, Berlin, Heidelberg (1989). https://doi.org/https://doi.org/10.1007/BFb0019850
26. Sun, Y., Liu, H., Yang, K., Yu, Y., Wang, X., Weng, C.: Committed vector oblivious linear evaluation and its applications. In: Huang, C.Y., Chen, J.C., Shieh, S.P., Lie, D., Cortier, V. (eds.) ACM CCS 2025. pp. 3635–3648. ACM Press (Oct 2025). https://doi.org/10.1145/3719027.3744887
27. Sun, Y., Liu, H., Yang, K., Yu, Y., Wang, X., Weng, C.: Committed vector oblivious linear evaluation and its applications. Cryptology ePrint Archive, Paper 2025/1037 (2025), https://eprint.iacr.org/2025/1037
28. Wang, T., Wang, A., Yang, K., Liu, H., Yu, Y., Zhang, J., Wang, X.: A hybrid algorithm for the regular syndrome decoding problem. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025, Part IV. LNCS, vol. 16248, pp. 466–497. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5113-2_15
29. Wang, T., Wang, A., Yang, K., Liu, H., Yu, Y., Zhang, J., Wang, X.: A hybrid algorithm for the regular syndrome decoding problem. Cryptology ePrint Archive, Report 2025/1284 (2025), https://eprint.iacr.org/2025/1284
30. Weng, C., Yang, K., Katz, J., Wang, X.: Wolverine: Fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits. In: 2021 IEEE Symposium on Security and Privacy. pp. 1074–1091. IEEE Computer Society Press (May 2021). https://doi.org/10.1109/SP40001.2021.00056
31. Yang, K., Weng, C., Lan, X., Zhang, J., Wang, X.: Ferret: Fast extension for correlated OT with small communication. In: Ligatti, J., Ou, X., Katz, J., Vigna, G. (eds.) ACM CCS 2020. pp. 1607–1626. ACM Press (Nov 2020). https://doi.org/10.1145/3372297.3417276

Supplementary A Experimental validations

Supplementary A.1 Experimental evidence for heuristic assumptions

Our complexity analysis relies on two heuristic assumptions. First, the submatrix $\mathbf{H}_{\bar{\mathsf{B}}_2}$ selected in Step 1 is assumed to be invertible with approximately the probability of a uniformly random square matrix over the underlying field. Second, when the combined zero guess is incorrect, the final overdetermined system $\mathcal{F}_2''$ is assumed to be inconsistent with high probability.

We provide experimental evidence for the heuristic assumptions used in our complexity analysis and verify the algebraic recovery procedure of the refined attack. Our experiments focus on RSD over the binary field $\mathbb{F}_2$.

This choice is deliberate. For the two heuristic assumptions considered here, the binary field provides a particularly informative test setting. First, the probability that a random square matrix over $\mathbb{F}_2$ is invertible converges to approximately 0.2889, whereas over a large field such as $\mathbb{F}_{2^{128}}$ it is essentially one. Second, for a random overdetermined linear system with an excess of $\xi(N)$ equations, the expected number of solutions is bounded by $q^{-\xi(N)}$. Hence, the predicted rejection of incorrect zero guesses becomes substantially stronger as the field size increases. Over $\mathbb{F}_2$, both phenomena therefore remain nontrivial and can be observed empirically over many random instances.

Accordingly, our experiments are intended to test whether the qualitative behavior assumed in the analysis is already visible in this more informative binary setting, rather than to provide an exhaustive empirical validation over all finite fields. The corresponding large-field estimates are based on the random-matrix and random-system analyses of Section 2.3.

Invertibility in Step 1. For $n \in \{100, 200, \ldots, 1000\}$, we generate 10,000 random RSD instances for each n, and use the fixed ratios $k/n = 0.2$ and $h/n = 0.04$ as an interior operating point for the experiments. These values lie away from the extreme parameter regimes appearing in Table 1 and give a fixed regular block size $\beta = n/h = 25$. Keeping these ratios constant allows us to vary n while preserving the same block structure and relative code/error rates, so that the experiments isolate the effect of increasing instance size. The purpose of this choice is not to represent all parameter sets in Table 1, but to provide a stable and computationally manageable setting in which to examine the qualitative behavior of the heuristic assumptions. The observed probability that $\mathbf{H}_{\bar{\mathsf{B}}_2}$ is invertible is approximately 0.289 throughout the tested range, in agreement with the random-matrix prediction $p(\infty, 2) \approx 0.2889$. This supports the constant-factor treatment of this event used in our complexity estimates.

Rejection of incorrect zero guesses. We next test the random-system heuristic underlying the rejection of incorrect zero guesses. As discussed in Section 2.3, when $m_2'' - N_2'' \geq \xi(N_2'')$, a random overdetermined system is expected to have at most $2^{-\xi(N_2'')}$ solutions on average. We therefore measure the probability that $\mathcal{F}_2''$ is inconsistent when constructed from an incorrect zero guess.

We perform two experiments, using 10,000 random instances for each parameter choice. First, for RSD$(100, 20, 2)$ with $(\eta_1, \eta, \eta_{\text{pre}}) = (9, 9, 6)$, we vary $a \in \{0.5, 0.6, \ldots, 3.0\}$ in $\xi(N) = aN$. As shown in Fig Supplementary A.1 (a), the inconsistency probability increases with a and approaches one as the excess number of equations increases. Second, we fix $a = 1$ and consider RSD$(n, 0.2n, 0.04n)$ for $n \in \{50, 75, \ldots, 300\}$, using the optimized attack parameters for each instance size. Fig Supplementary A.1 (b) shows that the inconsistency probability also increases with n and is close to one for the larger tested instances.

These experiments are consistent with the random-system heuristic used in our complexity analysis. They do not constitute a proof for the full parameter range considered in Table 1, but provide empirical evidence that incorrect zero guesses are rejected with high probability under the parameter choices used by the attack.

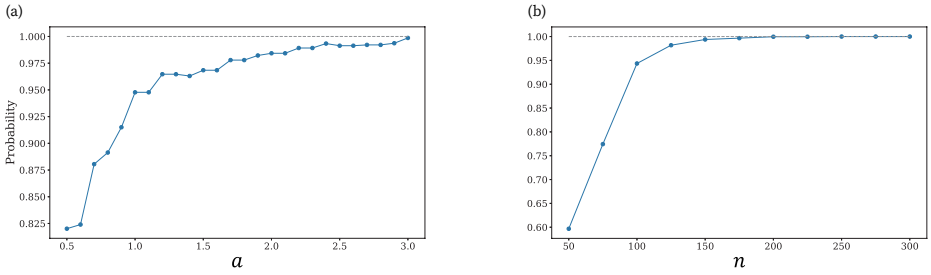


Fig. Supplementary A.1. Empirical probability that the final system $\mathcal{F}_2''$ is inconsistent under an incorrect zero guess, estimated from 10,000 random binary RSD instances per point. (a) RSD$(100, 20, 2)$ with $(\eta_1, \eta, \eta_{\text{pre}}) = (9, 9, 6)$, varying a in $\xi(N) = aN$. (b) RSD$(n, 0.2n, 0.04n)$ for $n \in \{50, 75, \ldots, 300\}$ with $a = 1$, using the optimized attack parameters.

Supplementary A.2 Recovery conditioned on correct zero guesses

We next verify that the algebraic stages of the refined attack recover a valid RSD solution when the zero guesses are correct. To isolate this part of the attack from the potentially expensive search for correct guesses, we condition both guessing stages on a known solution $\mathbf{e}$, choosing $\bar{\mathsf{B}}_1$ and $\bar{\mathsf{B}}_4$ such that $\mathbf{e}_{\bar{\mathsf{B}}_1 \cup \bar{\mathsf{B}}_4} = 0$. We then execute the remaining algebraic procedure, including syndrome-based elimination, preprocessing, linearization, and reconstruction, and verify the recovered vector against the original RSD instance.

We test 10 independently generated binary RSD$(512, 256, 16)$ instances. For these parameters, the estimated complexity of the full attack is approximately $2^{34.73}$ field operations with $(\eta_1, \eta, \eta_{\text{pre}}) = (224, 225, 14)$. Under the conditioned correct-guess setting, the attack successfully recovers a valid solution in all 10

instances. In each case, the recovered vector $\mathbf{x}$ satisfies $\mathbf{Hx} = \mathbf{s}$ and has exactly one nonzero coordinate in each regular block.

These experiments provide a direct sanity check of the algebraic recovery procedure under correct zero guesses. They do not test the cost of finding such guesses, and therefore should not be interpreted as an empirical measurement of the total attack complexity.

Supplementary A.3 A toy example of the refined attack

We illustrate the four stages of the refined attack described in Section 3.3, incorporating the binary modifications from Section 4.1 and the row budgets from Corollary 2. All computations are carried out over $\mathbb{F}_2$, so that $x_j^2 = x_j$. The example follows the sequence

$$\mathcal{F} \longrightarrow \mathcal{F}_1 \longrightarrow \mathcal{F}_2 \longrightarrow \mathcal{F}_2' \longrightarrow \mathcal{F}_2'',$$

with the subsystem $\mathcal{F}_1'$ shared across the second-stage guesses. For clarity, we display only the successful block partition and second guess. In the recorded run, three earlier partitions were rejected because the corresponding linear coefficient matrices were singular.

The binary instance. Let $(n, k, h) = (18, 10, 3)$. The coordinates are divided into three blocks

$$\mathsf{B}_i = \{(i-1)\beta + 1, \ldots, i\beta\}, \qquad i \in [h],$$

which correspond to the three column groups below where $\beta := n/h = 6$.

$$[\,\mathbf{H} \mid \mathbf{s}\,] = \left[\begin{array}{ccc|c} 000101 & 100011 & 111001 & 0 \\ 111100 & 000010 & 010111 & 0 \\ 111011 & 011111 & 111101 & 1 \\ 011111 & 101101 & 000001 & 0 \\ 000100 & 010011 & 000000 & 1 \\ 000110 & 000100 & 000101 & 1 \\ 110110 & 110100 & 100101 & 0 \\ 110111 & 100010 & 000010 & 1 \end{array}\right], \qquad \mathbf{e} = 000010\,000001\,100000.$$

The known solution satisfies $\mathrm{supp}(\mathbf{e}) = \{5, 12, 13\}$, with exactly one nonzero coordinate in each block. We use $\mathbf{e}$ only to verify the final output; it is not used when choosing the guesses or solving the intermediate systems.

The initial polynomial system is

$$\mathbf{Hx} = \mathbf{s}, \qquad \sum_{j \in \mathsf{B}_i} x_j = 1, \qquad x_u x_v = 0 \quad (u < v,\ u, v \in \mathsf{B}_i), \quad i \in [h]. \tag{11}$$

In the binary setting, the block-sum equations provide additional linear constraints. Together with the pairwise product equations, they enforce that each block contains exactly one nonzero coordinate.

Parameters and block partition. We choose

$$(\eta_1, \eta, \eta_{\mathrm{pre}}) = (2, 4, 2), \qquad \eta_2 = \eta - \eta_1 = 2.$$

The balanced allocation is determined by

$$(\omega, f_{\mathrm{lin}}) = (3, 2), \quad (\mu_1, f_1) = (0, 2), \quad (\mu, f) = (1, 1), \quad (\nu, f_{\mathrm{pre}}) = (0, 2).$$

The successful block partition is shown below. The sets $\bar{\mathsf{B}}_1, \bar{\mathsf{B}}_2$, and $\bar{\mathsf{B}}_3$ are fixed before preprocessing, whereas $\bar{\mathsf{B}}_4$ and $\bar{\mathsf{B}}_5$ depend on the second-stage guess.

i	$\mathsf{B}_{i,1}$	$\mathsf{B}_{i,2}$	$\mathsf{B}_{i,3}$	$\mathsf{B}_{i,4}$	$\mathsf{B}_{i,5}$
1	$\{4\}$	$\{1, 2, 3, 6\}$	$\{5\}$	$\varnothing$	$\varnothing$
2	$\varnothing$	$\{7, 8, 9, 12\}$	$\{10\}$	$\{11\}$	$\varnothing$
3	$\{18\}$	$\{13, 14, 16\}$	$\varnothing$	$\{17\}$	$\{15\}$
$\lvert\bar{\mathsf{B}}_j\rvert$	2	11	2	2	1

Thus,

$$\bar{\mathsf{B}}_1 = \{4, 18\}, \qquad \bar{\mathsf{B}}_3 = \{5, 10\},$$

while the successful second guess gives

$$\bar{\mathsf{B}}_4 = \{11, 17\}, \qquad \bar{\mathsf{B}}_5 = \{15\}.$$

The corresponding binary monomial counts are

$$n_1 = k - h - \eta_1 = 5, \qquad N_1 = \tbinom{n_1}{2} + n_1 = 15,$$
$$n_2 = k - h - \eta = 3, \qquad N_2 = \tbinom{n_2}{2} + n_2 = 6.$$

The numbers of quadratic equations before row selection are

$$m_1 = \sum_{i=1}^{h} \tbinom{\beta - \lvert\mathsf{B}_{i,1}\rvert}{2} = 10 + 15 + 10 = 35,$$
$$m_2 = \sum_{i=1}^{h} \tbinom{\beta - \lvert\mathsf{B}_{i,1}\rvert - \lvert\mathsf{B}_{i,4}\rvert}{2} = 10 + 10 + 6 = 26.$$

After the first guess, there are $\Delta_1 = \binom{1}{2} + \binom{2}{2} + \binom{2}{2} = 2$ free monomials, whereas $\Delta_2 = 0$ after the second guess. Since no two positions in $\bar{\mathsf{B}}_3$ belong to the same block, we have $\delta_1 = 2$ and $\delta_2 = 0$. Hence,

$$\gamma = \tbinom{\eta_{\mathrm{pre}}}{2} = 1, \qquad \tau = \sum_{i=1}^{h} \lvert\mathsf{B}_{i,2}\rvert \, \lvert\mathsf{B}_{i,4}\rvert = 4 + 3 = 7,$$
$$N_2' = N_2 - \gamma - \delta_2 = 5, \qquad N_2'' = N_2' - \eta_{\mathrm{pre}} = 3.$$

The corresponding untruncated row counts used in the complexity analysis are

$$m_2' = m_2 - \gamma - \delta_2 = 25, \qquad m_2'' = m_2' - \eta_{\mathrm{pre}} = 23.$$

These quantities should not be confused with the numbers of rows actually retained in this toy example.

Choosing the row budgets. For this small instance, we use a single excess equation, $\xi(N_2'') = 1$. This corresponds to the integer choice $\xi(N) = \lfloor \frac{1}{2} N \rfloor$ at $N = 3$.

The backward row budgets are therefore

$$\begin{aligned} \check{m}_2'' &= N_2'' + \xi(N_2'') = 4, \\ \check{m}_2' &= \check{m}_2'' + \eta_{\mathrm{pre}} = 6, \\ \check{m}_1' &= \check{m}_2' + \tau = 13, \\ \check{m}_1 &= \check{m}_1' + \gamma + \delta_1 = 16. \end{aligned}$$

Thus, only $\check{m}_1 - \Delta_1 = 14$ non-free equations are selected for preprocessing. The parameter constraints are satisfied:

$$\eta + \eta_{\mathrm{pre}} = 6 \le k - h = 7,$$

and

$$\left\lceil \frac{n - k + h + \eta + \eta_{\mathrm{pre}}}{h} \right\rceil = 6 \le \beta.$$

Step 1. First stage of zero guessing. We first set $\mathbf{x}_{\bar{\mathsf{B}}_1} = 0$. Combining the syndrome equations with the three block-sum equations yields eleven linear equations in the sixteen remaining coordinates; the three additional equations arise from the binary block-sum constraints.

Let L be the resulting 11×16 coefficient matrix and let

$$\mathbf{b} = (\mathbf{s}^{\mathsf{T}}, 1, 1, 1)^{\mathsf{T}}.$$

For the successful partition, the selected submatrix $\mathbf{L}_{\bar{\mathsf{B}}_2}$ is invertible. Writing $\mathbf{Q} = \mathbf{L}_{\bar{\mathsf{B}}_2}^{-1}$, we obtain

$$\mathbf{x}_{\bar{\mathsf{B}}_2} = \mathbf{Q}\mathbf{b} + \mathbf{Q}\mathbf{L}_{[n]\setminus(\bar{\mathsf{B}}_1 \cup \bar{\mathsf{B}}_2)} \mathbf{x}_{[n]\setminus(\bar{\mathsf{B}}_1 \cup \bar{\mathsf{B}}_2)}.$$

This is precisely the binary affine substitution used in Step 1 of the general attack.

Substituting these affine expressions into the pairwise product equations of (11) gives the system $\mathcal{F}_1$ in the five remaining variables

$$(x_5, x_{10}, x_{11}, x_{15}, x_{17}).$$

The two free equations are

$$x_{10}x_{11} = 0, \qquad x_{15}x_{17} = 0.$$

Using the linearization notation of the main text, write $\mathcal{F}_1$ as

$$\mathbf{M}_1 \mathbf{z}_1 = \mathbf{d}_1,$$

where $\mathbf{z}_1$ contains $N_1 = 15$ linear and square-free quadratic monomials.

We order the two free monomials first and eliminate their columns, obtaining

$$\hat{\mathbf{M}}_1 = \begin{pmatrix} \mathbf{I}_{\Delta_1} & \mathbf{O} \\ \mathbf{O} & \hat{\mathbf{M}}_1'' \end{pmatrix}, \qquad \Delta_1 = 2.$$

The lower subsystem has $m_1 - \Delta_1 = 33$ equations and $N_1 - \Delta_1 = 13$ monomial columns. We select $14 (= \check{m}_1 - \Delta_1)$ equations according to the row budget above. As an example, suppose that Step 1 yields the following system:

$$\begin{array}{lll}
x_{17} + x_{10}(x_{15} + x_{17}) = 0, & x_{11}x_{17} = 0, & x_{10}(x_{15} + x_{17}) = 0, \\
x_5(x_{11} + x_{15} + x_{17}) = 0 & x_{11}(x_{15} + x_{17}) = 0, & x_{10}(1 + x_{17}) = 0, \\
x_5(1+x_{15})+x_{10}(1+x_{15})+x_{11}+x_{15}=1, & x_5x_{11} = 0, & x_5x_{10} = 0, \\
x_{10}(1 + x_{15} + x_{17}) + x_{11}x_{17} = 0, & x_{17}(1 + x_5 + x_{10}) = 0, & x_{17}(x_{10} + x_{11}) = 0 \\
x_{15}(1 + x_5 + x_{10}) = 0, & x_{10}(x_5 + x_{15} + x_{17}) = 0 &
\end{array}$$

Step 2. Preprocessing via Gaussian elimination. Fixing $\bar{\mathsf{B}}_3$, define

$$\mathbf{x}_{\text{pre}} = (x_5, x_{10})^{\mathsf{T}}, \qquad \mathbf{x}_{\text{var}} = (x_{11}, x_{15}, x_{17})^{\mathsf{T}}.$$

The only quadratic monomial involving only preprocessing variables is x_5x_{10}. Gaussian elimination isolates this monomial in an equation that also contains the monomial x_5x_{10}. The remaining $13 (= \check{m}_1')$ equations form the selected shared subsystem $\mathcal{F}_1'$, with $12 (= N_1 - \gamma - \delta_1)$ monomial columns. Each equation has the form

$$\mathbf{c}_l^{\mathsf{T}}\mathbf{x}_{\text{pre}} + \mathbf{x}_{\text{pre}}^{\mathsf{T}}\mathbf{D}_l\mathbf{x}_{\text{var}} + \mathbf{x}_{\text{var}}^{\mathsf{T}}\mathbf{E}_l\mathbf{x}_{\text{var}} + \mathbf{e}_l^{\mathsf{T}}\mathbf{x}_{\text{var}} = d_l'. \tag{12}$$

After Step 2, we obtain the following system:

$$\begin{array}{lll}
x_{17} + x_{10}(x_{15} + x_{17}) = 0, & x_{11}x_{17} = 0, & x_{10}(x_{15} + x_{17}) = 0, \\
x_5(x_{11} + x_{15} + x_{17}) = 0 & x_{11}(x_{15} + x_{17}) = 0, & x_{10}(1 + x_{17}) = 0, \\
x_5(1+x_{15})+x_{10}(1+x_{15})+x_{11}+x_{15}=1, & x_5x_{11} = 0, & x_{17}(x_{10} + x_{11}) = 0 \\
x_{10}x_{15} + x_{11}x_{17} = 0, & x_{17}(1 + x_5 + x_{10}) = 0, & \\
x_{15}(1 + x_5 + x_{10}) = 0, & x_{10}(x_{15} + x_{17}) = 0 &
\end{array}$$

Note that, in the above system, there are at most $|\mathbf{x}_{\text{pre}}|$ equations satisfying $\mathbf{c}_l \neq 0$. This is achieved during preprocessing.

Step 3. Second stage of zero guessing. For the successful second guess, choose

$$\bar{\mathsf{B}}_4 = \{11, 17\}$$

and set $x_{11} = x_{17} = 0$. The remaining variable vector is

$$\mathbf{x}_{\text{var}}' = \mathbf{x}_{\bar{\mathsf{B}}_5} = (x_{15}).$$

Substitution into $\mathcal{F}_1$ gives $\mathcal{F}_2$, while substitution into the selected subsystem $\mathcal{F}_1'$ gives $\mathcal{F}_2'$.

In the chosen row basis, 5 of the 13 selected equations become identically zero, leaving eight nonzero equations involving $N_2' = 5$ monomials. Note that the structural quantity $\tau = 7$ appearing in the backward row budget is not the

number of visibly vanishing rows in this particular row-reduced representation. Moreover, the eight displayed rows need not be linearly independent; repeated equations are retained. After Step 3, we obtain the following system:

$$\begin{aligned} x_5 + x_{15} + x_5x_{15} &= 1, & x_{10} &= 0, \\ x_{10}x_{15} &= 0, & x_5x_{15} &= 0, \\ x_{15} + x_5x_{15} + x_{10}x_{15} &= 0 \end{aligned}$$

Note that, in this example, the monomial $x_{10}x_{15} = 0$ remains in four other equations.

Step 4. Solving the final system by linearization. Since the previous step yields $\check{m}_2''$ equations containing no linear terms in $\mathbf{x}_{\mathrm{pre}}$, the remaining subsystem $\mathcal{F}_2''$ consists of 4 equations and has $N_2'' = 3$ monomial ccolumns. Using the monomial ordering of the main text, let

$$\mathbf{z}_2'' = (x_5x_{15},\, x_{10}x_{15},\, x_{15})^{\mathsf{T}}.$$

There is no additional quadratic monomial supported entirely on $\bar{\mathsf{B}}_5$, since $x_{15}^2 = x_{15}$ over $\mathbb{F}_2$. Retaining $\check{m}_2'' = 4$ equations gives

$$\mathbf{M}_2''\mathbf{z}_2'' = \mathbf{d}_2'', \qquad \mathbf{M}_2'' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \qquad \mathbf{d}_2'' = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{13}$$

The coefficient matrix has rank $3 = N_2''$, and the corresponding augmented matrix reduces to

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right].$$

Hence,

$$\mathbf{z}_2'' = (0, 0, 0)^{\mathsf{T}}$$

is the unique linearized solution. Substituting this solution into the two recovery equations yields $x_5 = 1$ and $x_{10} = 0$. The recovered variables satisfy the required mixed-monomial identities, as well as the quadratic equation set aside during Step 2. Finally, applying the affine substitutions obtained in Step 1 gives

$$\mathbf{x} = \mathsf{000010\,000001\,100000} = \mathbf{e}.$$

Direct verification confirms that

$$\mathbf{Hx} = \mathbf{s} \qquad \text{and} \qquad |\mathbf{x}_{\mathsf{B}_i}| = 1 \quad \text{for every } i \in [h].$$

The final system also has the prescribed excess $\check{m}_2'' - N_2'' = 1$.

This example illustrates the complete recovery procedure for a successful guess. It is intended only to demonstrate the mechanics of the refined attack and does not, by itself, establish an overwhelming rejection probability for incorrect guesses. In particular, consistency of a selected linearized subsystem must not be regarded as a substitute for verifying the solution against the original RSD instance.

System sizes. For reference, the following table summarizes the dimensions of the main linearized systems encountered above. The column counts refer only to coefficient columns and should be distinguished from the untruncated equation counts m_1, m_2, m_2', and m_2'' used in the complexity analysis.

Step	System or operation		Rows × columns
1	Linear system after the first guess	11×16	$(n-k+h) \times (n-\eta_1)$
1	Non-free part of $\mathcal{F}_1$	33×13	$(m_1 - \Delta_1) \times (N_1 - \Delta_1 + 1)$
1	Selected shared of $\mathcal{F}_1$	14×13	$(\check{m}_1 - \Delta_1) \times (N_1 - \Delta_1 + 1)$
2	After preprocessing with selected $\mathcal{F}_1$	13×12	$(\check{m}_1 - \gamma - \delta_1) \times (N_1 - \gamma - \delta_1 + 1)$
3	$\mathcal{F}_2'$	8×5	$(\check{m}_2' + \epsilon) \times N_2'$
4	$\mathcal{F}_2''$	4×3	$\check{m}_2'' \times N_2''$

Note that, in the size of $\mathcal{F}_2'$, ϵ denotes the number of equations that remain after the second zero-guessing step due to the row-budget constraint. In Step 3, at most τ equations can remain nonvanishing; therefore, we have $\epsilon \le \tau$.

Supplementary B Asymptotic complexity

Supplementary B.1 Non-binary fields

We now analyze the asymptotic complexity of the refined attack over non-binary fields. Our analysis follows the same asymptotic viewpoint as in [28], but we keep the exact binomial term in the equation count.

Let $R = \frac{k}{n}, \rho = \frac{h}{n}$ and $\beta = \frac{n}{h} = \frac{1}{\rho}$. We count the number of field operations over $\mathbb{F}_q$. Here, we focus on the large-field setting of Corollary 1, where $\xi(N_2'') = c_{\text{excess}}$ for some constant $c_{\text{excess}} \in \mathbb{N}$.

For the asymptotic upper bound, it is sufficient to consider the simple choice $\eta_1 = 0, \eta_{\text{pre}} = 0$ and $\eta = rn$ for some $0 \le r \le R$. This means that all zero guesses are made in the second guessing stage. This choice is not necessarily optimal for concrete parameters, but it gives a clean asymptotic upper bound. With this choice, the number of remaining variables is $n_2 = k - \eta = (R-r)n$, and hence the number of linearized monomials is $N_2 = \frac{n_2(n_2+3)}{2} = \frac{(R-r)^2n^2+3(R-r)n}{2}$. On the other hand, by the definition of m_2 in Section 3.4, after $\eta = rn$ balanced zero guesses we have, ignoring only the integer rounding among blocks,

$$m_2 = h\binom{\beta - r/\rho}{2} = \frac{n}{2\rho}(1-r)(1-r-\rho).$$

The exact rounded expression is

$$m_2 = f\binom{\beta-\mu-1}{2} + (h-f)\binom{\beta-\mu}{2}, \qquad \mu = \left\lfloor \tfrac{\eta}{h} \right\rfloor, \quad f = \eta \bmod h.$$

The continuous expression above is used only to make the asymptotic dependence on R and ρ explicit.

A key point is that the preprocessing terms cancel in the final overdeterminedness condition. Indeed, by the definitions in Section 3.4,

$$m_2' = m_2 - \gamma - \delta_2,\ N_2' = N_2 - \gamma - \delta_2 \quad \text{and} \quad m_2'' = m_2' - \eta_{\text{pre}},\ N_2'' = N_2' - \eta_{\text{pre}}.$$

Therefore, it holds $m_2'' - N_2'' = m_2 - N_2$. Thus the condition $m_2'' - N_2'' \geq c_{\text{excess}}$ is implied by

$$\tfrac{n}{2\rho}(1-r)(1-r-\rho) - \tfrac{(R-r)^2n^2+3(R-r)n}{2} \geq c_{\text{excess}}.$$

Equivalently,

$$(R-r)^2\rho n + 3(R-r)\rho + \tfrac{2c_{\text{excess}}\rho}{n} \leq (1-r)(1-r-\rho). \tag{14}$$

We also record the asymptotic form of the guessing probability. Let $c = \lfloor r/\rho \rfloor$ and $\theta = r/\rho - c$. Then the exact rounded exponent of the balanced zero guessing is $p_\eta^{-1} = 2^{\rho n \Lambda_\beta(r) + O(1)}$ where

$$\Lambda_\beta(r) = \theta \log_2 \tfrac{1}{1-(c+1)/\beta} + (1-\theta)\log_2 \tfrac{1}{1-c/\beta}. \tag{15}$$

If the integer rounding is ignored, or if $r\beta$ is an integer, then this simplifies to

$$p_\eta = (1-r)^h, \quad p_\eta^{-1} = 2^{\rho n \log_2(1/(1-r))}. \tag{16}$$

Theorem 3. *Let $R = k/n$ and $\rho = h/n$. The refined attack solves* $\mathrm{RSD}(n,k,h)$ *over non-binary fields in polynomial time if*

$$R^2\rho n + 3R\rho \leq 1 - \rho \tag{17}$$

equivalent to $\frac{n}{2\rho}(1-\rho) \geq \frac{R^2n^2+3Rn}{2}$. In particular, this condition can be written as $R^2\rho \leq \frac{1-\rho-o(1)}{n}$. Following the coarse asymptotic convention, this is written as $R^2\rho \leq n^{-1}$.

Proof. Take $\eta_1 = \eta = \eta_{\text{pre}} = 0$. Then no zero guessing is performed, and hence $p_{\eta_1} = p_\eta = 1$. With this choice, after eliminating $n-k$ variables by the linear equations, the resulting quadratic system has $m_2 = h\binom{\beta}{2} = \frac{n}{2\rho}(1-\rho)$ equations, and the number of remaining variables is $n_2 = k = Rn$. Therefore, the number of linearized monomials is $N_2 = \frac{n_2(n_2+3)}{2} = \frac{R^2n^2+3Rn}{2}$.

Among the m_2 equations, exactly Δ_2 are free equations, and the corresponding Δ_2 monomials are free monomials. Removing them yields a linearized system with $m_2 - \Delta_2$ equations and $N_2 - \Delta_2$ monomials. Hence it is sufficient that

$m_2 - \Delta_2 \geq N_2 - \Delta_2$, which is equivalent to $m_2 \geq N_2$. Substituting the above expressions for m_2 and N_2, we obtain

$$\tfrac{n}{2\rho}(1-\rho) \geq \tfrac{R^2n^2+3Rn}{2},$$

equivalently, $R^2\rho n + 3R\rho \leq 1-\rho$. Under this condition, the resulting linearized system can be solved in polynomial time. Since all remaining steps consist of Gaussian eliminations and substitutions on matrices of polynomial size, the total complexity is polynomial in n. □

Theorem 4. *Assume that $k \leq n-h$. Then, for the choice $\eta_1 = 0$, $\eta = k$, and $\eta_{\mathrm{pre}} = 0$, the refined attack solves* $\mathrm{RSD}(n,k,h)$ *over non-binary fields in time*

$$\mathrm{poly}(n) \cdot 2^{\rho n \Lambda_\beta(R)+O(1)}$$

field operations.

Proof. Take $\eta_1 = 0$, $\eta = k$, and $\eta_{\mathrm{pre}} = 0$, equivalently $r = R$. A regular error vector has exactly one nonzero entry in each block of size β. Hence each block contains exactly $\beta - 1$ zero entries, and the whole vector contains exactly $h(\beta - 1) = n - h$ zero entries. Therefore, the choice $\eta = k$ can be correct only if $k \leq n-h$. Under balanced zero guessing, the k guessed-zero positions are distributed so that each block receives either $\lfloor k/h \rfloor$ or $\lceil k/h \rceil$ guesses. Since $k \leq h(\beta-1)$, no block is guessed entirely to zero. In particular, the success probability p_k is positive.

With this choice, $n_2 = k - \eta = 0$ and therefore $N_2 = \frac{n_2(n_2+3)}{2} = 0$. Hence the final linearized system is empty. For each fixed zero-guess pattern, all remaining coordinates are recovered from the linear equations obtained in the reduction step, and the resulting candidate is checked against the original RSD instance. Each trial therefore costs only polynomial time.

By (15), specialized to $r = R$, the inverse success probability of the balanced zero guessing is

$$p_k^{-1} = 2^{\rho n \Lambda_\beta(R)+O(1)}.$$

Multiplying this by the polynomial per-trial cost gives the desired result. □

Under the continuous approximation in which the blockwise rounding is ignored, the condition $k \leq n-h$ becomes $R \leq 1-\rho$, and the guessing probability satisfies $p_k = (1-R)^h$. Hence the above bound simplifies to

$$\mathrm{poly}(n) \cdot 2^{\rho n \log_2(1/(1-R))}.$$

Supplementary B.2 Binary fields

Theorem 5. *Let $R = k/n$ and $\rho = h/n$. The refined attack solves binary* $\mathrm{RSD}(n,k,h)$ *in polynomial time if $R \leq \rho$ or*

$$(R-\rho)^2\rho n + (R-\rho)\rho \leq 1-\rho. \tag{18}$$

Proof. If $R \le \rho$, then $k - h \le 0$. Hence the syndrome equation together with the h block-sum equations already leaves no asymptotic degree of freedom, and the instance can be handled in polynomial time.

Assume now that $R > \rho$, and take $\eta_1 = \eta = \eta_{\mathrm{pre}} = 0$. Then no zero guessing is performed, and hence $p_{\eta_1} = p_\eta = 1$. After eliminating the $n - k + h$ linear equations, the number of remaining variables is $n_2 = k - h = (R - \rho)n$. Therefore, the number of linearized monomials is $N_2 = \frac{n_2(n_2+1)}{2} = \frac{(R-\rho)^2 n^2 + (R-\rho)n}{2}$. On the other hand, the number of quadratic equations is $m_2 = h\binom{\beta}{2} = \frac{n}{2\rho}(1-\rho)$.

Among these, exactly Δ_2 equations are free equations, and the corresponding Δ_2 monomials are free monomials. Removing them yields a linearized system with $m_2 - \Delta_2$ equations and $N_2 - \Delta_2$ monomials. Hence it is sufficient that $m_2 - \Delta_2 \ge N_2 - \Delta_2$, which is equivalent to $m_2 \ge N_2$. Substituting the above expressions gives

$$\frac{n}{2\rho}(1-\rho) \ge \frac{(R-\rho)^2 n^2 + (R-\rho)n}{2},$$

equivalently, $(R-\rho)^2 \rho n + (R-\rho)\rho \le 1 - \rho$.

Under this condition, the resulting linearized system can be solved in polynomial time. Since all remaining steps consist of Gaussian eliminations and substitutions on matrices of polynomial size, the total complexity is polynomial in n. □

Under the continuous approximation, (18) becomes $(R-\rho)^2\rho \le \frac{1-\rho-o(1)}{n}$, and, in the coarser convention, this is written as $(R-\rho)^2\rho \le n^{-1}$.

Theorem 6. *Let $R = k/n$ and $\rho = h/n$, and assume $R > \rho$. Then, for the choice $\eta_1 = 0$, $\eta = k - h$, and $\eta_{\mathrm{fix}} = 0$, the refined attack solves binary* $\mathrm{RSD}(n, k, h)$ *in time*

$$\mathrm{poly}(n) \cdot 2^{\rho n \Lambda_\beta(R-\rho) + O(1)}$$

operations over $\mathbb{F}_2$.

Proof. Take $\eta_1 = 0, \eta = k - h$, and $\eta_{\mathrm{pre}} = 0$. Under balanced zero guessing, the $k - h$ guessed-zero positions are distributed so that each block receives either $\lfloor (k-h)/h \rfloor$ or $\lceil (k-h)/h \rceil$ guesses. Since $k - h \le n - h = h(\beta - 1)$, no block is guessed entirely to zero. In particular, the success probability p_{k-h} is positive.

With this choice, $n_2 = k - h - \eta = 0$, and therefore $N_2 = \frac{n_2(n_2+1)}{2} = 0$. Hence the final linearized system is empty. For each fixed zero-guess pattern, all remaining coordinates are recovered from the linear equations obtained in the reduction step, and the resulting candidate is checked against the original RSD instance. Each trial therefore costs only polynomial time.

By (15), specialized to $r = R - \rho$, the inverse success probability of the balanced zero guessing is $p_{k-h}^{-1} = 2^{\rho n \Lambda_\beta(R-\rho) + O(1)}$. Multiplying this by the polynomial per-trial cost gives $\mathrm{poly}(n) \cdot 2^{\rho n \Lambda_\beta(R-\rho) + O(1)}$. □

Under the continuous approximation, the guessing probability satisfies $p_{k-h} = (1 - R + \rho)^h$. Hence the above bound simplifies to

$$\mathrm{poly}(n) \cdot 2^{\rho n \log_2(1/(1-R+\rho))}.$$

Corollary 3. *Let* $\mathrm{T}_{\mathrm{total}}$ *denote the complexity of the refined attack. Under the continuous approximation used in the proof of Theorem 6, the following asymptotic upper bounds hold.*

1. *If* R *and* ρ *are constants, then* $\mathrm{T}_{\mathrm{total}} \le \mathrm{poly}(n) \cdot 2^{\rho n \log_2(1/(1-R+\rho))}$.
2. *If* R *is a constant and* $\rho = o(1)$*, then* $\mathrm{T}_{\mathrm{total}} \le \mathrm{poly}(n) \cdot 2^{\rho n \log_2(1/(1-R))+o(\rho n)}$. *In particular, in coarse* $O(\cdot)$ *notation,* $\mathrm{T}_{\mathrm{total}} = O\left(2^{\rho n \log_2(1/(1-R))}\right)$.
3. *If* $R = o(1)$ *and* $\rho = o(R)$*, then* $\log_2 \mathrm{T}_{\mathrm{total}} \le \frac{\rho n R}{\ln 2} + o(\rho n R) + \mathrm{polylog}(n)$, *and hence* $\mathrm{T}_{\mathrm{total}} = O\left(2^{\rho n R/\ln 2}\right)$.
4. *If* $R = o(1)$ *and* $\rho = \lambda R$ *for a constant* $0 < \lambda < 1$*, then* $\log_2 T_{\mathrm{total}} \le \frac{\rho n(1-\lambda)R}{\ln 2} + o(\rho n R) + \mathrm{polylog}(n)$, *and hence* $T_{\mathrm{total}} = O\left(2^{\rho n(1-\lambda)R/\ln 2}\right)$.

Proof. By above, the leading exponential term under the continuous approximation is $2^{\rho n \log_2(1/(1-R+\rho))}$. The first case is immediate.

If R is constant and $\rho = o(1)$, then $\log_2 \frac{1}{1-R+\rho} = \log_2 \frac{1}{1-R} + O(\rho)$, which gives the second case.

If $R = o(1)$ and $\rho = o(R)$, then $\log_2 \frac{1}{1-R+\rho} = \frac{R-\rho}{\ln 2} + O(R^2) = \frac{R}{\ln 2} + o(R)$, which gives the third case.

Finally, if $\rho = \lambda R$ with $0 < \lambda < 1$, then $\log_2 \frac{1}{1-R+\rho} = \log_2 \frac{1}{1-(1-\lambda)R} = \frac{(1-\lambda)R}{\ln 2} + O(R^2)$, which proves the fourth case. □