Reduce and Prange: Revisiting Prange's ISD for Solving SD/RSD over Large Fields

Jiseung Kim[1] and Changmin Lee[2]

[1] Department of Computer Science and Artificial Intelligence, Jeonbuk National University, Jeonju, Republic of Korea
jiseungkim@jbnu.ac.kr
[2] School of Cybersecurity, Korea University, Seoul, Republic of Korea
changminlee@korea.ac.kr

Abstract. Syndrome decoding (SD), together with its regular-noise variant (RSD), is fundamental to many cryptographic primitives; in the generator formulation used in cryptography, it is closely related to learning parity with noise (LPN). While recent proposals extend these problems to larger fields, the concrete security of SD over large fields remains comparatively less understood. This gap leaves room for more effective attacks against SD-based primitives over large fields. In this paper, we present an improved algorithm for solving SD over large fields. Our method modifies Prange's information-set decoding algorithm by iteratively applying Gaussian elimination to reduced-size matrices rather than to the full-size matrix. We call this the "Reduce and Prange (RP)" algorithm and demonstrate its effectiveness for both SD and its variant with regular noise over large finite fields (e.g., $\mathbb{F}_{2^{128}}$). On the comparison rows of our numerical section, RP lowers the SD estimate by up to 6 bits, and on the 128-bit recommended SD rows the drop is up to 7 bits, relative to Liu et al. (Eurocrypt'24). For SD with regular noise, regular-RP gives the lowest listed estimate on the small and medium tested parameter sets, improving on the best previously listed estimate by up to 15 bits; on the largest tested sets the algebraic estimate of Briaud and Øygarden (Eurocrypt'23) remains lower.

Keywords: Learning parity with noise · Regular syndrome decoding · Syndrome decoding

1 Introduction

As a central problem of coding theory and learning theory, the syndrome decoding (SD) problem underlies numerous cryptographic primitives such as secure arithmetic computations [7, 9, 17, 19, 20, 27, 35, 40, 42, 56, 62, 70], zero-knowledge proofs [10, 31, 38, 66] and more [4–6, 14, 25, 28, 34, 43, 44, 46, 51, 71, 72]. The SD problem is defined as follows.

Problem 1 (Syndrome Decoding (SD)). Let m, n, t be positive integers and $\mathcal{R}$ be a ring. Let $\mathcal{C}$ be a probabilistic code generation algorithm such that $\mathcal{C}(m, n, \mathcal{R})$

returns a matrix $\mathbf{A} \in \mathcal{R}^{m\times n}$, regarded as a generator matrix of a length-m, dimension-n linear code; we write $\mathbf{H} \in \mathcal{R}^{(m-n)\times m}$ for a parity-check matrix of that code, so that $\mathbf{HA} = \mathbf{0}$. Let $\chi(\mathcal{R}) = \{\chi_{m,t}(\mathcal{R})\}_{m,t\in\mathbb{N}}$ be a family of distributions that returns an error vector $\mathbf{e} \in \mathcal{R}^m$ of Hamming weight t.

In the syndrome form, the (m, n, t) syndrome decoding problem asks to recover the weight-t error $\mathbf{e}$ given the matrix $\mathbf{H}$ and the syndrome $\mathbf{H} \cdot \mathbf{e}$. Equivalently, in the generator (primal) form, which is the formulation known as learning parity with noise (LPN) in the cryptographic literature, one is given $(\mathbf{A}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e})$ with $\mathbf{A} \leftarrow \mathcal{C}(m, n, \mathcal{R})$, $\mathbf{s} \leftarrow \mathcal{R}^n$, and $\mathbf{e} \leftarrow \chi_{m,t}(\mathcal{R})$, and recovers $\mathbf{s}$; the two are equivalent since $\mathbf{Hb} = \mathbf{He}$ and $\mathbf{e} = \mathbf{b} - \mathbf{As}$ [55, Lemma 4.9], and we work with the generator form $(\mathbf{A}, \mathbf{b})$ throughout.[3] We simply say the (m, n, t)-SD problem over $\mathcal{R}$.[4]

We consider SD over a ring/field $\mathcal{R}$. In modern cryptography, the most common choices are $\mathcal{R} \in \{\mathbb{F}_2, \mathbb{F}_q\}$ and the ring $\mathbb{Z}/2^\lambda\mathbb{Z}$ for some $\lambda > 1$. For $\mathcal{R} = \mathbb{Z}/2^\lambda\mathbb{Z}$, Liu et al. [50] proved a reduction from (m, n, t)-SD over $\mathbb{Z}/2^\lambda\mathbb{Z}$ to $(m, n, t/2)$-SD over $\mathbb{F}_2$, so the field cases capture the core security questions we focus on.

When $\mathcal{R} = \mathbb{F}_2$, then it is called the standard SD problem or simply SD. When $\mathcal{R} = \mathbb{F}_q$ for $q \gg 2$, we call it *SD over large fields*, which is a natural generalization of standard SD. In particular, several post-quantum signature proposals instantiate SD over $\mathbb{F}_{2^8}$ [1–3, 38].

Beyond the choice of $\mathcal{R}$, many primitives based on pseudorandom correlation generators (PCGs) assume a *regular* error distribution to significantly improve efficiency [9,10,18,20,22,27,29,30,32,33,41,45,57,60–62,64–70]. This defines the variant known as SD with regular noise (RSD), recalled below. Moreover, these protocols often operate in a low-noise regime $t/m = 1/n^\varepsilon$ for some constant $\varepsilon > 0$, and with a bounded number of samples.

In coding-theoretic terms, m is the code length, n the dimension, $R = n/m$ the rate, and t the error weight, so that solving SD amounts to decoding a length-m, rate-R code with t errors. We retain the parametrization (m, n, t) throughout. Unlike the general decoding problem, the cryptographic regime fixes the code length, namely the number of samples, and keeps the rate R bounded away from zero.

[3] We keep $\mathbf{s}$ and $\mathbf{e}$ as column vectors, so that $\mathbf{A} \in \mathcal{R}^{m\times n}$ carries one row per sample and $\mathbf{A}_\mathcal{I}$ denotes the submatrix of $\mathbf{A}$ formed by the *rows* indexed by $\mathcal{I}$. Information-set selection then acts on rows. Part of the coding-theory literature instead writes vectors as rows and transposes the matrices accordingly. The two conventions are related by transposition, and no quantity in our analysis depends on the choice.

[4] Following the previous work [19, 20, 50], we adapt the definition for our purpose. We note that the definition of SD originally states that the goal is to find $\mathbf{s}$ using oracle access to $\mathcal{O}$, which returns a SD instance. To be specific, we only consider the finite-sample setting in which the number of oracle queries is limited to m. This can be viewed as a decoding problem for a length-m linear code. This constraint is reasonable because most cryptographic primitives built on SD use the parameter regime where the number of oracle queries are bounded.

Problem 2 (SD with regular noise, RSD). Let m, n, t, β be positive integers with $m = t \cdot \beta$ and $\mathcal{R}$ be a ring. Let $\mathcal{C}$ be a probabilistic code generation algorithm such that $\mathcal{C}(m, n, \mathcal{R})$ returns a matrix $\mathbf{A} \in \mathcal{R}^{m \times n}$, with parity-check matrix $\mathbf{H} \in \mathcal{R}^{(m-n) \times m}$ as above ($\mathbf{HA} = \mathbf{0}$). Let $\tau(\mathcal{R}) = \{\tau_{m,t}\}_{m,t \in \mathbb{N}}(\mathcal{R})$ be a family of distributions over $\mathcal{R}^m$. Given m, t, a distribution $\tau_{m,t}$ returns a vector $\mathbf{e} = (\mathbf{e}_1 \| \ldots \| \mathbf{e}_t) \in \mathcal{R}^m$ such that $\mathbf{e}_i \in \mathcal{R}^\beta$ is of Hamming weight $|\mathbf{e}_i| = 1$ for $1 \leq i \leq t$.

The (m, n, t) regular syndrome decoding problem is syndrome decoding with the regular error distribution $\tau_{m,t}$: in the syndrome form one recovers the regular error $\mathbf{e}$ from $\mathbf{H}$ and the syndrome $\mathbf{H} \cdot \mathbf{e}$, equivalently in the generator form one recovers $\mathbf{s}$ from $(\mathbf{A}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e})$ with $\mathbf{A} \leftarrow \mathcal{C}(m, n, \mathcal{R})$, $\mathbf{s} \leftarrow \mathcal{R}^n$, and $\mathbf{e} \leftarrow \tau_{m,t}(\mathcal{R})$. We simply say the (m, n, t)-RSD problem over $\mathcal{R}$.

In this paper, we mainly focus on $\mathcal{R} = \mathbb{F}_q$ with $q \gg 2$. For both problems, we assume throughout that the decomposition $\mathbf{b} = \mathbf{As} + \mathbf{e}$ with $|\mathbf{e}| \leq t$ is unique, so that the secret $\mathbf{s}$ is well defined; for a random code, this holds with high probability whenever t is below the Gilbert–Varshamov distance of the code.

Concrete Analysis for SD over Large Fields. Despite the development of various applications utilizing SD over large fields, cryptanalysis has primarily focused on standard SD. Accurately estimating the security of cryptographic primitives is crucial for evaluating their robustness.

The Blum-Kalai-Wasserman (BKW) algorithm [15] is highly effective for standard SD but becomes less efficient for SD over large fields when $q \gg 2$. This inefficiency occurs because BKW is designed for small-field settings, relying on a subexponential number of samples to achieve optimal performance. As q increases, the required number of samples grows exponentially, making the algorithm impractical.

Boyle et al. [17] initiated a study of concrete security estimates for SD over $\mathbb{F}_q$ in order to instantiate PCG-based protocols. However, Liu et al. [50] demonstrated that this estimation was oversimplified, leading to an overestimation of actual attack costs. Moreover, they demonstrate that to solve SD over $\mathbb{F}_2$ and $\mathbb{Z}/2^\lambda\mathbb{Z}$, advanced algorithms such as [12, 16, 52, 53] yield the most effective results, as expected. While these techniques perform well for SD over $\mathbb{F}_2$, their effectiveness significantly declines as q increases in SD over large fields.

Concrete Analysis for RSD (over Large Fields). For RSD, the work proposed by [21] describes a novel algebraic attack that exploits regular noise distributions on $\mathbb{F}_q$ for any $q \geq 2$. Specifically, the regular noise distribution is used to generate multivariate polynomials, which can reduce the concrete security of RSD. Subsequently, Esser and Santini [37] modified the well-known algorithms for solving SD into specialized algorithms for solving RSD on $\mathbb{F}_2$. They mainly exploit the specific structure derived from the regular distribution of the error vector. These results demonstrate the importance of addressing the regular noise distribution in SD. Unfortunately, similar to a SD case, if $\mathcal{R} = \mathbb{F}_q$ with large q,

algorithms in [37] are not suitable for solving RSD, except for the permutation-based algorithm.[5]

These two results have in common that, despite powerful applications in the case of SD over large fields, more advanced analysis algorithms are still lacking. According to [21,24,37,50,54,58], when the field size is sufficiently large, Prange's Information Set Decoding (ISD) algorithm has a comparable cost to other known algorithms.

1.1 Our Contribution

In this paper, we revisit Prange's ISD algorithm [59], and enhance its performance by reducing the computational cost of the Gaussian elimination. We refer to our modified version as the Reduce and Prange (RP) algorithm. We then extend this approach to develop a variant specifically for solving RSD, which we call regular-RP.

Furthermore, our RP analysis shows that the Scalable Multiparty Garbling scheme from [11] does not achieve its claimed security levels under our estimator. For instance, for the parameter sets $(Q, V, \tau) = (555, 127, 2^{-2})$ and $(785, 214, 2^{-2})$, claimed to provide 80-bit and 128-bit security respectively, the RP cost falls below the target for about 75% and 78% of the noise vectors; see Table 7.

We implement the estimators and use them to search for the best parameter choices found by our threshold-partition estimators.

Implicit Impacts on PCG-like protocols. Recently, Liu et al. [50] provided refined bit-security estimates for SD over $\mathbb{F}_q$, with an emphasis on PCG protocols. They recomputed the parameters from [17] and reported more accurate concrete estimates, together with recommended parameter sets targeting 128-bit security.

Applying RP to these parameter settings, we obtain lower estimates on the PCG-relevant rows. For RSD, by contrast, regular-RP is not below every previously listed estimate: the algebraic estimates remain lower on some of the larger rows. Table 3 summarizes the resulting estimates across various parameters. In particular, our estimates reduce the claimed bit-security of SD over $\mathbb{F}_{2^{128}}$ by up to 6 bits on the comparison rows of Table 3 and up to 7 bits on the recommended parameters of Table 4.

For RSD over $\mathbb{F}_q$ with $\log q = 128$, the results demonstrate that our algorithm gives the lowest listed estimate on many small and medium tested parameter settings, while algebraic estimates remain stronger on some larger rows.[6] For detailed numerical results on solving SD with regular noise, we refer to Section 4.3.

[5] This permutation-based algorithm is identical to Prange's ISD algorithm except for the guessing probability induced by regular noise assumption.

[6] Note that [37] primarily focuses on efficiently solving RSD over the binary field. Their algorithms are not well-suited for RSD over large fields.

We also evaluate RP on the parameter sets recommended in [50] for 128-bit security. For SD over $\mathbb{F}_q$ with $\log q = 128$, RP lowers the estimated security by 3 to 7 bits on the recommended rows; for RSD, regular-RP lowers the estimate by 5 to 7 bits on three of the six recommended rows, while on the remaining three the listed parameters already exceed the target under regular-RP; see Table 4 and Table 6.

Subsequent to This work. Following the first version of this manuscript [47], two works [48, 63] investigated algorithmic improvements for RSD. While our work addresses SD (and RSD) over large fields, these papers focus specifically on RSD. Technically, [48,63] refine algebraic attacks building on the polynomial-system framework [21], whereas we analyze attacks stemming from information-set decoding [59]. Our comparisons exhibit a crossover: our ISD-based estimates are lower for smaller n in our benchmarks, while the algebraic methods become preferable as n grows. For further details, we refer to these papers [48, 63].

1.2 Technical Idea

We first revisit Prange's Information Set Decoding (ISD) algorithm to describe our approach. To this end, we introduce some notations: for any vector $\mathbf{x} \in \mathbb{F}_q^m$, we denote $|\mathbf{x}|$ as its Hamming weight. Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{F}_q^{m \times n} \times \mathbb{F}_q^m$ be an SD instance and $\mathcal{I} \subset [m]$ an index set of size n. We write $\mathbf{A}_\mathcal{I}$ and $\mathbf{b}_\mathcal{I}$ for the submatrix and subvector obtained by collecting the i-th row of $\mathbf{A}$ and the i-th coordinate of $\mathbf{b}$, respectively, for all $i \in \mathcal{I}$.

Using this notation, we present Prange's Information Set Decoding algorithm in Algorithm 1.

Algorithm 1 Prange's ISD

Require: m, n, t: Parameters.
(m, n, t)-SD over $\mathbb{F}_q$ instances $(\mathbf{A}, \mathbf{b}) \in \mathbb{F}_q^{m \times n} \times \mathbb{F}_q^m$.
Ensure: A secret $\mathbf{s}$ such that $|\mathbf{b} - \mathbf{A} \cdot \mathbf{s}| = t$.
1: **repeat**
2: Choose a random index subset $\mathcal{I} \subset [m]$ of size n
3: Generate the submatrix/subvector $(\mathbf{A}_\mathcal{I}, \mathbf{b}_\mathcal{I})$ by collecting the i-th row of $\mathbf{A}$ and the i-th coordinate of $\mathbf{b}$ for $i \in \mathcal{I}$
4: **until** $\mathbf{A}_\mathcal{I}$ is invertible and $|\mathbf{b} - \mathbf{A} \cdot \mathbf{s}| = t$ where $\mathbf{s} = \mathbf{A}_\mathcal{I}^{-1} \cdot \mathbf{b}_\mathcal{I}$
5: **return** $(\mathbf{s}, \mathcal{I})$

Prange's algorithm terminates in expected time $\mathcal{G}/\mathcal{P}$, where $\mathcal{P}$ is the probability of collecting n error-free instances (i.e., $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle$) and $\mathcal{G}$ represents the time complexity of the Gaussian elimination step. Although the Gaussian elimination operation has a time complexity of only $O(n^\omega)$ for the linear algebra constant ω, where it is typically disregarded in the context of asymptotic analysis, its concrete time complexity is of significant consequence in estimating the

bit-security. For instance, we consider the $(1024, 652, 57)$-SD over $\mathbb{F}_q$. According to [50], the bit-size of the total cost associated with this problem is reported to be 111, with the Gaussian elimination accounting for a bit-size of 26 when we take $O(n^\omega) = n^{2.8}$. This highlights that the Gaussian elimination contributes to approximately one-fifth of the total complexity exponent involved in solving the problem. Consequently, to improve Prange's ISD, it is essential to reduce the cost of the Gaussian algorithm $\mathcal{G}$ since the probability that a single uniformly drawn information set is error-free cannot be changed; an improvement can still change how many such draws are effectively needed, and at what cost per draw.

Canteaut and Chabaud (CC) [23] and Bernstein, Lange and Peters (BLP) [13] proposed algorithms for solving SD through reducing the cost of Gaussian elimination via so-called *reusing existing pivots* and *forcing more existing pivots*. Both are techniques developed on top of Stern's algorithm. In [23] a single coordinate of the information set is exchanged per iteration, while [13] treats the number c of exchanged coordinates as a parameter. In this work we reduce the same factor $\mathcal{G}$ by a different mechanism, the Reduce and Prange (RP) method, which consists of two key steps:

1. Guess N zero positions in $\mathbf{e}$ and transform from (m, n, t)-SD instances to $(m - N, n - N, t)$-SD instances.
2. Solve the $(m - N, n - N, t)$-SD problem using Prange's ISD algorithm.

Since the secret dimension is reduced to $n - N$, the Gaussian elimination now operates on a smaller matrix of size $n - N$, which reduces its computational cost. This reduction can be iterated; we call the number of reduction steps the *level* of the RP algorithm. For example, when $k = 2$, then we first guess N_1 positions, to generate $(m - N_1, n - N_1, t)$-SD instances. Then, we additionally guess N_2 positions to get $(m - N_1 - N_2, n - N_1 - N_2, t)$-SD instances. As the final step, we solve the SD problem using Prange's ISD algorithm.

Advantages of Reduce and Prange's ISD. The RP algorithm lowers the per-iteration cost compared to Prange's original ISD method. While the original algorithm requires n^ω cost for each iteration due to performing Gaussian elimination on an n-dimensional matrix, the Reduce and Prange's algorithm strategically divides zero indices into sets, $\mathcal{I}_1$ and $\mathcal{I}_2$. If an error occurs in $\mathcal{I}_2$, only this subset is replaced rather than all n indices, reducing the iteration cost to $(n - N_1)^\omega$. Similarly, when $\mathcal{I}_1$ is incorrectly guessed, the Reduce and Prange's algorithm only repeats the Step 1 with a complexity $\mathcal{C}_{m,N_1}$. This approach lowers the concrete cost of the inner iterations, and the final target is still a clean information set of size n. Under a bounded-pass implementation, the global factor introduced below accounts for the pass-success probability separately.

Comparison to previous works [13,23]. We provide a conceptual comparison between previous works and our algorithm in terms of $\mathcal{G}$. The key idea behind the algorithms in [13, 23] involves reusing the pivots from the previous iteration.

An information-set decoder has two parts. One part maintains the information set and produces the next candidate. The other part searches for the error inside that set. Prange uses the trivial search, while Lee–Brickell, Stern, Dumer

and MMT use more elaborate ones. The algorithms of [13, 23] act on the first part, and they were developed on top of Stern's search. Indeed, [23, §I] describes its algorithm as combining "a heuristic proposed by Stern" with an iterative method stemming from linear programming. Our reduction also acts on the first part, and it does not depend on which search is used afterward.

In a nutshell, suppose that $(\mathbf{A}, \mathbf{b})$ with $\mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e}$ is given. The BLP/CC algorithms [13, 23] transform this into a pair $(\mathbf{A}', \mathbf{b}') \in \mathbb{F}_q^{m-n\times(n+1)}$ where $\mathbf{b}' = \mathbf{A}' \cdot \mathbf{e}' + \mathbf{e}''$ by reducing $\mathbf{s}$. If the partial noise vector $\mathbf{e}' \in \mathbb{F}_q^n$ is zero, then $\mathbf{b}'$ has a Hamming weight of t, leading to the algorithm's termination. If not, some components of $\mathbf{e}'$ will be zero, allowing the reuse of the $n - c$ zero coordinates in $\mathbf{e}'$ and collection of c new samples to obtain n error-free instances.

When guessing that $n - c$ coordinates of $\mathbf{e}'$ are zero, the corresponding columns of $\mathbf{A}'$ are not needed. As a result, the pair $(\mathbf{A}', \mathbf{b}')$ becomes $(\mathbf{A}'', \mathbf{b}') \in \mathbb{F}_q^{m-n\times(c+1)}$. Since Gaussian elimination is required over a smaller dimension $c < n$, the overall complexity is reduced compared to Prange's algorithm. This optimization can be seen as collecting error-free instances from two separate lists, one of size $n - c$ and the other of size c.

Our RP algorithm collects the final information set through several disjoint blocks $(N_1, N_2, \ldots, N_{k+1})$ with $\sum_i N_i = n$. After guessing N_1 error-free positions, RP converts the original (m, n, t)-SD instance into an (m_1, n_1, t)-SD instance with $m_1 = m - N_1$ and $n_1 = n - N_1$ by eliminating the corresponding secret coordinates. Repeating this reduction lowers the linear-algebra dimension of the inner search. The main optimization problem is then to choose the block sizes appropriately. For the detailed algorithmic description, we refer to Section 2.

The two maintenance strategies are nevertheless structurally different: RP is not simply a multi-level version of [13,23]. In [23], a neighboring information set is obtained by exchanging a single coordinate at each iteration [23, Def. 2], and [13] generalizes this move to an exchange of c coordinates. In both algorithms, the maintained information set has size n throughout, and the saving comes from updating a matrix that is already in systematic form rather than re-running a full elimination from scratch. By contrast, RP fixes a guessed prefix and uses it to form a genuinely reduced decoding instance. The prefix is kept only for a bounded inner search; if this search leads to a candidate passing the residual check, the candidate is accepted, and otherwise the prefix is discarded once its budget is exhausted. Thus RP reduces the dimension of the active solve, from n to $n - \mu_k$ after a prefix of length μ_k, rather than performing a pivot walk on a fixed-size information set. Table 1 summarizes these differences. For information-set decoding over $\mathbb{F}_q$ in general, which underlies these comparisons, we follow [58].

Both strategies also lose something on the success side. Reusing coordinates makes successive candidates dependent, and [13, §4] reports that at matched Stern parameters the single-coordinate exchange of [23] needs $9.85 \cdot 10^{11}$ iterations where independent sampling needs $5.78 \cdot 10^{11}$. RP instead pays the bounded-budget factor $\gamma_{\mathrm{RP}} = G_1 \leq 1 - e^{-1}$, which costs 1.4 to 2.7 bits on the parameters

Table 1. Information-set maintenance strategies. Here m is the code length, n the dimension, c the number of coordinates exchanged per iteration, and $\mu_k = \sum_{j \le k} N_j$ the number of coordinates fixed by the RP prefix.

	Prange	**CC [23] / BLP [13]**[†]	**RP (this work)**
Reused state	nothing	pivot work on an information set of fixed size	a solved, dimension-reduced instance
Working dimension	n	n throughout: $n-c$ retained, c fresh	decreases, $n \to n-\mu_k$
How long the retained part is kept	—	indefinitely, as the set keeps walking	for a bounded number r_j of attempts, then discarded
Successive candidates	independent	dependent, through a Markov chain on the number of errors in the set [23, Prop. 4]	independent given the prefix
Expected number of iterations	geometric, $1/\mathcal{P}$	absorbing chain, via its fundamental matrix [23, Thm. 1]	the recursion G_j of Section 2.2
Linear algebra cost per candidate	full elimination, n^ω	one pivot update, $\frac{1}{2}n(m-n)$ [23, §III-B]; linear in c	full elimination at the reduced dimension, $(n-\mu_k)^\omega$

[†]The exchange is fixed to a single coordinate in [23]. In [13] the number c of exchanged coordinates becomes a parameter, and intermediate values of c are found to be the best choice.

of Table 3. The two figures come from different settings, binary Stern parameters in the first case and $\mathbb{F}_{2^{128}}$ with a Prange search in the second, so we do not read them as a direct comparison. The two strategies do act on the same term. For a Prange search over a large field the per-iteration cost *is* the Gaussian elimination, and this is the regime our reduction targets.

2 Reduce and Prange Technique

This section introduces a new approach for solving SD over large fields, simply called the Reduce and Prange approach. Throughout this section and the cost analysis, we solve SD in its equivalent generator (primal) form: recovering the secret $\mathbf{s}$ from $(\mathbf{A}, \mathbf{b} = \mathbf{As}+\mathbf{e})$ is equivalent to recovering the error $\mathbf{e} = \mathbf{b}-\mathbf{As}$, and the central reduction step is most naturally described as a reduction of the code dimension. Unless stated otherwise, all concrete costs in this section and in the numerical tables count arithmetic operations over $\mathbb{F}_q$; reported bit-security values are $\log_2$ of this operation count, not bit-operation estimates. For the algorithm

description, we suppose that an SD instance of the form $(\mathbf{A}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e})$ is given, where $\mathbf{A} \in \mathbb{F}_q^{m \times n}$, $\mathbf{s} \in \mathbb{F}_q^n$, and $\mathbf{e} \leftarrow \chi_{m,t}$.

The probability of obtaining n error-free polynomials is given by $P = \frac{\binom{m-n}{t}}{\binom{m}{t}}$. Thus, the expected number of iterations required for Prange's ISD (Algorithm 1) to recover the secret vector is $\frac{1}{P} = \frac{\binom{m}{t}}{\binom{m-n}{t}}$. Hence, the time complexity of Prange's ISD algorithm is given by

$$O\left(\frac{\binom{m}{t}}{\binom{m-n}{t}} \cdot n^{\omega}\right)$$

where n^{ω} represents the Gaussian elimination cost, with the linear algebra constant $2 \leq \omega \leq 3$.

2.1 Reduce and Prange's ISD

The core idea of our approach is to first guess N_1 error-free samples to reduce the dimension of the SD problem. We then apply Prange's algorithm to the SD problem of dimension $n - N_1$.

Suppose that $(\mathbf{A}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e})$ is an SD instance, where $\mathbf{A} \in \mathbb{F}_q^{m \times n}$, $\mathbf{s} \in \mathbb{F}_q^n$, and $\mathbf{e} \leftarrow \chi_{m,t}$. We let $\mathbf{a}_i$ denote the i-th row of $\mathbf{A}$, and b_i and e_i the i-th entries of $\mathbf{b}$ and $\mathbf{e}$, respectively.

Step 1. Choose an index set $\mathcal{I}_1 \subset [m]$ of the size $N_1 (< n)$, which corresponds to the zero positions in $\mathbf{e}$. For simplicity, assume that the last N_1 positions in $\mathbf{e}$ are zero. In general, the zero positions can be shifted to the end through permutation. Let $\mathbf{A}_{\mathsf{top}}$ and $\mathbf{A}_{\mathsf{bot}}$ denote the submatrices of $\mathbf{A}$ consisting of the first $m - N_1$ rows and the last N_1 rows, respectively. We define $\mathbf{b}_{\mathsf{top}}$, $\mathbf{b}_{\mathsf{bot}}$, $\mathbf{e}_{\mathsf{top}}$, and $\mathbf{e}_{\mathsf{bot}}$ analogously. The above assumption then says that $\mathbf{e}_{\mathsf{bot}} = \mathbf{0}$. Using the notation, we express the system as

$$\begin{pmatrix} \mathbf{b}_{\mathsf{top}} \\ \mathbf{b}_{\mathsf{bot}} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{\mathsf{top}} \\ \mathbf{A}_{\mathsf{bot}} \end{pmatrix} \cdot \mathbf{s} + \begin{pmatrix} \mathbf{e}_{\mathsf{top}} \\ \mathbf{0} \end{pmatrix}.$$

We then apply the Gaussian elimination algorithm to obtain SD of less dimension. For this purpose, we parse $\mathbf{s}$ to $(\mathbf{s}^1 \| \mathbf{s}^2)$ such that $\mathbf{s}^1 \in \mathbb{F}_q^{n-N_1}$ and $\mathbf{s}^2 \in \mathbb{F}_q^{N_1}$. Similarly to the $\mathbf{s}$, we split $\mathbf{A}_{\mathsf{top}}$ (resp. $\mathbf{A}_{\mathsf{bot}}$) into $(\mathbf{A}_{\mathsf{top}}^{\mathsf{left}} \| \mathbf{A}_{\mathsf{top}}^{\mathsf{right}})$ (resp. $(\mathbf{A}_{\mathsf{bot}}^{\mathsf{left}} \| \mathbf{A}_{\mathsf{bot}}^{\mathsf{right}})$) as well. Assuming that the matrix $\mathbf{A}_{\mathsf{bot}}^{\mathsf{right}}$ is invertible over $\mathbb{F}_q$ (otherwise we permute columns until an invertible block is obtained), one can get

the following SD samples:

$$\begin{aligned}\mathbf{b}' :&= \left(\mathbf{I}_{m-N_1} \| -\mathbf{A}_{\mathsf{top}}^{\mathsf{right}} \cdot \left(\mathbf{A}_{\mathsf{bot}}^{\mathsf{right}}\right)^{-1}\right) \cdot \begin{pmatrix}\mathbf{b}_{\mathsf{top}} \\ \mathbf{b}_{\mathsf{bot}}\end{pmatrix} \\ &= \left(\mathbf{I}_{m-N_1} \| -\mathbf{A}_{\mathsf{top}}^{\mathsf{right}} \cdot \left(\mathbf{A}_{\mathsf{bot}}^{\mathsf{right}}\right)^{-1}\right) \\ &\quad \cdot \left(\begin{pmatrix}\mathbf{A}_{\mathsf{top}} \\ \mathbf{A}_{\mathsf{bot}}\end{pmatrix} \cdot \mathbf{s} + \begin{pmatrix}\mathbf{e}_{\mathsf{top}} \\ \mathbf{0}\end{pmatrix}\right) \\ &= \underbrace{\left(\mathbf{A}_{\mathsf{top}}^{\mathsf{left}} - \mathbf{A}_{\mathsf{top}}^{\mathsf{right}} \cdot \left(\mathbf{A}_{\mathsf{bot}}^{\mathsf{right}}\right)^{-1} \cdot \mathbf{A}_{\mathsf{bot}}^{\mathsf{left}}\right)}_{=:\mathbf{A}'} \cdot \mathbf{s}^1 + \mathbf{e}_{\mathsf{top}}.\end{aligned}$$

We denote this conversion by $(\mathbf{A}', \mathbf{b}') \leftarrow \mathsf{Conv}((\mathbf{A}, \mathbf{b}), \mathcal{I}_1)$ and retain the original row labels on the remaining coordinates; this notation will be used repeatedly.

The conversion Conv is independent of the decoder applied afterward: it maps an (m, n, t)-SD instance to a smaller $(m - N_1, n - N_1, t)$-SD instance using only Gaussian elimination. Thus Step 2 may use any information-set-decoding variant, such as Lee–Brickell, Stern, Dumer, or MMT, in place of Prange. We instantiate Step 2 with Prange and quantify the resulting speedup, which comes from the reduced per-iteration Gaussian-elimination cost. A full optimization of hybrids obtained by prepending RP to other ISD decoders is beyond the scope of the estimates reported here.

Complexity. In each guessing, this algorithm computes the matrix $\mathbf{A}'$, which consists of one matrix inversion and two matrix multiplications so it takes in time $\mathcal{C}_{m,N_1} := O(N_1^\omega + N_1^2 \cdot (m - N_1) + N_1 \cdot (m - N_1) \cdot (n - N_1))$, where ω is the linear algebra constant. The probability that attackers can correctly guess N_1 positions is $\mathcal{P}_{m,t,N_1} := \frac{\binom{m-t}{N_1}}{\binom{m}{N_1}}$, which directly implies that the time complexity to get SD samples of dimension $n - N_1$ is $\frac{1}{\mathcal{P}_{m,t,N_1}} \cdot \mathcal{C}_{m,N_1}$.

Step 2. For ease of representation, we assume that $(m - N_1, n - N_1, t)$-SD instances $(\mathbf{A}', \mathbf{b}')$ such that $\mathbf{b}' = \mathbf{A}' \cdot \mathbf{s}^1 + \mathbf{e}_{\mathsf{top}}$ are given, where $(\mathbf{A}', \mathbf{b}') \leftarrow \mathsf{Conv}((\mathbf{A}, \mathbf{b}), \mathcal{I}_1)$, $\mathbf{s}^1 \in \mathbb{F}_q^{n-N_1}$ and $\mathbf{e}_{\mathsf{top}} \leftarrow \chi_{m-N_1,t}$. In addition, the index of $\mathbf{e}_{\mathsf{top}}$ is given as $[m] \setminus \mathcal{I}_1$.

The second step just runs Prange's algorithm. That is, one first collects an index set $\mathcal{I}_2 \subset [m] \setminus \mathcal{I}_1$ of size $n - N_1$ from given $m - N_1$ SD instances. Then, one can recover the secret $\mathbf{s}^1$ via the Gaussian elimination algorithm. After recovering $\mathbf{s}^1$, one can recover the vector $\mathbf{s}^2$ from the identity $\mathbf{b}_{\mathsf{bot}} = \mathbf{A}_{\mathsf{bot}} \cdot (\mathbf{s}^1 \| \mathbf{s}^2)$.

Correctness and Verification. The simple two-level case follows the same verification principle as the general correctness lemma below. If both $\mathcal{I}_1$ and $\mathcal{I}_2$ are noise-free and all pivot blocks used by the conversion and final solve are invertible, then the vector reconstructed from $\mathcal{I}_1 \cup \mathcal{I}_2$ is the true secret $\mathbf{s}$ and satisfies $|\mathbf{b} - \mathbf{A}\mathbf{s}| = t$. If one of the selected sets contains noise, any accepted output must pass the full residual-weight check as a false acceptance;

Algorithm 2 Simple Reduce and Prange's ISD

Require: (m, n, t)-SD over $\mathbb{F}_q$ instance $\mathbf{A} \in \mathbb{F}_q^{m \times n}$ and $\mathbf{b} \in \mathbb{F}_q^m$, and an index size $\{N_1, N_2 := n - N_1\}$.
Ensure: A secret vector $\mathbf{s}$ such that $|\mathbf{b} - \mathbf{A} \cdot \mathbf{s}| = t$.

Compute $P_1 = \frac{\binom{m-t}{N_1}}{\binom{m}{N_1}}$ and $P_2 = \frac{\binom{m-N_1-t}{N_2}}{\binom{m-N_1}{N_2}}$, and set repetition budgets r_1, r_2 under the chosen convention
for $i = 1$ up to r_1 **do**
  Choose a random index set $\mathcal{I}_1$ of size N_1
  Compute $(\mathbf{A}^1, \mathbf{b}^1) \leftarrow \mathsf{Conv}((\mathbf{A}, \mathbf{b}), \mathcal{I}_1)$
  for $j = 1$ up to r_2 **do**
    Choose a random index set $\mathcal{I}_2 \subset [m] \setminus \mathcal{I}_1$ of size N_2
    $(\mathbf{A}^1_{\mathcal{I}_2}, \mathbf{b}^1_{\mathcal{I}_2}) \leftarrow [\mathbf{A}^1, \mathbf{b}^1]_{\mathcal{I}_2}$
    if $\mathbf{A}^1_{\mathcal{I}_2}$ is invertible **then**
      Set $\mathbf{s}^1 = (\mathbf{A}^1_{\mathcal{I}_2})^{-1} \cdot \mathbf{b}^1_{\mathcal{I}_2}$.
      if $|\mathbf{b}^1 - \mathbf{A}^1 \cdot \mathbf{s}^1| = t$ **then**
        Lift $\mathbf{s}^1$ through the stored conversion for $\mathcal{I}_1$ to obtain the candidate full secret $\mathbf{s}$.
        if $|\mathbf{b} - \mathbf{A} \cdot \mathbf{s}| = t$ **then**
          Set $\mathcal{I} \leftarrow \mathcal{I}_1 \cup \mathcal{I}_2$.
          return $(\mathbf{s}, \mathcal{I})$
        end if
      end if
    end if
  end for
end for
return failure

over random large-field instances this occurs only with the negligible probability bounded in Lemma 2.1. The reduced residual check in Algorithm 2 is charged in the final-body term. The full residual check after lifting is executed only after a reduced-instance return and can be included explicitly as an additional linear verification cost without changing the displayed exponents. A union bound over all checks in a bounded pass is included in the general lemma.

Time Complexity. The complexity of Prange's ISD for solving SD with parameters $(m - N_1, n - N_1, t)$ can be computed as follows.

$$\begin{aligned}\mathcal{S}_{m-N_1,n-N_1,t} &\stackrel{def}{=} \frac{\binom{m-N_1}{n-N_1}}{\binom{m-N_1-t}{n-N_1}} \\ &\quad \cdot ((n-N_1)^\omega + (m-N_1)\cdot(n-N_1)) \\ &= \frac{1}{\mathcal{P}_{m-N_1,t,n-N_1}} \\ &\quad \cdot ((n-N_1)^\omega + (m-N_1)\cdot(n-N_1)),\end{aligned}$$

where ω denotes the linear-algebra exponent. The last term, $(m-N_1)\cdot(n-N_1)$, represents the verification cost of verifying whether $|\mathbf{b}^1 - \mathbf{A}^1 \cdot \mathbf{s}^1| = t$. Although

this verification cost is typically negligible in Prange's original ISD, we include it here for an accurate computation. We emphasize that this verification cost can be large, depending on the size of N_1.

For a bounded-pass implementation, the bounded repetitions contribute one global pass-success factor. If the two stages are repeated with budgets r_1, r_2, this factor is

$$\begin{aligned} \gamma_{\mathrm{RP}}^{(1)} &= G_1(r_1, r_2), \\ G_2 &= 1 - (1 - \mathcal{P}_{m-N_1,t,n-N_1})^{r_2}, \\ G_1 &= 1 - (1 - \mathcal{P}_{m,t,N_1} G_2)^{r_1}. \end{aligned}$$

Thus the success correction is not a product of independent $(1-1/e)$ factors. In the numerical estimator we set $r_j = 1/\mathcal{P}(j)$ under the continuous fixed-budget convention and evaluate this global factor through the above recursion; for a one-stage Prange pass with success probability p and budget r, the fixed-budget value is $1-(1-p)^r$, which becomes $1-(1-p)^{1/p}$ under the continuous choice $r = 1/p$ and is approximated by $1-1/e$ in the standard exponential convention. This is an analytic fixed-budget estimator, not a claim that the implementation executes a fractional number of trials; a literal bounded-pass implementation instead chooses integral budgets and evaluates the same recursion with those integer values.

Putting them together, the SD problem is solved within

$$\frac{1}{\mathcal{P}_{m,t,N_1}} \cdot (\mathcal{C}_{m,N_1} + \mathcal{S}_{m-N_1,n-N_1,t})$$

with a bounded-pass success probability $\gamma_{\mathrm{RP}}^{(1)}$. Hence, under the continuous fixed-budget convention, its constant-success cost estimate is represented by

$$\frac{1}{\mathcal{P}_{m,t,N_1} \cdot \gamma_{\mathrm{RP}}^{(1)}} \cdot (\mathcal{C}_{m,N_1} + \mathcal{S}_{m-N_1,n-N_1,t}) \,.$$

Impact of the Reduce and Prange's ISD. We present the advantages of the Reduce and Prange's ISD algorithm. The key improvement lies in reducing the computational cost per iteration. In Prange's original ISD algorithm, each iteration requires Gaussian elimination on an n-dimensional square matrix, incurring a cost of n^ω.

The RP algorithm significantly reduces this per-iteration cost. The algorithm partitions the zero indices into two sets, $\mathcal{I}_1$ and $\mathcal{I}_2$, such that $\#\mathcal{I}_1 + \#\mathcal{I}_2 = n$ with $\#\mathcal{I}_1 = N_1$. When an incorrect zero index is found in $\mathcal{I}_2$, the RP algorithm only needs to replace $\mathcal{I}_2$ (Step 2), rather than all n zero indices. This optimization reduces the per-iteration cost to $(n-N_1)^\omega + (m-N_1)\cdot(n-N_1)$. Similarly, if $\mathcal{I}_1$ is incorrectly guessed, the algorithm repeats Step 1 with complexity $\mathcal{C}_{m,N_1}$.

While the RP algorithm still searches for the same final clean information set as Prange's ISD, its bounded-pass success probability is the global factor described above; its efficiency gain comes from reducing the cost of the inner iterations.

Indeed, in this fixed-budget cost model, the dominant RP term is

$$\frac{\mathcal{S}_{m-N_1,n-N_1,t}}{\mathcal{P}_{m,t,N_1}\cdot\gamma_{\mathrm{RP}}^{(1)}} = \frac{\binom{m}{n}}{\binom{m-t}{n}}\cdot\frac{(n-N_1)^\omega+(m-N_1)\cdot(n-N_1)}{\gamma_{\mathrm{RP}}^{(1)}}.$$

Under the same fixed-budget convention, plain Prange with budget $\mathsf{T} = \binom{m}{n}/\binom{m-t}{n}$ has success factor

$$\gamma_{\mathrm{P}} = 1-(1-1/\mathsf{T})^{\mathsf{T}} \approx 1-1/e$$

and cost estimate

$$T_{\mathrm{P}}^{\mathrm{fb}} = \mathsf{T}\cdot(n^\omega+mn)/\gamma_{\mathrm{P}}.$$

Compared to this baseline, our Reduce and Prange (RP) algorithm shares the fractional part but can feature, after optimizing N_1, a smaller multiplicative factor $\frac{(n-N_1)^\omega+(m-N_1)\cdot(n-N_1)}{\gamma_{\mathrm{RP}}^{(1)}}$ than $(n^\omega+mn)/\gamma_{\mathrm{P}}$. As a consequence, this reduction in the multiplier can lower the resulting cost estimate.

2.2 Iterative Reduce and Prange Technique

'Step 1' in Section 2 can be viewed as a self-reduction technique for SD. Naturally, this approach can be iterated to progressively reduce the dimension of SD samples. Thereafter, one can solve the reduced SD problems by Step 2. With respect to the number of reductions, we call this algorithm the level-k Reduce and Prange algorithm.

For instance, consider $k = 2$. The level-2 Reduce and Prange algorithm proceeds as follows: First, select three index sets $\{\mathcal{I}_1,\mathcal{I}_2,\mathcal{I}_3\}\subset[m]$ of size $\#\mathcal{I}_i = N_i$ and $\sum_{i=1}^3 N_i = n$. Then, one can get SD samples $(\mathbf{A}^{(2)},\mathbf{b}^{(2)})$ of dimension N_3 from $(\mathbf{A}^{(2)},\mathbf{b}^{(2)}) \leftarrow \mathsf{Conv}((\mathbf{A}^{(1)},\mathbf{b}^{(1)}),\mathcal{I}_2)$ and $(\mathbf{A}^{(1)},\mathbf{b}^{(1)}) \leftarrow \mathsf{Conv}((\mathbf{A},\mathbf{b}),\mathcal{I}_1)$. Then the new instance can be solved via the Prange algorithm on the less dimension N_3. The interesting fact is that since this auxiliary step proceeds after the first step occurs, the complexity of the level-2 Reduce and Prange algorithm is given as

$$\frac{1}{\mathcal{P}_{m,t,N_1}}\cdot\left(\mathcal{C}_{m,N_1} + \frac{1}{\mathcal{P}_{m-N_1,t,N_2}}\left(\mathcal{C}_{m-N_1,N_2}+\mathcal{S}_{m-N_1-N_2,N_3,t}\right)\right).$$

However, as previously mentioned, we must also account for the success probability of the nested guessing of $\mathcal{I}_1,\mathcal{I}_2$, and $\mathcal{I}_3$. As in the general case this contributes a single global factor $\gamma_{\mathrm{RP}}^{(2)}$, since the inner searches reuse the correctly

guessed outer sets. Consequently, the continuous fixed-budget cost estimate of the level-2 Reduce and Prange algorithm is represented as

$$\frac{1}{\gamma_{\mathrm{RP}}^{(2)}} \cdot \frac{1}{\mathcal{P}_{m,t,N_1}} \cdot \Bigg(\mathcal{C}_{m,N_1} + \frac{1}{\mathcal{P}_{m-N_1,t,N_2}} \left(\mathcal{C}_{m-N_1,N_2} + \mathcal{S}_{m-N_1-N_2,N_3,t} \right) \Bigg).$$

In general, the continuous fixed-budget cost estimate for level-k Reduce and Prange algorithm can be computed as follows. Given $\{N_i\}_{i=1}^{k+1}$, we sample disjoint index sets $(\mathcal{I}_1, \ldots, \mathcal{I}_{k+1})$ sequentially without replacement, i.e., $\mathcal{I}_j \subseteq [m] \setminus \bigcup_{\ell<j} \mathcal{I}_\ell$ and $|\mathcal{I}_j| = N_j$ for all j, with $\sum_{j=1}^{k+1} N_j = n$.

To simplify notation, we introduce the following terms $\mathcal{P}(j), \mathcal{C}(j), \mathcal{S}$ as follows:

$$\begin{aligned}
\widehat{\mathcal{C}}_{M,N} &:= N^\omega + N^2(M-N) + N(M-N)^2, \\
\mathcal{P}(j) &= \mathcal{P}_{m-\mu_{j-1},t,N_j} \quad (1 \le j \le k), \\
\mathcal{C}(j) &= \widehat{\mathcal{C}}_{m-\mu_{j-1},N_j} \quad (1 \le j \le k), \\
\mathcal{P}(k+1) &= \mathcal{P}_{m-\mu_k,t,n-\mu_k}, \\
\mathcal{V}_{k+1} &= (n-\mu_k)^\omega + (n-\mu_k) \cdot (m-\mu_k), \\
\mathcal{S} &= \frac{1}{\mathcal{P}(k+1)} \cdot \mathcal{V}_{k+1},
\end{aligned}$$

where $\mu_j = \sum_{i=1}^{j} N_i$ for every $j \ge 1$. The exact conversion expression with current column dimension D is $N^\omega + N^2(M-N) + N(M-N)(D-N)$; the numerical tables use the conservative body-cost convention $\widehat{\mathcal{C}}_{M,N}$ above, matching the public estimator output. Then, using this notation, the cost of one bounded pass under the standard continuous convention is

$$\mathcal{T} = \frac{1}{\mathcal{P}(1)} \Bigg(\mathcal{C}(1) + \frac{1}{\mathcal{P}(2)} \Bigg(\mathcal{C}(2) + \cdots + \frac{1}{\mathcal{P}(k)} \left(\mathcal{C}(k) + \mathcal{S} \right) \Bigg) \Bigg)$$

To simplify it, we also define $\mathcal{Q}(j)$ as $\mathcal{Q}(j) = \prod_{i=1}^{j} \mathcal{P}(i)$. We denote this bounded-pass cost, viewed as a function of the partition $\mathbf{N} = (N_1, \ldots, N_{k+1})$, by

$$B(\mathbf{N}) = \sum_{j=1}^{k} \frac{\mathcal{C}(j)}{\mathcal{Q}(j)} + \frac{\mathcal{S}}{\mathcal{Q}(k)}.$$

Its success probability relies on the assumption that the index sequence $\{\mathcal{I}_i\}_{i=1}^{k+1}$ is correctly guessed. The bounded searches contribute one global pass-success factor, which we denote by

$$\gamma_{\mathrm{RP}} = G_1(r_1, \ldots, r_{k+1}).$$

For fixed repetition budgets r_j, the quantities G_j are defined recursively by

$$G_{k+1} = 1 - (1 - \mathcal{P}(k+1))^{r_{k+1}},$$
$$G_j = 1 - (1 - \mathcal{P}(j)G_{j+1})^{r_j} \quad (1 \le j \le k).$$

This recursion captures that the inner searches reuse a fixed clean prefix, so the stage success probabilities are not multiplied as independent factors. In the revised numerical estimator we use the continuous fixed-budget convention $r_j = 1/\mathcal{P}(j)$ and evaluate γ_{RP} by this recursion. This convention is an analytic estimator, not a literal integer-loop schedule. A literal bounded-pass attack is obtained by fixing integral budgets, for example $r_j = \lceil 1/\mathcal{P}(j) \rceil$, and evaluating the same recursion with those integer values. The corresponding bounded-pass body cost is

$$B_r(\mathbf{N}) = \sum_{j=1}^{k} \mathcal{C}(j) \prod_{i=1}^{j} r_i + \mathcal{V}_{k+1} \prod_{i=1}^{k+1} r_i.$$

The Monte Carlo validation below uses integer budgets to check this recursion for the pass-success event; the table estimates use the continuous convention $B(\mathbf{N})/\gamma_{\mathrm{RP}}(\mathbf{N})$ so that the body cost and success correction follow the same fixed-budget interpolation. As usual for large-field ISD estimates, the formulas condition on all pivot blocks used in the conversions and the final information matrix being nonsingular, or equivalently absorb the $O((k+1)/q)$ rank-failure probability into negligible terms. Accordingly, for a candidate partition $\mathbf{N} = (N_1, \ldots, N_{k+1})$, the continuous fixed-budget constant-success cost estimate of the k-level RP algorithm is

$$T_{\mathrm{RP}}(\mathbf{N}) = \frac{B(\mathbf{N})}{\gamma_{\mathrm{RP}}(\mathbf{N})} = \frac{1}{\gamma_{\mathrm{RP}}(\mathbf{N})} \left(\sum_{j=1}^{k} \frac{\mathcal{C}(j)}{\mathcal{Q}(j)} + \frac{\mathcal{S}}{\mathcal{Q}(k)} \right).$$

Thus γ_{RP} is applied once to the whole bounded-pass cost, rather than being inserted separately into the prefix denominators. The threshold search below generates candidate partitions using the unnormalized terms $\mathcal{C}(j)/\mathcal{Q}(j)$, and each candidate is then evaluated with its own value of γ_{RP} from the recursion.

Given a sequence $\{N_i\}_{i=1}^{k+1}$ as input, the RP algorithm proceeds as Algorithm 3. An optimized construction of $\{N_i\}_{i=1}^{k+1}$ to minimize the algorithm's complexity will be discussed later.

Correctness and Verification. We now state correctness formally; verification is performed only by the residual-weight checks appearing in the algorithm.

Lemma 2.1 (Correctness) *Over $\mathbb{F}_q$, if the guessed index sets $\mathcal{I}_1, \ldots, \mathcal{I}_{k+1}$ are all noise-free and all conversion pivot blocks and the final information matrix are invertible, the level-k RP algorithm outputs the secret $\mathbf{s}$. If at least one guessed index set contains noise, then any output must pass a residual-weight check for an incorrect candidate; under the random-instance model, each such false acceptance occurs with probability at most $\binom{m-n}{t}(q-1)^t/q^{m-n}$. Hence, over one*

Algorithm 3 Reduce and Prange's ISD

Require: (m, n, t)-SD over $\mathbb{F}_q$ instance represented by the current pair $(\mathbf{A}, \mathbf{b})$,
conversion history for lifting candidates back to the original secret space,
block sizes $\{N_i\}_{i=1}^{k+1}$, precomputed stage probabilities P_i, and budgets r_i,
current level i,
an index set $\mathcal{I} = \cup_{j=1}^{i-1} \mathcal{I}_j$ with $\mathcal{I} = \emptyset$ when $i = 1$,
the rows of the current pair are indexed by the remaining original labels $[m] \setminus \mathcal{I}$.

Ensure: A secret vector $\mathbf{s}$ such that $|\mathbf{b} - \mathbf{A} \cdot \mathbf{s}| = t$.

Use the precomputed stage probabilities P_h and repetition budgets r_h
if $i < k + 1$ **then**
  for $j = 1$ up to r_i **do**
    Choose a random index set $\mathcal{I}_i \subset [m] \setminus \mathcal{I}$ of size N_i
    Set the local index copy $\bar{\mathcal{I}} := \mathcal{I} \cup \mathcal{I}_i$
    Compute $(\bar{\mathbf{A}}, \bar{\mathbf{b}}) \leftarrow \mathsf{Conv}((\mathbf{A}, \mathbf{b}), \mathcal{I}_i)$
    Call $\mathrm{RP}((\bar{\mathbf{A}}, \bar{\mathbf{b}}), \{N_h, P_h, r_h\}_{h=1}^{k+1}, i + 1, \bar{\mathcal{I}})$ and store the result as $(\hat{\mathbf{s}}, \hat{\mathcal{I}})$
    The variables $\mathcal{I}$ and $(\mathbf{A}, \mathbf{b})$ of the current loop iteration are not modified
    if $(\hat{\mathbf{s}}, \hat{\mathcal{I}}) \neq \mathsf{failure}$ **then**
      Lift $\hat{\mathbf{s}}$ through the stored conversion for $\mathcal{I}_i$ to obtain the candidate secret $\mathbf{s}$ for the current instance
      if $|\mathbf{b} - \mathbf{A} \cdot \mathbf{s}| = t$ **then**
        return $(\mathbf{s}, \hat{\mathcal{I}})$
      end if
    end if
  end for
end if
if $i = k + 1$ **then**
  for $j = 1$ up to r_{k+1} **do**
    Choose a random index set $\mathcal{I}_{k+1} \subset [m] \setminus \mathcal{I}$ of size N_{k+1}
    $(\mathbf{A}_{\mathcal{I}_{k+1}}, \mathbf{b}_{\mathcal{I}_{k+1}}) \leftarrow [\mathbf{A}, \mathbf{b}]_{\mathcal{I}_{k+1}}$
    if $\mathbf{A}_{\mathcal{I}_{k+1}}$ is invertible **then**
      Set $\mathbf{s} = (\mathbf{A}_{\mathcal{I}_{k+1}})^{-1} \cdot \mathbf{b}_{\mathcal{I}_{k+1}}$
      if $|\mathbf{b} - \mathbf{A} \cdot \mathbf{s}| = t$ **then**
        return $(\mathbf{s}, \mathcal{I} \cup \mathcal{I}_{k+1})$
      end if
    end if
  end for
  return failure
end if
return failure

bounded pass, the false-accept probability is at most $N_{\mathrm{chk}}\binom{m-n}{t}(q-1)^t/q^{m-n}$, *where* N_{chk} *is the number of residual checks performed in that pass. This remains negligible for the large-field parameter ranges considered here. Similarly, rank failures introduce only a* $1+O(1/q)$ *expected overhead per pivot attempt, and on a clean successful path the probability that one of the* $k+1$ *required pivot/final blocks is singular is* $O((k+1)/q)$.

Proof. If all $\mathcal{I}_i$ are noise-free, the conversions remove only error-free equations. The final reduced solve therefore recovers the remaining secret coordinates, and the stored conversions lift them algebraically to the original secret; equivalently, the union $\mathcal{I} = \cup_i \mathcal{I}_i$ is an error-free information set of size n, so conditioned on the required conversion pivots and final information matrix being invertible the lifted vector is the true $\mathbf{s}$ and satisfies $|\mathbf{b} - \mathbf{A} \cdot \mathbf{s}| = t$. Conversely, if some guessed set contains a noisy coordinate, any accepted vector is not the true secret unless it passes a residual-weight check as a false acceptance. For a wrong candidate $\mathbf{s}'$, the unchecked residual coordinates are uniform over $\mathbb{F}_q$ in the random-instance model, so the probability that the residual has exactly weight t is at most $\binom{m-n}{t}(q-1)^t/q^{m-n}$ for each check. A union bound over the N_{chk} residual checks in a bounded pass gives the displayed pass-level bound. The rank statement follows from the standard $O(1/q)$ singularity probability for each random square pivot/final block and a union bound along a clean successful path. For $q = 2^{128}$ these terms are negligible, so except for these events the residual check accepts only noise-free information sets. □

Optimal index size. To (approximately) optimize the cost, we observe $\mathcal{T}$ as two perspectives. First of all, for every j, we observe

$$\mathcal{Q}(j) = \frac{\binom{m-t}{\mu_j}}{\binom{m}{\mu_j}} = \prod_{i=1}^{j} \mathcal{P}(i)$$

where $\mu_j = \sum_{i=1}^{j} N_i$. From the observation, $\frac{\mathcal{C}(1)}{\mathcal{Q}(1)}$ is only determined by the N_1 when $j = 1$. Inductively, once $\sum_{i=1}^{j-1} N_i$ is fixed once, the ratio $\frac{\mathcal{C}(j)}{\mathcal{Q}(j)}$ is only determined by the N_j.

Likewise, since $\mathcal{S} = \mathcal{V}_{k+1}/\mathcal{P}(k+1)$ and $\mathcal{P}(k+1) \cdot \prod_{i=1}^{k} \mathcal{P}(i) = \binom{m-t}{n}/\binom{m}{n}$, the last term of $B(\mathbf{N})$ is

$$\frac{\mathcal{S}}{\mathcal{Q}(k)} = \mathcal{V}_{k+1} \cdot \frac{\binom{m}{n}}{\binom{m-t}{n}},$$

which is invariant under the level k for fixed μ_k.

Motivated by the usual balancing heuristic for sums of positive cost terms, we generate candidate partitions by comparing the unnormalized terms against $\Delta \cdot \mathsf{T}$, where $\mathsf{T} = \binom{m}{n}/\binom{m-t}{n}$. The global factor $\gamma_{\mathrm{RP}}(\mathbf{N})$ depends on the completed partition, so it is applied only after the candidate bounded-pass cost has been formed. We choose the largest integer N_i satisfying this comparison and $1 \leq N_i < n - \mu_{i-1}$, starting at $i = 1$; the strict upper bound reserves a nonempty final Prange block. If there is no integer N_j for an index j, we let $N_j = n - \mu_{j-1}$ and run Prange's algorithm on the SD instance of dimension N_j. Then the total cost is given as $\frac{1}{\gamma_{\mathrm{RP}}(\mathbf{N})}\left(\sum_{i=1}^{j-1} \frac{\mathcal{C}(i)}{\mathcal{Q}(i)} + \frac{\binom{m}{n}}{\binom{m-t}{n}}(N_j^{\omega} + (m - \mu_{j-1}) \cdot N_j)\right)$.

Algorithm 4 RP Estimation

Input: SD parameters (m, n, t)
Output: The continuous fixed-budget cost estimate of the Reduce and Prange algorithm.
Set $\mathsf{T} = \frac{\binom{m}{n}}{\binom{m-t}{n}}$ ▷ T is a threshold.
Set $g_{\mathrm{P}} = 1 - 1/e$ ▷ standard fixed-budget approximation for plain Prange
Set $Cost = \mathsf{T} \cdot (n^{2.8} + mn)/g_{\mathrm{P}}$
This initial value is the internal plain-Prange cost under the same fixed-budget convention, not the external Gauss/ISD baseline copied from prior work.
for Δ in the prescribed threshold set $\mathcal{D}$ **do**
    Set $k = 0$
    while $\exists N_{k+1}$ with $1 \le N_{k+1} < n - \sum_{i=1}^{k} N_i$ such that $\frac{\mathcal{C}(k+1)}{\mathcal{Q}(k+1)} < \Delta \cdot \mathsf{T}$ **do**
        Set largest N_{k+1} with $1 \le N_{k+1} < n - \sum_{i=1}^{k} N_i$ such that $\frac{\mathcal{C}(k+1)}{\mathcal{Q}(k+1)} < \Delta \cdot \mathsf{T}$
        Set $k = k + 1$
    end while
    Set $N_{k+1} = n - \sum_{i=1}^{k} N_i$
    Compute $\gamma_{\mathrm{RP}} = G_1(r_1, \ldots, r_{k+1})$ for this candidate partition.
    Set $B = \sum_{i=1}^{k} \frac{\mathcal{C}(i)}{\mathcal{Q}(i)} + \mathsf{T} \cdot (N_{k+1}^{2.8} + (m - \mu_k) \cdot N_{k+1})$.
    $Cost = \min\{Cost, B/\gamma_{\mathrm{RP}}\}$
end for
Return the best $Cost$ and its corresponding partition $\{N_i\}_{i=1}^{k+1}$

The idealized estimator may evaluate all integral Δ values up to $n^{2.8}$. In the implementation, the prescribed threshold set is

$$\mathcal{D} = \left\{ \exp\left(\log 2 + \frac{s}{259}(2.8 \log n - \log 2) \right) : \ 0 \le s < 260 \right\} \cup \{1, 1.25, 1.5, 2, 3, 4, 8, 16, 32\},$$

with duplicate values removed. Thus the implementation searches a logarithmic grid on $[2, n^{2.8}]$ together with several small threshold values where low-dimensional candidates often change. It reports the best candidate found in this threshold family. The full algorithm for estimating the Reduce and Prange will be given by Algorithm 4.

On the Threshold T. Recall that in standard Prange, the success probability of collecting n error-free indices by sampling without replacement is

$$1/\mathsf{T} = \Pr(n \text{ sampled indices are all noise-free}) = \frac{\binom{m-t}{n}}{\binom{m}{n}}.$$

Thus $1/\mathsf{T}$ serves as a natural *baseline* success probability for any ISD-style approach whose termination condition is still to obtain n error-free equations.

In level-k RP, we partition the search into stages and the overall success event can be viewed as successively selecting disjoint index sets whose union contains n error-free indices. Each additional stage reduces the linear-algebra

cost by lowering the effective dimension (hence decreasing S), but it also incurs an additional probabilistic filter: the conditional probability that a newly sampled block is entirely noise-free. A convenient way to decide whether adding one more stage is beneficial is to compare the *marginal gain* in the per-run cost S against the *marginal loss* in the per-run success probability. Accordingly, we use the baseline T and introduce a multiplicative tolerance parameter Δ capturing the required reduction in S to compensate for a smaller stage-success probability. This leads to a threshold of the form

$$\mathsf{T} \cdot \Delta = \frac{\binom{m}{n}}{\binom{m-t}{n}} \cdot \Delta,$$

i.e., we only add/allocate an extra stage when its conditional success probability is not too far below the baseline, up to the factor Δ that reflects the concrete linear-algebra savings. Heuristically, the resulting optimization is most effective when the stage-wise contributions to the expected cost are of comparable magnitude.

2.3 Graphical Results and Discussion

This section presents graphical results illustrating the performance improvements of the RP algorithm across different parameter sizes. To analyze the behavior, we plot the bit gain $\Delta = \log_2 T_{\mathsf{Prange}} - \log_2 T_{\mathsf{RP}}$, that is, the number of bits of estimated cost that RP saves. Both sides are evaluated with the estimator of Section 2.2 under the same convention, so that T_{Prange} is its $k = 0$ case: a single block of size n, with pass-success factor $1 - (1 - \mathcal{P}_{m,t,n})^{1/\mathcal{P}_{m,t,n}}$. The baseline here is plain Prange, whereas the comparison tables of Section 4.2 use the best estimate reported by [50], so the two sets of numbers are not directly comparable. By definition $\Delta = 0$ for the Prange baseline, and $\Delta > 0$ means RP has a lower estimated cost under the same fixed-budget model.

We analyze the parameter sets $(m, n, t) = (2^{10}, 652, 106)$ and $(m, n, t) = (2^{12}, 1589, 172)$, plotting the changes in Δ while varying a single parameter at a time. The gain shrinks as m grows with (n, t) fixed (Fig. 1, Fig. 2), and grows as either n or t increases with the other parameters fixed (Fig. 3–Fig. 6). We have not identified a closed-form expression for this behavior; these are empirical observations for the plotted sweeps rather than an asymptotic characterization.

A closed-form approximation of the observed gain would aid in estimating RP's asymptotic complexity. We did not identify one that matches the observed values across the full parameter range. Nevertheless, we find that graphical visualization effectively highlights RP's performance improvements.

3 Regular-RP: Application to RSD

This section applies RP to SD with regular noise (RSD). As in the standard SD case, we freely move between the syndrome and generator formulations of RSD.

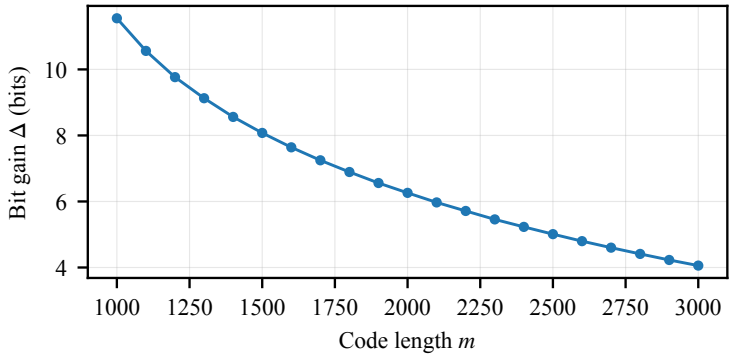


Fig. 1. Effect of increasing m on the bit gain Δ with a fixed $(n, t) = (652, 106)$, for SD over $\mathbb{F}_q$ with $q = 2^{128}$.

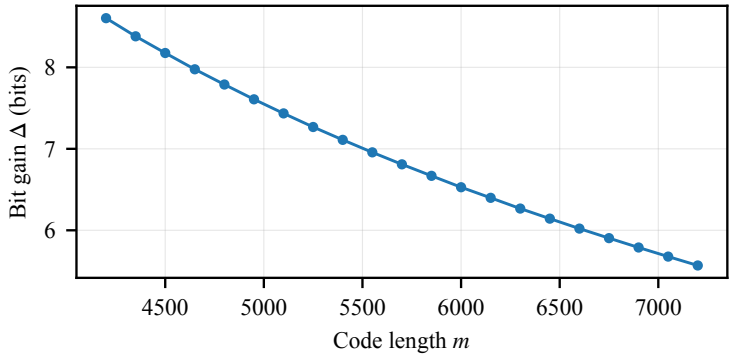


Fig. 2. Effect of increasing m on the bit gain Δ with a fixed $(n, t) = (1589, 172)$, for SD over $\mathbb{F}_q$ with $q = 2^{128}$.

We first introduce a definition of RSD, originating in [8]. The problem has numerous applications in PCG-like protocols [9,18,20,22,27,30,39,60–62,66,70]. We therefore switch to the RSD, where the unknown is the regular error vector, since it cleanly exposes the block structure exploited by our sampling strategy.

Problem 3 (Regular Syndrome Decoding over $\mathbb{F}_q$). Let m, n, t, β be positive integers with $m = t \cdot \beta$ and $\mathbb{F}_q$ be a finite field. Sample a full rank matrix $\mathbf{H} \leftarrow \mathbb{F}_q^{(m-n)\times m}$ and a column vector $\mathbf{e} := (\mathbf{e}_1 \| \mathbf{e}_2 \| \dots \| \mathbf{e}_t) \leftarrow \mathbb{F}_q^m$ such that $\mathbf{e}_i \in \mathbb{F}_q^\beta$ is of Hamming weight $|\mathbf{e}_i| = 1$ for $1 \le i \le t$. Given $(\mathbf{H}, \mathbf{y} = \mathbf{H} \cdot \mathbf{e})$, recover the error $\mathbf{e}$.

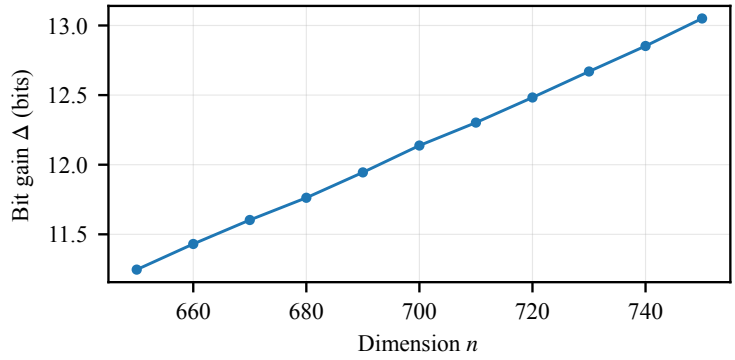


Fig. 3. Effect of increasing n on the bit gain Δ with a fixed $(m, t) = (2^{10}, 106)$, for SD over $\mathbb{F}_q$ with $q = 2^{128}$.

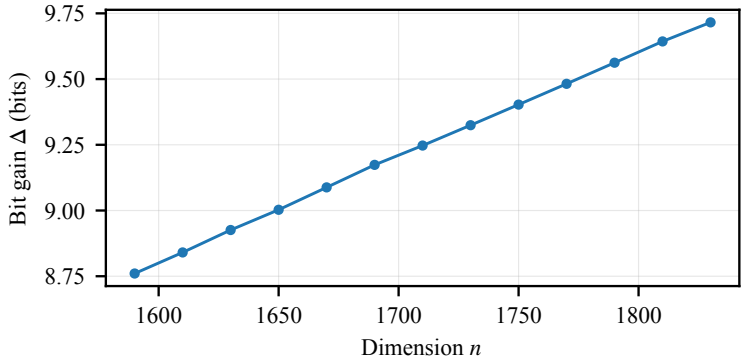


Fig. 4. Effect of increasing n on the bit gain Δ with a fixed $(m, t) = (2^{12}, 172)$, for SD over $\mathbb{F}_q$ with $q = 2^{128}$.

3.1 Algorithm for RSD over Large Fields

Esser and Santini [37] proposed algorithms for solving RSD by leveraging the structure of regular noise to identify error-free instances. We integrate this approach into our algorithm to refine security estimations for RSD over large fields.

This approach optimizes the guessing probability in the RP algorithm to solve RSD more efficiently in large fields, which we refer to as *regular-RP*. We follow the same reduction-and-ISD framework as RP, and only modify the sampling of the guessed zero positions to exploit the regular-noise constraint.

Regular-RP Algorithm. As the standard RP, suppose that $\{N_i\}_{i=1}^{k+1}$ is given and select index sets $\{\mathcal{I}_i\}_{i=1}^{k+1}$ such that $\#\mathcal{I}_i = N_i$ and $\sum_{i=1}^{k+1} N_i = n$. In the first step, we select $\mathcal{I}_1$ and compute $(\mathbf{A}', \mathbf{b}') \leftarrow \mathsf{Conv}((\mathbf{A}, \mathbf{b}), \mathcal{I}_1)$. We now focus on how to sample $\mathcal{I}_1$ to leverage the regular constraint. For a simple closed form,

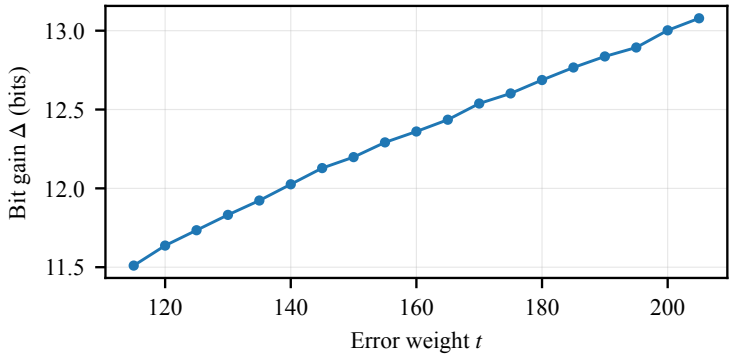


Fig. 5. Effect of increasing t on the bit gain Δ with a fixed $(m, n) = (2^{10}, 652)$, for SD over $\mathbb{F}_q$ with $q = 2^{128}$.

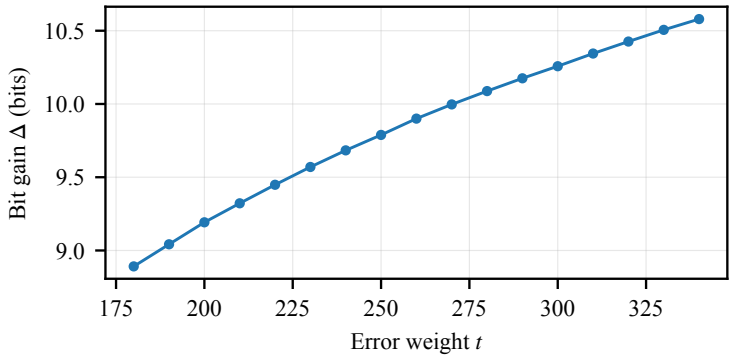


Fig. 6. Effect of increasing t on the bit gain Δ with a fixed $(m, n) = (2^{12}, 1589)$, for SD over $\mathbb{F}_q$ with $q = 2^{128}$.

first suppose that N_i is a multiple of t; the implementation and estimator use the balanced blockwise allocation described below when divisibility does not hold.

Since each $\mathbf{e}_j$ has exactly one nonzero entry, the probability of guessing a noise-free coordinate in the first regular block e_1 equals to $\frac{\beta-1}{\beta}$. To guess the second zero position, one considers two cases:

- E_1: Find another noise-free coordinate in $\mathbf{e}_1$.
- E_2: Find a noise-free coordinate in some $\mathbf{e}_j$ with $j \neq 1$.

The probability of event E_1 is $\frac{\beta-2}{\beta-1}$, whereas the probability of event E_2 is $\frac{\beta-1}{\beta}$. Thus, in the divisible case, the optimal strategy for selecting $\mathcal{I}_1$ is to choose N_1/t-zero coordinates from each $\mathbf{e}_j$. This balanced choice maximizes the product success probability for equal remaining block lengths, since $\sum_\ell \log(\beta - a_\ell)$ is Schur-concave under the constraint $\sum_\ell a_\ell = N_1$. The probability of correctly

selecting all N_1 zero positions is given by:

$$\mathcal{P}^R_{m,t,N_1} = \left(1 - \frac{N_1/t}{\beta}\right)^t = \left(1 - \frac{N_1}{m}\right)^t.$$

Similarly, when sampling $\mathcal{I}_i$, if the current prefix is equally balanced across the regular blocks and $t \mid N_i$, guessing N_i/t zero coordinates in each $\mathbf{e}_j$ yields an optimal probability. In general, we fix a deterministic cumulative balanced-prefix rule: at each prefix μ_i, choose integers $A_{\ell,i}$ as balanced as possible subject to $\sum_{\ell=1}^{t} A_{\ell,i} = \mu_i$ and $A_{\ell,i} \geq A_{\ell,i-1}$, and then set $a_{\ell,i} = A_{\ell,i} - A_{\ell,i-1}$. If $A_{\ell,i-1}$ coordinates have already been sampled from the ℓ-th block, then the stage success probability is

$$\mathcal{P}^R_i = \prod_{\ell=1}^{t} \frac{\beta - A_{\ell,i-1} - a_{\ell,i}}{\beta - A_{\ell,i-1}}.$$

The corresponding equal-block, divisible special case is $(1 - N_i/(m - \mu_{i-1}))^t$, which for $i = 1$ reduces to $(1 - N_1/m)^t$. To describe it formally, we denote a coordinate set of $\mathbf{e}_j$ as $[\mathbf{e}_j]$ and parse $\mathcal{I}_i = \cup_{j=1}^{t} \mathcal{I}_{i,j}$ such that $\mathcal{I}_{i,j} \subset [\mathbf{e}_j] \backslash \cup_{\ell=1}^{i-1} \mathcal{I}_{\ell,j}$ has size $a_{j,i}$. We define the RP algorithm with this modified sampling technique as the regular-RP algorithm, detailed in Algorithm 5.

For RSD instances, "regular weight t" means that the residual has exactly one nonzero entry in each regular block. This verification only decreases the false-accept probability compared with a plain Hamming-weight-t check and has negligible extra cost.

Complexity. Using the adjusted probability, we now compute the continuous fixed-budget cost estimate of the regular-RP algorithm. To this end, we define $\mathcal{P}^R(j), \mathcal{C}(j)$ and $\mathcal{S}^R$ as follows:

$$\begin{aligned}
\widehat{\mathcal{C}}_{M,N} &:= N^\omega + N^2(M-N) + N(M-N)^2, \\
\mathcal{P}^R(j) &= \mathcal{P}^R_{m-\mu_{j-1},t,N_j} \quad (1 \leq j \leq k) \\
\mathcal{P}^R(k+1) &= \mathcal{P}^R_{m-\mu_k,t,n-\mu_k} \\
\mathcal{V}_{k+1} &= (n-\mu_k)^\omega + (n-\mu_k) \cdot (m-\mu_k) \\
\mathcal{C}(j) &= \widehat{\mathcal{C}}_{m-\mu_{j-1},N_j} \\
\mathcal{S}^R &= \frac{1}{\mathcal{P}^R(k+1)} \cdot \mathcal{V}_{k+1}
\end{aligned}$$

where $\mu_j = \sum_{i=1}^{j} N_i$ for every j. Under the deterministic cumulative balanced-prefix rule, $\mathcal{P}^R(j)$ is determined by (μ_{j-1}, N_j); equivalently, without fixing this rule, the notation $\mathcal{P}^R_{m-\mu_{j-1},t,N_j}$ should be understood as depending on the blockwise prefix vector $(A_{\ell,j-1})_\ell$. In the divisible equal-block case it reduces to $(1 - N_j/(m - \mu_{j-1}))^t$. Then, using the notation, the bounded-pass cost of the

Algorithm 5 Regular-RP

Require: (m, n, t)-RSD over $\mathbb{F}_q$ instance represented by the current pair $(\mathbf{A}, \mathbf{b})$,
conversion history for lifting candidates back to the original secret space,
block sizes $\{N_i\}_{i=1}^{k+1}$, regular stage probabilities P_i^R, and budgets r_i,
current level i,
an index set $\mathcal{I} = \cup_{j=1}^{i-1} \mathcal{I}_j$ with $\mathcal{I} = \emptyset$ when $i = 1$,
blockwise prefixes $\mathcal{I}_{h,\ell}$ used for regular sampling; the rows of the current pair are indexed by $[m] \setminus \mathcal{I}$.

Ensure: A secret vector $\mathbf{s}$ such that $\mathbf{b} - \mathbf{A} \cdot \mathbf{s}$ has regular weight t.

Use the precomputed regular stage probabilities P_h^R and repetition budgets r_h
if $i < k + 1$ **then**
  for $j = 1$ up to r_i **do**
    Choose cumulative balanced counts $A_{\ell,i}$, set $a_{\ell,i} = A_{\ell,i} - A_{\ell,i-1}$, and sample local sets $\mathcal{I}_{i,\ell} \subset [\mathbf{e}_\ell] \setminus \bigcup_{h<i} \mathcal{I}_{h,\ell}$ of size $a_{\ell,i}$
    Set local $\mathcal{I}_i = \bigcup_\ell \mathcal{I}_{i,\ell}$, $\bar{\mathcal{I}} = \mathcal{I} \cup \mathcal{I}_i$, and updated local blockwise prefixes $\overline{\mathsf{pref}}$
    Compute $(\bar{\mathbf{A}}, \bar{\mathbf{b}}) \leftarrow \mathsf{Conv}((\mathbf{A}, \mathbf{b}), \mathcal{I}_i)$
    $(\hat{\mathbf{s}}, \hat{\mathcal{I}}) \leftarrow \textsc{RegularRP}((\bar{\mathbf{A}}, \bar{\mathbf{b}}), \{N_h, P_h^R, r_h\}_{h=1}^{k+1}, i + 1, \bar{\mathcal{I}}, \overline{\mathsf{pref}})$
    if $(\hat{\mathbf{s}}, \hat{\mathcal{I}}) \neq \mathsf{failure}$ **then**
      Lift $\hat{\mathbf{s}}$ through the stored conversion for $\mathcal{I}_i$ to obtain the candidate secret $\mathbf{s}$ for the current instance
      if $\mathbf{b} - \mathbf{A} \cdot \mathbf{s}$ has regular weight t **then**
        return $(\mathbf{s}, \hat{\mathcal{I}})$
      end if
    end if
  end for
end if
if $i = k + 1$ **then**
  for $j = 1$ up to r_{k+1} **do**
    Choose cumulative balanced counts $A_{\ell,k+1}$, set $a_{\ell,k+1} = A_{\ell,k+1} - A_{\ell,k}$, sample $\mathcal{I}_{k+1,\ell} \subset [\mathbf{e}_\ell] \setminus \bigcup_{h<k+1} \mathcal{I}_{h,\ell}$ of size $a_{\ell,k+1}$, and set $\mathcal{I}_{k+1} = \cup_\ell \mathcal{I}_{k+1,\ell}$
    $(\mathbf{A}_{\mathcal{I}_{k+1}}, \mathbf{b}_{\mathcal{I}_{k+1}}) \leftarrow [\mathbf{A}, \mathbf{b}]_{\mathcal{I}_{k+1}}$
    if $\mathbf{A}_{\mathcal{I}_{k+1}}$ is invertible **then**
      Set $\mathbf{s} = (\mathbf{A}_{\mathcal{I}_{k+1}})^{-1} \cdot \mathbf{b}_{\mathcal{I}_{k+1}}$
      if $\mathbf{b} - \mathbf{A} \cdot \mathbf{s}$ has regular weight t **then**
        return $(\mathbf{s}, \mathcal{I} \cup \mathcal{I}_{k+1})$
      end if
    end if
  end for
  return failure
end if
return failure

level-k regular-RP algorithm can be expressed as

$$\frac{1}{\mathcal{P}^R(1)}\left(\mathcal{C}(1)+\frac{1}{\mathcal{P}^R(2)}\left(\mathcal{C}(2)+\cdots\right.\right.$$
$$\left.\left.+\frac{1}{\mathcal{P}^R(k)}\left(\mathcal{C}(k)+\mathcal{S}^R\right)\right)\right)$$

To simplify it, we also define $\mathcal{Q}^R(j)$ as $\mathcal{Q}^R(j)=\prod_{i=1}^{j}\mathcal{P}^R(i)$. We denote this bounded-pass cost by

$$B^R(\mathbf{N})=\sum_{j=1}^{k}\frac{\mathcal{C}(j)}{\mathcal{Q}^R(j)}+\frac{\mathcal{S}^R}{\mathcal{Q}^R(k)}.$$

As in the RP algorithm, the bounded searches contribute one global factor $\gamma_{\mathrm{RRP}}=G_1^R(r_1,\ldots,r_{k+1})$, defined by the same recursion as above with $\mathcal{P}(j)$ replaced by $\mathcal{P}^R(j)$. The revised numerical estimator evaluates this factor through the recursion using the continuous fixed-budget convention $r_j=1/\mathcal{P}^R(j)$. Combining all components, the continuous fixed-budget constant-success cost estimate is expressed as

$$\begin{aligned}T_{\mathrm{RRP}}(\mathbf{N})&=\frac{B^R(\mathbf{N})}{\gamma_{\mathrm{RRP}}(\mathbf{N})}\\&=\frac{1}{\gamma_{\mathrm{RRP}}(\mathbf{N})}\left(\sum_{j=1}^{k}\frac{\mathcal{C}(j)}{\mathcal{Q}^R(j)}+\frac{\mathcal{S}^R}{\mathcal{Q}^R(k)}\right).\end{aligned}$$

Again, the success correction is applied once globally after the candidate bounded-pass cost is formed.

4 Concrete Estimation of SD via the Reduce and Prange Algorithm

This section reports concrete estimates for our algorithms over large fields, with emphasis on the parameter ranges used in PCG applications. The results are summarized as follows:

- Table 3: comparison of RP with previous estimates for solving SD over $\mathbb{F}_{2^{128}}$.
- Table 4: RP estimates on the 128-bit parameter recommendations of [50].
- Table 5: comparison of regular-RP with previous estimates for solving RSD.
- Table 6: regular-RP estimates on the 128-bit parameter recommendations of [50].
- Table 2: Monte-Carlo validation of the pass-success factor γ_{RP}.

– Table 7: RP estimates for the Scalable Multiparty Garbling parameters of [11].

Throughout this section, reported logarithmic values are $\log_2$ counts of arithmetic operations over $\mathbb{F}_q$ (field additions and multiplications), following [21,50]. They are not bit-operation counts. For the RP and regular-RP columns, the reported quantity is the continuous fixed-budget constant-success cost, measured in $\mathbb{F}_q$-operations. Our numerical estimations are derived from the candidate-wise costs $B(\mathbf{N})/\gamma_{\mathrm{RP}}(\mathbf{N})$ and $B^R(\mathbf{N})/\gamma_{\mathrm{RRP}}(\mathbf{N})$ defined above, rather than from a Las Vegas early-stopping expected-time model. For one-stage Prange this fixed-budget convention is conservative relative to the usual early-stopping expectation. For nested RP and regular-RP, we report it as a separate fixed-budget constant-success estimator rather than as a Las Vegas early-stopping expectation; the RP columns therefore should not be read as early-stopping timings. The RP table values also use the conservative body-cost convention $\widehat{\mathcal{C}}_{M,N}$ from Section 2.2. As a model-sensitivity check, the artifact also includes a 1035-row recursion-only grid comparing the continuous budgets used in the analytic estimator with literal integer budgets $r_j = \lceil 1/\mathcal{P}(j) \rceil$. This check is not used to alter the table values; it documents the difference between the analytic interpolation and a literal bounded-loop schedule for the pass-success factor. The adjusted-parameter checks also record the unrounded estimate, the pass-success factor, and the number of levels for every adjusted recommendation. We compare our results with recent attacks [21,37,50], using the appropriate baseline for each problem setting. For fair comparisons, the time complexity of the Gaussian elimination algorithm on a $d \times d$ matrix is evaluated using the same cost model. Specifically, we use a cost model of $d^{2.8}$, as in previous works [17, 50].[7]

Validation of the success-probability model. We additionally checked the recursive success factor G_1 on parameters where direct simulation is feasible. These checks are not used to extrapolate the concrete security estimates. We use them only to verify that the bounded-pass probability used in the estimator matches the nested RP experiment. In all checks we used the literal integer budgets $r_j = \lceil 1/\mathcal{P}(j) \rceil$ and seed 20260627. The first check samples only the nested index sets, with the error positions fixed, and declares success when a bounded pass reaches at least one clean leaf. It covers 19 cases with $24 \le m \le 256$, $12 \le n \le 128$, $3 \le t \le 20$, levels 2 through 8, and balanced, front-heavy, back-heavy, low-noise, and higher-noise partitions; the total number of sampled bounded passes is 765 000, and a local Python rerun took 104.8 seconds. The maximum absolute deviation from G_1 was 0.00589, and the observed deviations are consistent with Monte Carlo sampling error over the tested cases. The second check is a full end-to-end C++ implementation over the prime field of cardinality $q = 2147483647$, run on an Intel Core i9-10940X at 3.30 GHz using 28 threads. It uses 8 cases, totaling 237 000 bounded passes; the maximum absolute deviation from G_1 was 0.0045, and every deviation was within the reported 95% confidence interval. Each trial samples $(\mathbf{A}, \mathbf{s}, \mathbf{e})$, runs the nested bounded search, solves the

[7] The exponent is inspired by the Strassen matrix multiplication algorithm.

Table 2. Representative bounded-pass validation of the recursive success factor G_1. The table reports the predicted pass-success probability, the empirical success rate, and the full-RP diagnostics for selected cases; aggregate coverage and maximum deviations are stated in the text. Here MC stands for Monte Carlo: 'Index MC' rows sample only the nested index sets, and 'Full RP' rows run the end-to-end implementation. The column $\hat{p} \pm 95\%$ CI gives the empirical success rate and the half-width of its 95% confidence interval (CI).

Check	$(m, n, t; N_i)$	Passes	G_1	$\hat{p} \pm 95\%$CI	Diagnostic
Index MC	$(60, 30, 12;\ 10, 10, 10)$	60 000	0.4001	0.3988 ± 0.0039	clean leaf
Index MC	$(192, 96, 16;\ 24, 16, 16, 16, 12, 12)$	25 000	0.3341	0.3358 ± 0.0059	clean leaf
Index MC	$(256, 128, 20;\ 24, 20, 18, 16, 14, 14, 12, 10)$	20 000	0.2757	0.2717 ± 0.0062	clean leaf
Full RP	$(24, 12, 4;\ 4, 4, 4)$	50 000	0.6247	0.6236 ± 0.0042	return; false/rank 0/0
Full RP	$(40, 20, 6;\ 8, 6, 6)$	20 000	0.4843	0.4798 ± 0.0069	return; false/rank 0/0
Full RP	$(60, 30, 3;\ 10, 10, 10)$	12 000	0.5997	0.6031 ± 0.0088	return; false/rank 0/0

final information system, and accepts only if the full residual has Hamming weight t. The comparison with G_1 is meaningful because the recursion defining G_j is exactly the conditional probability that, given a clean prefix up to level $j-1$, the bounded search from level j reaches a clean final information set. Thus G_1 is the clean-leaf probability of one bounded pass. The full end-to-end check then verifies that, over a large field, this clean-leaf event agrees with the actual algorithmic return event up to false residual accepts and rank failures; neither occurred in the tested cases. Table 2 summarizes the results.

As shown in Table 4, the listed 128-bit recommendations from [50] fall below the target under our RP estimate. On the comparison rows of Table 3, RP is at or below the best previously listed estimate throughout, with margins between 1 and 6 bits. Within each block of the table the margin narrows as m grows. For RSD over large fields, regular-RP is competitive for many small and medium parameter settings, while algebraic estimates remain stronger for some larger parameters. For this purpose, we first recall the parameters for SD over large fields.

- q: the size of fields $= 2^{128}$.
- m: the code length.
- n: the dimension of SD secret $\mathbf{s}$.
- t: the number of nonzero entries in $\mathbf{e}$.

Our focus is mainly on the parameter regime $(m, q, t) = (poly(n), n^{w(1)}, o(m))$, which is widely used in cryptographic applications [9,10,17,19,20,27,31, 62,66,70].

For a more detailed analysis, the complete source code for our computational experiments is publicly available. The artifact contains the C++ and Python estimators used for the tables, the success-probability checks described above, and the merged raw outputs. The main estimator is a parallel C++ implementation of the threshold-partition search in Algorithm 4, run on the same machine. For each candidate partition it records the bounded-pass cost $B(\mathbf{N})$, the recursive factor $\gamma_{\mathrm{RP}}(\mathbf{N})$ or $\gamma_{\mathrm{RRP}}(\mathbf{N})$, and the final value B/γ. The

merged output contains 47 table rows. All active table entries reported below are obtained from this threshold-partition estimator or from the explicitly identified regular-RP estimator. The code can be accessed via the following link: `github.com/JiseungKim90/reduce-and-prange`.

- RP: The basic cost model proposed by Section 2.2.
- regular-RP: RP for RSD Section 3.

To evaluate our algorithm, we compare its performance against ISD and its variants. We begin by recalling the results from [50]:

- For the parameter regime $\mathcal{R} = \mathbb{F}_q$ with prime power $q = n^{w(1)}$, $t = o(m)$ and $(1+\beta) \cdot n \leq m = poly(n)$ for some constant β, the statistical decoding including the 2.0 variant [26] needs more cost than Prange's algorithm [59].
- For the above parameter setup, information set decoding and its variants outperform other algorithms.

Therefore, ISD and its variants serve as the most relevant benchmarks for evaluating our results. In the following, we then describe the complexity model of the algorithms.

4.1 Cost models of the Previous algorithms

This section introduces cost models for three major algorithms used to solve SD over large fields: Gauss ISD [59], ISD frameworks, and Statistical Decoding (StatD) [26, 50]. Throughout, complexity is measured as the number of arithmetic operations (additions and multiplications) over $\mathbb{F}_q$, the unit used in all comparison tables.

Gauss Algorithm. The Gauss algorithm is identical to Prange's algorithm (see Algorithm 1), but it covers an additional case: SD instances $(\mathbf{A}, \mathbf{b}) = (\mathbf{A}, \mathbf{A}\cdot\mathbf{s}+\mathbf{e})$ can be transformed into the problem of finding $\mathbf{e}$ when $\mathbf{H}\cdot\mathbf{e} = \mathbf{H}\cdot\mathbf{b}$ for a parity check matrix $\mathbf{H}$, and vice versa, as shown in [55, Lemma 4.9].

The computational cost in this case is: $(m-n)^{2.8} \cdot \left(\frac{\binom{m}{t}}{\binom{m-n}{t}} \right)$. Thus, the Gauss algorithm follows the cost model:

$$\texttt{Gauss_cost_model} = \log \left(\min\{n^{2.8}, (m-n)^{2.8}\} \cdot \left(\frac{\binom{m}{t}}{\binom{m-n}{t}} \right) \right)$$

Information Set Decoding. For SD over large fields and RSD problems, Liu et. al. [50] showed that the Stern–Dumer ISD (StD-ISD) is the most efficient algorithm. As a result, they estimate the hardness of SD over large fields and RSD based on the cost of StD-ISD. Thus, we introduce only the cost of the StD-ISD attack as a representative of ISD frameworks.

Lemma 4.1 (Cost of StD-ISD attack [50]) *The (m, n, t)-SD problem over $\mathbb{F}_q$ can be solved by the StD-ISD variant in expected time*

$$T_{\mathsf{StD}}(m,n,t) = \min_{p,\ell} \frac{1}{\mathcal{P}(p,\ell)} \left(T_{\mathsf{Gauss}} + 2L \cdot m \left(1 + \frac{L}{q^\ell} \right) \right)$$

where $T_{\mathsf{Gauss}} = \frac{(m-n-\ell)\cdot m^2}{\log(m-n-\ell)}$, $L = \binom{\frac{n+\ell}{2}}{p/2} \cdot (q-1)^{p/2}$ *and* $\mathcal{P}(p,\ell) = \frac{\binom{m-n-\ell}{t-p}}{\binom{m}{t}} \cdot \frac{L^2}{(q-1)^p}$.[8]

Statistical Decoding framework. While ISD algorithms aim to recover the secret element $\mathbf{s}$ directly, the Low Weight Parity Check (LWPC) algorithm focuses on distinguishing between random instances and SD instances. In other words, it determines whether $\mathbf{b}$ is equal to $\mathbf{A} \cdot \mathbf{s} + \mathbf{e}$ or just a random element. Although this algorithm is not well-suited for solving SD over large fields, we still provide its attack complexity as a comparison for decision problems. For $q \geq 4t$, the adapted StatD 2.0 algorithm is proposed in [50].

Lemma 4.2 ([26, 50]) *The adapted StatD 2.0 algorithm can solve the decisional (m, n, t)-SD over $\mathbb{F}_q$ in time T with constant advantage, where $w = w(s) \in \mathbb{N}$ and $q \geq 4t$. The formula for T is given by*

$$T = \min_s \left(T_1 \cdot \left(\frac{\binom{m}{t}}{\binom{m-w}{t}} \cdot \frac{q}{q-1} \right)^2 + s \cdot \log q \cdot q^s \right).$$

Here, T_1 represents the time it takes to find one parity check vector. Following [17, 50], we also set $w = n - s + 1$ and $T_1 = n + 1$. As a result, we have

$$T = \min_s \left((n+1) \cdot \left(\frac{\binom{m}{t}}{\binom{m-n+s-1}{t}} \cdot \frac{q}{q-1} \right)^2 + s \cdot \log q \cdot q^s \right).$$

4.2 Numerical Results of SD over Large Fields

This subsection reports the estimated RP costs for several SD parameter sets. For each parameter set, the estimator searches over candidate thresholds Δ under the constraint $\Delta \leq n^{2.8}$ and evaluates every completed candidate partition using the global factor $\gamma_{\mathrm{RP}}(\mathbf{N})$. The idealized version of Algorithm 4 loops over all integral thresholds; the implementation uses the threshold grid $\mathcal{D}$ defined above to locate the minimizing region and then evaluates the resulting candidate partitions with the displayed continuous fixed-budget formula. We therefore report the best values found within this threshold family, not a proof of global optimality over

[8] As the paper [50] only presents an estimation of SD over $\mathbb{F}_2$, we rely on the formulation provided by their Python script to describe our approach.

Table 3. Comparison results with previous results [50, Tab. 2] and ours. All values are logarithmic scale of arithmetic operations over $\mathbb{F}_q$ with the field size $\log q = 128$ and are rounded to the nearest integer. The column StatD indicates the result of the statistical decoding attack. In this work, the 'This work' column presents the continuous fixed-budget estimate of the Reduce and Prange algorithm using the conservative body-cost convention $\widehat{\mathcal{C}}_{M,N}$.

Parameters			[50]†			**This work**
m	n	t	Gauss	ISD	StatD	
2^{10}	652	57	111	111	184	**105**
2^{12}	1589	98	100	100	151	**94**
2^{14}	3482	198	101	101	149	**97**
2^{16}	7391	389	103	103	147	**99**
2^{18}	15336	760	105	105	146	**102**
2^{20}	32771	1419	107	107	145	**104**
2^{22}	67440	2735	108	108	144	**107**
2^{12}	$3/4 \cdot 2^{12}$	44	117	117	189	**111**
2^{14}	$3/4 \cdot 2^{14}$	39	111	111	170	**107**
2^{16}	$3/4 \cdot 2^{16}$	34	107	107	151	**103**
2^{18}	$3/4 \cdot 2^{18}$	32	108	108	145	**105**
2^{20}	$3/4 \cdot 2^{20}$	31	112	112	143	**109**
2^{22}	$3/4 \cdot 2^{22}$	30	116	116	141	**113**
2^{24}	$3/4 \cdot 2^{24}$	29	119	119	139	**117**

† [50] also presents numerical results for the statistical decoding 2.0 attack [26]. However, we did not take these results into account because they showed lower performance than the Gauss and ISD results in these parameters in the table.

all partitions. The corresponding partitions and success factors are recorded in the raw output. Table 3 gives the comparison with [50].

For several parameters used in [50], the estimated bit-security of SD over large fields is reduced by up to 6 bits when $\log q = 128$. The recommended 128-bit rows in Table 4 fall below the target under RP, so the rightmost columns give adjusted choices following the original table format.

Remark 4.3 *The optimized partitions may contain many levels: in our searches, k often exceeds 20 and can be much larger for the largest parameters. For readability, the tables report the resulting costs and parameter recommendations rather than listing every block size N_i. The full estimator output records the corresponding partitions; empirically, the first block is often a substantial fraction of n, and similar behavior is observed for RSD.*

Table 4. Comparison results with previous results [50, Tab. 3] and ours. All values are logarithmic scale of arithmetic operations over $\mathbb{F}_q$ with the field size $\log q = 128$ and are rounded to the nearest integer except for the adjusted-row estimates. In this work, the 'RP' column presents the continuous fixed-budget estimate of the Reduce and Prange algorithm. The rightmost columns keep the original table format and list adjusted parameter choices, obtained by increasing either n or t, for targeting 128-bit security under RP; each adjusted group also gives the corresponding $\log_2 T_{\mathrm{RP}}$ value to two decimals.

Parameters			[50]	RP	Recommend Parameters							
					Increase n				Increase t			
m	n	t			m	n	t	$\log_2 T$	m	n	t	$\log_2 T$
2^{12}	1321	172	128	**121**	2^{12}	1400	172	128.54	2^{12}	1321	188	130.49
2^{14}	2895	338		**122**	2^{14}	3053	338	128.02	2^{14}	2895	367	130.34
2^{16}	6005	667		**123**	2^{16}	6450	667	130.38	2^{16}	6005	720	130.35
2^{18}	12160	1312		**124**	2^{18}	12900	1312	129.53	2^{18}	12160	1400	129.77
2^{20}	25346	2467		**125**	2^{20}	26900	2467	129.99	2^{20}	25346	2617	129.70
2^{22}	50854	4788		**125**	2^{22}	53490	4788	129.53	2^{22}	50854	5050	129.58

4.3 Numerical Results of RSD over Large Fields

This subsection reports regular-RP estimates for RSD and compares them with [21, 50]. As above, we focus on large fields; the binary-field case is discussed separately in Section A.

The works [21,50] provide numerical results for RSD over $\mathbb{F}_q$ with $\log q = 128$, since the corresponding estimates have only mild dependence on the field size. By contrast, [37] is formulated for the binary setting. We therefore revisit the permutation-based estimate of [37] and re-estimate it over $\mathbb{F}_{2^{128}}$ for comparison.

[37] introduces three algorithms for RSD: permutation-based regular-ISD (P-ISD), enumeration-based regular-ISD (E-ISD), and representation-based regular-ISD (R-ISD). These adapt well-known ISD techniques [12, 52, 59] to the regular-noise structure. We focus on P-ISD, since the enumeration and representation variants require many collision vectors over $\mathbb{F}_q$, which increases their concrete cost in the large-field setting.

Indeed, numerical results from [50, Tab. 3, Tab. 4] indicate that the Gaussian elimination costs are consistently comparable to the ISD costs for RSD over $\mathbb{F}_{2^{128}}$. However, for RSD over $\mathbb{F}_2$, the ISD algorithms consistently outperform Gaussian elimination in terms of cost.

Table 5 and Table 6 present the numerical results. Regular-RP gives the lowest listed estimate for many small and medium parameter settings, while the algebraic estimates remain lower for some larger RSD parameters. The largest improvement over the best previously listed estimate is 15 bits, on the row $(m, n, t) = (2^{10}, 652, 106)$ of Table 5. As in the previous subsection, all values are logarithmic counts of arithmetic operations over $\mathbb{F}_{2^{128}}$. The subsequent RSD works mentioned in the introduction use benchmark choices and cost models that are not directly aligned with the rows of these tables; we therefore discuss

their crossover qualitatively rather than merging their numbers into the same comparison table.

Revisiting P-ISD for RSD over $\mathbb{F}_q$ with $q \gg 2$. As discussed in Section 3.1 and [36, Sec. 4.1], the large-field permutation-based regular-ISD estimate is

$$\left(1 - \frac{n}{m}\right)^{-t} \cdot \min\{n^{2.8}, (m-n)^{2.8}\}$$

for solving (m, n, t)-RSD. In the binary setting, [36, 37] instead use

$$\left(1 - \frac{n-t}{m}\right)^{-t} \cdot \min\{n^{2.8}, (m-n)^{2.8}\}$$

because the binary regularity gives t additional error-free linear equations, as recalled in Section A. These equations are not available in the same form over general large fields.

Acknowledgments. JK is supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2023-00273635, RS-2026-25477662). Changmin Lee is supported by the Korea Internet & Security Agency (KISA) grant funded by the Korean government (PIPC) (No. 2780000036, Masters and Doctoral Program for Advanced Talent Development in Personal Data Protection and Responsible Use) and by Korea University grant K2603241.

References

1. Aguilar-Melchor, C., et al.: The syndrome decoding in the head (SD-in-the-Head) signature scheme. Submission to the NIST call for additional post-quantum signatures (2023), `https://www.sdith.org/docs/sdith-v1.1.pdf`
2. Aguilar-Melchor, C., Gama, N., Howe, J., Hülsing, A., Joseph, D., Yue, D.: The return of the SDitH. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 564–596. Springer (2023)
3. Aguilar-Melchor, C., Hülsing, A., Joseph, D., Majenz, C., Ronen, E., Yue, D.: SDitH in the QROM. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 317–350. Springer (2023)
4. Alekhnovich, M.: More on average case vs approximation complexity. Computational Complexity **20**(4), 755–786 (2011). `https://doi.org/10.1007/s00037-011-0029-x`
5. Applebaum, B., Barak, B., Wigderson, A.: Public-key cryptography from different assumptions. In: Proceedings of the forty-second ACM symposium on Theory of computing. pp. 171–180 (2010)
6. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5677, pp. 595–618. Berlin, Heidelberg (2009)
7. Applebaum, B., Damgård, I., Ishai, Y., Nielsen, M., Zichron, L.: Secure arithmetic computation with constant computational overhead. In: Annual International Cryptology Conference. pp. 223–254. Springer (2017)

Table 5. Numerical results for RSD over $\mathbb{F}_{2^{128}}$ (rounded to the nearest integer). The 'This work' column reports the continuous fixed-budget regular-RP estimate obtained from the threshold-partition estimator.

Parameters			[49, 50]	[21]	[37]†	**This work**
m	n	t				
2^{10}	652	106	194	179	178	**163**
2^{12}	1589	172	155	150	151	**143**
2^{14}	3482	338	150	150	149	**141**
2^{16}	7391	667	151	150	151	**144**
2^{18}	15336	1312	153	**133**	154	148
2^{20}	32771	2467	155	**131**	155	150
2^{22}	67440	4788	152	**110**	156	153
2^{10}	652	57	111	111	107	**101**
2^{12}	1589	98	100	107	99	**92**
2^{14}	3482	198	101	110	101	**96**
2^{16}	7391	389	103	111	103	**99**
2^{18}	15336	760	105	107	105	**102**
2^{20}	32771	1419	107	**102**	107	104
2^{22}	67440	2735	108	**104**	108	107
2^{12}	$3/4 \cdot 2^{12}$	44	117	127	116	**109**
2^{14}	$3/4 \cdot 2^{14}$	39	111	127	111	**106**
2^{16}	$3/4 \cdot 2^{16}$	34	107	128	107	**103**
2^{18}	$3/4 \cdot 2^{18}$	32	108	132	108	**105**
2^{20}	$3/4 \cdot 2^{20}$	31	112	139	112	**109**
2^{22}	$3/4 \cdot 2^{22}$	30	116	145	116	**113**
2^{24}	$3/4 \cdot 2^{24}$	29	119	152	119	**117**

†This is our re-estimate of the permutation-based regular-ISD algorithm over $\mathbb{F}_{2^{128}}$.

8. Augot, D., Finiasz, M., Sendrier, N.: A family of fast syndrome based cryptographic hash functions. In: Mycrypt. vol. 3715, pp. 64–83. Springer (2005)
9. Baum, C., Braun, L., Munch-Hansen, A., Scholl, P.: Moz$\mathbb{Z}_2^k$arella: efficient vector-OLE and zero-knowledge proofs over z 2 k. In: Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part IV. pp. 329–358. Springer (2022)
10. Baum, C., Malozemoff, A.J., Rosen, M.B., Scholl, P.: Mac'n'cheese mac′n′ cheese: Zero-knowledge proofs for boolean and arithmetic circuits with nested disjunctions. In: Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part IV 41. pp. 92–122. Springer (2021)

Table 6. Comparison with the recommended RSD parameters of [50, Tab. 3]. All values are logarithmic counts of arithmetic operations over $\mathbb{F}_q$ with $\log q = 128$. The 'RP' column presents regular-RP estimates. The rightmost columns keep the original table format and list adjusted parameter choices, obtained by increasing either n or t, for targeting 128-bit security under regular-RP; each adjusted group also gives the corresponding regular-RP $\log_2 T$ value to two decimals. The dash rows already exceed the target under regular-RP. All logarithmic values in the original and RP columns are rounded to the nearest integer.

Parameters			[50]	RP	Recommend Parameters: Increase n				Increase t			
m	n	t			m	n	t	$\log_2 T$	m	n	t	$\log_2 T$
2^{12}	1377	172	128	**123**	2^{12}	1460	172	130.44	2^{12}	1377	187	131.96
2^{14}	2909	338		**121**	2^{14}	3165	338	129.81	2^{14}	2909	373	131.11
2^{16}	6091	667		**123**	2^{16}	6500	667	129.92	2^{16}	6091	713	130.25
2^{18}	14796	1312		144	-	-	-	-	-	-	-	-
2^{20}	30978	2467		144	-	-	-	-	-	-	-	-
2^{22}	75396	4788		166	-	-	-	-	-	-	-	-

Table 7. RP estimates for the Scalable Multiparty Garbling parameters discussed in the introduction. The estimates use the same continuous fixed-budget estimator as the SD tables. The noise vector of [11, Def. 1] is Bernoulli, so its weight T is distributed as $\mathrm{Binom}(Q, \tau)$ rather than being fixed. Here t^* is the largest weight for which the RP estimate stays below the claimed level, and the last column is the probability that the drawn weight is at most t^*. The fifth column evaluates the estimator at the mean weight $\lceil \tau Q \rceil$.

Q	V	τ	Claimed	$\log_2 T_{\mathrm{RP}}$ at $\lceil \tau Q \rceil$	t^*	$\Pr[T \le t^*]$
555	127	2^{-2}	80	76.61	145	0.75
785	214	2^{-2}	128	122.75	205	0.78

11. Beck, G., Goel, A., Hegde, A., Jain, A., Jin, Z., Kaptchuk, G.: Scalable multiparty garbling. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. pp. 2158–2172 (2023)
12. 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: Annual international conference on the theory and applications of cryptographic techniques. pp. 520–536. Springer (2012)
13. Bernstein, D.J., Lange, T., Peters, C.: Attacking and defending the McEliece cryptosystem. In: Buchmann, J., Ding, J. (eds.) Post-Quantum Cryptography. pp. 31–46. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
14. Blum, A., Furst, M.L., Kearns, M.J., Lipton, R.J.: Cryptographic primitives based on hard learning problems. In: Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings. Lecture Notes in Computer Science, vol. 773, pp. 278–291 (1993)

15. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. Journal of the ACM (JACM) **50**(4), 506–519 (2003)
16. Both, L., May, A.: Decoding linear codes with high error rate and its impact for LPN security. In: International Conference on Post-Quantum Cryptography. pp. 25–46. Springer (2018)
17. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y.: Compressing vector OLE. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018. pp. 896–912 (2018)
18. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Resch, N., Scholl, P.: Correlated pseudorandomness from expand-accumulate codes. In: Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part II. pp. 603–633. Springer (2022)
19. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Rindal, P., Scholl, P.: Efficient two-round OT extension and silent non-interactive secure computation. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 291–308 (2019)
20. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Scholl, P.: Efficient pseudorandom correlation generators: Silent OT extension and more. In: Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part III 39. pp. 489–518. Springer (2019)
21. Briaud, P., Øygarden, M.: A new algebraic approach to the regular syndrome decoding problem and implications for PCG constructions. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 391–422. Springer (2023)
22. Bui, D., Couteau, G.: Improved private set intersection for sets with small entries. In: Boldyreva, A., Kolesnikov, V. (eds.) Public-Key Cryptography – PKC 2023, Part II. Lecture Notes in Computer Science, vol. 13941, pp. 190–220. Springer (2023). https://doi.org/10.1007/978-3-031-31371-4_7
23. Canteaut, A., Chabaud, F.: A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense bch codes of length 511. IEEE Transactions on Information Theory **44**(1), 367–378 (1998)
24. Canto Torres, R., Sendrier, N.: Analysis of information set decoding for a sublinear error weight. In: International Workshop on Post-Quantum Cryptography. pp. 144–161. Springer (2016)
25. Carozza, E., Couteau, G., Joux, A.: Short signatures from regular syndrome decoding in the head. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 532–563. Springer (2023)
26. Carrier, K., Debris-Alazard, T., Meyer-Hilfiger, C., Tillich, J.P.: Statistical decoding 2.0: Reducing decoding to LPN. In: Advances in Cryptology–ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part IV. pp. 477–507. Springer (2022)
27. Couteau, G., Rindal, P., Raghuraman, S.: Silver: silent VOLE and oblivious transfer from hardness of decoding structured LDPC codes. In: Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part III. pp. 502–534. Springer (2021)

28. Damgård, I., Park, S.: How practical is public-key encryption based on LPN and Ring-LPN? Cryptology ePrint Archive, Paper 2012/699 (2012), https://eprint.iacr.org/2012/699
29. Damgård, I., Pastro, V., Smart, N., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Advances in Cryptology–CRYPTO 2012: 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings. pp. 643–662. Springer (2012)
30. Dittmer, S., Ishai, Y., Lu, S., Ostrovsky, R.: Authenticated garbling from simple correlations. In: Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part IV. pp. 57–87. Springer (2022)
31. Dittmer, S., Ishai, Y., Lu, S., Ostrovsky, R.: Improving line-point zero knowledge: Two multiplications for the price of one. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 829–841 (2022)
32. Dittmer, S., Ishai, Y., Lu, S., Ostrovsky, R.: Improving line-point zero knowledge: Two multiplications for the price of one. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 829–841 (2022)
33. Dittmer, S., Ishai, Y., Ostrovsky, R.: Line-point zero knowledge and its applications. In: Tessaro, S. (ed.) Proc. 2nd Conference on Information-Theoretic Cryptography (ITC 2021). LIPIcs, vol. 199, pp. 5:1–5:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.ITC.2021.5
34. Dodis, Y., Kiltz, E., Pietrzak, K., Wichs, D.: Message authentication, revisited. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7237, pp. 355–374 (2012)
35. Döttling, N., Ghosh, S., Nielsen, J.B., Nilges, T., Trifiletti, R.: Tinyole: Efficient actively secure two-party computation from oblivious linear function evaluation. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS. p. 2263–2276. Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3133956.3134024, https://doi.org/10.1145/3133956.3134024
36. Esser, A., Santini, P.: Not just regular decoding: Asymptotics and improvements of regular syndrome decoding attacks. Cryptology ePrint Archive, Paper 2023/1568 (2023), https://eprint.iacr.org/2023/1568
37. Esser, A., Santini, P.: Not just regular decoding: asymptotics and improvements of regular syndrome decoding attacks. In: Annual International Cryptology Conference. pp. 183–217. Springer (2024)
38. Feneuil, T., Joux, A., Rivain, M.: Syndrome decoding in the head: Shorter signatures from zero-knowledge proofs. In: Annual International Cryptology Conference. pp. 541–572. Springer (2022)
39. Franzese, N., Katz, J., Lu, S., Ostrovsky, R., Wang, X., Weng, C.: Constant-overhead zero-knowledge for ram programs. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. pp. 178–191 (2021)
40. Ghosh, S., Nielsen, J.B., Nilges, T.: Maliciously secure oblivious linear function evaluation with constant overhead. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 629–659. Springer (2017)
41. Hazay, C., Scholl, P., Soria-Vazquez, E.: Low cost constant round MPC combining BMR and oblivious transfer. Journal of Cryptology **33**(4), 1732–1786 (2020)

42. Ishai, Y., Prabhakaran, M., Sahai, A.: Secure arithmetic computation with no honest majority. In: Theory of Cryptography Conference. pp. 294–314. Springer (2009)
43. Jain, A., Lin, H., Sahai, A.: Indistinguishability obfuscation from well-founded assumptions. Journal of the ACM **73**(1), 1–30 (2026). https://doi.org/10.1145/3785007
44. Jain, A., Krenn, S., Pietrzak, K., Tentes, A.: Commitments and efficient zero-knowledge proofs from learning parity with noise. In: Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7658, pp. 663–680 (2012)
45. Keller, M., Orsini, E., Scholl, P.: Mascot: faster malicious arithmetic secure computation with oblivious transfer. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 830–842 (2016)
46. Kiltz, E., Pietrzak, K., Venturi, D., Cash, D., Jain, A.: Efficient authentication from hard learning problems. J. Cryptol. **30**(4), 1238–1275 (2017)
47. Kim, J., Lee, C.: Reduce and Prange: Revisiting Prange's ISD for solving LPN/RSD over large fields. Cryptology ePrint Archive, Paper 2024/276 (2024), https://eprint.iacr.org/2024/276
48. 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, Paper 2025/1476 (2025), https://eprint.iacr.org/2025/1476
49. Liu, H., Wang, X., Yang, K., Yu, Y.: The hardness of LPN over any integer ring and field for PCG applications. Cryptology ePrint Archive, Paper 2022/712 (2022), https://eprint.iacr.org/2022/712
50. Liu, H., Wang, X., Yang, K., Yu, Y.: The hardness of LPN over any integer ring and field for PCG applications. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 149–179. Springer (2024)
51. Lyubashevsky, V., Masny, D.: Man-in-the-middle secure authentication schemes from LPN and weak prfs. In: Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II. Lecture Notes in Computer Science, vol. 8043, pp. 308–325 (2013)
52. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 107–124. Springer (2011)
53. May, A., Ozerov, I.: On computing nearest neighbors with applications to decoding of binary linear codes. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 203–228. Springer (2015)
54. Meurer, A.: A Coding-Theoretic Approach to Cryptanalysis. Ph.D. thesis, Ruhr-Universität Bochum (2013)
55. Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: Advances in Cryptology–CRYPTO 2011: 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings 31. pp. 465–484. Springer (2011)
56. Naor, M., Pinkas, B.: Oblivious polynomial evaluation. SIAM Journal on Computing **35**(5), 1254–1281 (2006)
57. Nielsen, J.B., Nordholt, P.S., Orlandi, C., Burra, S.S.: A new approach to practical active-secure two-party computation. In: Advances in Cryptology–CRYPTO 2012: 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings. pp. 681–700. Springer (2012)

58. Peters, C.: Information-set decoding for linear codes over $\mathfrak{U}_q$. In: International Workshop on Post-Quantum Cryptography. pp. 81–94. Springer (2010)
59. Prange, E.: The use of information sets in decoding cyclic codes. IRE Transactions on Information Theory **8**(5), 5–9 (1962)
60. Raghuraman, S., Rindal, P.: Blazing fast PSI from improved OKVS and subfield VOLE. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 2505–2517 (2022)
61. Rindal, P., Schoppmann, P.: VOLE-PSI: fast OPRF and circuit-PSI from vector-OLE. In: Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part II. pp. 901–930. Springer (2021)
62. Schoppmann, P., Gascón, A., Reichert, L., Raykova, M.: Distributed vector-OLE: improved constructions and implementation. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 1055–1072 (2019)
63. 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.) Advances in Cryptology – ASIACRYPT 2025, Part IV. Lecture Notes in Computer Science, vol. 16248, pp. 466–497. Springer (2025). https://doi.org/10.1007/978-981-95-5113-2_15
64. Wang, X., Ranellucci, S., Katz, J.: Authenticated garbling and efficient maliciously secure two-party computation. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. pp. 21–37 (2017)
65. Wang, X., Ranellucci, S., Katz, J.: Global-scale secure multiparty computation. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp. 39–56 (2017)
66. 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 (SP). pp. 1074–1091. IEEE (2021)
67. Weng, C., Yang, K., Yang, Z., Xie, X., Wang, X.: Antman: Interactive zero-knowledge proofs with sublinear communication. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 2901–2914 (2022)
68. Yang, K., Sarkar, P., Weng, C., Wang, X.: Quicksilver: Efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. pp. 2986–3001 (2021)
69. Yang, K., Wang, X., Zhang, J.: More efficient MPC from improved triple generation and authenticated garbling. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. pp. 1627–1646 (2020)
70. Yang, K., Weng, C., Lan, X., Zhang, J., Wang, X.: Ferret: Fast extension for correlated OT with small communication. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. pp. 1607–1626 (2020)
71. Yu, Y., Steinberger, J.P.: Pseudorandom functions in almost constant depth from low-noise LPN. In: Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9666, pp. 154–183 (2016)

72. Yu, Y., Zhang, J.: Cryptography with auxiliary input and trapdoor from constant-noise LPN. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9814, pp. 214–243 (2016), https://doi.org/10.1007/978-3-662-53018-4_9

A Algorithm for RSD over Binary Field

Over $\mathbb{F}_2$, the regularity of the error distribution provides additional linear relations. For any $1 \leq i \leq t$, we observe that

$$1 - \sum_{j=1}^{\beta} e_{i,j} = 0,$$

where $e_{i,j}$ is the j-th entry of a vector $\mathbf{e}_i$. Consequently, one obtains t linear equations from the regularity of $\tau_{m,t}$.

Writing $\mathbf{b} = (\mathbf{b}_1 \| \cdots \| \mathbf{b}_t)$ with $\mathbf{b}_i \in \mathbb{F}_2^{\beta}$, the attack exploits this fact as follows: for every $1 \leq i \leq t$,

$$\begin{aligned} \sum_{j=1}^{\beta} b_{i,j} &= (\sum_{j=1}^{\beta} \mathbf{a}_j^T) \cdot \mathbf{s} + \sum_{j=1}^{\beta} e_{i,j} \bmod 2 \\ &= (\sum_{j=1}^{\beta} \mathbf{a}_j^T) \cdot \mathbf{s} + 1 \bmod 2 \end{aligned}$$

Here, $b_{i,j}$ is the j-th entry of $\mathbf{b}_i$, and $\mathbf{a}_j^T$ is the j-th row vector of $\mathbf{A}$. Thus, we obtain an error-free linear equation g_i of the form

$$g_i(\mathbf{x}) = \sum_{j=1}^{\beta} b_{i,j} - 1 - (\sum_{j=1}^{\beta} \mathbf{a}_j^T) \cdot \mathbf{x}.$$

These relations can significantly strengthen the attack in the binary case. Specifically, RP aims to collect error-free equations, and the t equations obtained from the regularity of $\tau_{m,t}$ reduce the number of additional error-free equations that must be collected. Consequently, we need only $n - t$ additional error-free equations to solve RSD.

The remaining task is to obtain these $n - t$ additional error-free equations by the same RP reduction strategy, with the final target size changed from n to $n - t$.