

A Generalized ISD Framework for Large-Weight Syndrome Decoding

Shintaro Narisada¹, Hiroki Okada^{1,2},
Yusuke Aikawa², and Kazuhide Fukushima¹

¹ KDDI Research, Inc., Japan
{sh-narisada,ir-okada,ka-fukushima}@kddi.com

² The University of Tokyo, Japan
aikawa@mist.i.u-tokyo.ac.jp

Abstract. Syndrome decoding (SD) over $\mathbb{F}_2$ has long been a central problem in code-based cryptography. Large-weight syndrome decoding (LWSD) is a recent variant whose ternary case was introduced by Debris-Alazard, Sendrier and Tillich (ASIACRYPT '19) in the security analysis of the *Wave* signature. It has different hardness characteristics from the classical low-weight SD problem. However, the security analysis of LWSD is still limited, and existing work mostly focuses on ternary instances. In this work, we study information set decoding (ISD) algorithms for LWSD over non-binary fields. Building on the ternary work of Briceout et al. (SAC '19), we give a generalized ISD framework for LWSD over $\mathbb{F}_q$. The framework is formulated as a single binary merge tree that unifies splitting-based, representation-based, and hybrid ISD algorithms. In particular, it captures known large-weight decoding algorithms as special cases and allows the merge pattern to be optimized together with the other ISD parameters.

We evaluate the framework both asymptotically and concretely. For the asymptotic analysis, we give algorithmic landscapes for the whole LWSD regime and show which ISD algorithm gives the smallest asymptotic complexity for each instance. We also provide representative asymptotic complexities for several LWSD instances, including those corresponding to existing work. For the concrete analysis, we provide bit complexity estimates for the *Wave* parameter sets. These estimates remain above the claimed security levels.

Keywords: code-based cryptography · syndrome decoding · information set decoding

1 Introduction

In coding theory, syndrome decoding (SD) is a fundamental computational problem. Given a parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$, a syndrome vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$, and a target weight w , the goal is to find an error vector $\mathbf{e} \in \mathbb{F}_q^n$ of Hamming weight w such that $\mathbf{H}\mathbf{e}^\top = \mathbf{s}^\top$. The problem is known to be NP-hard [6] and is believed to remain hard even for quantum computers. Most classical code-based

schemes rely on the low-weight binary SD problem, corresponding to the regime $w \ll n$ and $q = 2$. The hardness of this regime is measured by the complexity of the best known solvers, which are based on Information Set Decoding (ISD), initiated by Prange [26] and refined in many subsequent works [11, 18, 5, 19, 7]. This problem underlies many code-based constructions, from the McEliece cryptosystem [20] to recent post-quantum schemes including HQC [21], Classic McEliece [1], and SDitH [22].

Low-weight SD over non-binary fields has also been studied mainly by generalizing these ISD algorithms. Peters generalized Stern’s ISD algorithm to $\mathbb{F}_q$ [25]. Meurer extended representation-based ISD techniques to $\mathbb{F}_q$ and observed that their improvement over simpler ISD variants decreases as q grows [23]. Gueye, Klamti, and Hirose later generalized the Becker–Joux–May–Meurer and May–Ozerov ISD algorithms to arbitrary finite fields [14]. Thus, the low-weight SD regime has a developed algorithmic literature, both over $\mathbb{F}_2$ and over $\mathbb{F}_q$.

On the other hand, it is natural to consider the large-weight syndrome decoding (LWSD) problem, where $w \approx n$. For $q = 2$, this regime is essentially equivalent to the low-weight regime. Indeed, if $\mathbf{e} \in \mathbb{F}_2^n$ has weight w , then $\mathbf{e} + \mathbf{1}$ has weight $n - w$ and syndrome $\mathbf{s} + \mathbf{1H}^\top$. Thus, a binary large-weight instance can be converted into a binary low-weight instance.

For non-binary fields, this symmetry no longer gives the same reduction. Debris-Alazard, Sendrier, and Tillich introduced large-weight syndrome decoding over $\mathbb{F}_3$ in the security analysis of the code-based signature scheme Wave [10]. As summarized in the later parameter analysis of Wave [29], its security is considered against both forgery attacks based on generic decoding and structural key attacks against the indistinguishability of generalized $(U, U + V)$ codes. In this work, we focus on the former, namely generic decoding attacks against the underlying LWSD problem. Bricout et al. [8] studied ISD algorithms for ternary LWSD and combined Wagner’s generalized birthday algorithm (GBA) [30] with partial representations. Chailloux, Debris-Alazard, and Etinski [9] considered Wagner’s GBA for more general weight functions and alphabet sizes, including classical and quantum analyses. Karpman and Lefevre [16] later studied time-memory tradeoffs for ternary LWSD using dissection-based subset-sum algorithms. These works naturally raise the following questions.

- Q1.** How does the hardness of the LWSD problem change when we increase q or vary the relative parameters k/n and w/n ?
- Q2.** What is the best ISD algorithm for each LWSD instance among splitting-based, representation-based, and hybrid algorithms?

In this paper, we address these questions by developing a generalized ISD framework for LWSD over non-binary fields.

1.1 Contributions

Our main contributions are summarized as follows.

A unified ISD framework for large-weight syndrome decoding. We formulate a generalized ISD framework for large-weight syndrome decoding over $\mathbb{F}_q$. The framework first generalizes the shifting technique used for the ternary alphabet [8]. After shifting, the information set search is performed with a flexible shifted weight $0 \leq \tilde{p} \leq k + \ell$, instead of the full-weight constraint $p = k + \ell$.

Under the shifted alphabet, we use a binary merge tree with two merge modes. In split mode, the support is partitioned into two disjoint parts. In rep mode, a parent vector is represented as the coordinate-wise sum of two child vectors. The all-split tree captures Dumer’s ISD [11] and Wagner’s generalized birthday decoding [30], while the all-rep tree captures the representation tree of BJMM ISD [5]. We call these algorithm classes **All-Split** and **All-Rep**, respectively.

Furthermore, inspired by partial representations [8], we also consider Hybrid algorithms between **All-Split** and **All-Rep**. In these algorithms, each depth can use either a concatenation merge or a representation merge. This allows us to optimize the merge pattern together with the other ISD parameters.

Asymptotic and concrete evaluation for large-weight syndrome decoding. We evaluate the proposed framework asymptotically over the large-weight regime. For representative field sizes, Fig. 4 shows which ISD algorithm among **All-Rep**, Hybrid, and **All-Split** gives the smallest asymptotic time complexity for each pair of $R = k/n$ and $W = w/n$. Table 1 reports asymptotic complexities for representative LWSD instances, including instances corresponding to existing work.

The results show that, as q increases, the polynomial-time Prange region covers a larger part of the large-weight range. In the remaining large-weight regime, **All-Split** attains the smallest asymptotic complexity in many regions, while Hybrid algorithms can be best in some intermediate regions.

For the concrete analysis, we provide bit complexity estimates for the Wave parameter sets. Table 2 reports the concrete estimates. The selected algorithms are finite-depth **All-Split** trees, and the estimates remain more than 10 bits above the corresponding reference security levels.

Artifacts. To ensure reproducibility, the code used in our experiments is publicly available at https://github.com/kr-isg-projects/generalized_lwsd.

1.2 Organization

The rest of this paper is organized as follows. Section 2 reviews preliminaries on LWSD and information set decoding. Section 3 classifies the large-weight regime where a generalized Prange algorithm does not run in polynomial time. Section 4 presents the generalized ISD framework. Section 5 gives asymptotic and concrete evaluations, including finite-length estimates for Wave parameters. Section 6 concludes the paper.

2 Preliminaries

2.1 Notation

Let q be a prime power, and let $\mathbb{F}_q$ denote the finite field with q elements. Vectors and matrices are written in bold lowercase and bold capital letters, respectively. Vectors are regarded as row vectors unless transposed. The transpose of a vector or matrix A is denoted by $A^\top$. The zero matrix and the identity matrix of size $k \times k$ are denoted by $\mathbf{O}$ and $\mathbf{I}_k$, respectively. For an integer n , we write $[1, n] := \{1, \dots, n\}$. For a vector $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{F}_q^n$, its Hamming weight is denoted by $|\mathbf{x}| := |\{i \in [1, n] : x_i \neq 0\}|$. For a set $\mathcal{A}$, its cardinality is denoted by $|\mathcal{A}|$, and sampling an element uniformly at random from $\mathcal{A}$ is denoted by $x \leftarrow \mathcal{A}$. The set of vectors of length n and Hamming weight w is denoted by $\mathcal{B}_w^n := \{\mathbf{x} \in \mathbb{F}_q^n; |\mathbf{x}| = w\}$.

2.2 Syndrome Decoding over $\mathbb{F}_q$

Definition 1 (SD problem over $\mathbb{F}_q$). Let n , k , and w be positive integers satisfying $k, w \leq n$. Given a parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$ and a syndrome $\mathbf{s} \in \mathbb{F}_q^{n-k}$, find a vector $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{H}\mathbf{e}^\top = \mathbf{s}^\top$ and $|\mathbf{e}| = w$.

An instance of the SD problem is denoted by $\text{SD}(\mathbf{H}, \mathbf{s}, w)$. Unless otherwise stated, we consider decodable random instances: $\mathbf{H}$ is chosen uniformly at random among full-rank matrices in $\mathbb{F}_q^{(n-k) \times n}$, and $\mathbf{s}$ is sampled uniformly at random from $\{\mathbf{H}\mathbf{e}^\top \mid \mathbf{e} \in \mathbb{F}_q^n, |\mathbf{e}| = w\}$. This guarantees that every instance $\text{SD}(\mathbf{H}, \mathbf{s}, w)$ has at least one solution.

For comparison, if $\mathbf{s}$ is sampled uniformly at random from $\mathbb{F}_q^{n-k}$, the expected number of weight- w solutions is given by the following proposition.

Proposition 1 ([8]). Let $\mathbf{H}$ be a random full-rank matrix in $\mathbb{F}_q^{(n-k) \times n}$, and let $\mathbf{s}$ be sampled uniformly at random from $\mathbb{F}_q^{n-k}$. Then the expected number of solutions to $\text{SD}(\mathbf{H}, \mathbf{s}, w)$ is

$$\frac{|\mathcal{B}_w^n|}{q^{n-k}} = \frac{\binom{n}{w}(q-1)^w}{q^{n-k}}.$$

Let $R := k/n$ be the code rate and $W := w/n$ be the relative weight. For $0 \leq x \leq 1$, let $H_q(x) := x \log_q(q-1) - x \log_q x - (1-x) \log_q(1-x)$ be the q -ary entropy function. Asymptotically, the threshold $\binom{n}{w}(q-1)^w = q^{n-k}$ corresponds to the equation $H_q(W) = 1 - R$. This equation has two solutions except for boundary cases. We denote the lower solution by W_{GV} , which corresponds to the classical q -ary Gilbert–Varshamov bound [15]. We denote the upper solution by W_{GV}^+ . In particular, when $0 < W < W_{\text{GV}}$ or $W_{\text{GV}}^+ < W \leq 1$, a random decodable SD instance is expected to have a unique solution. For $W_{\text{GV}} \leq W \leq W_{\text{GV}}^+$, it is expected to have multiple solutions.

In the binary setting, the following regime is referred to as *low-weight syndrome decoding* and has been extensively studied in the literature [20,21]:

$$q = 2, \quad R \in [0, 1], \quad W < \frac{1-R}{2}.$$

The regime $\frac{1-R}{2} \leq W \leq \frac{1+R}{2}$ can be solved in polynomial time by Prange's algorithm [26]. For the binary large-weight regime $\frac{1+R}{2} < W \leq 1$, an instance $\text{SD}(\mathbf{H}, \mathbf{s}, w)$ can be converted to $\text{SD}(\mathbf{H}, \mathbf{s} + \mathbf{1H}^\top, n-w)$ by replacing $\mathbf{e}$ with $\mathbf{e} + \mathbf{1}$. Thus the relative weight becomes $1 - W < (1 - R)/2$.

For $q \geq 3$, decoding over non-binary fields has also been studied, especially in the low-weight regime [25,14]. However, there are relatively few studies on the large-weight regime. In particular, existing works [8,10,16,29] mainly focus on the large-weight regime over $\mathbb{F}_3$, which is motivated by the ternary code-based signature scheme *Wave* [10]. We next recall the ISD framework used in this paper.

2.3 Information Set Decoding

Information Set Decoding (ISD) is a general framework originally developed for binary syndrome decoding and later extended to non-binary fields by Peters [25]. We review the ISD framework over $\mathbb{F}_q$ used in this paper. To this end, we first introduce two integer parameters

$$0 \leq p \leq w, \quad 0 \leq \ell < n - k.$$

Then, the following four steps are repeated until a solution is found.

1. (*Permutation*) Choose a random permutation matrix $\mathbf{P}$ and compute the column-permuted matrix $\mathbf{HP}$.
2. (*Partial Gaussian Elimination*) Transform $\mathbf{HP}$ into the systematic form

$$\bar{\mathbf{H}} := \mathbf{SHP} = \begin{pmatrix} \mathbf{I}_{n-k-\ell} & \mathbf{H}' \\ \mathbf{O} & \mathbf{H}'' \end{pmatrix},$$

where $\mathbf{H}' \in \mathbb{F}_q^{(n-k-\ell) \times (k+\ell)}$, $\mathbf{H}'' \in \mathbb{F}_q^{\ell \times (k+\ell)}$, and $\mathbf{S} \in \mathbb{F}_q^{(n-k) \times (n-k)}$ is an invertible matrix. If such a transformation is impossible, return to Step 1.

Define

$$\bar{\mathbf{s}}^\top := \mathbf{Ss}^\top, \quad \bar{\mathbf{e}} := \mathbf{eP}.$$

We can then rewrite the system $\mathbf{He}^\top = \mathbf{s}^\top$ as

$$\begin{aligned} \mathbf{He}^\top = \mathbf{s}^\top &\iff \bar{\mathbf{H}}\bar{\mathbf{e}}^\top = \bar{\mathbf{s}}^\top \\ &\iff \begin{pmatrix} \mathbf{I}_{n-k-\ell} & \mathbf{H}' \\ \mathbf{O} & \mathbf{H}'' \end{pmatrix} \begin{pmatrix} \mathbf{e}'^\top \\ \mathbf{e}''^\top \end{pmatrix} = \begin{pmatrix} \mathbf{s}'^\top \\ \mathbf{s}''^\top \end{pmatrix} \\ &\iff \begin{cases} \mathbf{e}'^\top + \mathbf{H}'\mathbf{e}''^\top = \mathbf{s}'^\top, \\ \mathbf{H}''\mathbf{e}''^\top = \mathbf{s}''^\top. \end{cases} \end{aligned} \tag{1}$$

where

$$\bar{\mathbf{e}} = (\mathbf{e}' \ \mathbf{e}''), \quad \mathbf{e}' \in \mathbb{F}_q^{n-k-\ell}, \quad \mathbf{e}'' \in \mathbb{F}_q^{k+\ell},$$

and

$$\bar{\mathbf{s}} = (\mathbf{s}' \ \mathbf{s}''), \quad \mathbf{s}' \in \mathbb{F}_q^{n-k-\ell}, \quad \mathbf{s}'' \in \mathbb{F}_q^\ell.$$

The first $n - k - \ell$ coordinates corresponding to $\mathbf{e}'$ are called the *parity part*, while the remaining $k + \ell$ coordinates corresponding to $\mathbf{e}''$ are called the *information set*. Hence, the goal is now to find $\mathbf{e}'' \in \mathbb{F}_q^{k+\ell}$ satisfying $\mathbf{H}'' \mathbf{e}''^\top = \mathbf{s}''^\top$ and $|\mathbf{e}''| = p$.

3. (*Information Set Search*) Compute a list $\mathcal{L}$ of candidates satisfying

$$\mathcal{L} \subseteq \left\{ \mathbf{e}'' \in \mathbb{F}_q^{k+\ell} \mid \mathbf{H}'' \mathbf{e}''^\top = \mathbf{s}''^\top, |\mathbf{e}''| = p \right\}.$$

This step is the core component of ISD and depends on the specific algorithm used. In general, this step is implemented via a search tree.

4. (*Parity-Part Test*) For each $\mathbf{e}'' \in \mathcal{L}$, compute

$$\mathbf{e}'^\top = \mathbf{s}'^\top - \mathbf{H}' \mathbf{e}''^\top.$$

If $|\mathbf{e}'| = w - p$, then $\bar{\mathbf{e}} = (\mathbf{e}' \ \mathbf{e}'')$ is a solution of $\text{SD}(\bar{\mathbf{H}}, \bar{\mathbf{s}}, w)$ by Eq. (1). Hence, $\bar{\mathbf{e}} \mathbf{P}^\top$ is a solution of $\text{SD}(\mathbf{H}, \mathbf{s}, w)$. If no solution is found, return to Step 1.

Complexity of ISD. First, we analyze the success probability of Step 4.

Proposition 2 ([8]). *Assume that $\mathbf{e}''$ satisfies $\mathbf{H}'' \mathbf{e}''^\top = \mathbf{s}''^\top$ and $|\mathbf{e}''| = p$ at Step 3. Let $\mathbf{e}'^\top = \mathbf{s}'^\top - \mathbf{H}' \mathbf{e}''^\top$. Then, up to a polynomial factor, the probability that $(\mathbf{e}' \ \mathbf{e}'') \mathbf{P}^\top$ is a solution to $\text{SD}(\mathbf{H}, \mathbf{s}, w)$ is given by*

$$P_{p,\ell} := \Pr[|\mathbf{e}'| = w - p \mid |\mathbf{e}''| = p] = \frac{\binom{n-k-\ell}{w-p} (q-1)^{w-p}}{\min(q^{n-k-\ell}, \binom{n}{w} (q-1)^w q^{-\ell})}. \quad (2)$$

Then, the average time complexity of ISD is given as follows.

Proposition 3 ([8]). *Assume that Step 3 constructs a candidate list $\mathcal{L}$ in time T . Then, up to a polynomial factor, the average time complexity of ISD is*

$$T \cdot \max \left(1, \frac{1}{|\mathcal{L}| \cdot P_{p,\ell}} \right). \quad (3)$$

3 Hardness Classification of the Large-Weight Regime

In this section, we recall the hardness classification of LWSD used in [10]. The classification is based on whether an instance can be solved in polynomial time by a generalization of Prange's algorithm.

The classical Prange algorithm [26] for low-weight syndrome decoding corresponds to the special case $p = 0$ and $\ell = 0$. After Gaussian elimination, it checks whether $|\mathbf{s}'| = w$.

For large-weight decoding, we relax this constraint by allowing $0 \leq p \leq w$ while keeping $\ell = 0$. This generalization can be viewed as a one-sample variant of Lee–Brickell’s algorithm [17]. For each permutation, the algorithm samples one vector $\mathbf{e}''$ of weight p in Step 3, instead of constructing a list of such vectors. The parameter p is chosen to maximize the success probability in Eq. (2). Namely,

$$p_{\text{Prange}}^* := \arg \max_p \binom{n-k}{w-p} (q-1)^{w-p},$$

where the maximum is taken over all feasible integers p satisfying

$$0 \leq p \leq \min(w, k), \quad 0 \leq w-p \leq n-k.$$

Since the algorithm samples only one candidate $\mathbf{e}''$ per iteration, we have $|\mathcal{L}| = 1$ and $T = 1$ up to polynomial factors. Then, the average time complexity of Prange’s algorithm is obtained by substituting $\ell = 0$, $p = p_{\text{Prange}}^*$, and $T = 1$ into Eq. (3), namely,

$$\max \left(1, \frac{\min \left(q^{n-k}, \binom{n}{w} (q-1)^w \right)}{\binom{n-k}{w-p_{\text{Prange}}^*} (q-1)^{w-p_{\text{Prange}}^*}} \right). \quad (4)$$

Therefore, whenever

$$\binom{n-k}{w-p_{\text{Prange}}^*} (q-1)^{w-p_{\text{Prange}}^*} \geq \min \left(q^{n-k}, \binom{n}{w} (q-1)^w \right),$$

Prange’s algorithm solves SD in polynomial time, since the expected number of iterations becomes constant. The following proposition characterizes this regime asymptotically.

Proposition 4 ([10]). *For an SD instance with code rate R and relative weight W , the generalized Prange algorithm runs in polynomial time in the following region:*

$$\frac{(q-1)(1-R)}{q} \leq W \leq \frac{q-1+R}{q}.$$

Outside this region, its expected number of iterations is exponential.

Fig. 1 illustrates this classification for $q = 3$. From Proposition 4 and the symmetry property of binary syndrome decoding, we define the cryptographically meaningful large-weight syndrome decoding regime as

$$q \geq 3 \quad \text{and} \quad \frac{q-1+R}{q} < W \leq 1. \quad (5)$$

In this region, the weight is beyond the polynomial-time interval and makes it the natural large-weight analogue of the low-weight decoding regime. The framework applies to arbitrary finite fields $\mathbb{F}_q$ with $q > 2$.

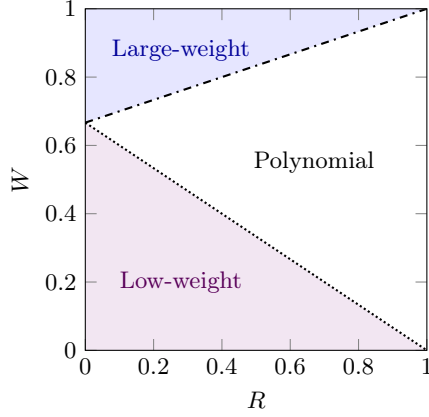


Fig. 1. Hardness classification for syndrome decoding over $\mathbb{F}_3$ in the (R, W) plane. The dash-dotted line is the boundary $W = (q - 1 + R)/q$, and the densely dotted line is the boundary $W = (q - 1)(1 - R)/q$. The white region between these two lines is the polynomial-time regime of the generalized Prange algorithm. The blue region above the dash-dotted line is the large-weight regime, and the violet region below the densely dotted line is the low-weight regime.

4 Generalized ISD Framework for Large-Weight Regime

In this section, we introduce a generalized ISD framework for the large-weight regime defined in Eq. (5). A natural case in this regime is the full-weight choice $p = k + \ell$, where the information set part is all nonzero. Then Step 3 of ISD becomes an all-nonzero search over $(\mathbb{F}_q^\times)^{k+\ell}$. This form is not well suited to the direct use of representation-based ISD algorithms such as MMT and BJMM [18,5], whose internal searches are designed for lower-weight information set vectors.

To handle this full-weight search, previous large-weight decoding algorithms use a shifting technique. This idea was used by Bricout et al. [8] for ternary large-weight syndrome decoding. The shift changes the alphabet in the information set part. For example, over $\mathbb{F}_3$, shifting by 1 maps 1 to 0 and 2 to 1. Thus, an all-nonzero search is rewritten as a shifted-weight search over an alphabet containing zero. We begin by generalizing this shifting step.

4.1 Shifting

We focus on the full-weight choice $p = k + \ell$. Then the information set part is all nonzero, and the numerator in Eq. (2) becomes $\binom{n-k-\ell}{n-w}(q-1)^{w-k-\ell}$. This choice turns Step 3 into an all-nonzero search over the information set part, while the value of ℓ is chosen by the overall tree-complexity optimization.

Thus, in Step 3, one has to search for candidates

$$\mathcal{L} \subseteq \left\{ \mathbf{e}'' \in (\mathbb{F}_q^\times)^{k+\ell} \mid \mathbf{H}'' \mathbf{e}''^\top = \mathbf{s}''^\top \right\}.$$

The shifting technique rewrites this all-nonzero search as a search over an alphabet containing zero.

Fix $x \in \mathbb{F}_q^\times$ and set $\mathbf{x} = (x, \dots, x) \in \mathbb{F}_q^{k+\ell}$. For each candidate $\mathbf{e}''$, define $\tilde{\mathbf{e}} := \mathbf{e}'' - \mathbf{x}$ and $\tilde{\mathbf{s}} := \mathbf{s}'' - \mathbf{x}\mathbf{H}''^\top$. Then

$$\mathbf{H}''\mathbf{e}''^\top = \mathbf{s}''^\top \iff \mathbf{H}''\tilde{\mathbf{e}}^\top = \tilde{\mathbf{s}}^\top.$$

Moreover, $\mathbf{e}'' \in (\mathbb{F}_q^\times)^{k+\ell}$ if and only if $\tilde{\mathbf{e}} \in (E_x^0)^{k+\ell}$, where $E_x^0 := \{a - x : a \in \mathbb{F}_q^\times\} = \mathbb{F}_q \setminus \{-x\}$. We also write $E_x := E_x^0 \setminus \{0\}$.

For a shifted weight $0 \leq \tilde{p} \leq k + \ell$, the shifted search constructs the list

$$\tilde{\mathcal{L}}_{x,\tilde{p}} \subseteq \left\{ \tilde{\mathbf{e}} \in (E_x^0)^{k+\ell} \mid \mathbf{H}''\tilde{\mathbf{e}}^\top = \tilde{\mathbf{s}}^\top, |\tilde{\mathbf{e}}| = \tilde{p} \right\}. \quad (6)$$

The original candidate is recovered by the inverse shift $\mathbf{e}'' = \tilde{\mathbf{e}} + \mathbf{x}$. Thus the full-weight information set search can be performed by choosing a shifted weight $\tilde{p}$ and constructing $\tilde{\mathcal{L}}_{x,\tilde{p}}$. In the following, we fix $x = 1$ for simplicity.

4.2 Overview of the Generalized ISD Tree

After the shifting step, the information set search is reduced to constructing the list $\tilde{\mathcal{L}}_{x,\tilde{p}}$ in Eq. (6). This is a restricted-alphabet subset-sum problem over $\mathbb{F}_q$.

Our construction generalizes the partial-representation approach of Bricout et al. [8]. We use a binary merge tree in which each level is assigned one of two modes: **split** or **rep**. In **split** mode, the support is partitioned into two disjoint parts. In **rep** mode, a parent vector is represented as the coordinate-wise sum of two child vectors over the shifted alphabet.

Thus, the all-**split** case gives a splitting tree such as the K -sum tree in Wagner's generalized birthday decoding, the all-**rep** case gives a representation-based ISD algorithm, and mixed patterns give hybrid algorithms.

4.3 Tree Construction

In Fig. 2, we present the two merge modes in the generalized ISD tree.

Level parameters. Let $a \geq 1$ be the tree depth, where level 0 is the root and level a consists of the leaves. For each transition from level i to level $i + 1$, choose a mode $\rho_i \in \{\text{split}, \text{rep}\}$, with $\rho_{a-1} = \text{split}$. We use the same merge mode ρ_i for all nodes at level i . Allowing node-dependent merge modes may further reduce the complexity, but their optimization is outside the scope of this work. Starting from $N_0 = k + \ell$ and $p_0 = \tilde{p}$, define

$$N_{i+1} = \begin{cases} N_i/2, & \rho_i = \text{split}, \\ N_i, & \rho_i = \text{rep}, \end{cases} \quad p_{i+1} = \begin{cases} p_i/2, & \rho_i = \text{split}, \\ (p_i + \varepsilon_{i+1})/2, & \rho_i = \text{rep}. \end{cases}$$

Here $\varepsilon_{i+1} = 2p_{i+1} - p_i$ denotes the number of nonzero-overlap positions at a representation merge.

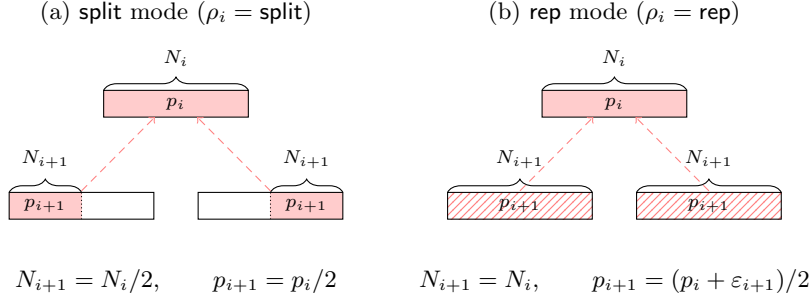


Fig. 2. Two merge modes $\rho_i \in \{\text{split}, \text{rep}\}$ in the generalized ISD tree. In **split** mode, the support is partitioned into left and right halves. In **rep** mode, both children use the same coordinate set.

We also associate each node of the tree with a coordinate set. The root node is assigned $I_1^{(0)} := [1, k + \ell]$. For a node j at level i , let $I_j^{(i)}$ denote its coordinate set, with $|I_j^{(i)}| = N_i$. If $\rho_i = \text{split}$, then $I_j^{(i)} = I_{2j-1}^{(i+1)} \sqcup I_{2j}^{(i+1)}$, where $\sqcup$ denotes a disjoint union. If $\rho_i = \text{rep}$, then $I_{2j-1}^{(i+1)} = I_{2j}^{(i+1)} = I_j^{(i)}$. For each node, $\mathbf{H}_j^{(i)}$ denotes the submatrix of $\mathbf{H}''$ formed by the columns indexed by $I_j^{(i)}$.

Finally, we choose integers $\ell_0, \dots, \ell_{a-1}$ with $\sum_{i=0}^{a-1} \ell_i = \ell$ and disjoint projections $\pi_i : \mathbb{F}_q^\ell \rightarrow \mathbb{F}_q^{\ell_i}$.

Leaf lists. At level a , for each $j \in [1, 2^a]$, construct the leaf list

$$\mathcal{L}_j^{(a)} \subseteq \left\{ (\mathbf{u}, \mathbf{y}) \mid \mathbf{u} \in (E_1^0)^{N_a}, |\mathbf{u}| = p_a, \mathbf{y} = \mathbf{H}_j^{(a)} \mathbf{u}^\top \right\}.$$

Merging. For $i = a - 1, \dots, 0$, the list $\mathcal{L}_j^{(i)}$ is obtained by merging $\mathcal{L}_{2j-1}^{(i+1)}$ and $\mathcal{L}_{2j}^{(i+1)}$. For child candidates $(\mathbf{u}_L, \mathbf{y}_L)$ and $(\mathbf{u}_R, \mathbf{y}_R)$, define

$$\mathbf{u} = \begin{cases} (\mathbf{u}_L \parallel \mathbf{u}_R), & \rho_i = \text{split}, \\ \mathbf{u}_L + \mathbf{u}_R, & \rho_i = \text{rep}, \end{cases} \quad \mathbf{y} = \mathbf{y}_L + \mathbf{y}_R.$$

The pair is kept only if $\mathbf{u} \in (E_1^0)^{N_i}$, $|\mathbf{u}| = p_i$, and $\pi_i(\mathbf{y}) = \mathbf{t}_j^{(i)}$, where the targets $\mathbf{t}_j^{(i)}$ are chosen so that $\sum_{j=1}^{2^i} \mathbf{t}_j^{(i)} = \pi_i(\tilde{\mathbf{s}})$. Thus,

$$\mathcal{L}_j^{(i)} \subseteq \left\{ (\mathbf{u}, \mathbf{y}) \mid \mathbf{u} \in (E_1^0)^{N_i}, |\mathbf{u}| = p_i, \mathbf{y} = \mathbf{H}_j^{(i)} \mathbf{u}^\top, \pi_i(\mathbf{y}) = \mathbf{t}_j^{(i)} \right\}.$$

Root list. Since the projections cover all ℓ coordinates, the root list satisfies $\mathcal{L}_1^{(0)} \subseteq \tilde{\mathcal{L}}_{1, \tilde{\mathbf{p}}}$.

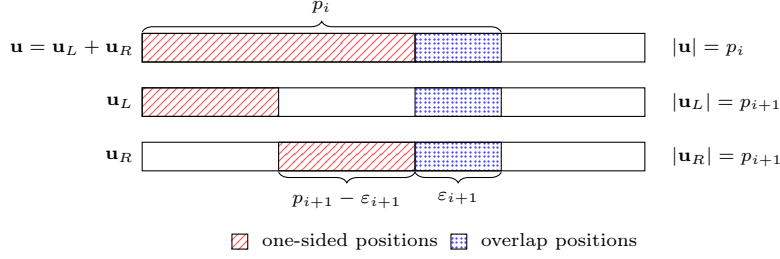


Fig. 3. Illustration of a representation merge from level $i + 1$ to level i . Red hatched positions are one-sided positions, while blue dotted positions are nonzero-overlap positions counted by ε_{i+1} .

4.4 Representation

At representation levels, a parent vector may have several decompositions as a sum of two child vectors. We first quantify the coordinate-wise additivity of the shifted alphabet. For $\gamma \in E_1$, define

$$\alpha_{E_1}(\gamma) := |\{(b, c) \in E_1 \times E_1 : b + c = \gamma\}|, \quad \bar{\alpha}_{E_1} := \frac{\sum_{\gamma \in E_1} \alpha_{E_1}(\gamma)}{|E_1|}.$$

Since $\alpha_{E_1}(\gamma)$ may depend on γ , the exact number of representations depends on the nonzero symbols of the parent vector. In the complexity model, we replace the resulting product of coordinate-wise representation counts by its average over uniformly distributed nonzero parent symbols. This gives an average-additivity heuristic. For the full large-weight setting with $E_1 = \mathbb{F}_q \setminus \{0, -1\}$, the values of $\alpha_{E_1}(\gamma)$ are explicit. For odd characteristic, one has $\alpha_{E_1}(\gamma) = q - 4$ for all $\gamma \in E_1$ except $\gamma = -2$, where $\alpha_{E_1}(-2) = q - 3$. Hence $\bar{\alpha}_{E_1} = q - 4 + 1/(q - 2)$. For characteristic two, $-2 = 0 \notin E_1$, and $\alpha_{E_1}(\gamma) = q - 4$ for every $\gamma \in E_1$. Hence $\bar{\alpha}_{E_1} = q - 4$.

Example 1. For $q = 5$, we have $E_1 = \{1, 2, 3\}$. Then $\alpha_{E_1}(1) = 1$, $\alpha_{E_1}(2) = 1$, and $\alpha_{E_1}(3) = 2$, since $1 = 3 + 3$, $2 = 1 + 1$, and $3 = 1 + 2 = 2 + 1$ over $\mathbb{F}_5$. Thus, $\bar{\alpha}_{E_1} = 4/3 = q - 4 + 1/(q - 2)$.

Example 2. For $q = 4$, let $\mathbb{F}_4 = \{0, 1, \omega, \omega + 1\}$ with $\omega^2 = \omega + 1$. Since the characteristic is two, $-1 = 1$, and hence $E_1 = \{\omega, \omega + 1\}$. Moreover, $\omega + \omega = 0$, $(\omega + 1) + (\omega + 1) = 0$, and $\omega + (\omega + 1) = 1$. Therefore, $\alpha_{E_1}(\omega) = \alpha_{E_1}(\omega + 1) = 0$, and thus $\bar{\alpha}_{E_1} = 0 = q - 4$.

Fig. 3 illustrates the support pattern used in a representation merge, including the one-sided positions and the nonzero-overlap positions counted by ε_{i+1} .

Lemma 1 (Representation count). *Consider a representation merge from level $i + 1$ to level i . Let $\mathbf{u} \in (E_1^0)^{N_i}$ be a parent vector of weight p_i . Assume that child vectors have weight $p_{i+1} = (p_i + \varepsilon_{i+1})/2$, where ε_{i+1} is the number*

of nonzero-overlap positions. Then, under the average-additivity heuristic, the number of decompositions $\mathbf{u} = \mathbf{u}_L + \mathbf{u}_R$ with $\mathbf{u}_L, \mathbf{u}_R \in (E_1^0)^{N_i}$ and $|\mathbf{u}_L| = |\mathbf{u}_R| = p_{i+1}$ is estimated by

$$r_i \approx \binom{p_i}{p_{i+1}} \binom{p_{i+1}}{\varepsilon_{i+1}} \bar{\alpha}_{E_1}^{\varepsilon_{i+1}}.$$

For split levels, we set $r_i = 1$.

Proof. First choose the p_{i+1} nonzero positions of the left child among the p_i nonzero positions of $\mathbf{u}$. Among them, choose ε_{i+1} nonzero-overlap positions, where both children are nonzero and their sum equals the parent symbol. For each such coordinate, the average number of choices is $\bar{\alpha}_{E_1}$. The remaining nonzero coordinates are represented in the one-sided forms $\gamma = \gamma + 0$ or $\gamma = 0 + \gamma$, and the zero coordinates are represented as $0 = 0 + 0$. This gives the claimed expression. For split levels, the child supports are disjoint, so each parent has a unique decomposition. $\square$

Remark 1 (Representation model). We use a support-preserving representation model. Nonzero parent coordinates may be represented by one-sided decompositions or by nonzero–nonzero overlaps, whereas zero coordinates are represented only as $0 = 0 + 0$. Thus, cancellations $0 = b + c$ with $b, c \in E_1$ are not used. Pairs whose sum equals -1 are discarded, since $-1 \notin E_1^0$.

One may also consider more general variants, such as allowing -1 in intermediate lists or allowing partially overlapping child coordinate sets. Such variants require additional parameters and accounting for sums outside E_1^0 , but did not improve the asymptotic exponents in our experiments. A more systematic exploration of such generalized representation models is left for future work.

4.5 Complexity Analysis

We now estimate the cost of the information set search performed by the generalized tree. The following complexity estimates rely on the average-additivity heuristic introduced in Section 4.4. We also assume that the projected list values are sufficiently uniformly distributed and that the list behavior across successive merges is approximately independent. Experimental validation of these assumptions is left for future work. Let $L^{(i)}$ denote the expected size of each list at level i . We also define

$$\Delta_i := \binom{N_i}{p_i} |E_1|^{p_i} = \binom{N_i}{p_i} (q-2)^{p_i},$$

which is the number of shifted vectors of length N_i and weight p_i , where the nonzero entries are chosen from E_1 . At the leaf level, we have $L^{(a)} = \Delta_a$.

Consider a merge from level $i+1$ to level i . The two child lists contain $(L^{(i+1)})^2$ pairs. After filtering on ℓ_i projection coordinates, the expected number of colliding pairs is $(L^{(i+1)})^2 q^{-\ell_i}$. Among these colliding pairs, only a fraction θ_i

is expected to yield a well-formed level- i candidate. We define

$$\theta_i := \begin{cases} 1, & \rho_i = \text{split}, \\ r_i \Delta_i / \Delta_{i+1}^2, & \rho_i = \text{rep}. \end{cases}$$

Here $\theta_i = 1$ in split mode because concatenating two well-formed child vectors always gives a well-formed parent vector. In representation mode, $r_i \Delta_i / \Delta_{i+1}^2$ estimates the fraction of pairs of well-formed child vectors whose coordinate-wise sum is a well-formed parent vector of weight p_i .

Therefore, before applying any cap, the expected list size satisfies

$$\widehat{L}^{(i)} = (L^{(i+1)})^2 q^{-\ell_i} \theta_i.$$

However, the list at level i cannot exceed the number of well-formed candidates satisfying the projection constraints already imposed. Let $A_i := \sum_{h=i}^{a-1} \ell_h$ be the number of projection coordinates fixed from level a up to level i . We therefore use the capped estimate

$$L^{(i)} = \min \left\{ \Delta_i q^{-A_i}, \widehat{L}^{(i)} \right\}.$$

In particular, at the root we have

$$L^{(0)} \leq \Delta_0 q^{-\ell} = \binom{k+\ell}{\tilde{p}} \frac{(q-2)^{\tilde{p}}}{q^\ell}.$$

Theorem 1 (Complexity of the generalized ISD framework). *Let $L^{(a)} = \Delta_a$, and for $i = a-1, \dots, 0$ define θ_i , $\widehat{L}^{(i)}$, and $L^{(i)}$ as above. Then $L^{(0)}$ is the expected size of the root list. Up to polynomial factors, the time complexity of one information set search is*

$$T_{\text{tree}} = L^{(a)} + \sum_{i=0}^{a-1} \left(L^{(i+1)} + (L^{(i+1)})^2 q^{-\ell_i} \right),$$

and the memory complexity is $M_{\text{tree}} = \max_{1 \leq i \leq a} L^{(i)}$. Consequently, by Eq. (3), the total ISD complexity is, up to polynomial factors,

$$T_{\text{ISD}} = T_{\text{tree}} \cdot \max \left(1, \frac{1}{L^{(0)} P_{k+\ell, \ell}} \right). \quad (7)$$

Proof. At level a , the base lists contain all well-formed vectors of length N_a and weight p_a , hence $L^{(a)} = \Delta_a$. Consider a merge from level $i+1$ to level i . The child lists are first scanned or organized according to their ℓ_i projected coordinates, which costs $L^{(i+1)}$ up to polynomial factors. Using hashing or sorting, the merge enumerates colliding pairs with matching projection value. Assuming that the projected values are uniformly distributed over $\mathbb{F}_q^{\ell_i}$, the expected number of colliding pairs is $(L^{(i+1)})^2 q^{-\ell_i}$. Among these pairs, a fraction θ_i is expected to give a well-formed level- i candidate. This gives $\widehat{L}^{(i)} = (L^{(i+1)})^2 q^{-\ell_i} \theta_i$.

Since the list cannot contain more than all well-formed candidates satisfying the already imposed Λ_i projection constraints, we cap it by $\Delta_i q^{-\Lambda_i}$. Summing the base-list construction cost and the merge costs over all levels gives the stated time bound.

The memory bound follows from storing the largest non-root list. Root candidates can be tested on the fly against the parity part, and hence the root list does not have to be stored. Finally, after undoing the shift, the root list is exactly the candidate list used in Eq. (3). Substituting $T = T_{\text{tree}}$ and $|\mathcal{L}| = L^{(0)}$ gives the stated ISD complexity. $\square$

Remark 2 (Special cases). The tree construction covers several known ISD algorithms as special cases. If $\rho_i = \text{split}$ for all i , $q = 3$, and $p_0 = (k + \ell)/2$, the construction recovers Wagner’s generalized birthday decoding [30], by enumerating binary vectors in $\{0, 1\}^{k+\ell}$ with a K -sum tree, where $K = 2^a$. If $a = 1$ and $\rho_0 = \text{split}$, it recovers Dumer’s ISD [11]. If $\rho_0 = \dots = \rho_{a-2} = \text{rep}$ and $\rho_{a-1} = \text{split}$, it recovers the representation tree of the BJMM algorithm [5].

5 Asymptotic and Concrete Evaluation

In this section, we evaluate the generalized ISD framework both asymptotically and concretely.

5.1 Asymptotic Landscapes

We first evaluate the framework asymptotically. For fixed q , $R = k/n$, and $W = w/n$, all internal parameters are normalized by n . We evaluate binomial coefficients using the entropy approximation $\binom{m}{xm} = 2^{mH(x)+o(m)}$, where $H(x) = -x \log_2 x - (1-x) \log_2 (1-x)$. The quantities in Section 4 are written as

$$T_{\text{tree}} = 2^{\tau_{\text{tree}}n+o(n)}, \quad L^{(i)} = 2^{\lambda^{(i)}n+o(n)}, \quad M_{\text{tree}} = 2^{\mu n+o(n)}, \quad P_{k+\ell, \ell} = 2^{\pi n+o(n)}.$$

For each point (R, W) , we optimize over the depth $a \in \{1, \dots, 7\}$, the merge pattern $\rho_i \in \{\text{split}, \text{rep}\}$, and the following normalized parameters:

$$c_{\tilde{p}}, \quad (c_{\ell_0}, \dots, c_{\ell_{a-1}}), \quad (c_{\varepsilon_1}, \dots, c_{\varepsilon_{a-1}}),$$

where $c_{\tilde{p}} = \tilde{p}/n$, $c_{\ell_i} = \ell_i/n$, $c_{\varepsilon_i} = \varepsilon_i/n$, and $\sum_{i=0}^{a-1} c_{\ell_i} = c_{\ell} = \ell/n$. Following Eq. (7), the objective is to minimize the total time exponent

$$\tau = \tau_{\text{tree}} + \max\{0, -\lambda^{(0)} - \pi\}. \quad (8)$$

In this evaluation, we search for the optimal parameters minimizing Eq. (8) over the entire LWSD regime defined in Eq. (5), namely

$$0 \leq R \leq 1, \quad \frac{q-1+R}{q} < W \leq 1,$$

using a grid precision of 0.01 for both R and W . For each grid point and each pair (a, ρ) , where $\rho = (\rho_0, \dots, \rho_{a-1})$, we perform numerical optimization using the `scipy` library from multiple initial points and keep the smallest value of τ found. Since the parameter search uses numerical optimization, the reported values are the smallest exponents found in our search and are not guaranteed to be global optima.

We classify each algorithm by its merge pattern ρ into the three classes All-Split, All-Rep, and Hybrid. The class All-Split corresponds to $\rho_0 = \dots = \rho_{a-1} = \text{split}$. The class All-Rep corresponds to $a \geq 2$, $\rho_0 = \dots = \rho_{a-2} = \text{rep}$, and $\rho_{a-1} = \text{split}$. All other patterns correspond to Hybrid. If two or more classes attain the same asymptotic exponent, we choose the simpler class in the order All-Split, All-Rep, and Hybrid.

Fig. 4 shows, for $q = 3, 5, 7, 31$, the ISD algorithm class selected at each grid point by the optimization. The colored regions above the line $W = (q - 1 + R)/q$ show the best class among All-Rep, Hybrid, and All-Split. The white region between the two lines $W = (q - 1 + R)/q$ and $W = (q - 1)(1 - R)/q$ is the polynomial-time regime of the generalized Prange algorithm. The gray region below the line $W = (q - 1)(1 - R)/q$ is the low-weight regime, which is out of scope of this paper. The red curves $W_{\text{GV}}(R)$ and $W_{\text{GV}}^+(R)$ are the lower and upper solutions of the Gilbert–Varshamov equation $H_q(W) = 1 - R$, respectively.

The results show that the selected algorithm class depends on the field size q , the code rate R , and the relative weight W . In these plots, the optimizer tends to select simpler splitting-based algorithms as q increases or W decreases.

For $q = 3$, the All-Rep algorithm gives the smallest asymptotic complexity in the high-weight, low-rate region, including the hardest point $(R, W) = (1 - \log_q(q - 1), 1) = (0.369, 1)$, consistently with the behavior reported in [8]. In contrast, for smaller weights or larger rates, All-Split algorithms often attain the smallest asymptotic complexity.

In the intermediate region, we find that Hybrid algorithms can give the smallest complexity. For example, at $(R, W) = (0.6, 1)$, a Hybrid algorithm with $a = 4$ and $\rho = (\text{split}, \text{rep}, \text{rep}, \text{split})$ achieves asymptotic complexity $2^{0.110n}$, improving on All-Split with $a = 3$, which gives $2^{0.117n}$, and on All-Rep with $a = 4$, which gives $2^{0.114n}$.

For $q \geq 5$, the All-Rep region disappears, and the optimizer mostly selects All-Split algorithms. As q grows, the large-weight regime becomes narrower and the optimized exponent decreases. In this range, All-Rep has larger asymptotic complexity than All-Split. However, Hybrid algorithms can still be beneficial for moderate q and in intermediate regions. For example, at $(q, R, W) = (7, 0.3, 1)$, Hybrid gives complexity $2^{0.120n}$ with $\rho = (\text{split}, \text{rep}, \text{rep}, \text{rep}, \text{split})$, while All-Split gives $2^{0.128n}$ and All-Rep gives $2^{0.131n}$.

To complement the region plot, Fig. 5 compares the asymptotic time exponents of the three classes along the slice $q = 3$ and $R = 0.4$. For smaller W , All-Split gives the smallest exponent. In an intermediate range of W , Hybrid gives a strictly smaller exponent than both All-Split and All-Rep. For larger W , Hybrid attains the same exponent as All-Rep.

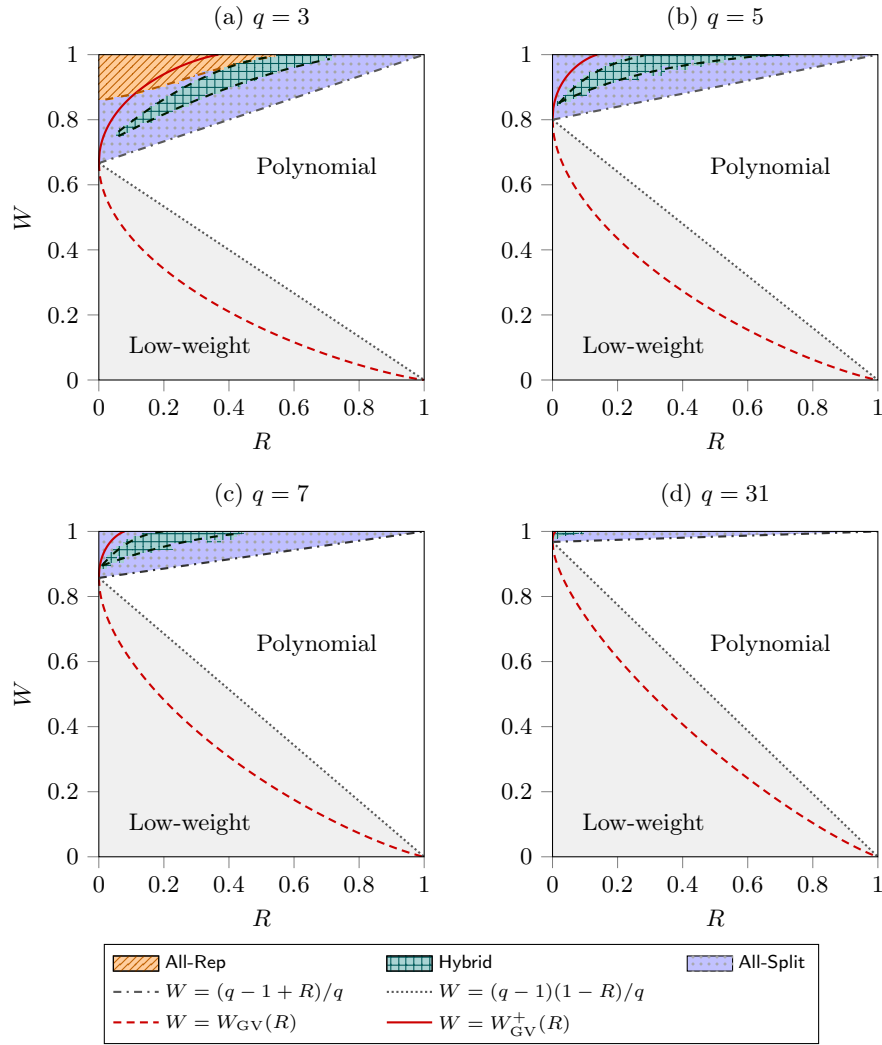


Fig. 4. ISD algorithm attaining the smallest exponent τ in the asymptotic time complexity $2^{\tau n + o(n)}$ among All-Rep, Hybrid, and All-Split, for LWSD over $\mathbb{F}_q$ with (a) $q = 3$, (b) $q = 5$, (c) $q = 7$, and (d) $q = 31$. The white region between the two lines $W = (q-1+R)/q$ and $W = (q-1)(1-R)/q$ is the polynomial-time regime. The gray region below the line $W = (q-1)(1-R)/q$ is the low-weight regime, which is out of scope of this paper. The curves $W_{GV}(R)$ and $W_{GV}^+(R)$ are the lower and upper solutions of the Gilbert–Varshamov equation $H_q(W) = 1 - R$, respectively.

Fig. 6 gives the analogous comparison for the full-weight slice $q = 5$ and $W = 1$. When $R \leq 1 - \log_5 4$, the three algorithm classes have the same asymptotic

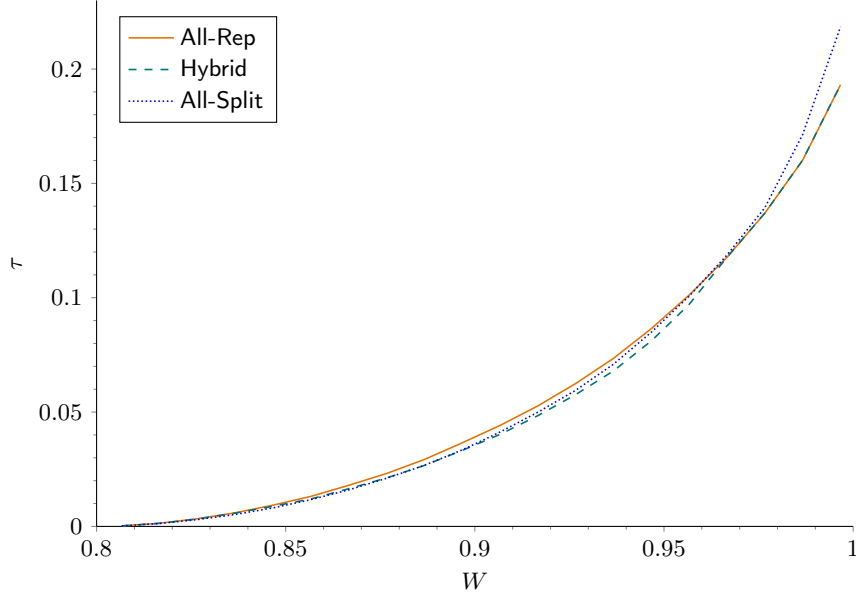


Fig. 5. Asymptotic time exponents τ for $q = 3$ and $R = 0.4$ in the asymptotic time complexity $2^{\tau n + o(n)}$ of three ISD algorithms All-Rep, Hybrid, and All-Split.

exponent. When $R > 1 - \log_5 4$, All-Rep is no better than the other two classes. In particular, Hybrid gives the smallest exponent in the range $0.3 \leq R \leq 0.7$.

Comparison with unshifted ISD. Our framework uses the shifting technique by default. We also compare it with the corresponding *unshifted* variant, where the information set search is performed directly over $(\mathbb{F}_q^\times)^{k+\ell}$ with $p = k + \ell$, instead of using the shifted weight $0 \leq \tilde{p} \leq k + \ell$.

In the unshifted coordinates, the number of information set candidates is $(q - 1)^{k+\ell}$, and its exponent is $(k + \ell) \log_2(q - 1)$. After shifting, candidates of shifted weight $\tilde{p}$ are counted by $\binom{k+\ell}{\tilde{p}} (q - 2)^{\tilde{p}}$. Writing $u = \tilde{p}/(k + \ell)$, the exponent is

$$(k + \ell) (H(u) + u \log_2(q - 2)).$$

Since $\frac{d}{du} (H(u) + u \log_2(q - 2)) = \log_2 \frac{1-u}{u} + \log_2(q - 2)$, the maximum is attained at $u = (q - 2)/(q - 1)$, where $H(u) + u \log_2(q - 2) = \log_2(q - 1)$. Thus, by setting $\tilde{p} = (k + \ell)(q - 2)/(q - 1)$, the shifted All-Split variant has the same asymptotic complexity as the unshifted All-Split.

For All-Rep and Hybrid, shifting also affects repeated representations. In the unshifted variant, the full-weight constraint can produce many duplicated parent candidates from different colliding pairs. These duplications are removed by the cap on $L^{(i)}$, but the colliding pairs still contribute to the merge cost $(L^{(i+1)})^2 q^{-\ell_i}$. After shifting, smaller shifted weights $p_i < N_i$ reduce such duplications and give smaller asymptotic exponents in our experiments.

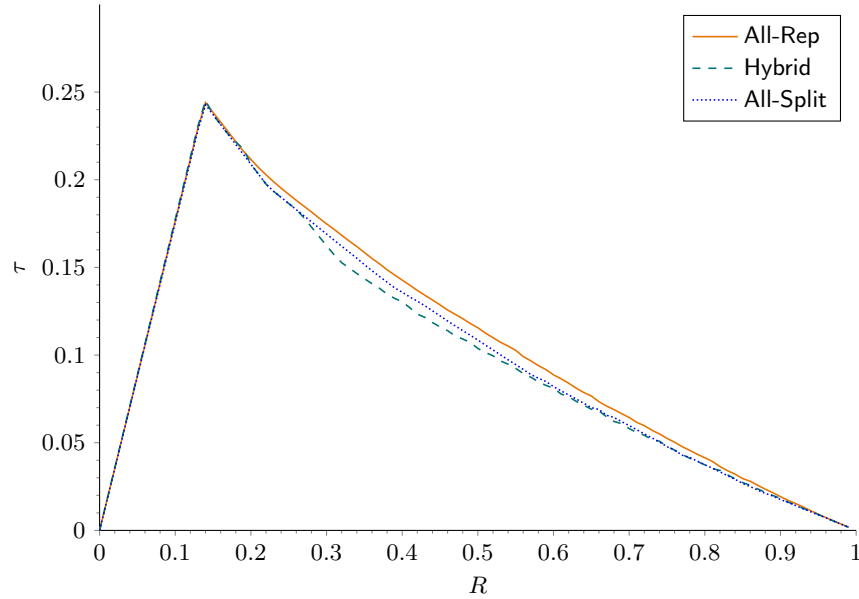


Fig. 6. Asymptotic time exponents τ for $q = 5$ and $W = 1$ in the asymptotic time complexity $2^{\tau n + o(n)}$ of three ISD algorithms All-Rep, Hybrid, and All-Split.

5.2 Asymptotic Complexities for Representative Instances

In this subsection, we present asymptotic complexities and algorithmic parameters for representative LWSO instances, including the hardest instances and the LWSO instance used in the Wave signature [4].

In the large-weight regime, the hardest points are attained on the unique-solution boundary. This boundary is W_{GV}^+ shown in Fig. 4, namely the upper solution of the Gilbert–Varshamov equation $H_q(W) = 1 - R$. For the full-weight case $W = 1$, this gives

$$R = 1 - H_q(1) = 1 - \log_q(q - 1).$$

This is consistent with the hardest ternary instances reported in [8]. The Wave signature is based on the LWSO instance with $(q, R, W) = (3, 1/2, 0.894)$, which is close to the half-gap parameter selection $W = 0.89412$ in [29].

Table 1 reports the optimized time exponent τ , memory exponent μ , and parameters for representative hardest instances with $q \in \{3, 5, 7, 31, 127\}$, together with the Wave instance.

For $q = 3$, the best time exponent is $\tau = 0.243$, obtained by All-Rep with $a = 3$. This is slightly smaller than the previous estimate $\tau = 0.247$ reported in [8, Table 4] for the depth-2 Wagner algorithm with one representation layer. The main difference is the relative shifted weight $c_{\bar{p}}/(R + c_{\ell})$, which is fixed to $1/2$ in the Wagner algorithm and optimized to 0.440 in our framework. Bricout et

Table 1. Asymptotic time exponent τ and memory exponent μ , along with algorithmic parameters for representative LWSD instances.

Type	q	R	W	Algorithm	a	$c_{\tilde{p}}$	c_{ℓ}	$\frac{c_{\tilde{p}}}{R+c_{\ell}}$	τ	μ
Hardest cases	3	0.369	1.000	All-Rep	3	0.305	0.324	0.440	0.243	0.194
	5	0.139	1.000	All-Split	2	0.183	0.104	0.753	0.244	0.122
	7	0.0792	1.000	All-Split	2	0.122	0.0676	0.831	0.190	0.0949
	31	0.00955	1.000	All-Split	2	0.0183	0.00937	0.967	0.0464	0.0232
	127	0.00163	1.000	All-Split	2	0.00323	0.00163	0.991	0.0114	0.00568
Wave [4,29]	3	0.500	0.894	All-Split	6	0.275	0.0489	0.501	0.0151	0.0151

al. [8] also considered partial representations, where splitting and representation are combined within a merge. Although this variant is not explicitly included in our framework, the above comparison suggests that its additional asymptotic improvement is limited for this instance.

For $q \geq 5$, the best time exponent is attained by All-Split. The same time exponent can be obtained with $a = 1$ or $a = 2$. The depth-1 version is Dumer’s algorithm [11] and satisfies $\mu = \tau$, while the depth-2 version keeps the same time exponent and reduces the memory exponent to $\mu = \tau/2$. It can be interpreted as Dumer’s algorithm with the Shamir–Schroeppel memory-saving technique [27]. In these cases, the optimized value satisfies $c_{\tilde{p}}/(R + c_{\ell}) \approx (q - 2)/(q - 1)$, which maximizes the number of information set candidates $\binom{k+\ell}{\tilde{p}}(q - 2)^{\tilde{p}}$.

For the Wave instance, the optimizer selects All-Split with $a = 6$ and gives $\tau = 0.0151$. This is close to the exponent 0.0150016 reported for the half-gap setting in [29, Table 1]. The small difference is expected, since [29] uses a smoothed GBA-based ISD model. More precisely, it combines an unshifted version of Wagner’s GBA [30] with the Decoding-One-Out-of-Many (DOOM) attack [28], and chooses a real-valued depth a satisfying

$$q^{\ell/a} = \left(\binom{k+\ell}{p} (q-1)^p \right)^{\frac{1}{2^a-1}}.$$

In contrast, our optimization uses an explicit integer-depth shifted All-Split tree.

5.3 Bit Security Estimates for Wave Parameters

We finally evaluate concrete bit complexities for the Wave parameters. The parameter selection in [29] is based on asymptotic estimates. For example, [29, Table 1] reports $W = 0.89412$ and $\tau = 0.0150016$, giving $n \simeq 128/\tau \simeq 8532$ for $\lambda = 128$, which is close to $n = 8576$ for Wave I. Following the bit complexity estimates in [3], we estimate the concrete bit security of the Wave parameters³.

³ The syndrome decoding estimator of [12] and `CryptographicEstimators` [13] target low-weight syndrome decoding over $\mathbb{F}_q$, and estimates for LWSD are not available.

Karpman and Lefevre [16] studied time–memory tradeoffs for ternary LWSD using dissection techniques. Their analysis considers an earlier Wave parameter regime with $R \simeq 0.676$ and $W \simeq 0.948$, whereas the parameters considered here have $R = 0.5$ and $W \simeq 0.894$. Moreover, their concrete estimates assume constant-cost memory access, while we include a logarithmic memory-access cost. Therefore, the concrete estimates are not directly comparable.

The optimizer selects All-Split for the Wave point (Table 1), so we state the concrete cost model for an a -level All-Split tree. We do not include the DOOM attack [28] in this estimate. For the Wave parameters, the smoothed GBA-based estimate of [29], which incorporates DOOM, differs from our asymptotic exponent by about 10^{-4} . This corresponds to less than two bits even at the largest concrete Wave length $n = 16512$.

Proposition 5 (Concrete cost of the All-Split tree). *Consider a Wave instance and an a -level All-Split tree after applying the shifting step. Let $N_i = (k + \ell)/2^i$, $p_i = \tilde{p}/2^i$,*

$$L_a = \binom{N_a}{p_a} (q - 2)^{p_a}, \quad L_i = L_{i+1}^2 q^{-\ell_i} \quad (i = a - 1, \dots, 0).$$

Then the memory cost in bits, namely the bit length of the stored lists, is lower-bounded by

$$M_{\text{tree}}^{\text{bit}} = \sum_{i=1}^a L_i (p_i \log_2 N_i + \ell_{i-1} \log_2 q).$$

Moreover, the bit complexity of one information set search is lower-bounded by $C_{\text{tree}} = \sum_{i=0}^a C_i$, where

$$C_i \geq \begin{cases} 2^i L_i (p_i \log_2 N_i + \ell_{i-1} \log_2 q), & 1 \leq i \leq a, \\ L_0 p_0 \log_2 N_0, & i = 0. \end{cases}$$

Including the logarithmic memory-access cost, the total bit complexity is lower-bounded by

$$T = C_{\text{tree}} \log_2 M_{\text{tree}}^{\text{bit}} \cdot \max \left\{ 1, \frac{1}{L_0 P_{k+\ell, \ell}} \right\}.$$

Proof. For $i = a, \dots, 1$, each entry of a level- i list is stored as the support of $\mathbf{u}$ together with the syndrome part $\pi_{i-1}(\mathbf{H}_j^{(i)} \mathbf{u}^\top)$ used in the next merge. Since $q = 3$ for Wave, the vector $\mathbf{u}$ is determined by its support. Thus one entry requires at least $p_i \log_2 N_i + \ell_{i-1} \log_2 q$ bits.

Following Wagner’s space-optimization technique [30], the generalized birthday step can be implemented by keeping only one list at each depth. Hence the memory cost is

$$M_{\text{tree}}^{\text{bit}} = \sum_{i=1}^a L_i (p_i \log_2 N_i + \ell_{i-1} \log_2 q).$$

The time cost is counted over all lists at each depth. In All-Split, the 2^i lists at level i have disjoint coordinate sets, so their entries cannot be reused across

nodes. The leaf lists are constructed by enumerating all shifted vectors of length N_a and weight p_a . For each such vector, the syndrome part $\pi_{a-1}(\mathbf{H}_j^{(a)} \mathbf{u}^\top)$ is computed by adding p_a columns restricted to ℓ_{a-1} coordinates. Using intermediate sums, the reusing addition technique [25] generates each p_a -column sum with an amortized constant number of column additions. Since the resulting entries also have to be stored, we get

$$C_a \geq 2^a L_a (p_a \log_2 N_a + \ell_{a-1} \log_2 q).$$

For $1 \leq i \leq a-1$, the list at each node of level i is obtained by merging two level- $(i+1)$ lists. The syndrome part of a parent entry is obtained from the two child syndrome parts, and we charge at least the cost of producing the stored entries. Since there are 2^i lists at level i , we get

$$C_i \geq 2^i L_i (p_i \log_2 N_i + \ell_{i-1} \log_2 q).$$

At the root, candidates are tested on the fly, so no syndrome part for a next merge is stored. We use

$$C_0 \geq L_0 p_0 \log_2 N_0.$$

Summing these costs over $i = 0, \dots, a$ gives $C_{\text{tree}} = \sum_{i=0}^a C_i$. Finally, the repetition factor is $\max\{1, (L_0 P_{k+\ell, \ell})^{-1}\}$ by Eq. (7), and the conservative logarithmic memory-access model [12,13] contributes the factor $\log_2 M_{\text{tree}}^{\text{bit}}$. $\square$

Table 2. Concrete complexity estimates for Wave parameters.

Instance	n	k	w	NIST [24]	Algorithm	a	$\log_2 T$	$\log_2 M_{\text{tree}}^{\text{bit}}$	$\log_2(L_0 P_{k+\ell, \ell})$
Wave I	8576	4288	7668	143	All-Split	5	158	146	0
Wave III	12544	6272	11226	207	All-Split	6	226	216	0
Wave V	16512	8256	14784	272	All-Split	6	290	277	0

Table 2 presents the results. For all three Wave parameter sets, the selected algorithm is All-Split with depth $a = 5$ or $a = 6$. The optimized points satisfy $L_0 P_{k+\ell, \ell} \approx 1$. The difference from the asymptotic results mainly comes from finite-depth factors and integrality constraints, which are hidden in the asymptotic analysis. The estimates are more than 10 bits above the NIST security levels [24] for the attacks considered in this work.

6 Conclusion

We presented a generalized ISD framework for large-weight syndrome decoding over non-binary fields. Our asymptotic and concrete evaluations identify the best algorithms within the proposed framework, which captures the known ISD

algorithms considered in this work. These results may help assess parameter choices for schemes based on this problem.

Future work includes incorporating nearest-neighbor ISD algorithms [19,7], which may be effective for smaller q . Another direction is to extend the framework to large-weight restricted syndrome decoding, which underlies the CROSS signature [2].

Acknowledgments. This work was supported by JST K Program Grant Number JPMJJP24U2, Japan.

References

1. Albrecht, M.R., Bernstein, D.J., Chou, T., Cid, C., Gilcher, J., Lange, T., Maram, V., von Maurich, I., Misoczki, R., Niederhagen, R., Paterson, K.G., Persichetti, E., Peters, C., Schwabe, P., Sendrier, N., Szefer, J., Tjhai, C.J., Tomlinson, M., Wang, W.: Classic McEliece: conservative code-based cryptography. <https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/mceliece-Round4.tar.gz> (2022)
2. Baldi, M., Barengi, A., Battagliola, M., Bitzer, S., Gianvecchio, M., Karl, P., Manganiello, F., Pavoni, A., Pelosi, G., Santini, P., Schupp, J., Signorini, E., Slaughter, F., Wachter-Zeh, A., Weger, V.: CROSS: Codes and Restricted Objects Signature Scheme. <https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-2/submission-pkg/cross-submission-round2.zip> (2025)
3. Baldi, M., Barengi, A., Battagliola, M., Bitzer, S., Gianvecchio, M., Karl, P., Manganiello, F., Pavoni, A., Pelosi, G., Santini, P., Schupp, J., Signorini, E., Slaughter, F., Wachter-Zeh, A., Weger, V.: CROSS Security Details Version 2.2. https://www.cross-crypto.com/CROSS_SecurityDetails_v2.2.pdf (2025)
4. Banegas, G., Carrier, K., Chailloux, A., Couvreur, A., Debris-Alazard, T., Gabarit, P., Karpman, P., Loyer, J., Niederhagen, R., Sendrier, N., Smith, B., Tillich, J.P.: Wave Round 1 Submission. <https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/wave-spec-web.pdf> (2023)
5. 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: EUROCRYPT 2012. pp. 520–536 (2012)
6. Berlekamp, E., McEliece, R., Van Tilborg, H.: On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory* **24**(3), 384–386 (1978)
7. Both, L., May, A.: Decoding linear codes with high error rate and its impact for LPN security. In: PQCrypto 2018. pp. 25–46 (2018)
8. Bricout, R., Chailloux, A., Debris-Alazard, T., Lequesne, M.: Ternary Syndrome Decoding with Large Weight. In: SAC. pp. 437–466. *Lecture Notes in Computer Science*, Springer (2019)
9. Chailloux, A., Debris-Alazard, T., Etinski, S.: Classical and Quantum Algorithms for Generic Syndrome Decoding Problems and Applications to the Lee Metric. In: PQCrypto. *Lecture Notes in Computer Science*, vol. 12841, pp. 44–62. Springer (2021)
10. Debris-Alazard, T., Sendrier, N., Tillich, J.: Wave: A New Family of Trapdoor One-Way Preimage Sampleable Functions Based on Codes. In: ASIACRYPT 2019. *Lecture Notes in Computer Science*, vol. 11921, pp. 21–51. Springer (2019)

11. Dumer, I.: On minimum distance decoding of linear codes. In: Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory. pp. 50–52 (1991)
12. Esser, A., Bellini, E.: Syndrome Decoding Estimator. In: Public Key Cryptography (1). Lecture Notes in Computer Science, vol. 13177, pp. 112–141. Springer (2022)
13. Esser, A., Verbel, J.A., Zweydinger, F., Bellini, E.: SoK: CryptographicEstimators - a Software Library for Cryptographic Hardness Estimation. In: AsiaCCS. ACM (2024)
14. Gueye, C.T., Klamti, J.B., Hirose, S.: Generalization of BJMM-ISD using May-Ozerov nearest neighbor algorithm over an arbitrary finite field. In: International conference on codes, cryptology, and information security. pp. 96–109. Springer (2017)
15. Huffman, W.C., Pless, V.: Fundamentals of error-correcting codes. Cambridge university press (2010)
16. Karpman, P., Lefevre, C.: Time-memory tradeoffs for large-weight syndrome decoding in ternary codes. In: Public Key Cryptography (1). pp. 82–111. Lecture Notes in Computer Science, Springer (2022)
17. Lee, P.J., Brickell, E.F.: An observation on the security of McEliece’s public-key cryptosystem. In: Workshop on the Theory and Application of Cryptographic Techniques. pp. 275–280. Springer (1988)
18. May, A., Meurer, A., Thomae, E.: Decoding Random Linear Codes in $\tilde{O}(2^{0.054n})$. In: ASIACRYPT 2011. pp. 107–124 (2011)
19. May, A., Ozerov, I.: On computing nearest neighbors with applications to decoding of binary linear codes. In: EUROCRYPT 2015. pp. 203–228 (2015)
20. McEliece, R.J.: A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report **44**, 114–116 (Jan 1978)
21. Melchor, C.A., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Bos, J., Deneuville, J.C., Dion, A., Gaborit, P., Lacan, J., Persichetti, E., Robert, J.M., Véron, P., Zémor, G.: Hamming Quasi-Cyclic (HQC) – Fourth round version (Updated version 01/10/2022). <https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/HQC-Round4.zip> (2022)
22. Melchor, C.A., Bettaieb, S., Bidoux, L., Feneuil, T., Gaborit, P., Gama, N., Gueron, S., Howe, J., Hülsing, A., Joseph, D., Joux, A., Kulkarni, M., Persichetti, E., Randrianarisoa, T.H., Rivain, M., Yue, D.: The Syndrome Decoding in the Head (SD-in-the-Head) Signature Scheme. <https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-2/submission-pkg/sdith-submission-round2.zip> (2025)
23. Meurer, A.: A Coding-Theoretic Approach to Cryptanalysis. Ph.D. thesis, Ruhr-Universität Bochum (Nov 2012)
24. National Institute of Standards and Technology: PQC Security (Updated May 12, 2025). [https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-\(evaluation-criteria\)](https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria)) (2025)
25. Peters, C.: Information-set decoding for linear codes over $\mathbb{F}_q$. In: International Workshop on Post-Quantum Cryptography. pp. 81–94. Springer (2010)
26. Prange, E.: The use of information sets in decoding cyclic codes. IRE Transactions on Information Theory **8**(5), 5–9 (1962)
27. Schroepfel, R., Shamir, A.: A $T = O(2^{n/2})$, $S = O(2^{n/4})$ Algorithm for Certain NP-Complete Problems. SIAM J. Comput. **10**(3), 456–464 (1981). <https://doi.org/10.1137/0210033>

28. Sendrier, N.: Decoding One Out of Many. In: Post-Quantum Cryptography. pp. 51–67 (2011)
29. Sendrier, N.: Wave Parameter Selection. In: Post-Quantum Cryptography. pp. 91–110. Springer Nature Switzerland, Cham (2023)
30. Wagner, D.A.: A Generalized Birthday Problem. In: CRYPTO. Lecture Notes in Computer Science, vol. 2442, pp. 288–303. Springer (2002)