Solving the Linear Equivalence Problem from Single Codeword Matching

Magali Bardet[1], Charles Brion[1], Ayoub Otmani[1], Mohamed Saeed[1], Nicolas Sendrier[2]

[1] LITIS, University of Rouen Normandie, France
[2] Inria, Paris, France
{magali.bardet,charles.brion,ayoub.otmani}@univ-rouen.fr,
saeedoon1@gmail.com, nicolas.sendrier@inria.fr

Abstract. In the Linear Code Equivalence (LCE) problem, one seeks the isometry of the Hamming space $\mathbb{F}_q^n$ relating two given linear codes. This problem is considered computationally hard for any alphabet size $q \geqslant 5$. The matching codewords framework currently serves as the benchmark for evaluating the security of cryptographic schemes whose security relies on the hardness of LCE, such as the LESS signature scheme. The framework operates by sampling multiple low-weight codewords in both target codes until matching codewords are discovered, thereby revealing the underlying isometry. Recent advancements improved this framework by introducing a new matching algorithm to decide whether two pairs of codewords match. While this technique offers better scalability than previous approaches and has led to enhanced attacks on LESS, it remains computationally intensive for certain parameter ranges. In this work, we propose a novel method to determine if a single pair of codewords matches. Our approach is based on the Schur product by an inverse vector, $\mathbf{v}^{-1} \star \mathcal{C}$, defined component-wisely. We demonstrate that if the codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are linearly equivalent and the codewords $\mathbf{v}_1$ and $\mathbf{v}_2$ match, then the resulting products $\mathbf{v}_1^{-1} \star \mathcal{C}_1$ and $\mathbf{v}_2^{-1} \star \mathcal{C}_2$ are permutation equivalent. Since the permutation equivalence problem is efficiently solvable, this provides a highly effective match-testing algorithm. By leveraging this idea, we propose several algorithms that improve the asymptotic exponent of LCE solvers for all parameters. Notably, our method reaches the optimal asymptotic exponent of the framework for a wide range of parameters. When applied to LESS, our technique substantially reduces the best-known bit complexity; for instance, reducing the security of LESS-1 from 127 to 120 bits.

1 Introduction

The problem of determining whether two linear codes are isometric under the Hamming distance –and, if so, computing an isometry between them– is a fundamental computational problem underpinning the security of several cryptographic primitives. This problem is known as the Linear Code Equivalence (LCE) problem, while its restriction to permutations is referred to as Permutation Code

Equivalence (PCE). In this paper, we introduce new algorithms that achieve improved time complexities for the LCE problem.

Our algorithms instantiate the *matching codewords* framework introduced by Beullens [6]. This methodology, which builds upon the earlier work of Leon [20], consists in computing a small invariant set of codewords for each code—such as the set of minimum-weight codewords. The isometries mapping these two sets can be recovered via exhaustive search and include those—ideally there is only one—mapping the codes. Beullens [6] improved this approach by considering larger invariant sets (e.g., codewords with weight slightly above the minimum distance). By computing only a small subset for each code, it suffices to obtain a few couples of matching codewords. While this strategy successfully broke an early version of LESS [2], the cost of the attack remains exponential, allowing a rescaling of LESS parameters.

Within the matching codewords framework, while the computation of numerous low-weight codewords is a well-understood hard problem, the identification of matching pairs remains a significant challenge. Indeed, efficiently determining whether a couple of codewords—one from each code—match often constitutes the computational bottleneck. The technique employed in [6] fails to scale for a fixed alphabet size and increasing blocklength. Although a recent matching algorithm [10] addressed this scaling issue to enable asymptotic analysis, all existing techniques [6,5,10] still require checking couples of codewords two at a time to identify a match.

In this work, we propose a novel approach based on the Schur product (the component-wise multiplication of vectors), which enables the identification of matches for a single couple of codewords in polynomial time. This result allows us to improve the asymptotic complexity for solving the LCE problem. When applied to LESS, we demonstrate that our approach outperforms the current state-of-the-art attack [10].

1.1 Related Works

Two primary approaches exist for solving the permutation code equivalence problem. The first is the support splitting algorithm [28], which solves PCE in empirical polynomial time for uniformly distributed instances. The second approach, introduced in [4], provides a reduction from PCE to the (edge-labeled) Graph Isomorphism problem, under the condition that the PCE instances have trivial hulls. Combined with Babai's landmark result [1], this reduction establishes that PCE instances with trivial hulls are provably solvable in quasi-polynomial time. This theoretical backing also reinforces the empirical success of [28], as the vast majority of linear codes have a small hull with high probability [30].

The support splitting algorithm also addresses the LCE problem for small alphabets where $q \in \{2, 3, 4\}$. However, for larger alphabets, the problem is generally considered hard for most instances [29]. Currently, two main efficient techniques exist for solving LCE: (1) the matching codewords framework and its variants [5,10], and (2) the approach proposed in [15,16] and subsequently

improved in [22], which relies on the enumeration of all information sets combined with the resolution of a matrix equivalence problem. While the former is particularly effective for small to moderate alphabet sizes, the latter becomes more advantageous as the alphabet size increases.

Worst Case. Our primary focus is the average-case complexity, as instances arising from cryptanalytic contexts are typically assumed to be uniformly distributed. It is noteworthy, however, that in the worst case, LCE and PCE are computationally equivalent [29,14]. Furthermore, PCE is harder than the Graph Isomorphism problem unless the NP hierarchy collapses [24].

1.2 Overview

We focus on uniformly distributed instances of LCE. We limit our study to $q \geqslant 5$, else [28] provides a solution, and to code rates $R = k/n \leqslant 0.5$, else we can consider the dual code, which has rate $1 - R$.

The core of our results is the $\boldsymbol{c}^{-1} \star \mathcal{C}$ construction and the reduction from LCE to PCE it allows. For any linear code $\mathcal{C}$ of length n and dimension k over the finite field $\mathbb{F}_q$ and any word $\boldsymbol{c} \in \mathbb{F}_q^n$ of Hamming weight w, the code $\boldsymbol{c}^{-1} \star \mathcal{C}$ is obtained by dividing component-wise all codewords of $\mathcal{C}$ by $\boldsymbol{c}$, the zero coordinates of $\boldsymbol{c}$ being removed. It results in a linear code of length w and of dimension at most k. If $w > k$, the dimension is almost always k and if $w \leqslant k$, the code is almost always trivial.

For all isometry φ of the Hamming space $\mathbb{F}_q^n$, the code $\boldsymbol{c}^{-1} \star \mathcal{C}$ is permutation equivalent to $\varphi(\beta \cdot \boldsymbol{c})^{-1} \star \varphi(\mathcal{C})$ for any $\beta \in \mathbb{F}_q^*$. Hence if $\mathcal{C}_1$ and $\mathcal{C}_2$ are linearly equivalent and if $\boldsymbol{c}_1 \in \mathcal{C}_1$ and $\boldsymbol{c}_2 \in \mathcal{C}_2$ match, then $\boldsymbol{c}_1^{-1} \star \mathcal{C}_1 \overset{\mathsf{PCE}}{\sim} \boldsymbol{c}_2^{-1} \star \mathcal{C}_2$ which can be efficiently tested with the support splitting algorithm [28].

If $w = |\boldsymbol{c}_1| = |\boldsymbol{c}_2| > k$ the reduced codes are non trivial, and solving the underlying PCE problem will provide enough information to recover the whole isometry. Otherwise, if $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ match, then[3] $\mathsf{Shorten}(\mathcal{C}_1, \boldsymbol{c}_1) \overset{\mathsf{LCE}}{\sim} \mathsf{Shorten}(\mathcal{C}_2, \boldsymbol{c}_2)$ and those shortened codes are non trivial and smaller, of length $n - w$ and dimension at least $k - w$. Hopefully this second LCE problem is easy enough and its cost does not exceed the cost for obtaining $(\boldsymbol{c}_1, \boldsymbol{c}_2)$. Finally, if both fail, we can still fall back to an approach with two couples of codewords, if both $(\boldsymbol{c}_1, \boldsymbol{c}_2)$ and $(\boldsymbol{c}_1', \boldsymbol{c}_2')$ match then $(\boldsymbol{c}_1 + \boldsymbol{c}_1')^{-1} \star \mathcal{C}_1 \overset{\mathsf{PCE}}{\sim} (\boldsymbol{c}_2 + \beta \cdot \boldsymbol{c}_2')^{-1} \star \mathcal{C}_2$ for some $\beta \in \mathbb{F}_q^*$. And those codes are non trivial if $|\boldsymbol{c}_1 + \boldsymbol{c}_1'| > k$.

Contributions:

- A reduction (exponential) from LCE to PCE, using the $\boldsymbol{c}^{-1} \star \mathcal{C}$ construction.
- Leon/Beullens matching codewords framework revisited.
  - Formalization of the concept of asynchronous instances, using an invariant rather than a matching algorithm.

[3] $\mathsf{Shorten}(\mathcal{C}, \boldsymbol{c}) \subset \mathcal{C}$ the codewords with zeroes on coordinates where $\boldsymbol{c}$ is non-zero.

  - Distinction between 2 types of instances, using one or two couples of matching codewords.
  - Provide the optimal asymptotic behavior for both types.
- Three new algorithms leveraging the reduction.
  - Improving the state-of-the-art asymptotic behavior for all parameters.
  - Reaching the framework's optimal asymptotic behavior for a wide range of parameters.
- Improved attack on LESS. Using the same cost metric as [10] we reduce the cost of the attack on LESS-1 from 2^{127} to 2^{120}, similarly, we reduce respectively by 10 and 17 the security bits of LESS-3 and LESS-5 compared with [10]. See Table 4 in §8.

Outline: The paper is organized as follows, we first give in §2 basic notions from coding theory. In §3 we give the reduction to PCE which is the core of our contribution. We revisit and give some new perspectives about the matching codewords framework in §4 and describe our proposed new instances in §5. The section §6 provides the technical ground for the correctness of our algorithms and §7 explains how to devise an efficient PCE solver to serve as building block. In §8 we apply our technique to improve the best know attacks on LESS.

Table 1. Notation

$\mathbb{F}_q$	the finite field with q elements
$[\![a, b[\![$	the integers i such that $a \leqslant i < b$
$\mathcal{S}_n$	the group of permutations of $[\![0, n[\![$
$\boldsymbol{v} \in \mathbb{F}_q^n$	q-ary vector are lowercase and boldface italic coordinates are indexed by $[\![0, n[\![$, $\boldsymbol{v} = (v_0, \ldots, v_{n-1})$
$\lvert\boldsymbol{v}\rvert$	the Hamming weight of $\boldsymbol{v}$
v_j or $\boldsymbol{v}(j)$	the j-th coordinate of $\boldsymbol{v}$
$\beta \cdot \boldsymbol{v}$	$\beta \cdot \boldsymbol{v} = (\beta v_i)_{0 \leqslant i < n}$ scales the vector $\boldsymbol{v} \in \mathbb{F}_q^n$ by a factor $\beta \in \mathbb{F}_q$
$\boldsymbol{u} \star \boldsymbol{v}$	$\boldsymbol{u} \star \boldsymbol{v} = (u_i v_i)_{0 \leqslant i < n}$ the componentwise (Schur) product in $\mathbb{F}_q^n$
$\boldsymbol{A} \in \mathbb{F}_q^{k \times n}$	q-ary $k \times n$ matrices are uppercase and boldface italic row and columns are indexed by $[\![0, k[\![$ and $[\![0, n[\![$ respectively
$A_{i,j}$ or $\boldsymbol{A}(i, j)$	the coefficient in i-th row and j-th column in $\boldsymbol{A}$
$\langle \boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell \rangle$	the vector space spanned by $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_\ell$
$\begin{bmatrix} n \\ k \end{bmatrix}_q$	the Gaussian binomial coefficient, counting the number of $[n, k]_q$ linear codes

2 Preliminaries

2.1 Linear codes

An $[n, k]_q$ code $\mathcal{C}$ is a linear code over $\mathbb{F}_q$ of length n and dimension k. A matrix $\boldsymbol{G} \in \mathbb{F}_q^{k \times n}$ is a *generator matrix* of $\mathcal{C}$ if $\mathcal{C} = \{\boldsymbol{x}\boldsymbol{G} \mid \boldsymbol{x} \in \mathbb{F}_q^k\}$. A matrix $\boldsymbol{H} \in$

$\mathbb{F}_q^{(n-k)\times n}$ is a *parity check matrix* of $\mathcal{C}$ if $\mathcal{C} = \{\boldsymbol{y} \in \mathbb{F}_q^n \mid \boldsymbol{y}\boldsymbol{H}^T = \boldsymbol{0}\}$. The dual $\mathcal{C}^\perp$ of $\mathcal{C}$ is spanned by any parity check matrix of $\mathcal{C}$. The *hull* of $\mathcal{C}$ is defined as $\mathcal{H}(\mathcal{C}) \triangleq \mathcal{C} \cap \mathcal{C}^\perp$. From [27, Theorem 4.19] we can derive the following result, giving the proportion $H_0(q)$ of codes with a hull reduced to $\{0\}$.

Lemma 1. *For $\mathcal{C}$ a uniformly distributed $[n, \lfloor Rn \rfloor]_q$ code, $0 < R < 1$, we have*

$$H_0(q) \triangleq \lim_{n\to\infty} \mathbb{P}(\mathcal{H}(\mathcal{C}) = \{0\}) = \prod_{i>0} \frac{1 - q^{-i}}{1 - q^{-2i}}.$$

A generator matrix is in *reduced row echelon form* if

- the leading entry in each row is 1 and is to the right of the leading entry of every row above;
- each column containing a leading 1 has zeroes in all its other entries.

It is unique for any given linear code and can be obtained in polynomial time by a Gaussian elimination.

2.2 Isometries of the Hamming Space

A *linear isometry* φ of the Hamming space $\mathbb{F}_q^n$, *a.k.a. monomial transformation*, is an element $\varphi = (\boldsymbol{v}; \pi)$ of the semidirect product $\mathbb{F}_q^{*n} \rtimes \mathcal{S}_n$ such that $\varphi(\boldsymbol{c}) = \boldsymbol{v} \star \pi(\boldsymbol{c})$ and $\pi(\boldsymbol{c}) = (c_{\pi^{-1}(i)})_{0 \leqslant i < n}$ for all $\boldsymbol{c} \in \mathbb{F}_q^n$. We denote $\varphi(\mathcal{C}) = \{\varphi(\boldsymbol{c}) \mid \boldsymbol{c} \in \mathcal{C}\}$ the action of φ on a code.

Definition 1. *Two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are linearly equivalent, denoted $\mathcal{C}_1 \overset{\mathsf{LCE}}{\sim} \mathcal{C}_2$ if there exists a linear isometry φ such that $\mathcal{C}_2 = \varphi(\mathcal{C}_1)$. They are permutation equivalent, denoted $\mathcal{C}_1 \overset{\mathsf{PCE}}{\sim} \mathcal{C}_2$ if there exists a permutation π such that $\mathcal{C}_2 = \pi(\mathcal{C}_1)$.*

When $\mathbb{F}_q$ is not a prime field, *i.e.* $q = p^m$ with p a prime and $m > 1$, there exists the slightly more general notion of semi-linear isometry, which involves the field automorphisms of $\mathbb{F}_q$. They are not considered in this work.

The LCE and PCE problems consist respectively in recovering the isometry or the permutation relating two codes.

2.3 Hamming Weight Distribution in Linear Codes

The expected number of codewords of weight w in an $[n, k]_q$ code is equal to

$$N(n, k, w) = N_w = \frac{\binom{n}{w}(q-1)^w}{q^{n-k}}. \tag{1}$$

The notation N_w will serve if there is no ambiguity. The Gilbert-Varshamov distance $d_{\mathrm{GV}}(n, k)$ is the smallest integer w such that $N_w \geqslant 1$. Asymptotically, we write $\omega = w/n$, the relative weight, and $R = k/n$, the code rate. We have

$$\nu(R, \omega) \triangleq \lim_{n\to\infty} \frac{1}{n} \log_2 N(n, nR, n\omega) = h_q(\omega) - (1 - R)\log_2 q \tag{2}$$

where $h_q(\omega) = -\omega \log_2 \frac{\omega}{q-1} - (1-\omega)\log_2(1-\omega)$ is the q-ary entropy function. The relative Gilbert-Varshamov, denoted $\delta_{\mathrm{GV}}(R)$ is the smallest $\omega > 0$ such that $\nu(R, \omega) = 0$.

2.4 Generic Decoding in Linear Codes

The decoding problem is defined for any finite field $\mathbb{F}_q$ as follows:

Problem 1 (Decoding Problem). Given $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ and $\boldsymbol{s} \in \{\boldsymbol{y}\boldsymbol{H}^T \mid \boldsymbol{y} \in \mathbb{F}_q^n, |\boldsymbol{y}| = w\}$, find $\boldsymbol{e} \in \mathbb{F}_q^n$ of weight w such that $\boldsymbol{e}\boldsymbol{H}^T = \boldsymbol{s}$.

A generic decoder solves the above problem for an input $(\boldsymbol{H}, \boldsymbol{s})$ uniformly distributed in $\mathbb{F}_q^{(n-k)\times n} \times \{\boldsymbol{y}\boldsymbol{H}^T \mid \boldsymbol{y} \in \mathbb{F}_q^n, |\boldsymbol{y}| = w\}$, with $\omega = w/n$ and $R = k/n$, the complexity of all known algorithms write as

$$T_{\mathrm{dec}}(n, nR, n\omega) = 2^{n\tau_{\mathrm{dec}}(R,\omega)(1+o(1))}. \tag{3}$$

The *decoding exponent* $\tau_{\mathrm{dec}}(R, \omega)$ differs for each algorithm. Generic decoders can be used to find codewords of given weight by setting $\boldsymbol{s} = \boldsymbol{0}$ with the same asymptotic complexity for all parameter sets.

For any fixed q and R, $0 < R < 1$, all known algorithm have an asymptotic exponent $\tau_{\mathrm{dec}}(R, \omega) > 0$ for $\omega \leqslant \frac{q-1}{q}(1 - R)$ and maximal for $\omega = \delta_{\mathrm{GV}}(R)$ the (relative) Gilbert-Varshamov distance.

In the binary case, the best decoding exponent is given by a variant of ISD (Information Set Decoding) [7] for $R > 0.42$ and a variant of statistical decoding [12] for $R \leqslant 0.42$. The q-ary case has been much less studied, when q grows the advantage of advanced ISD variant over Prange's basic ISD [25] seems to reduce [11]. A q-ary variant was proposed in [23] and was recently improved and applied to a particular case in [13].

Decoding Exponents for $\boldsymbol{\omega \geqslant \delta_{\mathrm{GV}}(R)}$. In this work we are interested in a regime of parameters where the decoded weight is larger than the Gilbert-Varshamov distance. In that case the decoding exponent for Prange is

$$\tau_{\mathrm{prange}}(R, \omega) = (1 - R)\log_2 q - (1 - R)h_q\left(\frac{\omega}{1 - R}\right). \tag{4}$$

For our purpose, Prange's algorithm will provide estimates that are qualitatively relevant. Occasionally, we will also use asymptotic exponents for other ISD variants, namely Dumer's algorithm [18] and (a variant of) MMT algorithm [21]. We will use the following asymptotic formulas, valid when the weight is above the Gilbert-Varshamov distance:

$$\tau_{\mathrm{dumer}}(R, \omega) = \min_{\rho}\left((1 - R - \lambda)\left(\log_2 q - h_q\left(\tfrac{\omega-\rho}{1-R-\lambda}\right)\right)\right)$$

where λ is implicitly defined as a function of ρ by $\lambda = \frac{R+\lambda}{2\log_2 q}h_q\left(\frac{\rho}{R+\lambda}\right)$, and

$$\tau_{\mathrm{mmt}}(R, \omega) = \min_{\rho}\left((1 - R - \lambda)\left(\log_2 q - h_q\left(\tfrac{\omega-\rho}{1-R-\lambda}\right)\right)\right)$$

where λ is implicitly defined as a function of ρ by $\lambda = \frac{R+\lambda}{\log_2 q}h_q\left(\frac{\rho/2+\varepsilon}{R+\lambda}\right)$ and $\varepsilon = \frac{(\rho/2+\varepsilon)^2}{2q(R+\lambda)}$.

2.5 Punctured and Shortened codes

The support of a word $\boldsymbol{v} \in \mathbb{F}_q^n$ is the set of its non-zero coordinates,

$$\mathrm{supp}(\boldsymbol{v}) \triangleq \{i \in [\![0, n[\![\mid v_i \neq 0\} \subset [\![0, n[\![.$$

The $[n, k]_q$ code $\mathcal{C}$ punctured or shortened in $I \subset [\![0, n[\![$, are defined as

$$\begin{aligned}\mathsf{Puncture}(\mathcal{C}, I) &\triangleq \{(c_i)_{i \notin I} \mid \boldsymbol{c} \in \mathcal{C}\},\\ \mathsf{Shorten}(\mathcal{C}, I) &\triangleq \{(c_i)_{i \notin I} \mid \boldsymbol{c} \in \mathcal{C}, \mathrm{supp}(\boldsymbol{c}) \cap I = \emptyset\}.\end{aligned}$$

For convenience, we denote $\mathsf{Puncture}(\mathcal{C}, \boldsymbol{v}) = \mathsf{Puncture}(\mathcal{C}, \mathrm{supp}(\boldsymbol{v}))$, likewise for shortening. Punctured/shortened codes have length $n - |I|$ and dimension respectively $\leqslant k$ and $\geqslant k - |I|$ with equality most of the time when $|I| < n - k$ and $|I| < k$ respectively.

2.6 Schur product

The *Schur product* of two words $\boldsymbol{u}$ and $\boldsymbol{v}$ in $\mathbb{F}_q^n$ is defined as

$$\boldsymbol{u} \star \boldsymbol{v} \triangleq (u_i v_i)_{0 \leqslant i < n}\,.$$

The *Schur inverse* of $\boldsymbol{v}$ is defined as $\boldsymbol{v}^{-1} \triangleq (v_1^{-1}, \ldots, v_{n-1}^{-1})$ with the convention that $0^{-1} = 0$. Remark that for any linear isometry $\varphi = (\boldsymbol{v}; \pi)$, we have

$$\varphi(\boldsymbol{c}^{-1}) = \boldsymbol{v} \star \pi(\boldsymbol{c}^{-1}) = \boldsymbol{v} \star \pi(\boldsymbol{c})^{-1}, \tag{5}$$

$$\varphi(\boldsymbol{c})^{-1} = \boldsymbol{v}^{-1} \star \pi(\boldsymbol{c})^{-1}. \tag{6}$$

For any $[n, k]_q$ code $\mathcal{C}$, its *Schur product* by $\boldsymbol{v}$ is defined as

$$\boldsymbol{v} \star \mathcal{C} = \boldsymbol{v} \star \mathcal{C} \triangleq \{\boldsymbol{v} \star \boldsymbol{c} \mid \boldsymbol{c} \in \mathcal{C}\}.$$

Note that $\boldsymbol{v} \star \mathcal{C}$, or $\boldsymbol{v} \star \mathcal{C}^{-1}$, is a linear code of dimension $\leqslant \min(w, k)$, where $w = |\boldsymbol{v}|$. It is almost always of dimension k if $w > k$ and almost always trivial, equivalent to $\mathbb{F}_q^w \times \{0\}^{n-w}$ if $w \leqslant k$.

The code $\boldsymbol{v} \star \mathcal{C}$, or $\boldsymbol{v} \star \mathcal{C}^{-1}$, formally has length n, but since its codewords have null coordinates outside the support of $\boldsymbol{v}$, we view it as a code of length $w = |\boldsymbol{v}|$. Hence

$$\boldsymbol{v} \star \mathcal{C} \equiv \mathsf{Puncture}(\boldsymbol{v} \star \mathcal{C}, [\![0, n[\![\setminus \mathrm{supp}(\boldsymbol{v}))$$

This convention is benign and simplifies the algorithms and statements coming later in the paper. However, the action of permutation must be clarified to avoid ambiguity.

Note About the Action of Permutations. For any $I \subset [\![0, n[\![$ of cardinality ℓ, the code $\mathcal{C}' = \mathsf{Shorten}(\mathcal{C}, I)$, or $\mathcal{C}' = \mathsf{Puncture}(\mathcal{C}, I)$, has length $n - \ell$. On a few occasion it will be convenient to lift $\pi \in \mathcal{S}_{n-\ell}$ on $\mathcal{S}_n$ or to restrict $\pi \in \mathcal{S}_n$ on $\mathcal{S}_{n-\ell}$. This is unambiguous when I is known, but leads to a uselessly cumbersome formalism. We will mention those lifting or restriction, without giving full details.

3 A Reduction from LCE to PCE

The core idea of this paper is the following reduction. From a LCE instance we are able to produce various PCE instances related to the LCE instance. This is achieved by computing Schur products between each code and some elements. Theorem 1 provides constraints on the elements for our Schur product construction to produce a related PCE instance. Theorem 2 provides, under some hypotheses, a polynomial time algorithm retrieving the solution to the original LCE instance from a solution of a related PCE instance. The analysis of the probability for an instance to satisfy the hypotheses of Theorem 2 will be conducted in §6.

Theorem 1. *Let $\varphi = (\boldsymbol{v}; \pi) \in \mathbb{F}_q^{*n} \rtimes \mathcal{S}_n$ be a monomial transformation, $\mathcal{C}_1, \mathcal{C}_2$ two $[n,k]_q$ codes and $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{F}_q^n$. If $\mathcal{C}_2 = \varphi(\mathcal{C}_1)$ and $\boldsymbol{u}_2 = \beta \cdot \varphi(\boldsymbol{u}_1)$ for any $\beta \in \mathbb{F}_q^*$ then $\boldsymbol{u}_1^{-1} \star \mathcal{C}_1$ and $\boldsymbol{u}_2^{-1} \star \mathcal{C}_2$ are permutation equivalent. Moreover, we have $\boldsymbol{u}_2^{-1} \star \mathcal{C}_2 = \pi(\boldsymbol{u}_1^{-1} \star \mathcal{C}_1)$ where here π is implicitly restricted to the support of $\boldsymbol{u_1}$.*

Proof. Let $(\boldsymbol{c_1}, \ldots, \boldsymbol{c_k})$ be a basis for $\mathcal{C}$. Then $(\varphi(\boldsymbol{c_1}), \ldots, \varphi(\boldsymbol{c_k}))$ is a basis for $\mathcal{C}_2$. Generating sets for $\boldsymbol{u}_1^{-1} \star \mathcal{C}$ and $\boldsymbol{u}_2^{-1} \star \mathcal{C}_2$ are respectively $\big(\boldsymbol{u}_1^{-1} \star \boldsymbol{c}_1, \ldots, \boldsymbol{u}_1^{-1} \star \boldsymbol{c}_k\big)$ and $\big(\boldsymbol{u}_2^{-1} \star \varphi(\boldsymbol{c}_1), \ldots, \boldsymbol{u}_2^{-1} \star \varphi(\boldsymbol{c}_k)\big)$. According to (5)–(6), from $\boldsymbol{u}_2 = \beta \cdot \boldsymbol{v} \star \pi(\boldsymbol{u}_1)$ we get $\boldsymbol{u}_2^{-1} = \beta^{-1} \cdot \boldsymbol{v}^{-1} \star \pi(\boldsymbol{u}_1^{-1})$ and for i in $[\![0, k[\![$ we have:

$$\boldsymbol{u}_2^{-1} \star \varphi(\boldsymbol{c}_i) = \beta^{-1} \cdot \boldsymbol{v}^{-1} \star \pi(\boldsymbol{u}_1^{-1}) \star \boldsymbol{v} \star \pi(\boldsymbol{c}_i) = \beta^{-1} \cdot \pi(\boldsymbol{u}_1^{-1} \star \boldsymbol{c}_i).$$

Then :

$$\big(\boldsymbol{u}_2^{-1} \star \varphi(\boldsymbol{c}_1), \ldots, \boldsymbol{u}_2^{-1} \star \varphi(\boldsymbol{c}_k)\big) = \beta^{-1} \cdot \big(\pi(\boldsymbol{u}_1^{-1} \star \boldsymbol{c}_1), \ldots, \pi(\boldsymbol{u}_1^{-1} \star \boldsymbol{c}_k)\big)$$

This shows that a generating set for $\boldsymbol{u}_2^{-1} \star \mathcal{C}_2$ can be obtained from a generating set of $\boldsymbol{u}_1^{-1} \star \mathcal{C}$ by applying the permutation π. Note that since the Schur product with $\boldsymbol{u}_1^{-1}$ is puncturing outside the positions $\mathrm{supp}(\boldsymbol{u}_1^{-1})$, only the restriction of π to those positions is involved in the equivalence.

Remark 1. The independence of $\boldsymbol{u}_1$ with respect to $\mathcal{C}_1$ in the previous proposition provides significant flexibility in selecting such an element. Indeed, the only constraint for the reduction to hold is the existence of a pair $(\boldsymbol{u}_1, \boldsymbol{u}_2) \in \mathbb{F}_q^n \times \mathbb{F}_q^n$ such that $\boldsymbol{u}_2$ is, up to a scalar factor, the image of $\boldsymbol{u}_1$ under the isometry. For instance, the reduction holds for any pair $(\boldsymbol{c}_1, \boldsymbol{c}_2) \in \mathcal{C}_1 \times \mathcal{C}_2$ satisfying $\boldsymbol{c}_2 = \varphi(\boldsymbol{c}_1)$. It similarly holds for $(\boldsymbol{h}_1^{-1}, \boldsymbol{h}_2^{-1})$ where $\boldsymbol{h}_1 \in {\mathcal{C}_1}^{\perp}$ and $\boldsymbol{h}_2 = \varphi'(\boldsymbol{h}_1) \in {\mathcal{C}_2}^{\perp}$ with $\varphi' \triangleq (\boldsymbol{v}^{-1}; \pi)$ which is an isometry[4] mapping ${\mathcal{C}_1}^{\perp}$ to ${\mathcal{C}_2}^{\perp}$. Thanks to the linearity of φ, any combination such as $\boldsymbol{c}_1 + \boldsymbol{h}_1^{-1}$ and its image under φ is also valid. In the remainder of this paper, we focus on elements $\boldsymbol{c}_1 \in \mathcal{C}_1$, as this choice enables an analysis of the conditions required to satisfy the hypotheses of the subsequent theorem. Such instances are referred to as *proper* instances hereafter.

[4] Indeed, $\varphi(\boldsymbol{h}_1^{-1}) = \boldsymbol{v} \star \pi(\boldsymbol{h}_1^{-1}) = (\boldsymbol{v}^{-1} \star \pi(\boldsymbol{h}_1))^{-1} = \varphi'(\boldsymbol{h}_1)^{-1} = \boldsymbol{h}_2^{-1}$.

To state our main theorem, we need the notion of *decomposable* codes. We admit here that the proportion of indecomposable codes is exponentially close to 1.

Definition 2 (Decomposable codes). *A code $\mathcal{C}$ is said to be* decomposable *if it is the direct sum $\mathcal{C}_1 \oplus \mathcal{C}_2$ of two codes with non intersecting support* $\mathrm{supp}(\mathcal{C}_1) \cap \mathrm{supp}(\mathcal{C}_2) = \emptyset$.

We admit the following lemma which is a corollary of [26, Prop. 2.11]

Lemma 2. *Let $\mathcal{C}$ be a non-decomposable $[n,k]_q$ code and let $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathcal{C}$ with no zero entries. If $\boldsymbol{c}_1 \star \mathcal{C} = \boldsymbol{c}_2 \star \mathcal{C}$ then $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are proportional.*

Theorem 2. *Let $(\mathcal{C}_1, \mathcal{C}_2)$ be a LCE instance having a unique solution $\mu = (\boldsymbol{v}; \pi) \in \mathbb{F}_q^{*n} \rtimes \mathcal{S}_n$. Let $I \subseteq [\![0, n[\![$ be a set of cardinal $w \geqslant k+1$ containing an information set of $\mathcal{C}$. Assume* $\mathsf{Puncture}(\mathcal{C}_1, [\![0, n[\![\backslash I)$ *is indecomposable.*

Then μ can be computed from a description of $\mathcal{C}_1, \mathcal{C}_2$ and $\{\pi(i) : i \in I\}$ in polynomial time $\mathrm{O}(w^\Omega)$ where Ω is the constant for linear algebra.

Proof. Let $\boldsymbol{G_1}, \boldsymbol{G_2}$ be any generator matrices of $\mathcal{C}_1, \mathcal{C}_2$. We denote by $\boldsymbol{P}, \boldsymbol{D}$ the respective matrices of the linear applications π and $\boldsymbol{c} \mapsto \boldsymbol{v} \star \boldsymbol{c}$. There exists an invertible matrix $\boldsymbol{S}$ verifying: $\boldsymbol{G_2} = \boldsymbol{S G_1 P D}$. Without loss of generality, thanks to the knowledge of $\{\pi(i) : i \in I\}$ we can assume $\boldsymbol{P} = \begin{pmatrix} \tilde{\boldsymbol{P}} & 0 \\ 0 & \boldsymbol{I_w} \end{pmatrix}$ where $\tilde{\boldsymbol{P}}$ is the still unknown part of $\boldsymbol{P}$ of size $(n-w) \times (n-w)$. The unknowns of the system are $\boldsymbol{S}, \tilde{\boldsymbol{P}}$ and $\boldsymbol{D}$ while $\boldsymbol{G_1}$ and $\boldsymbol{G_2}$ are known. We can split the matrices $\boldsymbol{D}, \boldsymbol{G_1}, \boldsymbol{G_2}$ into two blocks of $n-w$ and w columns as follows:

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{D_1} & 0 \\ 0 & \boldsymbol{D_2} \end{pmatrix}, \boldsymbol{G_1} = \begin{pmatrix} \boldsymbol{G_1^{(1)}} & \boldsymbol{G_1^{(2)}} \end{pmatrix}, \boldsymbol{G_2} = \begin{pmatrix} \boldsymbol{G_2^{(1)}} & \boldsymbol{G_2^{(2)}} \end{pmatrix}.$$

This leads to the following system:

$$\boldsymbol{G_2} = \boldsymbol{S G_1 P D} \iff \begin{cases} \boldsymbol{G_2^{(1)}} = \boldsymbol{S G_1^{(1)} \tilde{P} D_1}, \\ \boldsymbol{G_2^{(2)}} = \boldsymbol{S G_1^{(2)} D_2}. \end{cases}$$

To recover $\boldsymbol{D_2}$ we can get rid of $\boldsymbol{S}$ by introducing $\boldsymbol{H_0^{(2)}}$ a kernel matrix of $\boldsymbol{G_2^{(2)}}$ which satisfies the linear system

$$\boldsymbol{G_1^{(2)} D_2 (H_0^{(2)})^\top} = 0. \tag{7}$$

This system has a unique non zero solution $\boldsymbol{D_2}$. Indeed, for any such solution $\boldsymbol{D_2}$, we can uniquely solve in $\boldsymbol{S}$ the equation $\boldsymbol{G_2^{(2)}} = \boldsymbol{S G_1^{(2)} D_2}$ since $rk(\boldsymbol{G_1^{(2)}}) = rk(\boldsymbol{G_2^{(2)}}) = k$ as I, the support of $\boldsymbol{G_1^{(2)}}$ contains an information set. This equality can be rewritten as $\boldsymbol{S^{-1} G_2^{(2)}} = \boldsymbol{G_1^{(2)} D}_2$. According to Lemma 2, the assumption $\boldsymbol{G_1^{(2)}}$ being indecomposable ensures the uniqueness of $\boldsymbol{D_2}$ and thus the uniqueness of $\boldsymbol{S}$.

The only part of the secret which stays unknown is $\tilde{\boldsymbol{Q}} \triangleq \tilde{\boldsymbol{P}}\boldsymbol{D}$ which is solution to the system:

$$\boldsymbol{G}_{\mathbf{2}}^{(\mathbf{1})} = \boldsymbol{S}\boldsymbol{G}_{\mathbf{1}}^{(\mathbf{1})}\tilde{\boldsymbol{Q}}$$

$\tilde{\boldsymbol{Q}}$ can be found by using monomial properties. Indeed, $\boldsymbol{S}$ being known, $\boldsymbol{G}_{\mathbf{2}}^{(\mathbf{1})}$ and $\boldsymbol{S}\boldsymbol{G}_{\mathbf{1}}^{(\mathbf{1})}$ possess the same columns up to permutation and multiplication by a scalar. Assuming several matches for one column of $\boldsymbol{G}_{\mathbf{2}}^{(\mathbf{1})}$ arise, it would imply that several columns of $\boldsymbol{G}_{\mathbf{2}}^{(\mathbf{1})}$ are proportional and thus, the automorphism group of $\boldsymbol{G}_{\mathbf{1}}^{(\mathbf{1})}$ would not be trivial as the one of $\boldsymbol{G}_{\mathbf{2}}$ which implies several solutions to the LCE instance. This means that each column of $\boldsymbol{S}\boldsymbol{G}_{\mathbf{1}}^{(\mathbf{1})}$ can be uniquely matched to a column of $\boldsymbol{G}_{\mathbf{2}}^{(\mathbf{1})}$ and $\tilde{\boldsymbol{Q}}$ can be uniquely recovered.

We have shown that the retrieving the monomial transformation only involves linear algebra operations on matrices of size at most w.

4 The Codeword Matching Framework for Solving Code Equivalence

Definition 3. *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ denote two $[n,k]_q$ codes. The pair $(\boldsymbol{c}_1, \boldsymbol{c}_2) \in (\mathbb{F}_q^n)^2$ is a* matching couple of codewords for $(\mathcal{C}_1, \mathcal{C}_2)$ *if $(\boldsymbol{c}_1, \boldsymbol{c}_2) \in \mathcal{C}_1 \times \mathcal{C}_2$ and there exists and isometry φ such that $\mathcal{C}_2 = \varphi(\mathcal{C}_1)$ and $\boldsymbol{c}_2 = \beta \cdot \varphi(\boldsymbol{c}_1)$ for some $\beta \in \mathbb{F}_q^*$.*

When there is no ambiguity, we may speak simply of "matching codewords". This shortcut implicitly assumes a couple of codewords belonging to equivalent codes and an isometry which simultaneously relates the codes and the codewords.

4.1 Generic Framework

We consider the following idealized framework for code equivalence which depends on two parameters, the number of codewords L and their weight w:

1. Compute L codewords of weight w in both codes, say $\mathcal{L}_1 \times \mathcal{L}_2 \subset \mathcal{C}_1 \times \mathcal{C}_2$.
2. Find a set $\mathcal{M} \subset \mathcal{L}_1 \times \mathcal{L}_2$ of matching couples of codewords for $(\mathcal{C}_1, \mathcal{C}_2)$.
3. From $\mathcal{M}$ reconstruct the isometry φ.

The weight w must be such that there are sufficiently many codewords of that weight. In particular $w \geqslant d_{\mathrm{GV}}$ the Gilbert-Varshamov distance. The first step has an exponential cost which is well understood, codewords of prescribed weight can be obtained with generic decoding technique. The final reconstruction step is easily achieved from only a few matching couples of codewords, moreover in many instances, as in [10] or all those proposed here, the isometry comes as a side product of the matching step. In contrast, efficiently discovering the matching codewords is not straightforward and is the key feature to properly instantiate the framework.

4.2 Type 1: Asynchronous Instances with a Single Matching Couple

An approach to deal with the matching problem of the second step is to index the codewords found at the first step with an invariant. In this context, by invariant, we mean a mapping Inv() taking as argument a word of $\mathbb{F}_q^n$ and a q-ary linear code of length n and such that for all matching couple of codewords $(\boldsymbol{u}_1, \boldsymbol{u}_2)$, we have $\mathrm{Inv}(\boldsymbol{u}_1, \mathcal{C}_1) = \mathrm{Inv}(\boldsymbol{u}_2, \mathcal{C}_2)$. The framework is modified as follows:

1. Compute L codewords of weight w in both codes, say $\mathcal{L}_1 \times \mathcal{L}_2 \subset \mathcal{C}_1 \times \mathcal{C}_2$.
2. (a) For all $\boldsymbol{u}_1 \in \mathcal{L}_1$ add $(\boldsymbol{u}_1, \mathrm{Inv}(\boldsymbol{u}_1, \mathcal{C}_1))$ in a data structure.
   (b) For all $\boldsymbol{u}_2 \in \mathcal{L}_2$, if $(\boldsymbol{u}_1, \mathrm{Inv}(\boldsymbol{u}_2, \mathcal{C}_2))$ is found in the data structure, add $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ in $\mathcal{M}$.
3. From $\mathcal{M}$ reconstruct the isometry φ.

This approach only requires the computation of $2L$ invariants. The downside, if the invariant is not discriminant enough, is a possibility of *false positive*, that is a non matching couple of codewords with the same invariant.

Asynchronous vs. *Synchronous.* Instead, a synchronous instance would require a matching function, assuming it is available, and the matching step would be:

2. For all $(\boldsymbol{u}_1, \boldsymbol{u}_2) \in \mathcal{L}_1 \times \mathcal{L}_2$, if $\mathrm{Match}(\boldsymbol{u}_1, \boldsymbol{u}_2, \mathcal{C}_1, \mathcal{C}_2)$, add $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ in $\mathcal{M}$.

which requires L^2 match checking.

4.3 Type 2: Asynchronous Instances with Two Matching Couples

To improve its discriminability, the invariant may take a pair codewords as input and be such that $\mathrm{Inv}(\boldsymbol{u}_1, \boldsymbol{v}_1, \mathcal{C}_1) = \mathrm{Inv}(\boldsymbol{u}_2, \boldsymbol{v}_2, \mathcal{C}_2)$ for every pair of matching couples of codewords $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $(\boldsymbol{v}_1, \boldsymbol{v}_2)$. The framework is modified as follows:

1. Compute L codewords of weight w in both codes, say $\mathcal{L}_1 \times \mathcal{L}_2 \subset \mathcal{C}_1 \times \mathcal{C}_2$.
2. (a) For all $(\boldsymbol{u}_1, \boldsymbol{v}_1) \in \mathcal{L}_1^2$, add $(\boldsymbol{u}_1, \boldsymbol{v}_1, \mathrm{Inv}(\boldsymbol{u}_1, \boldsymbol{v}_1, \mathcal{C}_1))$ in a data structure.
   (b) For all $(\boldsymbol{u}_2, \boldsymbol{v}_2) \in \mathcal{L}_2^2$, if $(\boldsymbol{u}_1, \boldsymbol{v}_1, \mathrm{Inv}(\boldsymbol{u}_2, \boldsymbol{v}_2, \mathcal{C}_2))$ is found in the data structure, add $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $(\boldsymbol{v}_1, \boldsymbol{v}_2)$ in $\mathcal{M}$.
3. From $\mathcal{M}$ reconstruct the isometry φ.

It requires the computation of $2\binom{L}{2}$ invariants. The synchronous counterpart would feature a matching algorithm taking as input two couples of codewords, to be called $\binom{L}{2}^2$ times.

4.4 Optimal Instances – Complexity Analysis and Optimization

To estimate complexity and choose the optimization parameters (w, L), we consider the following simplifications:

- fixed weight codewords are sampled uniformly and independently,
- all costs besides decoding and invariant/match computation are negligible,
- the framework succeeds if there are matching codewords in $\mathcal{L}_1 \times \mathcal{L}_2$.

This is essentially verified for all existing instances and for most inputs (structured codes defeat the process, but are unlikely). We assume that, unless specified otherwise, those simplifications hold in the rest of the paper.

Proper Instances. In practical instances, the following may happen:

- some calls to the invariant (or to the matching algorithm) fail,
- false positives hinder the algorithm or the isometry reconstruction.

Instances of the framework may incur a penalty because of the above behavior. We refer to *proper instances*, instances for which this penalty can be neglected.

In practice, this properness assumption may hold only for a limited range of parameters. Or the penalty might be a small constant factor in the complexity. This must be assessed for each particular instance.

Complexity. We will denote below by $T_{\mathrm{dec}}(n,k,w)$ the average running time for sampling a codeword of weight w in an $[n,k]_q$ code, and by $T_{\mathrm{inv}}(n,k,w)$ the average running time for computing an invariant in asynchronous instances of type $s \in \{1,2\}$. We denote $P^{(s)}_{\mathrm{match}}(n,k,w,L)$ the probability to obtain at least s matching couples of codewords when L codewords of weight w are drawn uniformly and independently in two equivalent $[n,k]_q$ codes.

Proposition 1 (Complexity for Proper Asynchronous Instances). *We consider proper instances of the framework for $[n,k]_q$ codes with parameter (w,L) and $w > d_{\mathrm{GV}}(n,k)$. We define*

$$T^{(1)}_{\mathrm{LCE}}(n,k,w,L) = L \cdot T_{\mathrm{dec}}(n,k,w) + L \cdot T_{\mathrm{inv}}(n,k,w), \tag{8}$$

$$T^{(2)}_{\mathrm{LCE}}(n,k,w,L) = L \cdot T_{\mathrm{dec}}(n,k,w) + \binom{L}{2} \cdot T_{\mathrm{inv}}(n,k,w). \tag{9}$$

If the input codes are equivalent and uniformly distributed, the average running time for a proper type s instance, $s \in \{1,2\}$, is equal to $2T^{(s)}_{\mathrm{LCE}}(n,k,w,L)$ with a success probability of $P^{(s)}_{\mathrm{match}}(n,k,w,L)$.

Optimization. The ratio $T^{(s)}_{\mathrm{LCE}}(n,k,w,L)/P^{(s)}_{\mathrm{match}}(n,k,w,L)$, $s \in \{1,2\}$, must be minimized over all choices of (w,L). When $w > d_{\mathrm{GV}}(n,k)$, the probability $P^{(s)}_{\mathrm{match}}(n,k,w,L)$ is well approximated by $1 - \left(1 - \frac{q-1}{s \cdot N_w}\right)^{L^2}$ where $N_w = N(n,k,w)$ is the expected number of codewords of weight w in an $[n,k]_q$ code. The optimal value of L, up to a small constant factor, is $\sqrt{\frac{s \cdot N_w}{q-1}}$, for which the probability is $\approx 1 - \exp(-1) \approx 0.63$. The formula for w_{opt} minimize T_{LCE} for this choice of L and depends of the exact expression of $T_{\mathrm{dec}}(n,k,w)$ and $T_{\mathrm{inv}}(n,k,w)$ for each particular instance. For all q,n,k and $s \in \{1,2\}$, we define the optimal weight as

$$w^{(s)}_{\mathrm{opt}}(n,k) \triangleq \underset{w > d_{\mathrm{GV}}(n,k)}{\operatorname{argmin}} T^{(s)}_{\mathrm{LCE}}\left(n,k,w,\sqrt{\frac{s \cdot N_w}{q-1}}\right). \tag{10}$$

Note that the optimal value of (w,L) might be different from $\left(w^{(s)}_{\mathrm{opt}}, \sqrt{\frac{s \cdot N_w}{q-1}}\right)$, but will be close enough to be meaningful.

Optimal Asymptotic Complexity.

Definition 4. *For any quantity $T(n)$ growing exponentially with n, we define its* asymptotic exponent *as $\lim_{n\to\infty} \frac{1}{n}\log_2 T(n)$, if this limit exists.*

The asymptotic exponent of $T(n) = 2^{(n+o(n))\tau}$ is equal to τ. The asymptotic exponent of an algorithm is the asymptotic exponent of its average running time.

To define the optimal exponents of the framework, we further assume that the invariant can be computed in polynomial time and we admit that, for a given q and w the optimal value of L is proportional to $\sqrt{N(n,k,w)}$. We denote $R = k/n$ and $\omega = w/n$, and take $L = \sqrt{N(n, nR, n\omega)}$. The asymptotic exponents of $T_{\mathrm{dec}}(n, nR, n\omega)$ and $N(n, nR, n\omega)$ are denoted $\tau_{\mathrm{dec}}(R,\omega)$ and $\nu(R,\omega)$. We define the optimal (relative) weight as the value of ω which minimizes the asymptotic exponent for instances of type 1 and 2. They are respectively equal to

$$\omega^{(1)}_{\mathrm{opt}}(R) = \underset{\omega \geqslant \delta_{\mathrm{GV}}(R)}{\operatorname{argmin}} \frac{1}{2}\nu(R,\omega) + \tau_{\mathrm{dec}}(R,\omega), \tag{11}$$

$$\omega^{(2)}_{\mathrm{opt}}(R) = \underset{\omega \geqslant \delta_{\mathrm{GV}}(R)}{\operatorname{argmin}} \max\left(\frac{1}{2}\nu(R,\omega) + \tau_{\mathrm{dec}}(R,\omega), \nu(R,\omega)\right). \tag{12}$$

We define the optimal asymptotic exponent for solving LCE for type $s \in \{1,2\}$ in the framework as the asymptotic exponent of $T^{(s)}_{\mathrm{LCE}}(n, nR, n\omega, L)$ when $\omega = \omega^{(s)}_{\mathrm{opt}}(R)$ and $L = 2^{\frac{n}{2}\nu(R,\omega)}\ \tau^{(1)}_{\mathrm{LCE}}(R)$.

Proposition 2 (Best Asymptotic Exponents for Proper Instances). *The asymptotic exponent of any proper type 1 or 2 instance of the framework with an invariant running in polynomial time is equal respectively to*

$$\tau^{(1)}_{\mathrm{LCE}}(R) = \frac{1}{2}\nu(R, \omega^{(1)}_{\mathrm{opt}}(R)) + \tau_{\mathrm{dec}}(R, \omega^{(1)}_{\mathrm{opt}}(R)) \tag{13}$$

$$\tau^{(2)}_{\mathrm{LCE}}(R) = \max\left(\frac{1}{2}\nu(R, \omega^{(2)}_{\mathrm{opt}}(R)) + \tau_{\mathrm{dec}}(R, \omega^{(2)}_{\mathrm{opt}}(R)), \nu(R, \omega^{(2)}_{\mathrm{opt}}(R))\right) \tag{14}$$

The exact optimization depends on the decoding exponent τ_{dec}. The study is qualitatively the same for all choice of generic decoding algorithm. Below, and later unless specified otherwise, we use Prange's asymptotic exponent as in (4). The exponents for the two types are very close for small alphabets, in particular when rate is close to 0.5. For instance, for $q = 5$, $\tau^{(1)}_{\mathrm{LCE}}(0.5) = 0.233361$ and $\tau^{(2)}_{\mathrm{LCE}}(0.5) = 0.233389$. The difference is larger when the alphabet grows, see Figure 1. The first exponent $\tau^{(1)}_{\mathrm{LCE}}(R)$ is the best that can be achieved in the framework, it is given in [10] where it is referred to as a lower bound. The second exponent $\tau^{(2)}_{\mathrm{LCE}}(R)$ is a lower bound for all instances using two matching couples of codewords. In Figure 2, we observe that the decoder choice has more impact for a small alphabet ($q = 5$) than for a large one ($q = 127$).

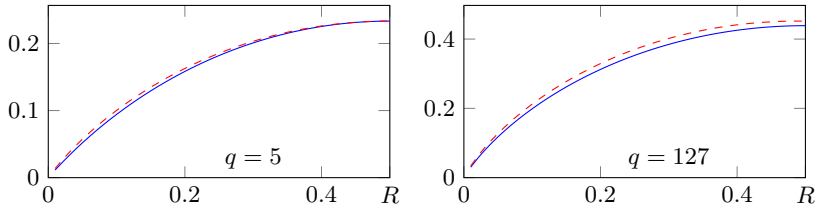


Fig. 1. Framework's Optimal Asymptotic Exponents for $q \in \{5, 127\}$ with Prange's Decoding Algorithm and $0 < R \leqslant 0.5$. Type 1 in blue, type 2 in dashed red

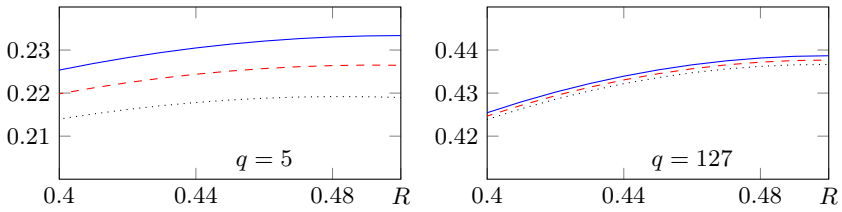


Fig. 2. Framework's (Type 1) Optimal Asymptotic Exponents for $q \in \{5, 127\}$ and $0.4 \leqslant R \leqslant 0.5$. Decoder is Prange (blue), Dumer (red, dashed) or MMT (black, dotted)

4.5 Connection with the State-of-the-Art

First Instances. The framework was introduced in [6], in its asynchronous form. A type 1 instance was proposed for permutation equivalence, using as invariant the multiset of coordinates up to a scalar multiple, denoted $\mathbf{lex}(\boldsymbol{u})$. For linear equivalence, the algorithm proposed in [6] is close to type 2, but instead of pairs of codewords, 2-dimensional subcodes[5] V with a support of size w are considered. The invariant, denoted $\mathbf{lex}(V)$, is the smallest basis of $\mu(V)$ over all choices of isometries μ. In both cases, the invariant typically spreads over a number of values polynomial in n (though exponential in q). This induces a condition on the parameters to limit the proportion of false positives, and this technique does not scale for fixed q and growing length. Hence, asymptotic exponents (for fixed q) are not meaningful. Still, this approach is well suited to the LESS signature scheme for which the alphabet size grows with the blocklength, setting a reference security standard for signatures based on code equivalence.

A New Matching Algorithm. In [10] a new approach to find pairs of matching couples of codewords is considered. Two weight-w codewords of $\mathcal{C}_1$ are used to identify a set of coordinates close to an information set of $\mathcal{C}_1$. Two codewords in $\mathcal{C}_2$ match with those two codewords of $\mathcal{C}_1$ if they identify the matching information set of $\mathcal{C}_2$, something that can be checked in polynomial time, thanks to LRL equivalence [16,22]. This approach overcomes the parameter constraint

[5] The improved variant proposed in [5] builds the 2-dimensional spaces with two codewords of appropriate weight, and is even closer to a type 2 instance

of [6], allowing a scaling for fixed q and growing blocksize. The matching algorithm runs in time $2^{\varepsilon}\text{poly}(n)$, and the overhead exponent ε seems to be a small, though constant, proportion of n, upper bounded in [10, Lemma 3] by $\frac{w^2}{(q-1)n}$. The asymptotic exponent is thus slightly above the optimal exponent for type 2 (14). To be thorough, let us add that the technique involves a compatibility criterion using only the codewords, similar to [6], which filters out all but a polynomial proportion of the tuples. This filtering has no impact on the asymptotic exponent, but has a significant practical impact, in particular when the technique is applied to the cryptanalysis of LESS, improving the attack of [6] on LESS.

Finally, as it is implemented in [10], it corresponds to a synchronous instance of type 2, which would lead to a higher asymptotic exponent, but the instance could be made asynchronous by changing the matching algorithm into an invariant, and would then be in line with the setting we are considering here and would correspond to the "Optimistic Runtime Exponent of Algorithm 2" which is mentioned in [10, Corollary 2].

4.6 Asymptotic Exponents for our Contribution.

In this paper we give effective instances of the framework whose asymptotic exponent meet the optimal exponent (type 1) for a wide range of parameters: (1) for all q if the code rate is not too close to 0.5 (the exact threshold depends of q) and (2) for all code rate if $q \geqslant 61$. Otherwise, except for $q = 5$, our asymptotic exponent never exceeds the optimal type 2 exponent.

5 New Instances Based on the Reduction to PCE

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two equivalent $[n,k]_q$ codes. From Theorem 1, if $(\boldsymbol{c}_1, \boldsymbol{c}_2)$ is a matching couple of codewords for $(\mathcal{C}_1, \mathcal{C}_2)$, the codes $\boldsymbol{c}_1^{-1} \star \mathcal{C}_1$ and $\boldsymbol{c}_2^{-1} \star \mathcal{C}_2$ are permutation equivalent. We use this reduction to define new instances of the matching codewords framework. Let $w = w_{\text{opt}}^{(1)}(n,k)$ as defined in (10) and assume we have built two lists of weight-w codewords $\mathcal{L}_1 \times \mathcal{L}_2 \subset \mathcal{C}_1 \times \mathcal{C}_2$, we consider the following guidelines:

1. If $w > k$ and $(\boldsymbol{c}_1, \boldsymbol{c}_2) \in \mathcal{L}_1 \times \mathcal{L}_2$ is a matching couple of codewords then the $[w,k]_q$ codes $\boldsymbol{c}_1^{-1} \star \mathcal{C}_1$ and $\boldsymbol{c}_2^{-1} \star \mathcal{C}_2$ are permutation equivalent.
   $\rightarrow$ Solve an easy permutation equivalence problem.
2. If $w < k$ and $(\boldsymbol{c}_1, \boldsymbol{c}_2) \in \mathcal{L}_1 \times \mathcal{L}_2$ is a matching couple of codewords then the $[n-w, k-w]_q$ codes $\mathsf{Shorten}(\mathcal{C}_1, \boldsymbol{c}_1)$ and $\mathsf{Shorten}(\mathcal{C}_2, \boldsymbol{c}_2)$ are equivalent.
   $\rightarrow$ Make a nested call to the framework, hopefully falling in the first case.
3. If $(\boldsymbol{c}_1, \boldsymbol{c}_2)$ and $(\boldsymbol{c}_1', \boldsymbol{c}_2')$ in $\mathcal{L}_1 \times \mathcal{L}_2$ are two matching couples of codewords then $(\boldsymbol{c}_1 + \boldsymbol{c}_1')^{-1} \star \mathcal{C}_1$ and $(\boldsymbol{c}_2 + \beta \cdot \boldsymbol{c}_2')^{-1} \star \mathcal{C}_2$ are permutation equivalent for some $\beta \in \mathbb{F}_q^*$.
   $\rightarrow$ Solve an easy permutation equivalence problem.

In the rest of this section, we provide asynchronous implementations for each the above guidelines.

To leverage the reduction to PCE, we need to assume that PCE is easy to solve on uniformly distributed instances. Though this is widely admitted since [28], it remains empirical.

Assumption 1 *There exists a code permutation-signature Ψ whose associated canonical form (as in Theorem 5), denoted* CanonicalForm *here, is defined for all but a vanishing proportion of the $[n,k]_q$ codes with a hull of dimension 0.*

Note. A canonical form can be associated to any equivalence relation and maps any element to a unique representative of its equivalence class. In this work the canonical forms are associated to PCE (see §7): two generator matrices are equivalent if they span permutation equivalent codes. In the context of code equivalence, canonical forms are also defined in [15], but for a different equivalence relation.

5.1 One Matching Couple, One Level

Rationale. Algorithm 1 is a straightforward implementation the type 1 framework. The invariant associated to $(\boldsymbol{c}_1, \mathcal{C}_1)$ is the canonical form for code permutation equivalence of $\boldsymbol{c}_1^{-1} \star \mathcal{C}_1$ as defined in §7.

Algorithm 1 LCP solver - Asynchronous - One Matching Couple - One Level

Input: Two $[n,k]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$, a weight w, a list length L.
Output: An isometry or $\perp$

repeat L times
  $\boldsymbol{c}_1 \leftarrow$ a codeword of $\mathcal{C}_1$ of weight w
  $\texttt{index} \leftarrow \mathsf{CanonicalForm}(\boldsymbol{c}_1^{-1} \star \mathcal{C}_1)$
  if $\texttt{index} \neq \perp$ **then**
    $T[\texttt{index}] \leftarrow \boldsymbol{c}_1$ ▷ table T initially empty
repeat L times
  $\boldsymbol{c}_2 \leftarrow$ a codeword of $\mathcal{C}_2$ of weight w
  $\texttt{index} \leftarrow \mathsf{CanonicalForm}(\boldsymbol{c}_2^{-1} \star \mathcal{C}_2)$
  if $T[\texttt{index}]$ exists **then**
    $\boldsymbol{c}_1 \leftarrow T[\texttt{index}]$
    $\varphi \leftarrow \mathsf{Isometry}(\boldsymbol{c}_1, \boldsymbol{c}_2, \mathcal{C}_1, \mathcal{C}_2)$ ▷ Algorithm 4
    if $\varphi \neq \perp$ **then return** φ
return $\perp$

Threshold Rate. Asymptotically, with $R = k/n$, we must have $\omega_{\text{opt}}^{(1)}(R) > R$, else the canonical form fails. We define the *threshold rate* $R_0(q)$ as the unique R, $0 < R \leqslant 0.5$, such that $\omega_{\text{opt}}^{(1)}(R) = R$ (see Figure 3):

$$R_0(q) = \underset{0<R\leqslant 0.5}{\text{rootOf}} \left(\omega_{\text{opt}}^{(1)}(R) = R \right) \tag{15}$$

Its value depends of the decoder used in the algorithm. We can observe in Table 2 that the impact of the decoder choice is limited, even negligible when q grows.

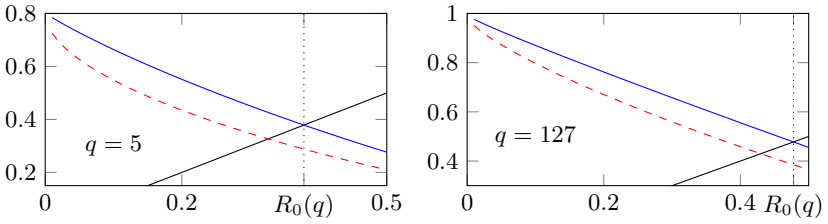


Fig. 3. $R \mapsto \omega^{(1)}_{\mathrm{opt}}(R)$ (blue), $R \mapsto \delta_{\mathrm{GV}}(R)$ (red, dashed) and $R \mapsto R$ (black)

	Decoding Algorithm		
	Prange	Dumer	MMT
$R_0(5)$	0.3787	0.3782	0.3727
$R_0(127)$	0.4778	0.4777	0.4776

Table 2. Threshold Rate $R_0(q)$ for $q \in \{5, 127\}$ and Various Decoders

Proposition 3. *For any R, $0 < R \leqslant 0.5$, and all $n > 0$, let $k = \lfloor nR \rfloor$, $w = \lfloor n\omega^{(1)}_{\mathrm{opt}}(R) \rfloor$ and $L = \sqrt{\frac{N_w}{q-1}}$. For growing n, under Assumption 1, the Algorithm 1 taking as input equivalent $[n,k]_q$ codes, with parameters (w, L) achieves the asymptotic exponent $\tau^{(1)}_{\mathrm{LCE}}(R)$ for solving* LCE *if $R < R_0(q)$, where $R_0(q)$ denotes the threshold rate* (15) *corresponding to the generic decoder used in Algorithm 1.*

Proof. If $R < R_0(q)$, then for sufficiently large n we have $w > k$ and from §6 and Assumption 1 the canonical form is defined with a probability larger than a positive constant and the proportion of false positives is negligible. Hence the instance is proper, in the sense of §4.4, up to a small constant factor. Moreover, with $L = \sqrt{\frac{N_w}{q-1}}$, there is a matching couple among all the $(\boldsymbol{c}_1, \boldsymbol{c}_2)$ with a probability larger than a positive constant.

Finally, since the canonical form can be computed in polynomial time, the algorithm reaches the exponent $\tau^{(1)}_{\mathrm{LCE}}(R)$ given in (13).

5.2 One Matching Couple, Two Levels

Rationale. If the code rate is above $R_0(q)$, the optimal weight is too low and the code $\boldsymbol{c}_1^{-1} \star \mathcal{C}_1$ is trivial. But then, since $w < k$, we can instead consider the

shortened code $\mathsf{Shorten}(\mathcal{C}_1, \boldsymbol{c}_1)$ which is an $[n-w, k-w]_q$ code. The new rate $(k-w)/(n-w)$ is always below the threshold rate $R_0(q)$ and we can apply the one level framework for this code. Instead of computing one invariant for each $\boldsymbol{c}_1$, we end up computing many invariants for a smaller instance. Hopefully, the cost for computing all those invariants will not be higher than the cost for obtaining $\boldsymbol{c}_1$, and we still have an optimal instance. This idea is implemented in Algorithm 2.

Algorithm 2 LCP solver - Asynchronous - One Matching Couple - Two levels

Input: Two $[n,k]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$, two weights w, w', two list lengths L, L'.
Output: An isometry or $\perp$

repeat L times
  $\boldsymbol{c}_1 \leftarrow$ a codeword of $\mathcal{C}_1$ of weight w
  $\mathcal{C} \leftarrow \mathsf{Shorten}(\mathcal{C}_1, \boldsymbol{c}_1)$
  repeat L' times
    $\boldsymbol{v}_1 \leftarrow$ a codeword of $\mathcal{C}$ of weight w'
    $\mathtt{index} \leftarrow \mathsf{CanonicalForm}(\boldsymbol{v}_1^{-1} \star \mathcal{C})$
    if $\mathtt{index} \neq \perp$ **then**
      $T[\mathtt{index}] \leftarrow (\boldsymbol{v}_1, \boldsymbol{c}_1)$ ▷ table T initially empty
repeat L times
  $\boldsymbol{c}_2 \leftarrow$ a codeword of $\mathcal{C}_2$ of weight w
  $\mathcal{C} \leftarrow \mathsf{Shorten}(\mathcal{C}_2, \boldsymbol{c}_2)$
  repeat L' times
    $\boldsymbol{v}_2 \leftarrow$ a codeword of $\mathcal{C}$ of weight w'
    $\mathtt{index} \leftarrow \mathsf{CanonicalForm}(\boldsymbol{v}_2^{-1} \star \mathcal{C})$
    if $T[\mathtt{index}]$ exists **then**
      $\boldsymbol{c}_1, \boldsymbol{v}_1 \leftarrow T[\mathtt{index}]$
      $\varphi \leftarrow \mathsf{Isometry}(\boldsymbol{c}_1, \boldsymbol{v}_1, \boldsymbol{c}_2, \boldsymbol{v}_2, \mathcal{C}_1, \mathcal{C}_2)$ ▷ Algorithm 5
      if $\varphi \neq \perp$ **then return** φ
return $\perp$

Complexity Analysis. The Algorithm 2 implements two successive nested instances of the framework, the first with parameters (n, k, w, L) and the second with parameters $(n' = n-w, k' = k-w, w'L')$. The second level can be viewed as a (somewhat expensive) invariant which consists in computing L' weight-w' codeword in an $[n', k']_q$ codes and the corresponding L' canonical forms for $[w', k']_q$ codes. Assuming proper instances, from Proposition 1 we have a running time of $2T_{\mathrm{LCE}}^{(1.2)}(n, k, w, L, w', L')$, where

$$T_{\mathrm{LCE}}^{(1.2)}(n, k, w, L, w', L') = L \cdot T_{\mathrm{dec}}(n, k, w) + L \cdot T_{\mathrm{LCE}}^{(1)}(n', k', w', L') \tag{16}$$

and a success probability of $P_{\mathrm{match}}^{(1)}(n, k, w, L) \cdot P_{\mathrm{match}}^{(1)}(n', k', w', L')$. Asymptotically, with the optimal parameters at each level, we define: $R = k/n$, $\omega = w/n = \omega_{\mathrm{opt}}(R)$ and $L = 2^{\frac{n}{2}\nu(R,\omega)}$ for the first level; and $R' = (k-w)/(n-k)$,

$\omega' = w'/(n-w) = \omega_{\text{opt}}(R')$ and $L' = 2^{\frac{n-w}{2}\nu(R',\omega')}$ for the second level. The asymptotic exponent is

$$\tau^{(1.2)}_{\text{LCE}}(R) = \max\left(\tau^{(1)}_{\text{LCE}}(R), \frac{1}{2}\nu(R,\omega) + (1-\omega)\cdot\tau^{(1)}_{\text{LCE}}\left(\frac{R-\omega}{1-\omega}\right)\right) \tag{17}$$

where $\omega = \omega_{\text{opt}}(R)$.

Threshold Alphabet Size. Let $\mu = \frac{1}{2}\nu(R,\omega) + (1-\omega)\tau^{(1)}_{\text{LCE}}\left(\frac{R-\omega}{1-\omega}\right)$ denote the second argument of max in (17) with $\omega = \omega_{\text{opt}}(R)$. We observed experimentally that for a given R and varying q, the curve $q \mapsto \mu$ intersects at most once with the curve $q \mapsto \tau^{(1)}_{\text{LCE}}(R)$ and with the curve $q \mapsto \tau^{(2)}_{\text{LCE}}(R)$. We define two threshold values for the alphabet size:

$$q_1(R) = \min\left\{q \geqslant 5, \frac{1}{2}\nu(R,\omega) + (1-\omega)\cdot\tau^{(1)}_{\text{LCE}}\left(\frac{R-\omega}{1-\omega}\right) \leqslant \tau^{(1)}_{\text{LCE}}(R)\right\} \tag{18}$$

$$q_2(R) = \min\left\{q \geqslant 5, \frac{1}{2}\nu(R,\omega) + (1-\omega)\cdot\tau^{(1)}_{\text{LCE}}\left(\frac{R-\omega}{1-\omega}\right) > \tau^{(2)}_{\text{LCE}}(R)\right\} \tag{19}$$

For $R = 0.5$ and with Prange decoder those thresholds are respectively $q_1(0.5) = 61$ and $q_2(0.5) = 37$, see Figure 4. With Dumer and MMT, the same thresholds are respectively $71, 41$ and $79, 43$.

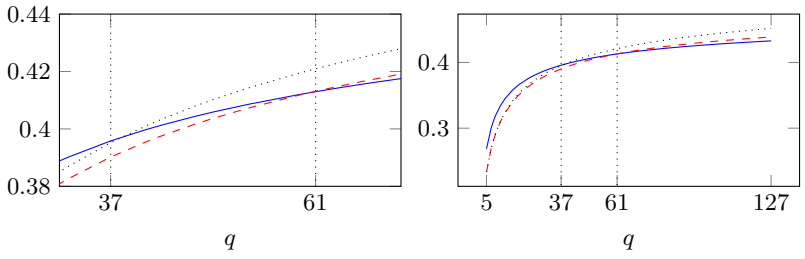


Fig. 4. Comparing $\tau^{(1)}_{\text{LCE}}(R)$ (red, dashed), $\tau^{(2)}_{\text{LCE}}(R)$ (black, dotted) and $\frac{1}{2}\nu(R,\omega)+(1-\omega)\tau^{(1)}_{\text{LCE}}\left(\frac{R-\omega}{1-\omega}\right)$ (blue) for $R = 0.5$ and $\omega = \omega_{\text{opt}}(R)$ with Prange Decoder

Proposition 4. *Let R be a code rate $0 < R \leqslant 0.5$. For all $n > 0$ let $k = \lfloor nR \rfloor$, $\omega = \omega^{(1)}_{\text{opt}}(R)$, $w = \lfloor n\omega \rfloor$, $L = \sqrt{\frac{N(n,k,w)}{q-1}}$, $R' = \frac{R-\omega}{1-\omega}$, $w' = \lfloor (n-w)\omega^{(1)}_{\text{opt}}(R') \rfloor$ and $L' = \sqrt{\frac{N(n',k',w')}{q-1}}$. For growing n, the Algorithm 2 taking as input equivalent $[n,k]_q$ codes, with parameters (w, w', L, L')*

- *achieves the asymptotic exponent $\tau^{(1)}_{\text{LCE}}(R)$ for solving* LCE *if $R > R_0(q)$ and $q \geqslant q_1(R)$,*

– *achieves an asymptotic exponent* $\leqslant \tau^{(2)}_{\mathrm{LCE}}(R)$ *for solving* LCE *if* $R > R_0(q)$ *and* $q > q_2(R)$,

where $R_0(q)$, $q_1(R)$ *and* $q_2(R)$ *are defined in* (15), (18) *and* (19) *with the generic decoder used in Algorithm 2.*

Proof. If $R > R_0(q)$, then, for n large enough we have $w < k$ and the code $\mathcal{C}_1$ can be shortened in w positions resulting in a code of length $n' = n - w$. This code has dimension $k' = k - w$ with probability larger than a positive constant. It is easily verified experimentally that $R' = \frac{R-\omega_{\mathrm{opt}}(R)}{1-\omega_{\mathrm{opt}}(R)} < R_0(q)$ and we are in the favorable case of Proposition 3.

Now, with $L = \sqrt{\frac{N(n,k,w)}{q-1}}$, there is at least one matching couple among all the $(\boldsymbol{c}_1, \boldsymbol{c}_2)$ computed at instructions 2: and 10: with constant probability. For each such matching couple the codes $\mathsf{Shorten}(\mathcal{C}_1, \boldsymbol{c}_1)$ and $\mathsf{Shorten}(\mathcal{C}_2, \boldsymbol{c}_2)$ are equivalent $[n', k']_q$ codes. Thus, with $L' = \sqrt{\frac{N(n',k',w')}{q-1}}$, there is a matching couple among all the $(\boldsymbol{v}_1, \boldsymbol{v}_2)$ computed at instructions 5: and 13: with constant probability.

From §6 and Assumption 1 the canonical form is defined with a probability larger than a constant positive and the proportion of false positives is negligible. Hence the two nested instances are proper, in the sense of §4.4, up to a small constant factor.

Finally, since the canonical form can be computed in polynomial time, the algorithm reaches the exponent given in (17)

$$\tau^{(1.2)}_{\mathrm{LCE}}(R) = \max\left(\tau^{(1)}_{\mathrm{LCE}}(R), \mu\right)$$

where

$$\mu = \frac{1}{2}\nu(R,\omega) + (1-\omega)\cdot\tau^{(1)}_{\mathrm{LCE}}\left(\frac{R-\omega}{1-\omega}\right) \text{ and } \omega = \omega^{(1)}_{\mathrm{opt}}(R).$$

Following the alphabet size threshold analysis $\mu \leqslant \tau^{(1)}_{\mathrm{LCE}}(R)$ if and only if $q \geqslant q_1(R)$ and $\mu > \tau^{(2)}_{\mathrm{LCE}}(R)$ if and only if $q \leqslant q_2(R)$ and we get the result.

5.3 Two Matching Couples, One Level

Rationale. If Algorithm 1 fails, rather than having two levels as in Algorithm 2, we may also consider two codewords $\boldsymbol{c}_1$ and $\boldsymbol{c}'_1$ in $\mathcal{C}_1$ and use as an invariant the canonical form for the sum $\mathsf{CanonicalForm}\left((\boldsymbol{c}_1 + \boldsymbol{c}'_1)^{-1} \star \mathcal{C}_1\right)$. This is essentially a type 2 instance which may reach the type 2 optimal exponent if the sum of two codewords of weight w is larger than k. For parameters such that $R > R_0(q)$ and $q \leqslant q_2(R)$ the exponent may improve on Algorithm 2.

Algorithm 3 LCP solver - Asynchronous - Two Matching Couples - One Level

Input: Two $[n,k]_q$ codes $\mathcal{C}_1, \mathcal{C}_2$, a weight w, a list length L.
Output: An isometry or $\perp$

$\mathcal{L}_1 \leftarrow$ a list of L codewords of $\mathcal{C}_1$ of weight w
for all pairs $(\boldsymbol{c}_1, \boldsymbol{c}'_1)$ in $\mathcal{L}_1^2$ **do**
  $\texttt{index} \leftarrow \mathsf{CanonicalForm}\left((\boldsymbol{c}_1 + \boldsymbol{c}'_1)^{-1} \star \mathcal{C}_1\right)$
  if $\texttt{index} \neq \perp$ **then**
    $T[\texttt{index}] \leftarrow (\boldsymbol{c}_1, \boldsymbol{c}'_1)$ ▷ table T initially empty
$\mathcal{L}_2 \leftarrow$ a list of L codewords of $\mathcal{C}_2$ of weight w
for all pairs $(\boldsymbol{c}_2, \boldsymbol{c}'_2)$ in $\mathcal{L}_2^2$ **do**
  for β in $\mathbb{F}_q \setminus \{0\}$ **do**
    $\texttt{index} \leftarrow \mathsf{CanonicalForm}\left((\boldsymbol{c}_2 + \beta\boldsymbol{c}'_2)^{-1} \star \mathcal{C}_2\right)$
    if $T[\texttt{index}]$ exists **then**
      $\boldsymbol{c}_1, \boldsymbol{c}'_1 \leftarrow T[\texttt{index}]$
      $\varphi \leftarrow \mathsf{Isometry}(\boldsymbol{c}_1 + \boldsymbol{c}'_1, \boldsymbol{c}_2 + \beta\boldsymbol{c}'_2, \mathcal{C}_1, \mathcal{C}_2)$ ▷ Algorithm 4
      if $\varphi \neq \perp$ **then return** φ
return $\perp$

Threshold Alphabet Size. The requirement now is that the sum of two codewords of optimal weight w has a weight larger than k. The average weight of such a sum is $2w - \frac{q+1}{q}\frac{w^2}{n}$, moreover, asymptotically, most sums will have this weight. If we set $R = k/n$ and $\omega = \omega^{(2)}_{\text{opt}}(R)$, then the asymptotic condition becomes $2\omega - \frac{q+1}{q}\omega^2 > R$. We define

$$q_3(R) = \min\left\{q \geqslant 5, 2\omega^{(2)}_{\text{opt}}(R) - \frac{q+1}{q}\omega^{(2)}_{\text{opt}}(R)^2 \leqslant R\right\}. \tag{20}$$

We have $q_3(0.5) = 5$ for Prange and Dumer, and $q_3(0.5) = 7$ for MMT.

Proposition 5. *Let R be a code rate $0 < R \leqslant 0.5$. For all $n > 0$ let $k = \lfloor nR \rfloor$, $w = \lfloor n\omega^{(2)}_{\text{opt}}(R) \rfloor$ and $L = \sqrt{\frac{2N_w}{q-1}}$. For growing n, the Algorithm 3 taking as input equivalent $[n,k]_q$ codes, with parameters (w, L) achieves the asymptotic exponent $\tau^{(2)}_{\text{LCE}}(R)$ for solving* LCE *if $q > q_3(R)$ as defined in* (20).

Proof. The proof is the same as the proof of Algorithm 3. It differs when weight of $\boldsymbol{c}_1 + \boldsymbol{c}'_1$ is concerned and for the ability to detect matching couples.

We have $|\boldsymbol{c}_1 + \boldsymbol{c}'_1| = 2w - \frac{q+1}{q}\frac{w^2}{n}$ on average: the two words have an average of w^2/n common non-zero coordinates, among those, on average, $w^2/(qn)$ coordinates will cancel in the sum. We easily deduce that, for n large enough, we have $|\boldsymbol{c}_1 + \boldsymbol{c}'_1| > k$ if $q > q_3(R)$.

If $(\boldsymbol{c}_1, \boldsymbol{c}_2)$ and $(\boldsymbol{c}'_1, \boldsymbol{c}'_2)$ are two matching couples, we have $\boldsymbol{c}_2 = \beta \cdot \varphi(\boldsymbol{c}_1)$ and $\boldsymbol{c}_2 = \beta' \cdot \varphi(\boldsymbol{c}_1)$ for some β and β' in $\mathbb{F}_q^*$, thus $\boldsymbol{c}_2 + \beta\beta'^{-1} \cdot \boldsymbol{c}'_2 = \beta \cdot \varphi(\boldsymbol{c}_1 + \boldsymbol{c}'_1)$. Thus, one of all $\mathsf{CanonicalForm}\left((\boldsymbol{c}_2 + \beta \cdot \boldsymbol{c}'_2)^{-1} \star \mathcal{C}_1\right)$ for $\beta \in \mathbb{F}_q^*$ is equal to $\mathsf{CanonicalForm}\left((\boldsymbol{c}_1 + \boldsymbol{c}'_1)^{-1} \star \mathcal{C}_1\right)$ and the match will be detected.

5.4 Recovering the Isometry

Each of our algorithms will call the Isometry routine once, possibly more if there are false positives. We will see that Isometry runs in polynomial time and since there are few false positives §6.3, there is no issue about this.

If we disregard false positives, when Isometry is called from one of our three algorithms, it means that, not only we have found matching couples of codewords, but also that the corresponding canonical forms were defined, hence we always recover the isometry. The algorithms use two main building blocks, both running in polynomial time:

- $\mathsf{Extend}(\pi, I, \mathcal{C}_1, \mathcal{C}_2)$, with π a permutation of $[\![0, n[\![$, $I \subset [\![0, n[\![$, $\mathcal{C}_1$ and $\mathcal{C}_2$ two equivalent $[n, k]_q$ codes, which returns an isometry φ such that $\mathcal{C}_2 = \varphi(\mathcal{C}_1)$ if $|I| \geqslant k + 2$ and if for all $i \in I$, $\pi(i) = \pi'(i)$ where π' is the underlying permutation in φ. This correct behavior is guaranteed by Theorem 2.
- $\mathsf{PCESolve}(\mathcal{C}_1, \mathcal{C}_2)$, with $\mathcal{C}_1$ and $\mathcal{C}_2$ two permutation equivalent $[n, k]_q$ codes, which returns a permutation π such that $\mathcal{C}_2 = \pi(\mathcal{C}_1)$. This correct behavior is guaranteed by Theorem 5.

Algorithm 4 Isometry Reconstruction

Input: $\boldsymbol{c}_1, \boldsymbol{c}_2, \mathcal{C}_1, \mathcal{C}_2$
▷ $\mathcal{C}_1$ and $\mathcal{C}_2$ are $[n, k]_q$ codes, $|\boldsymbol{c}_1| = |\boldsymbol{c}_2| = w$
Output: an isometry or $\perp$
1: $\pi \leftarrow \mathsf{PCESolve}(\boldsymbol{c}_1^{-1} \star \mathcal{C}_1, \boldsymbol{c}_2^{-1} \star \mathcal{C}_2)$ ▷ §7 (22)
▷ *π is a permutation of w elements, we lift it as a permutation of $[\![0, n[\![$ such that $\pi(i) = j$ for all $i \in \mathrm{supp}(\boldsymbol{c}_1)$ matching with $j \in \mathrm{supp}(\boldsymbol{c}_2)$, entries out of $\mathrm{supp}(\boldsymbol{c}_1)$ can take any value as long as π remains a permutation*
2: **return** $\mathsf{Extend}(\pi, \mathrm{supp}(\boldsymbol{c}_1), \mathcal{C}_1, \mathcal{C}_2)$ ▷ Theorem 2

Algorithm 5 Isometry Reconstruction - Two Levels

Input: $\boldsymbol{c}_1, \boldsymbol{v}_1, \boldsymbol{c}_2, \boldsymbol{v}_2, \mathcal{C}_1, \mathcal{C}_2$
▷ $\mathcal{C}_1$ and $\mathcal{C}_2$ are $[n, k]_q$ codes, $|\boldsymbol{c}_1| = |\boldsymbol{c}_2| = w$, $|\boldsymbol{v}_1| = |\boldsymbol{v}_2| = w'$
Output: an isometry or $\perp$
1: $\mu \leftarrow \mathsf{Isometry}(\boldsymbol{v}_1, \boldsymbol{v}_2, \mathsf{Shorten}(\mathcal{C}_1, \boldsymbol{c}_1), \mathsf{Shorten}(\mathcal{C}_2, \boldsymbol{c}_2))$ ▷ Algorithm 4
2: $\boldsymbol{v}, \pi \leftarrow \mathsf{Decompose}(\mu)$ ▷ $\mu = (\boldsymbol{v}; \pi)$ a linear isometry
▷ *π is a permutation of $n - w$ elements, we lift it as a permutation of $[\![0, n[\![$ such that $\pi(i) = j$ for all $i \notin \mathrm{supp}(\boldsymbol{v}_1)$ matching with $j \notin \mathrm{supp}(\boldsymbol{v}_2)$, entries in $\mathrm{supp}(\boldsymbol{v}_1)$ can take any value as long as π remains a permutation*
3: **return** $\mathsf{Extend}(\pi, [\![0, n[\![\setminus \mathrm{supp}(\boldsymbol{c}_1), \mathcal{C}_1, \mathcal{C}_2)$ ▷ Theorem 2

Proposition 6. *If* $\mathsf{CanonicalForm}(\boldsymbol{c}_1^{-1}\star\mathcal{C}_1) = \mathsf{CanonicalForm}(\boldsymbol{c}_2^{-1}\star\mathcal{C}_2) \neq \perp$ *then, on input* $(\boldsymbol{c}_1, \boldsymbol{c}_2, \mathcal{C}_1, \mathcal{C}_2)$*, the Algorithm 4 returns an isometry* φ *such* $\mathcal{C}_2 = \varphi(\mathcal{C}_1)$ *with probability close to one.*

Proof. From §6.3, the equality of the canonical forms guarantees matching couples with a probability exponentially close to 1. From Theorem 5, if the canonical form is defined there is a unique permutation between $\boldsymbol{c}_1^{-1}\star\mathcal{C}_1$ and $\boldsymbol{c}_2^{-1}\star\mathcal{C}_2$ revealing $w > k$ matching coordinates between $\mathcal{C}_1$ and $\mathcal{C}_2$, enough to apply Theorem 2 and recover the whole isometry with a probability exponentially close to 1.

Proposition 7. *If* $\mathsf{CanonicalForm}(\boldsymbol{c}_1^{-1}\star\mathcal{C}_1) = \mathsf{CanonicalForm}(\boldsymbol{c}_2^{-1}\star\mathcal{C}_2) \neq \perp$ *and* $\mathsf{CanonicalForm}(\boldsymbol{v}_1^{-1}\star\mathsf{Shorten}(\mathcal{C}_1, \boldsymbol{c}_1)) = \mathsf{CanonicalForm}(\boldsymbol{v}_2^{-1}\star\mathsf{Shorten}(\mathcal{C}_2, \boldsymbol{c}_2)) \neq \perp$ *then, on input* $(\boldsymbol{c}_1, \boldsymbol{v}_1, \boldsymbol{c}_2, \boldsymbol{v}_2, \mathcal{C}_1, \mathcal{C}_2)$*, the Algorithm 5 returns an isometry* φ *such* $\mathcal{C}_2 = \varphi(\mathcal{C}_1)$ *with probability close to one.*

Proof. Same as Proposition 6.

5.5 Memory Complexity

The memory complexity of all the algorithms is essentially the size of the data structure holding the codewords. The `index` can be short, *e.g.* a hash of the canonical form. Respectively $O(n \cdot L)$, $O(n \cdot L \cdot L')$ and $O(n \cdot \binom{L}{2})$ for Algorithms 1, 2 and 3.

6 Correctness of the algorithms

In this section we analyze theoretically the fraction of *proper* instances for which our algorithms will succeed with at least a constant probability. In order to prove the correctness of our algorithms we will need to quantify the following phenomenons:

1. the distribution of the Schur products $\boldsymbol{c}^{-1} \star \mathcal{C}$,
2. the distribution of the dimension of the hull of these codes,
3. the proportion of decomposable codes among all the codes and
4. the probability of having *false positive solutions* for PCE instances (i.e. probability for a produced PCE instance to have a solution that is not a restriction of the LCE solution).

Notation We use gothic calligraphy for random variables as $\mathfrak{C}, \mathfrak{C}_0, \mathfrak{X}, \mathfrak{c}$. The matrices are still uppercase and boldface italic and the vectors are still boldface italic.

Probabilistic modeling All the new algorithms presented in this paper can be boiled down to the codewords matching framework. In this framework, some code $\mathcal{C}$ is given and we draw uniformly elements in it. These elements are modeled by a random variable $\mathfrak{c}$ uniformly distributed in the codewords of $\mathcal{C}$ of a prescribed weight w.

For example, we can estimate on average how many codewords $\boldsymbol{c}$ of a weight w in a code $\mathcal{C}$ will verify that $\boldsymbol{c}^{-1} \star \mathcal{C}$ has a non trivial hull. A bound on the variance of this phenomenon will enable us to use the Bienaymé-Chebychev inequality and to give lower bounds for the probability $\mathbb{P}\left(\mathcal{H}\left(\mathfrak{c}^{-1} \star \mathcal{C}\right) = \{0\}\right)$ for a large fraction of codes $\mathcal{C}$. Those results are summarized in Theorem 4.

For technical details, some random variable $\mathfrak{C}$ representing a uniformly distributed $[n,k]_q$ code will be considered. However, to conduct the theoretical analysis, we make the same assumption as in [10, Prop. 1], and instead of uniformly drawing an $[n,k]_q$ code $\mathfrak{C}$ and a word $\mathfrak{c} \in \mathfrak{C}$ of weight w, we will draw uniformly a word $\mathfrak{c} \in \mathbb{F}_q^n$ of weight w and an $[n,k-1]_q$ code $\mathfrak{C}'$ not containing $\mathfrak{c}$ (as long as w is larger than the minimal distance of $\mathfrak{C}$). If w is large enough, it is possible to prove that the two distributions are exponentially close.

The proofs for the results of this section are available in Appendix A.

6.1 Distribution of $\boldsymbol{c^{-1} \star \mathcal{C}}$

It will be very useful to prove that this Schur product behaves like some uniformly distributed code in order to apply general results on the probabilities of having a non trivial hull, having a non-trivial automorphism group, being non-decomposable, or the probabilities for two independent random codes to be permutation equivalent.

Theorem 3. *Let $\mathfrak{C}' = \langle \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k \rangle$ a uniformly distributed $[n,k-1]_q$ code and let $\mathfrak{c}$ be a codeword uniformly distributed in $\mathbb{F}_q^n \backslash \mathfrak{C}'$ of weight $w > k$. Then the probability $\mathbb{P}(\mathfrak{c}^{-1} \star \mathfrak{C}' = \mathcal{C}_0 \mid \dim(\mathfrak{C}') = k-1)$ for a fraction of the $[w,k-1]_q$ code $\mathcal{C}_0$ close to 1, up to a term $O\left(q^{-\frac{1}{2}[R-\log_q(\frac{q}{q-1})]n}\right)$, verifies*

$$\mathbb{P}(\mathfrak{c}^{-1} \star \mathfrak{C}' = \mathcal{C}_0 \mid \dim(\mathfrak{c}^{-1} \star \mathfrak{C}') = k-1) = \frac{1}{\begin{bmatrix} w \\ k-1 \end{bmatrix}_q} \left(1 + O(q^{-\frac{R}{2}w})\right).$$

Remark 2. The $O\left(q^{-\frac{1}{2}[R-\log_q(\frac{q}{q-1})]n}\right)$ fraction of codes $\mathcal{C}_0$ that do not satisfy the results of the theorem are codes such that the number of full weight codewords in $\mathcal{C}_0$ is far from the expected value for $[w,k-1]_q$ codes.

Remark 3. This theorem ensure that the Schur product between the code $\mathfrak{C} = \langle \mathfrak{c}_1, \ldots, \mathfrak{c}_k \rangle$ and ${\mathfrak{c}_1}^{-1}$ can be seen as a (not necessarily direct) sum between $\langle \mathbb{1}_w \rangle$ and a uniformly distributed $[w,k-1]_q$ code.

6.2 Hull distribution among $\boldsymbol{c}^{-1} \star \mathcal{C}$ codes

Thanks to the previous subsection, we can determine the proportion of Schur products that will have a trivial hull and prove that this happens with high probability. Table 3 gives numerical values for concrete q using $F_0(q) \triangleq 1 - H_0(q)$ the asymptotic value of the probability of a code to *not* have a trivial hull (see Lemma 1). Our experiments in §8.1 show that for concrete LESS parameters, these probabilities are even lower.

Theorem 4. *We fix q the size of the field, and consider $n, k \to \infty$ and $w > k$. For at least a constant fraction $f_q \triangleq 1 - F_0(q)^{\frac{1}{2}} \left(1 + O\left(q^{-\frac{1}{4}k}\right)\right)$ of $[n,k]_q$ codes $\mathcal{C}$, the fraction of codewords $\boldsymbol{c}$ of weight w in $\mathcal{C}$ such that $\boldsymbol{c}^{-1} \star \mathcal{C}$ has a non trivial hull is upper bounded by $\delta_q \triangleq F_0(q)^{\frac{2}{3}} \left(1 + O\left(q^{-\frac{1}{3}k}\right)\right)$.*

q	f_q	δ_q	$F_0(q)$
5	54.54	34.96	20.77
7	61.87	27.65	14.54
61	87.19	6.45	1.64
127	91.13	3.96	0.79

Table 3. For various q, $f_q \triangleq 1 - F_0(q)^{\frac{1}{2}} \left(1 + O\left(q^{-\frac{1}{4}k}\right)\right)$ is a lower bound on the proportion of $[n,k]_q$ codes $\mathcal{C}$ with a fraction upper bounded by $\delta_q \triangleq F_0(q)^{\frac{2}{3}} \left(1 + O\left(q^{-\frac{1}{3}k}\right)\right)$ of codewords $\boldsymbol{c} \in \mathcal{C}$ of weight w such that $\boldsymbol{c}^{-1} \star \mathcal{C}$ has a non trivial hull ($F_0(q)$ being the asymptotic fraction of codes not having a trivial hull).

Remark 4. The given upper bounds are very conservative. In our experiments, we observed for several codes a fraction of codewords producing a trivial hull very close to $H_0(q)$.

6.3 False positive probability

This paper relies on a reduction from LCE to PCE. Here, *false positive* designates PCE instances having a solution from which no solution can be retrieved for the original LCE instance by Theorem 2. This implies that we produced an unfortunate solvable PCE instance whose solution is not related to the original LCE instance.

In order to quantify this phenomenon, we analyze the behavior of false positives where $\mathfrak{C}$ is a uniformly distributed $[n,k]_q$ code and $\mathfrak{c}, \mathfrak{c}'$ are uniformly distributed in the codewords of weight w of $\mathfrak{C}$. This study provides bounds for the average phenomenon of false positives over all the possible codes. The results of this study will enable us to refine its statement through Bienaymé-Chebychev inequality, as we have done for the distribution of hulls for $\mathfrak{c}^{-1} \star \mathcal{C}$ codes. For the

sake of conciseness, only the results on the average phenomenon are presented here.

Proposition 8 provides an upper bound for two codewords to produce a false positive depending on the size of their support intersection meanwhile Proposition 9 provides an asymptotic behavior of this upper bound.

Proposition 8. *Let $\mathfrak{C}_1, \mathfrak{C}_2$ two uniformly distributed $[n,k]_q$ codes, φ an isometry such that $\varphi(\mathfrak{C}_1) = \mathfrak{C}_2$, and $\mathfrak{c}, \mathfrak{c}'$ two uniformly drawn codewords of weight $w > k$ in $\mathfrak{C}_1$. Assume the intersection of their support contains an information set and is of size ℓ. Then the probability for having false positive is upper bounded by:*

$$\mathbb{P}\left(\mathfrak{c}^{-1} \star \mathfrak{C}_1 \overset{\mathsf{PCE}}{\sim} \varphi(\mathfrak{c}'^{-1}) \star \mathfrak{C}_2\right) \leqslant \frac{q^N - 1}{q^{\lfloor Rn \rfloor} - 1} \frac{N!}{\begin{bmatrix} N \\ \lfloor Rn \rfloor \end{bmatrix}_q}$$

where $N \triangleq w - (\ell - k)$.

In order to study an asymptotic behavior for this upper bound $\frac{q^N-1}{q^k-1} \frac{N!}{\begin{bmatrix} N \\ \lfloor Rn \rfloor \end{bmatrix}_q}$, assume $N = \lfloor \alpha n \rfloor$ with $\alpha \in]0,1[$. In this case, the following proposition ensures the false positive probability is exponentially small.

Proposition 9. *The probability for two uniformly distributed $[\lfloor \alpha n \rfloor, \lfloor Rn \rfloor]_q$ codes containing $\mathbb{1}_{\lfloor \alpha n \rfloor}$ with R and α in $]0,1[$, to be permutation equivalent is upper bounded by $\frac{q^{\lfloor \alpha n \rfloor}-1}{q^{\lfloor Rn \rfloor}-1} \frac{\lfloor \alpha n \rfloor !}{\begin{bmatrix} \lfloor \alpha n \rfloor \\ \lfloor Rn \rfloor \end{bmatrix}_q}$ and:*

$$\frac{q^{\lfloor \alpha n \rfloor} - 1}{q^{\lfloor Rn \rfloor} - 1} \frac{\lfloor \alpha n \rfloor !}{\begin{bmatrix} \lfloor \alpha n \rfloor \\ \lfloor Rn \rfloor \end{bmatrix}_q} \underset{n \to \infty}{=} O\left(\sqrt{2\pi\alpha n}\; q^{\alpha n[(1-R)+\log(\frac{\alpha n}{e})-(1-R)R\alpha n]}\right).$$

A remaining question is: will this $N = w - (\ell - k)$ be proportional to n for most of the pairs $\mathfrak{c}, \mathfrak{c}'$? If it is, then we have an exponentially small probability to create a false positive for most of the pairs. Here is an answer, given in a nutshell.

Denote $\mathfrak{L}$ the length of the support intersection between $\mathfrak{c}$ and $\mathfrak{c}'$. One can assume that $\mathfrak{L}$ follows an hypergeometric distribution for which the mean value $\mathbb{E}[\mathfrak{L}]$ and the variance $\mathbb{V}[\mathfrak{L}]$ are known. Basic probabilistic analysis shows that, in most cases (i.e., except for an $O(n^{-1})$ fraction of them), $\mathfrak{L}$ is smaller than $\mathbb{E}[\mathfrak{L}] + \mathbb{V}[\mathfrak{L}]$, which is proportional to n. Then, in most cases, N/n is bigger than a constant (which lies in $]0,1[$).

7 Solving Permutation Equivalence

7.1 Signatures, Discriminability and Canonical Forms

Definition 5 ([28]). *A* code permutation-signature *is a function Ψ taking as arguments a $[n,k]_q$ code $\mathcal{C}$ and a coordinate index $i \in [\![0, n[\![$ such that for all permutation π of $[\![0, n[\![$, we have $\Psi(\mathcal{C}, i) = \Psi(\pi(\mathcal{C}), \pi(i))$.*

Definition 6 (Discriminability of a Signature). *For any code permutation-signature Ψ and any $[n,k]_q$ code $\mathcal{C}$:*

(i) Ψ discriminates $i \in [\![0,n[\![$ *for* $\mathcal{C}$ *if* $\Psi(\mathcal{C},i) \neq \Psi(\mathcal{C},j)$ *for all* $j \in [\![0,n[\![\setminus \{i\}$.
(ii) Ψ *is* fully discriminant for $\mathcal{C}$ *if it discriminates all* $i \in [\![0,n[\![$.

A (PCE) canonical form for $\mathcal{C}[n,k]_q$ is a uniquely defined matrix, say $\boldsymbol{G}_0 \in \mathbb{F}_q^{k\times n}$, which spans a code $\mathcal{C}_0$ permutation equivalent to $\mathcal{C}$, and such that any code permutation equivalent to $\mathcal{C}$ admits the same canonical form. The corresponding canonical ordering of the coordinates of $\mathcal{C}$ is a permutation π of $[\![0,n[\![$ such that $\pi(\mathcal{C}) = \mathcal{C}_0$. The following theorem states that a canonical form (and ordering) can be associated to any signature (as in Definition 5) and is defined for any code $\mathcal{C}$, as long as the signature is discriminant enough for that code.

7.2 Canonical Form for Permutation Equivalence

Theorem 5 (Canonical Form and Canonical Ordering for Permutation Equivalence). *For any code permutation-signature Ψ and for any $[n,k]_q$ code $\mathcal{C}$ we define the following set*

$$I_\Psi(\mathcal{C}) = \{i \in [\![0,n[\![\mid \Psi(\mathcal{C},i) \notin \{\Psi(\mathcal{C},j), j \in [\![0,n[\![\setminus \{i\}\}\}$$

which consists of all coordinates of $\mathcal{C}$ discriminated by Ψ. If $I_\Psi(\mathcal{C})$ does not contain an information set for $\mathcal{C}$, we set $\mathrm{CF}_\Psi(\mathcal{C}) = \mathrm{CO}_\Psi(\mathcal{C}) = \perp$, else we define $\mathrm{CF}_\Psi(\mathcal{C}) = \boldsymbol{G}_0 \in \mathbb{F}_q^{k\times n}$ and $\mathrm{CO}_\Psi(\mathcal{C}) = \pi$ a permutation of $[\![0,n[\![$ as follows:

- *From any generator matrix $\boldsymbol{G} \in \mathbb{F}_q^{k\times n}$ of $\mathcal{C}$, build a matrix $\boldsymbol{G}_1 \in \mathbb{F}_q^{k\times n}$ as follows:*
  - *the first $\ell = |I_\Psi(\mathcal{C})|$ columns of $\boldsymbol{G}_1$ are the columns of $\boldsymbol{G}$ indexed by $I_\Psi(\mathcal{C})$ in the decreasing lexicographic order given by $\Psi(\mathcal{C},i)$, say $I_\Psi(\mathcal{C}) = \{i_0,\dots,i_{\ell-1}\}$ with*

$$\Psi(\mathcal{C},i_0) <_{\mathrm{lex}} \Psi(\mathcal{C},i_1) <_{\mathrm{lex}} \dots <_{\mathrm{lex}} \Psi(\mathcal{C},i_{\ell-1}).$$

  - *the last $n-\ell$ columns of $\boldsymbol{G}_1$ are those of $\boldsymbol{G}$ indexed by $[\![0,n[\![\setminus I_\Psi(\mathcal{C}) = \{i_\ell,\dots,i_{n-1}\}$ in any order, say $\boldsymbol{G}_1 = \pi_1(\boldsymbol{G})$ with $\pi_1(i_j) = j$ for all $j \in [\![0,n[\![$.*
- *Compute $\boldsymbol{G}_2$ the reduced row echelon form of $\boldsymbol{G}_1$.*
- *Order the columns of $\boldsymbol{G}_2$ indexed by $[\![\ell,n[\![= \{i'_\ell,\dots,i'_{n-1}\}$ in decreasing lexicographic order*

$$\mathrm{col}(\boldsymbol{G}_2,i'_\ell) <_{\mathrm{lex}} \mathrm{col}(\boldsymbol{G}_2,i'_{\ell+1}) <_{\mathrm{lex}} \dots <_{\mathrm{lex}} \mathrm{col}(\boldsymbol{G}_2,i'_{n-1}),$$

  and let $\pi_2(i) = i$, $0 \leqslant i < \ell$ and $\pi_2(i'_j) = j$, $\ell \leqslant j < n$.
- *Define $\boldsymbol{G}_0 = \pi_2(\boldsymbol{G}_2)$ and $\pi = \pi_2 \circ \pi_1$.*

If $(\mathrm{CF}_\Psi(\mathcal{C}), \mathrm{CO}_\Psi(\mathcal{C})) = (\boldsymbol{G}_0, \pi)$ are defined then $\boldsymbol{G}_0$ spans $\pi(\mathcal{C})$. Moreover, for all code $\mathcal{C}'$ permutation equivalent to $\mathcal{C}$, we have $\mathrm{CF}_\Psi(\mathcal{C}') = \boldsymbol{G}_0$ and $\mathcal{C}' = \pi'^{-1} \circ \pi(\mathcal{C})$ where $\pi' = \mathrm{CO}_\Psi(\mathcal{C}')$.

Proof. Let $\mathcal{C}$ be a $[n,k]_q$ code admitting $\boldsymbol{G}$ as generator matrix. We assume that $I_\psi(\mathcal{C})$ contains an information set and has cardinality ℓ. Let $\boldsymbol{G}_1$, $\boldsymbol{G}_2$, $\boldsymbol{G}_0$, π_1, π_2, π be defined as in the statement and let $\mathcal{C}_b = \langle \boldsymbol{G}_b \rangle$ for $b = 0, 1, 2$. It is readily observed that the codes $\mathcal{C}$, $\mathcal{C}_1 = \mathcal{C}_2 = \pi_1(\mathcal{C})$ and $\mathcal{C}_0 = \pi_2(\mathcal{C}_1) = \pi(\mathcal{C})$ are permutation equivalent.

Let now $\mathcal{C}'$ be a code permutation equivalent to $\mathcal{C}$. We have $\mathcal{C}' = \sigma(\mathcal{C})$ for some permutation σ of $[\![0, n[\![$. We define $\boldsymbol{G}'$, $\boldsymbol{G}'_1$, $\boldsymbol{G}'_2$, $\boldsymbol{G}'_0$, $\mathcal{C}'_1$, $\mathcal{C}'_2$, $\mathcal{C}'_0$, π'_1, π'_2, π' as above, using $\mathcal{C}'$ instead of $\mathcal{C}$.

Let $I_\Psi(\mathcal{C}) = \{i_0, i_1, \ldots, i_{\ell-1}\}$ with

$$\Psi(\mathcal{C}, i_0) <_{\text{lex}} \Psi(\mathcal{C}, i_1) <_{\text{lex}} \ldots <_{\text{lex}} \Psi(\mathcal{C}, i_{\ell-1}),$$

because $\Psi(\mathcal{C}, i) = \Psi(\mathcal{C}', \sigma(i))$ for all i, we have $I_\Psi(\mathcal{C}') = \{\sigma(i_0), \sigma(i_1), \ldots, \sigma(i_{\ell-1})\}$ and

$$\Psi(\mathcal{C}', \sigma(i_0)) <_{\text{lex}} \Psi(\mathcal{C}', \sigma(i_1)) <_{\text{lex}} \ldots <_{\text{lex}} \Psi(\mathcal{C}', \sigma(i_{\ell-1})).$$

The codes $\mathcal{C}_1$ and $\mathcal{C}'_1$ are permutation equivalent, and from the above, their first ℓ coordinates match, that is $\mathcal{C}'_1 = \sigma_1(\mathcal{C}_1)$ with $\sigma_1(i) = i$ for $i \in [\![0, \ell[\![$. The codes $\mathcal{C}_1$ and $\mathcal{C}'_1$ punctured in their last $n - \ell$ coordinates are identical, let $\boldsymbol{G}_{\text{rref}} \in \mathbb{F}_q^{k \times \ell}$ denote their generator matrix in reduced row echelon form. For some $\boldsymbol{R}_2, \boldsymbol{R}'_2 \in \mathbb{F}_q^{k \times (n-\ell)}$, we have

$$\boldsymbol{G}_2 = \left(\boldsymbol{G}_{\text{rref}} \middle| \boldsymbol{R_2} \right),$$
$$\boldsymbol{G}'_2 = \left(\boldsymbol{G}_{\text{rref}} \middle| \boldsymbol{R'_2} \right).$$

Moreover, again because of the unicity of the reduced row echelon form, the columns of $\boldsymbol{R}_2$ and $\boldsymbol{R}'_2$ are identical up to the order. After reordering the last $n-\ell$ columns in lexicographic order, we obtain $\boldsymbol{G}_0 = \boldsymbol{G}'_0$ and $\mathcal{C}_0 = \pi(\mathcal{C}) = \pi'(\mathcal{C}')$, hence $\mathcal{C}' = \pi'^{-1} \circ \pi(\mathcal{C})$.

In [28], with the support splitting algorithm, the existence of a code permutation-signature which can be computed in polynomial time and which is fully discriminant for most codes was empirically established. This will allow us to define the building blocks used in the algorithms of §5:

$$\mathsf{CanonicalForm}(\mathcal{C}) = \mathrm{CF}_\Psi(\mathcal{C}) \tag{21}$$
$$\mathsf{PCESolve}(\mathcal{C}_1, \mathcal{C}_2), = \mathrm{CO}_\Psi(\mathcal{C}_2)^{-1} \circ \mathrm{CO}_\Psi(\mathcal{C}_1). \tag{22}$$

It is additionally shown in Appendix C that a signature (which is sketched below), much simpler than [28], can be designed to successfully produce a canonical form for PCE for almost all codes with a hull of dimension 0. There is a constant proportion of such codes, and this is sufficient for the LCE solvers of §5. Discarding the codewords such that $\mathcal{H}\left(\boldsymbol{c}^{-1} \star \mathcal{C}\right) \neq \{0\}$ will simply increase the number of required low weight codewords (see §6.2), increasing the complexity by a small constant multiplicative factor.

A Simple Canonical Form for PCE (sketch). Let $\mathcal{C}$ be a $[n,k]_q$ code with a hull of dimension 0. Take any generator matrix of $\mathcal{C}$ of the form $\boldsymbol{G} = (\boldsymbol{I}_k \mid \boldsymbol{R})\,\boldsymbol{P}$ with $\boldsymbol{P}$ an $n \times n$ permutation matrix. It can be proven that the following $n \times n$ matrix is defined independently of the choice of $\boldsymbol{R}$ and $\boldsymbol{P}$

$$\boldsymbol{U} = \mathrm{HSM}(\mathcal{C}) \triangleq \boldsymbol{P}^T \left(\left(\begin{array}{c|c} \boldsymbol{I}_k & \boldsymbol{R} \\ \hline \boldsymbol{R}^T & -\boldsymbol{I}_{n-k} \end{array} \right)^{-1} - \left(\begin{array}{c|c} \boldsymbol{I}_k & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} \end{array} \right) \right) \boldsymbol{P}. \tag{23}$$

We define $\Psi(\mathcal{C}, i) = \mathrm{cw}(\mathrm{row}(\boldsymbol{U}, i))$ where the complete weight is defined as $\mathrm{cw}(\boldsymbol{v}) = (c_\beta)_{\beta \in \mathbb{F}_q}$ with c_β the number of entries equal to β in $\boldsymbol{v} \in \mathbb{F}_q^n$. It is shown in Appendix C that Ψ is a permutation-signature which discriminates an information set, thus fulfilling the requirement of Theorem 5, for almost all codes when $q \geqslant 5$.

8 Application to LESS

LESS parameters always correspond to a case where we can use Algorithm 2. To provide complexity estimates, we need to specify the parameters (n, k, w, L, w', L') as well as formulas for the running time of the decoder and of the canonical form:

- *Decoding Complexity.* For the generic decoding running time, we used T_{ISD} given by (24) in §8.2. Note that we took exactly the formulas as they are given in [2,10]. In particular, it means the numbers that we give below can be fairly and accurately compared to those given in [10].
- *Canonical Form.* For the canonical form, we use the signature derived from [28] given at the end of §7. For codes over $\mathbb{F}_{127}$, the canonical form computation fails with a small probability (see §6.2) which we neglect, and we take $T_{\mathrm{cf}}(n, k, w) = n^3$ as an estimate for its running time.

In addition, with w exceeding the Gilbert-Varshamov distance, the probability to get at least one match is well approximated by

$$P^{(1)}_{\mathrm{match}}(n, k, w, L) = 1 - \left(1 - \frac{q-1}{N(n,k,w)}\right)^{L^2}.$$

Finally the estimated running time to recover the isometry (with $n' = n - w$ and $k' = k - w$) is

$$T = \frac{2L \cdot \Big(T_{\mathrm{ISD}}(n,k,w) + L' \cdot \big(T_{\mathrm{ISD}}(n',k',w') + T_{\mathrm{cf}}(n',k',w') \big) \Big)}{P^{(1)}_{\mathrm{match}}(n,k,w,L) \cdot P^{(1)}_{\mathrm{match}}(n',k',w',L')}.$$

- LESS-1 $(n,k,q) = (252, 126, 127)$. We choose $w = 114$, $L = 2^{77.11}$, $w' = 120$ and $L' = 2^{13.00}$ and obtain $T = 2^{120.12}$.
- LESS-3 $(n,k,q) = (400, 200, 127)$. We choose $w = 180$, $L = 2^{122.00}$, $w' = 192$ and $L' = 2^{27.64}$ and obtain $T = 2^{184.28}$.
- LESS-5 $(n,k,q) = (548, 274, 127)$. We choose $w = 248$, $L = 2^{174.19}$, $w' = 264$ and $L' = 2^{39.03}$ and obtain $T = 2^{248.79}$.

	Best known attack [10]	This work
LESS-1	127	120
LESS-3	195	184
LESS-5	266	249

Table 4. Bit Complexity Estimate for LCE Solvers for LESS Parameters

8.1 Experimental Results

We implemented experiments for LESS-1, LESS-3 and LESS-5 with w and w' as described previously. The experiments were done for more than 100,000 codes that are generated at random with a known matching couple. We used our algorithm 2 (One Matching Couple - Two levels) assuming that we know the matching couples in the two levels. In other words we retrieved the isometry in the second level as described in Algorithm 5 (Isometry Reconstruction).

Properness. As expected from §6, we observed no properness defect in the sense of §4.4. No decomposable codes nor false positives. The proportion of $\boldsymbol{c}^{-1} \star \mathcal{C}$ codes with non-trivial hull is about 0.008. The corresponding penalty, a factor ≈ 1.008, hardly impacts the exponents of Table 4. Every time the hull was reduced to $\{0\}$, the canonical form could be successfully computed, *i.e.* the complete weights of the rows of the matrix $\boldsymbol{U}$ defined in (23) could always discriminate an information set, enabling Theorem 5.

8.2 Formulas for the Decoding Complexity

For practical decoding complexity estimates, we used formulas that derive from [23]. It is essentially a generalization of Stern algorithm [31], similar to [18]. Formula (24) is used for security estimates in the LESS specification document [2] and reused later in [10] to estimate the cost of an attack against LESS. Recent approaches, *e.g.* [13], might give marginally better numbers but would not allow a fair comparison. The average cost, in elementary operations, of the ISD variant in [23], to sample one codeword of weight w in an $[n,k]_q$ is equal to

$$T_{\mathrm{ISD}}(n,k,w) = \min_{p,\ell} \frac{C_{\mathrm{ISD}}(n,k,w,p,\ell)}{P_{\mathrm{ISD}}(n,k,w,p,\ell)}, \tag{24}$$

where, for any optimization parameters p (even) and ℓ, the success probability of one iteration of the algorithm is equal to

$$P_{\mathrm{ISD}}(n,k,w,p,\ell) = \min\left(\frac{\binom{k/2}{p/2}^2 \binom{n-k-\ell}{w-p} N_w}{\binom{n}{w}}, 1\right)$$

and a cost of one iteration is equal to

$$C_{\mathrm{ISD}}(n,k,w,p,\ell) = \ell\left((k/2-p/2+1)+2(q-1)^{p/2}\tbinom{k/2}{p/2}\right) + \frac{\frac{pq(w-p+1)}{q-1}\left(1+\frac{q-2}{q}\right)\tbinom{k/2}{p/2}^2(q-1)^p}{q^\ell} + \frac{(n-k)^2(n+k)}{2}.$$

The above formulas are exactly those employed in [2,10], we just changed the names of some variables and assumed that k was even.

Acknowledgment. This research was partially funded by the French Agence Nationale de la Recherche and plan France 2030 program under grant ANR-22-PETQ-0008 PQ-TLS.

References

1. Babai, L.: Graph isomorphism in quasipolynomial time [extended abstract]. In: Wichs, D., Mansour, Y. (eds.) STOC. pp. 684–697. ACM (2016). https://doi.org/10.1145/2897518.2897542,
2. Baldi, M., Barenghi, A., Beckwith, L., Chou, T., Biasse, J.F., Esser, A., Gaj, K., Karl, P., Mohajerani, K., Pelosi, G., Persichetti, E., Saarinen, M.J., Santini, P., Wallace, R., Zweydinger, F.: Less: Linear equivalence signature scheme. Second round submission to NIST additional signatures call (Feb 2025), https://www.less-project.com/, version 2.0
3. Bardet, M., Brion, C., Otmani, A., Saeed, M., Sendrier, N.: Solving the linear code equivalence problem from single codeword matching. In: Heninger, N., Rosulek, M. (eds.) Advances in Cryptology - CRYPTO 2026. LNCS (2026)
4. Bardet, M., Otmani, A., Saeed-Taha, M.: Permutation code equivalence is not harder than graph isomorphism when hulls are trivial. In: Proc. IEEE Int. Symposium Inf. Theory - ISIT 2019. pp. 2464–2468 (Jul 2019), http://arxiv.org/abs/1905.00073
5. Barenghi, A., Biasse, J., Persichetti, E., Santini, P.: On the computational hardness of the code equivalence problem in cryptography. Adv. Math. Commun. **17**(1), 23–55 (2023). https://doi.org/10.3934/AMC.2022064,
6. Beullens, W.: Not enough LESS: an improved algorithm for solving code equivalence problems over $\mathbb{F}_q$. In: Dunkelman, O., Jr., M.J.J., O'Flynn, C. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 12804, pp. 387–403. Springer (2020). https://doi.org/10.1007/978-3-030-81652-0_15,
7. Both, L., May, A.: Decoding linear codes with high error rate and its impact for LPN security. In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography 2018. LNCS, vol. 10786, pp. 25–46. Springer, Fort Lauderdale, FL, USA (Apr 2018), https://doi.org/10.1007/978-3-319-79063-3_2
8. Budroni, A., Esser, A.: One pair to rule them all: An optimal algorithm for solving code equivalence via codeword search. Cryptology ePrint Archive, Paper 2026/268 (2026), https://eprint.iacr.org/2026/268

9. Budroni, A., Esser, A.: One pair to rule them all: An optimal algorithm for solving code equivalence via codeword search. In: Heninger, N., Rosulek, M. (eds.) Advances in Cryptology - CRYPTO 2026. LNCS (2026)
10. Budroni, A., Esser, A., Franch, E., Natale, A.: Two is all it takes: Asymptotic and concrete improvements for solving code equivalence. IACR Cryptol. ePrint Arch. p. 227 (2025), https://eprint.iacr.org/2025/227
11. Canto Torres, R.: Asymptotic analysis of ISD algorithms for the $q-$ary case. In: Proceedings of the Tenth International Workshop on Coding and Cryptography WCC 2017 (Sep 2017)
12. Carrier, K., Debris-Alazard, T., Meyer-Hilfiger, C., Tillich, J.P.: Reduction from sparse LPN to LPN, dual attack 3.0. In: Joye, M., Leander, G. (eds.) Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI. LNCS, vol. 14656, pp. 286–315. Springer (2024). https://doi.org/10.1007/978-3-031-58754-2_11, artifact available at https://artifacts.iacr.org/eurocrypt/2024/a10/
13. Carrier, K., Hatey, V., Tillich, J.P.: Projective space Stern decoding and application to SDitH. In: Applied Cryptography and Network Security Workshops: ACNS 2024 Satellite Workshops, AIBlock, AIHWS, AIoTS, SCI, AAC, SiMLA, LLE, and CIMSS, Abu Dhabi, United Arab Emirates, March 5–8, 2024, Proceedings, Part II. p. 29–52. Springer-Verlag, Berlin, Heidelberg (2024). https://doi.org/10.1007/978-3-031-61489-7_3,
14. Cheraghchi, M., Shagrithaya, N., Veliche, A.: Reductions between code equivalence problems. In: Proc. IEEE Int. Symposium Inf. Theory - ISIT. pp. 1–6 (2025)
15. Chou, T., Persichetti, E., Santini, P.: On linear equivalence, canonical forms, and digital signatures. Cryptology ePrint Archive, Paper 2023/1533 (2023), https://eprint.iacr.org/2023/1533
16. Chou, T., Persichetti, E., Santini, P.: On linear equivalence, canonical forms, and digital signatures. Des. Codes Cryptogr. (2025). https://doi.org/10.1007/s10623-025-01576-1, first appeared as IACR ePrint 2023/1533
17. Debris-Alazard, T.: Code-based cryptography: Lecture notes (2023). https://doi.org/10.48550/arXiv.2304.03541
18. Dumer, I.: On minimum distance decoding of linear codes. In: Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory. pp. 50–52. Moscow (1991)
19. Howgrave-Graham, N., Joux, A.: New generic algorithms for hard knapsacks. In: Gilbert, H. (ed.) Advances in Cryptology - EUROCRYPT 2010. LNCS, vol. 6110. Sringer (2010)
20. Leon, J.: Computing automorphism groups of error-correcting codes. IEEE Trans. Inform. Theory **28**(3), 496–511 (1982)
21. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $O(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology - ASIACRYPT 2011. LNCS, vol. 7073, pp. 107–124. Springer (2011)
22. Nowakowski, J.: An improved algorithm for code equivalence. In: Niederhagen, R., Saarinen, M.O. (eds.) Post-Quantum Cryptography. LNCS, vol. 15577, pp. 71–103. Springer (2025). https://doi.org/10.1007/978-3-031-86599-2_3
23. Peters, C.: Information-set decoding for linear codes over $\mathbf{F}_q$. In: Post-Quantum Cryptography 2010. LNCS, vol. 6061, pp. 81–94. Springer (2010)
24. Petrank, E., Roth, R.: Is code equivalence easy to decide? IEEE Trans. Inform. Theory **43**(5), 1602–1604 (1997)

25. Prange, E.: The use of information sets in decoding cyclic codes. IRE Transactions on Information Theory **8**(5), 5–9 (1962). https://doi.org/10.1109/TIT.1962.1057777
26. Randriambololona, H.: On products and powers of linear codes under componentwise multiplication. In: Proc. 14th International Conference on Arithmetic, Geometry, Cryptography, and Coding Theory (AGCT-14), Luminy, France. pp. 3–7 (2015)
27. Sendrier, N.: On the dimension of the hull. SIAM J. Discrete Math. **10**(2), 282–293 (1997)
28. Sendrier, N.: Finding the permutation between equivalent linear codes: The support splitting algorithm. IEEE Trans. Inform. Theory **46**(4), 1193–1203 (2000)
29. Sendrier, N., Simos, D.E.: The hardness of code equivalence over $\mathbf{F}_q$ and its application to code-based cryptography. In: Post-Quantum Cryptography 2013. LNCS, vol. 7932, pp. 203–216. Springer (2013)
30. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
31. Stern, J.: A method for finding codewords of small weight. In: Cohen, G.D., Wolfmann, J. (eds.) Coding Theory and Applications. LNCS, vol. 388, pp. 106–113. Springer (1988)

The following appendices are addition to the published version [3] of this paper.

A Distribution of $\boldsymbol{c}^{-1} \star \boldsymbol{\mathcal{C}}$

A.1 Generalities on vector spaces and their distribution

We collect here several lemmas that will be used in the following sections. We will use the fact that uniformly drawing an $[n,k]_q$ code $\mathcal{C}$ is equivalent to uniformly drawing a matrix $\boldsymbol{G} \in \mathbb{F}_q^{k\times n}$ of rank k.

Lemma 3. *The number of $[n,k]_q$ codes $\mathcal{C}_2$ having a trivial intersection with a given $[n,\ell]_q$ code $\mathcal{C}_1$ is:*

$$q^{\ell k} \begin{bmatrix} n-\ell \\ k \end{bmatrix}_q .$$

Proof. To construct a basis of a $[n,k]_q$ code $\mathcal{C}_2$ having a trivial intersection with $\mathcal{C}_1$, we have $q^n - q^\ell$ choices for the first basis vector, $q^n - q^{\ell+1}$ for the second one, to $q^n - q^{\ell+k-1}$ for the k-st vector. The number of different basis producing the same $[n,k]_q$ linear code is $\prod_{i=0}^{k-1}(q^k - q^i)$. The total number of $\mathcal{C}_2$ codes is then

$$\prod_{i=0}^{k-1} \frac{(q^n - q^{\ell+i})}{(q^k - q^i)} = q^{\ell k} \prod_{i=0}^{k-1} \frac{(q^{n-\ell} - q^i)}{(q^k - q^i)} = q^{\ell k} \begin{bmatrix} n-\ell \\ k \end{bmatrix}_q .$$

Lemma 4. *The number of $[n,k]_q$ codes not containing some given non-null vector $\boldsymbol{c}$ is $q^k \begin{bmatrix} n-1 \\ k \end{bmatrix}_q$ for $k < n$.*

Proof. This is a particular case of Lemma 3 with $\ell = 1$.

Lemma 5. *Let $\mathcal{C}_1 \subseteq \mathcal{C}$ with dimension $dim(\mathcal{C}_1) = \ell$ and $dim(\mathcal{C}) = k$. Then $\mathcal{C}_1$ has $q^{\ell(k-\ell)}$ different complementary spaces in $\mathcal{C}$.*

Proof. Up to a change of basis we can consider that $\mathcal{C}$ is $\mathbb{F}_q^k$. Then, we want to count the number of $[k, k-\ell]$-codes $\mathcal{C}_2$ that do not intersect with a given $[k, \ell]$ code $\mathcal{C}_1$. This is a particular case of Lemma 3 with $(k, k-\ell, \ell)$ instead of (n, k, ℓ), this gives $q^{\ell(k-\ell)} \begin{bmatrix} k-\ell \\ k-\ell \end{bmatrix}_q = q^{\ell(k-\ell)}$.

Lemma 6. *Let $\mathfrak{C}_1, \mathfrak{C}_2$ two random codes of length n and of respective dimension ℓ and $k-\ell$ uniformly distributed. Then:*

$$\mathbb{P}(\mathfrak{C}_1 \cap \mathfrak{C}_2 = \{0\}) = \frac{\begin{bmatrix} n-\ell \\ k-\ell \end{bmatrix}_q q^{\ell(k-\ell)}}{\begin{bmatrix} n \\ k-\ell \end{bmatrix}_q}$$

Proof. The total number of pairs of $[n, \ell]_q$ and $[n, k-\ell]_q$ codes $(\mathcal{C}_1, \mathcal{C}_2)$ is $\begin{bmatrix} n \\ \ell \end{bmatrix}_q \begin{bmatrix} n \\ k-\ell \end{bmatrix}_q$. If we fix $\mathcal{C}_1$ a $[n, \ell]_q$ code, the number of $[n, k-\ell]_q$ codes $\mathcal{C}_2$ having a trivial intersection with $\mathcal{C}_1$ is, according to Lemma 3: $q^{\ell(k-\ell)} \begin{bmatrix} n-\ell \\ k-\ell \end{bmatrix}_q$. Then the probability is

$$\mathbb{P}(\mathfrak{C}_1 \cap \mathfrak{C}_2 = \{0\}) = \frac{\begin{bmatrix} n \\ \ell \end{bmatrix}_q q^{\ell(k-\ell)} \begin{bmatrix} n-\ell \\ k-\ell \end{bmatrix}_q}{\begin{bmatrix} n \\ \ell \end{bmatrix}_q \begin{bmatrix} n \\ k-\ell \end{bmatrix}_q} = \frac{\begin{bmatrix} n-\ell \\ k-\ell \end{bmatrix}_q q^{\ell(k-\ell)}}{\begin{bmatrix} n \\ k-\ell \end{bmatrix}_q}$$

Lemma 7. *Let $I \subseteq [\![0, n[\![$ of size k and $\mathfrak{M}$ be uniformly distributed among the $k \times n$ matrix having I as an information set. Then puncturing $\mathfrak{M}$ on I returns a uniformly distributed $k \times (n-k)$ matrix.*

Lemma 8. *Let $\mathcal{C}$ be a code of length n and dimension k. Take any word $\boldsymbol{c}$ in $\mathbb{F}_q^n$ and a basis $\langle \boldsymbol{c}_1, \ldots, \boldsymbol{c}_k \rangle$ for $\mathcal{C}$. Then $\langle \boldsymbol{c} \star \boldsymbol{c}_1, \ldots, \boldsymbol{c} \star \boldsymbol{c}_k \rangle$ is a generating set for $\boldsymbol{c} \star \mathcal{C}$ and if $\mathrm{supp}(\boldsymbol{c})$ contains an information set for $\mathcal{C}$, then it is a basis.*

Proof. It is clear that $\langle \boldsymbol{c} \star \boldsymbol{c}_1, \ldots, \boldsymbol{c} \star \boldsymbol{c}_k \rangle$ is a generating set for $\boldsymbol{c} \star \mathcal{C}$.

Let $I \subset \mathrm{supp}(\mathcal{C})$ be an information set for $\mathcal{C}$, and $\boldsymbol{G}$ be the matrix whose ist row is $\boldsymbol{c}_i$. Up to a multiplication by an invertible matrix $\boldsymbol{S}$, the restriction of $\boldsymbol{SG}$ on the columns in I is the identity. Then, $\boldsymbol{c} \star \boldsymbol{SG} = \boldsymbol{S}(\boldsymbol{c} \star \boldsymbol{G})$ is a $k \times n$ matrix which is $\mathrm{diag}(\mathsf{Puncture}(\boldsymbol{c}, \overline{I}))$ on the columns in I, i.e. has rank k as $\boldsymbol{c}$ has non-zero entries on $I \subset \mathrm{supp}(\boldsymbol{c})$.

Lemma 9. *Let $\mathfrak{c}_1, \ldots, \mathfrak{c}_k$ be k linearly independent words in $\mathbb{F}_q^n$ with $\mathfrak{c}_1$ of weight w. Assume $n-w \geqslant k-1$ and $\overline{\mathrm{supp}(\mathfrak{c}_1)}$ contains an information set of $\langle \mathfrak{c}_2, \ldots, \mathfrak{c}_k \rangle$. Then the matrix whose rows are the concatenation of the vectors $\mathfrak{c}_1^{-1} \star \mathfrak{c}_2, \ldots, \mathfrak{c}_1^{-1} \star \mathfrak{c}_k$ is uniformly drawn among the $(k-1) \times w$ matrices.*

Proof. The code $\mathfrak{C}' = \langle \mathfrak{c}_2, \ldots, \mathfrak{c}_k \rangle$ is uniform among $[n, k-1]_q$ codes having an information set included in $\overline{\mathrm{supp}(\mathfrak{c}_1)}$ (this condition ensure that $\mathfrak{c}_1$ cannot belong to $\mathfrak{C}'$). When we puncture those codes on $\overline{\mathrm{supp}(\mathfrak{c}_1)}$, the resulting code is uniform in all codes $[w, \leqslant k-1]_q$.

A.2 Decomposable codes

Definition and characterization of decomposable codes We recall Definition 2: a code $\mathcal{C}$ is said to be *decomposable* if it is the direct sum $\mathcal{C}_1 \oplus \mathcal{C}_2$ of two codes with non intersecting supports $\mathrm{supp}(\mathcal{C}_1) \cap \mathrm{supp}(\mathcal{C}_2) = \emptyset$.

Example 1. We provide two simple examples of decomposable codes.

1. A $[n, k]$ code generated by $\left(\boldsymbol{I_k}\ \boldsymbol{G_0}\ \boldsymbol{0_{k,w}}\right)$ where $w < n - k$ can be seen as the direct sum of a $[n - w, k]$-code and w copies of the $[1, 0]$-code.
2. $\begin{pmatrix} \boldsymbol{G_1} & \boldsymbol{0_{\ell,n-w}} \\ \boldsymbol{0_{k-\ell,w}} & \boldsymbol{G_2} \end{pmatrix}$ generates a $[n, k]$ code and is the direct sum code built from a $[n - w, \ell]$ code generated by $\boldsymbol{G_1}$ and a $[w, k - \ell]$ code generated by $\boldsymbol{G_2}$.

Lemma 10. *Let $\mathcal{C}$ be a $[n, k]_q$ code generated by some generator matrix $\boldsymbol{G}$, and $\boldsymbol{X}_0, \ldots, \boldsymbol{X}_{n-1}$ be its columns. Then the following claims are equivalent:*

1. *$\mathcal{C}$ is decomposable,*
2. *There exists two complementary vector spaces $U_1 \oplus U_2 = \mathbb{F}_q^k$ and a partition of $[\![0, n[\![$, $I_1 \cup I_2 = [\![0, n[\![$ s.t $\boldsymbol{X}_i \in U_\ell$ for $i \in I_\ell$, $\ell \in \{1, 2\}$.*

Proof. It is pretty clear that if $\mathcal{C}$ is decomposable, then there exists such subspaces U_1, U_2 in $\mathbb{F}_q^k$. Indeed, there exists a generator matrix of the shape

$$\begin{pmatrix} \boldsymbol{G_1} & \boldsymbol{0_{\ell,n-w}} \\ \boldsymbol{0_{k-\ell,w}} & \boldsymbol{G_2} \end{pmatrix}$$

up to permutation of the columns and $U_1 \triangleq \langle \boldsymbol{e_1}, \ldots, \boldsymbol{e_\ell} \rangle, U_2 \triangleq \langle \boldsymbol{e_{\ell+1}}, \ldots, \boldsymbol{e_k} \rangle$ verify the desired inclusion.

Assume there exists $U_1 \oplus U_2 = \mathbb{F}_q^k$ and $I_1 \cup I_2 = [\![0, n[\![$ s.t $\boldsymbol{X}_i \in U_\ell$ for $i \in I_\ell$ and $\ell \in \{1, 2\}$. We introduce $(\boldsymbol{u}_i^{(1)})_{i \in I_1}$ a basis for U_1, $(\boldsymbol{u}_i^{(2)})_{i \in I_2}$ a basis for U_2 and $(\boldsymbol{e}_i)_{i \in [\![0,n[\![}$ the basis in which the generator matrix of $\mathcal{C}$ is $\boldsymbol{G}$. We note $\boldsymbol{S}$ the invertible $k \times k$ matrix to change the basis from $(\boldsymbol{e}_i)_{i \in [\![0,n[\![}$ to $(\boldsymbol{u}_i^{(1)})_{i \in I_1} \cup (\boldsymbol{u}_i^{(2)})_{i \in I_2}$. In this basis, by definition $\boldsymbol{S}\boldsymbol{X}_i \in U_1$ for $i \in I_1$ and $\boldsymbol{S}\boldsymbol{X}_i \in U_2$ for $i \in I_2$. We also have that $\mathsf{Submatrix}(\boldsymbol{G}, [\![0, \ell[\![, I_1)$ generates a code $\mathcal{C}_1$ and $\mathsf{Submatrix}(\boldsymbol{G}, [\![\ell, k[\![, I_2)$ generates a code $\mathcal{C}_2$ verifying $\mathcal{C}_1 \oplus \mathcal{C}_2 = \mathcal{C}$ and $\mathrm{supp}(\mathcal{C}_1) \cap \mathrm{supp}(\mathcal{C}_2) = \emptyset$.

We assume in this paper that the fraction of decomposable codes is exponentially small.

Schur product and decomposable codes Some technical results on decomposable codes are requested for the proof of Lemma 13 and Theorem 2. Those results are deduced from [26].

Definition 7 (Decomposition under a partition [26, Def. 2.8]). *A code $\mathcal{C}$ decomposes under a partition $\mathcal{P}$ of its support if there exists for all $P \in \mathcal{P}$ a subcode $\mathcal{C}_P \subseteq \mathcal{C}$ with $supp(\mathcal{C}_P) = P$ and*

$$\mathcal{C} = \bigoplus_{P \in \mathcal{P}} \mathcal{C}_P.$$

Definition 8 (Proper Stabilizing Algebra [26, Def. 2.6]). *Let the proper stabilizing algebra of a code $\mathcal{C}$ be the algebra $(\mathcal{A}(\mathcal{C}), +, \star, \cdot)$ where*

$$\mathcal{A}(\mathcal{C}) \triangleq \{\boldsymbol{a} \in \mathbb{F}_q^n \mid a \star \mathcal{C} \subseteq \mathcal{C}, \mathrm{supp}(\boldsymbol{a}) \subseteq \mathrm{supp}(\mathcal{C})\}.$$

Lemma 11 (Finest Partition of a Code [26, Lem.-Def. 2.10]). *There exists a (unique) finest partition of a code $\mathcal{C}$, denoted $\mathcal{P}(\mathcal{C})$, under which $\mathcal{C}$ is decomposed: for any partition $\mathcal{P}$ decomposing $\mathcal{C}$ and any P in $\mathcal{P}(\mathcal{C})$, there exists P' in $\mathcal{P}$ such that $P \subseteq P'$.*

In particular, we are interested in corollaries of the following proposition:

Proposition 10. *[26, Prop. 2.11] Let $\mathcal{C}$ be a $[n,k]_q$ code. Then:*

$$dim(\mathcal{A}(\mathcal{C})) = |\mathcal{P}(\mathcal{C})|$$

More precisely, if $\mathcal{P}(\mathcal{C}) = \{A_1, \ldots, A_r\}$ with $r \in \mathbb{N}^*$, then:

$$\mathcal{A}(\mathcal{C}) = \langle \mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_r} \rangle = \langle \mathbb{1}_{A_1} \rangle \oplus \ldots \oplus \langle \mathbb{1}_{A_r} \rangle$$

Lemma 12. *Let $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathcal{C}$ of full support where $\mathcal{C}$ is a non-decomposable $[n,k]_q$ code such that $\boldsymbol{c}_1 \star \mathcal{C} = \boldsymbol{c}_2 \star \mathcal{C}$. Then*

$$\langle \boldsymbol{c}_1 \rangle = \langle \boldsymbol{c}_2 \rangle$$

Proof. Since $\mathcal{C}$ is non-decomposable, Proposition 10 ensures that $\mathcal{A}(\mathcal{C}) = \langle \mathbb{1}_n \rangle$. Then

$$\boldsymbol{c}_1 \star \mathcal{C} = \boldsymbol{c}_2 \star \mathcal{C} \Leftrightarrow \boldsymbol{c}_1 \star \boldsymbol{c}_2^{-1} \star \mathcal{C} = \mathcal{C} \Leftrightarrow \boldsymbol{c}_1 \star \boldsymbol{c}_2^{-1} \in \langle \mathbb{1}_n \rangle \Leftrightarrow \langle \boldsymbol{c}_1 \rangle = \langle \boldsymbol{c}_2 \rangle.$$

The two corollaries 1 and 2 states results on $\mathcal{C}$ a full support $[n,k]_q$ code decomposable in $\mathcal{C}_1 \oplus \ldots \oplus \mathcal{C}_r$ where the $(\mathcal{C}_i)_{i \in [\![1,r]\!]}$ are non-decomposable and there support non intersecting.

Corollary 1. *Under the previous assumptions, there are $(q-1)^r$ different $\boldsymbol{c}$ such $\boldsymbol{c} \star \mathcal{C} = \mathcal{C}$.*

Proof. The previous Proposition 10 describes all possible such $\boldsymbol{c}$. Numbering them gives $(q-1)^r$ for all possible α_i with $1 \leqslant i \leqslant r$.

Corollary 2. *Under the previous assumptions, there are $(q-1)^{n-r}$ different $[n,k]_q$ code $\mathcal{C}'$ such there exists $\boldsymbol{c} \in \mathbb{F}_q^*$ verifying $\boldsymbol{c} \star \mathcal{C}' = \mathcal{C}$.*

Proof. For all $(q-1)^n$ full support codewords $\boldsymbol{c}$, there exists $\mathcal{C}' \triangleq \boldsymbol{c}^{-1} \star \mathcal{C}$ such that $\boldsymbol{c} \star \mathcal{C}' = \mathcal{C}$. Nevertheless, by Corollary 1, there are $(q-1)^r$ codewords $\boldsymbol{c}'$ stabilizing $\mathcal{C}$ by Schur product: $\boldsymbol{c}' \star \mathcal{C} = \mathcal{C}$. If $\boldsymbol{c}'$ stabilize $\mathcal{C}$ and $\mathcal{C}' \triangleq \boldsymbol{c}^{-1} \star \mathcal{C}$, then $\boldsymbol{c}' \star \mathcal{C}' = \boldsymbol{c}' \star \boldsymbol{c}^{-1} \star \mathcal{C} = \boldsymbol{c}^{-1} \star \boldsymbol{c}' \star \mathcal{C} = \boldsymbol{c}^{-1} \star \mathcal{C} = \mathcal{C}'$. Thus the number of such different codes $\mathcal{C}'$ is $(q-1)^{n-r}$.

A.3 Distribution of $\boldsymbol{c^{-1} \star \mathfrak{C}}$

Let $\mathfrak{C}$ be a $[n,k]_q$ code, and $\mathfrak{c} \in \mathfrak{C}$ a codeword of weight w. To analyze the distribution of the Schur product $\mathfrak{c}^{-1} \star \mathfrak{C}$, we can consider a specific generator matrix for $\mathfrak{C}$ having its first row representing $\mathfrak{c}$ as follows (we draw the matrices as if the support of $\mathfrak{c}$ is $[\![n-w, n[\![$):

$$\begin{pmatrix} \mathbf{0_{1\times(k-\ell)}} & \mathfrak{c} \\ \boldsymbol{I_{k-\ell}} & \mathfrak{G_1} \\ \mathbf{0_{(\ell-1)\times(k-\ell)}} & \mathfrak{G_2} \end{pmatrix} \qquad \text{if } \ell \triangleq k-(n-w) \geqslant 0,$$

$$\begin{pmatrix} \mathbf{0_{1\times(k-1)}} & \mathbf{0_{1\times(n-w-k+1)}} & \mathfrak{c} \\ \boldsymbol{I_k} & \mathfrak{G_3} & \mathfrak{G_1} \end{pmatrix} \qquad \text{if } k < n-w.$$

Thus a generator matrix (not necessary full rank) for $\mathfrak{c}^{-1} \star \mathfrak{C}$ is

$$\begin{pmatrix} \mathbb{1}_w \\ \mathfrak{c}^{-1} \star \mathfrak{G_1} \\ \mathfrak{c}^{-1} \star \mathfrak{G_2} \end{pmatrix} \text{ or } \begin{pmatrix} \mathbb{1}_w \\ \mathfrak{c}^{-1} \star \mathfrak{G_1} \end{pmatrix}.$$

The distribution of the two matrices $\mathfrak{c}^{-1} \star \mathfrak{G_1}$ and $\mathfrak{c}^{-1} \star \mathfrak{G_2}$ are different. Indeed, one comes from a matrix of rank $\ell - 1$ punctured on null columns, ensuring the punctured matrix to be of same rank, and the other one is built from a matrix of rank $k-\ell$ punctured on an information set which might decrease its rank. Thanks to Lemma 7 and 9 we will see that $\boldsymbol{c}^{-1} \star \boldsymbol{G_1}$ is uniformly distributed among the $(k-\ell) \times w$ matrices. $\boldsymbol{c}^{-1} \star \boldsymbol{G_2}$ has a very particular distribution among the $(\ell-1)\times w$ matrices which rows do not generate $\mathbb{1}_w$. The distribution for the codes generated by these matrices is exponentially close to the uniform distribution among the codes not containing the vector $\mathbb{1}_w$. Lemma 13 will describe how much close this distribution is to be uniform.

In order to merge those results and to understand the distribution of $\boldsymbol{c}^{-1} \star \mathcal{C}$ we will need more Lemma which are used in the proof of Theorem 3.

Let $N_w(\mathcal{C})$ be the number of codewords in a code $\mathcal{C}$ of weight w.

Lemma 13. *Let $\mathfrak{c}$ be a uniformly distributed word of full support, $\mathfrak{C}_0$ be a uniformly distributed $[n, k-1]_q$ code not containing $\mathfrak{c}$.*

If $\mathcal{C}_0$ is a $[n, k-1]_q$ code not containing $\mathbb{1}_n$, then

$$\mathbb{P}\left(\mathfrak{c}^{-1} \star \mathfrak{C}_0 = \mathcal{C}_0\right) = \frac{(q-1)^{n-1}}{\begin{bmatrix} n \\ k-1 \end{bmatrix}_q} \; \frac{q-1}{(q-1)^n - N_n(\mathcal{C}_0)}$$

where $N_n(\mathcal{C}_0)$ is the number of codewords of full support in $\mathcal{C}_0$.

Proof. Assume $\mathcal{C}_0$ is decomposed as a sum of r non-decomposable codes.

By Corollary 2 , there are $(q-1)^{n-r}$ elements in $\mathcal{P}(\mathcal{C}_0)$ the set of all codes $\mathcal{C}'$ such that there exists $\boldsymbol{c}$ of full support verifying $\boldsymbol{c}^{-1} \star \mathcal{C}' = \mathcal{C}_0$.

We note $\mathcal{P}(\mathcal{C}_0)$ the set of codes $\mathcal{C}$ for which there exists (at least) a codeword $\boldsymbol{c} \notin \mathcal{C}$ such that $\boldsymbol{c}^{-1} \star \mathcal{C} = \mathcal{C}_0$. By Corollary 1 there exists $(q-1)^r$ such codeword

$\boldsymbol{c}$ verifying $\boldsymbol{c}^{-1} \star \mathcal{C}' = \mathcal{C}_0$ for a given $\mathcal{C}' \in \mathcal{P}(\mathcal{C}_0)$. Moreover, if $\mathbb{1}_w \notin \mathcal{C}_0$, we can't have $\boldsymbol{c} \in \mathcal{C}'$. Then all those codewords are among the full support codewords not in $\mathcal{C}'$. Yet $N_n(\mathcal{C}_0) = N_n(\mathcal{C}')$. Thus there are $(q-1)^n - N_n(\mathcal{C}_0)$ full support codewords not in $\mathcal{C}'$.

Thanks to those remarks we get:

$$\begin{aligned}\mathbb{P}\left(\mathfrak{c}^{-1} \star \mathfrak{C}_0 = \mathcal{C}_0\right) &= \sum_{\mathcal{C} \in \mathcal{P}(\mathcal{C}_0)} \mathbb{P}\left(\mathfrak{C}_0 = \mathcal{C}\right)\ \mathbb{P}\left(\mathfrak{c}^{-1} \star \mathcal{C} = \mathcal{C}_0\right) \\ &= \frac{(q-1)^{n-r}}{\begin{bmatrix} n \\ k-1 \end{bmatrix}_q}\ \frac{(q-1)^r}{(q-1)^n - N_n(\mathcal{C}_0)}\end{aligned}$$

In order to use this previous general Lemma, we need to approximate the quantity $(q-1)^n - N_n\left(\mathcal{C}_0\right)$. The following lemma claims that the fraction of codes $\mathcal{C}$, such that $N_n(\mathcal{C})$ is close to the average value (among all codes), is exponentially close to 1 when $n \to \infty$.

Lemma 14 ([17, Prop. 2.3.3]). *Let $\mathfrak{C}$ be a uniformly distributed $[n, \lfloor Rn \rfloor]_q$-code with $0 < R < 1$. Then:*

$$\mathbb{P}\left(|N_n(\mathfrak{C}) - E[N_n(\mathfrak{C})]| \geqslant \left(E[N_n(\mathfrak{C})]\right)^{3/4}\right) \leqslant (q-1)\ q^{-\frac{1}{2}[R - \log_q\left(\frac{q}{q-1}\right)]n}.$$

Whenever $R > \log_q\left(\frac{q}{q-1}\right)$ (for $q = 2$, there is no such R), the fraction $\frac{N_n(\mathfrak{C})}{E[N_n(\mathfrak{C})]}$ is exponentially close to one:

$$\frac{N_n(\mathfrak{C})}{E[N_n(\mathfrak{C})]} = 1 + O\left(q^{-\frac{1}{4}[R - \log_q(\frac{q}{q-1})]n}\right)$$

with a probability exponentially close to one:

$$1 - O\left(q^{-\frac{1}{2}[R - \log_q\left(\frac{q}{q-1}\right)]n}\right).$$

Notation 1 *Let $H_q^{n,k}(\boldsymbol{c})$ the number of $[n,k]_q$ code not containing $\langle \boldsymbol{c} \rangle$.*

Corollary 3. *Let $\mathfrak{C}$ be a uniformly distributed $[n,k]_q$ code and $\mathfrak{c}$ a full support codeword not in $\mathfrak{C}$. Then for $1 - O\left(q^{-\frac{1}{2}[R - \log_q\left(\frac{q}{q-1}\right)]n}\right)$ percent of $[n,k]_q$ code $\mathcal{C}_0$ not containing $\mathbb{1}_n$:*

$$\mathbb{P}\left(\mathfrak{c}^{-1} \star \mathfrak{C} = \mathcal{C}_0\right) = H_q^{n,k}(\mathbb{1}_w)^{-1}\ \left(1 + O\left(q^{-\frac{1}{2}mn}\right)\right)$$

where $m \triangleq 1 - \frac{1}{4}[3R + \log_2(\frac{q}{q-1})]$. The hidden constant in the O is in $[-1, 2]$.

Proof. By Lemma 13:

$$\begin{aligned}\mathbb{P}\left(\mathfrak{c}^{-1}\star\mathfrak{C}=\mathcal{C}_0\right) &= \frac{(q-1)^{n-1}}{\begin{bmatrix}n\\k-1\end{bmatrix}_q}\ \frac{q-1}{(q-1)^n-N_n(\mathcal{C}_0)}\\ &= \frac{1}{\begin{bmatrix}n\\k-1\end{bmatrix}_q}\ \frac{1}{1-\frac{N_n(\mathcal{C}_0)}{(q-1)^n}}\\ &= \frac{1}{\begin{bmatrix}n\\k-1\end{bmatrix}_q}\ \frac{1}{1-\frac{N_n(\mathcal{C}_0)}{E[N_n(\mathfrak{C})]}\ q^{-(n-k)}}\end{aligned}$$

By Lemma 14, for $1-O\left(q^{-\frac{1}{2}[R-\log_q\left(\frac{q}{q-1}\right)]n}\right)$ of $[n,k]_q$ code $\mathcal{C}_0$, we have:

$$\frac{N_n(\mathfrak{C})}{E[N_n(\mathfrak{C})]}-1=O\left(q^{-\frac{1}{4}[R-\log_q(\frac{q}{q-1})]n}\right)$$

Then we have for $\gamma\triangleq\frac{1}{4}[R-\log_q(\frac{q}{q-1})]$:

$$\begin{aligned}\frac{1}{1-\frac{N_n(\mathcal{C}_0)}{E[N_n(\mathfrak{C})]}\ q^{-(n-k)}} &= \frac{1}{1-q^{-(n-k)}+(1-\frac{N_n(\mathcal{C}_0)}{E[N_n(\mathfrak{C})]})\ q^{-(n-k)}}\\ &= \frac{1}{1-q^{-(n-k)}}\ \frac{1}{1+(1-\frac{N_n(\mathcal{C}_0)}{E[N_n(\mathfrak{C})]})\ \frac{q^{-(n-k)}}{1-q^{-(n-k)}}}\\ &= \frac{1}{1-q^{-(n-k)}}\ \left(1+O\left(\sqrt{(1-\frac{N_n(\mathcal{C}_0)}{E[N_n(\mathfrak{C})]})\ \frac{q^{-(n-k)}}{1-q^{-(n-k)}}}\right)\right)\\ &= \frac{1}{1-q^{-(n-k)}}\ \left(1+O\left(\sqrt{q^{-\gamma n}\ \frac{q^{-(n-k)}}{1-q^{-(n-k)}}}\right)\right)\\ &= \frac{1}{1-q^{-(n-k)}}\ \left(1+O\left(\sqrt{q^{-(\gamma+1-R)n}}\right)\right)\\ &= \frac{1-q^{-n}}{1-q^{-(n-k)}}\ \left(1+O\left(q^{-\frac{1}{2}mn}\right)\right)\end{aligned}$$

where $m\triangleq 1-\frac{1}{4}[3R+\log_2(\frac{q}{q-1})]$.

Remark 5. This corollary shows how close is the distribution of our Schur product $\mathfrak{c}^{-1}\star\mathfrak{C}$ whenever $\mathfrak{c}$ is of full support to the uniform distribution among the $[n,k]_q$ code not containing $\mathbb{1}_n$.

Lemma 15. *Let $\mathfrak{C}$ a random code uniformly drawn in the $[n,k]_q$ code, $\mathfrak{c}$ uniformly distributed in the codewords of $\mathfrak{C}$ of weight $w>k$ such that $\overline{\mathrm{supp}\,(\mathfrak{c})}$ belongs to an information set, $\mathfrak{C}'=\langle \mathbf{c}_2,\ldots,\mathbf{c}_k\rangle$ a basis of a complementary space of $\langle\mathfrak{c}\rangle$ in $\mathfrak{C}$ and $\mathcal{C}_0$ a given $[n,k-1]_q$ code. Then the probability $\mathbb{P}(\mathfrak{c}^{-1}\star\mathfrak{C}'=\mathcal{C}_0\mid dim(\mathfrak{c}^{-1}\star\mathfrak{C}')=k-1)$ verifies*

1. *either* $\mathbb{P}(\mathfrak{c}^{-1}\star\mathfrak{C}' = \mathcal{C}_0 \mid dim(\mathfrak{c}^{-1}\star\mathfrak{C}') = k-1) = \frac{(q^w-1)\left(1+O(q^{-Mw})\right)}{(q^w-q^{\ell-1})\begin{bmatrix} w \\ k-1 \end{bmatrix}_q}$ *if* $\mathbb{1}_w \notin \mathcal{C}_0$
2. *or* $\mathbb{P}(\mathfrak{c}^{-1} \star \mathfrak{C}' = \mathcal{C}_0 \mid dim(\mathfrak{c}^{-1} \star \mathfrak{C}') = k-1) = \frac{q^{k-1}-q^{\ell-1}}{q^{k-1}-1} \frac{(q^w-1)\left(1+O(q^{-Mw})\right)}{(q^w-q^{\ell-1})\begin{bmatrix} w \\ k-1 \end{bmatrix}_q}$ *whenever* $\mathbb{1}_w \in \mathcal{C}_0$,

where $\ell \triangleq k-(n-w)$, $R_\ell \triangleq \ell/w$, $\gamma \triangleq \frac{1}{4}[R_\ell - \log_q(\frac{q}{q-1})]$, $M \triangleq \gamma + 1 - R_\ell$ *and the hidden constant in the* O *is in* $[-1, 2]$.

Proof. Let split $\mathfrak{C}'$ in $\mathfrak{C}'_1 \oplus \mathfrak{C}'_2$ where $\mathfrak{C}'_1$ has an information set in $\overline{\mathrm{supp}(\mathfrak{c})}$ and $\mathrm{supp}(\mathfrak{C}'_2) \subseteq \mathrm{supp}(\mathfrak{c})$. Thus $\mathfrak{c}^{-1} \star \mathfrak{C}'_1$ is uniformly distributed in the $[w, n-w]_q$-codes by Lemma 9 and $\mathfrak{c}^{-1} \star \mathfrak{C}'_2$ is distributed in the $[w, \ell-1]_q$-codes not containing $\mathbb{1}_w$ as described in Lemma 13.

With this modeling, $\mathfrak{c}^{-1}\star\mathfrak{C}' = \mathcal{C}_0$ if and only if $\mathfrak{c}^{-1}\star\mathfrak{C}'_2$ is equal to one of the subcodes of $\mathcal{C}_0$ not containing $\mathbb{1}_w$ and $\mathfrak{c}^{-1}\star\mathfrak{C}'_1$ is equal to one of the $q^{(\ell-1)(n-w)}$ complementary space of $\mathfrak{c}^{-1} \star \mathfrak{C}'_2$ in $\mathfrak{c}^{-1} \star \mathfrak{C}'$ (see Lemma 5 for the number of complementary space).

We define p_2 the probability for $\mathfrak{c}^{-1} \star \mathfrak{C}'_2$ to be one of those subcodes of $\mathcal{C}_0$ and p_1 the probability for $\boldsymbol{c}^{-1} \star \mathfrak{C}'_1$ to be a complementary space of $\boldsymbol{c}^{-1} \star \mathfrak{C}'_2$ in $\mathcal{C}_0$. Thus

$$\mathbb{P}(\mathfrak{c}^{-1} \star \mathfrak{C}' = \mathcal{C}_0 \mid dim(\mathfrak{c}^{-1} \star \mathfrak{C}') = k-1) = p_1.p_2$$

By Lemma 4, there are $\frac{q^{k-1}-q^{\ell-1}}{q^{k-1}-1} \begin{bmatrix} k-1 \\ \ell-1 \end{bmatrix}_q$ subcodes of $\mathcal{C}_0$ not containing $\mathbb{1}_w$ of dimension $\ell-1$ if $\mathbb{1}_w \in \mathcal{C}_0$ and $\begin{bmatrix} k-1 \\ \ell-1 \end{bmatrix}_q$ such subcodes if $\mathbb{1}_w \notin \mathcal{C}_0$. Then thanks to Corollary 3, if $\mathbb{1}_w \notin \mathcal{C}_0$, p_2 is equal to:

$$p_2 = \frac{(q^w-1)\begin{bmatrix} k-1 \\ \ell-1 \end{bmatrix}_q}{(q^w-q^{\ell-1})\begin{bmatrix} w \\ \ell-1 \end{bmatrix}_q} \left(1 + O\left(q^{-\frac{1}{2}mw}\right)\right)$$

and if $\mathbb{1}_w \in \mathcal{C}_0$, p_2 is equal to:

$$p_2 = \frac{q^{k-1}-q^{\ell-1}}{q^{k-1}-1} \frac{(q^w-1)\begin{bmatrix} k-1 \\ \ell-1 \end{bmatrix}_q}{(q^w-q^{\ell-1})\begin{bmatrix} w \\ \ell-1 \end{bmatrix}_q} \left(1 + O\left(q^{-\frac{1}{2}mw}\right)\right)$$

where $m = 1 - \frac{1}{4}[3R_{w,\ell-1} + \log_2(\frac{q}{q-1})]$ and $R_{w,\ell-1} = \frac{\ell-1}{w} \simeq R - 1 + \tau$.

We are assuming that $dim(\mathfrak{C}'_1 \oplus \mathfrak{C}'_2) = k-1$ and $\boldsymbol{c}^{-1} \star \mathfrak{C}'_2 \subseteq \mathcal{C}_0$. Now p_1 is the number of complementary space of $\boldsymbol{c}^{-1} \star \mathfrak{C}'_2$ in $\mathcal{C}_0$ divided by the number of complementary space of $\boldsymbol{c}^{-1} \star \mathfrak{C}'_2$ in $\mathbb{F}_q^w$:

$$p_1 = \frac{q^{(\ell-1)(k-\ell)}}{q^{(\ell-1)(k-\ell)}\begin{bmatrix} w-\ell+1 \\ k-\ell \end{bmatrix}_q} = \begin{bmatrix} w-\ell+1 \\ k-\ell \end{bmatrix}_q^{-1}$$

Applying the equality $\begin{bmatrix} w \\ k-1 \end{bmatrix}_q^{-1} = \frac{\begin{bmatrix} k-1 \\ \ell-1 \end{bmatrix}_q}{\begin{bmatrix} w-(\ell-1) \\ k-\ell \end{bmatrix}_q \begin{bmatrix} w \\ \ell-1 \end{bmatrix}_q}$ to $p_1 p_2$ will provide the expected results.

Proof (of Theorem 3). The theorem is deduced from the previous Lemma 15 and some remarks on the following fractions : $\frac{q^w-1}{q^w-q^{\ell-1}} = \left(1+O\left(q^{-\frac{(1-R)}{2}n}\right)\right)$ and $\frac{q^{k-1}-q^{\ell-1}}{q^{k-1}-1} = \left(1+O\left(q^{-\frac{R}{2}n}\right)\right)$ if $1-\tau \geqslant R/2$.

Indeed, the exponents verify $\frac{R}{2} \leqslant \frac{(1-R)}{2}$ since $R \leqslant 1/2$ and if $\frac{R}{8} \geqslant \frac{1}{8(q-1)}$

$$\begin{aligned}\frac{1}{2}m = \frac{1}{2}[1-\frac{1}{4}[3R+\log_2(\frac{q}{q-1})]] &\geqslant \frac{1-\frac{3R}{4}}{2} - \frac{1}{8(q-1)} \\ &\geqslant \frac{1-R}{2} + \frac{R}{8} - \frac{1}{8(q-1)} \geqslant \frac{R}{2}.\end{aligned}$$

A.4 Distribution of a hull for $\boldsymbol{c^{-1}\star\mathcal{C}}$ codes

Thanks to the previous subsection, we will manage to reduce the analysis of the dimension of the hull to codes described as a sum $\mathcal{C} \triangleq \langle \mathbb{1}_w \rangle + \mathcal{C}_0$ where $\mathcal{C}_0$ is uniformly distributed in the $[w, k-1]_q$ codes.

We prove here Theorem 4. The only particularity of our code compared to the existing results on hulls is the sum with $\langle \mathbb{1}_w \rangle$. From the decomposition of the hull as

$$\begin{aligned}\mathcal{H}\left(\langle \mathbb{1}_w \rangle + \mathcal{C}_0\right) &= \left(\langle \mathbb{1}_w \rangle + \mathcal{C}_0\right) \cap \left(\langle \mathbb{1}_w \rangle + \mathcal{C}_0\right)^\perp \\ &= \underbrace{\langle \mathbb{1}_w \rangle \cap \langle \mathbb{1}_w \rangle^\perp}_{\mathcal{H}(\langle \mathbb{1}_w \rangle)} \cap \mathcal{C}_0^\perp + \langle \mathbb{1}_w \rangle^\perp \cap \underbrace{\mathcal{C}_0 \cap \mathcal{C}_0^\perp}_{\mathcal{H}(\mathcal{C}_0)}\end{aligned}$$

we derive the upper bounds:

Lemma 16. *Let $\mathfrak{C}_0$ uniformly distributed in the $[n, k-1]_q$ codes. Then:*

$$\mathbb{P}\left(\mathcal{H}\left(\langle \mathbb{1}_w \rangle + \mathfrak{C}_0\right) \neq \{0\}\right) \leqslant \begin{cases} \mathbb{P}(\mathcal{H}\left(\mathfrak{C}_0\right) \neq \{0\}) & , \text{ if } q \not| w \\ \mathbb{P}(\mathbb{1}_{\boldsymbol{w}} \in \mathfrak{C}_0^\perp) + \mathbb{P}(\mathcal{H}\left(\mathfrak{C}_0\right) \neq \{0\}) & , \text{ if } q | w \end{cases}$$

Remark 6. It is pretty clear that $\mathcal{H}\left(\langle \mathbb{1}_w \rangle\right) = \langle \mathbb{1}_w \rangle$ or $\{0\}$

Let $\mathfrak{D}$ a uniformly distributed $[n, k]_q$ code. We define the function

$$prop \ : \ \mathcal{C} \in Gr^k(\mathbb{F}_q^n) \ \mapsto \ \mathbb{P}_{\substack{\mathfrak{c} \xleftarrow{\$} \mathcal{C} \\ |\mathfrak{c}|=w}} \left(\mathcal{H}\left(\mathfrak{c}^{-1} \star \mathcal{C}\right) \neq \{0\}\right)$$

and we introduce the random variable $\mathfrak{X} \triangleq prop(\mathfrak{D})$. If $\mathfrak{D} = \mathcal{C}$, then $\mathfrak{X}$ *is the proportion of codewords $\boldsymbol{c}$ of $\mathcal{C}$ such that $\boldsymbol{c}^{-1} \star \mathcal{C}$ has a non-trivial hull* and we have the following results on $\mathfrak{X}$:

Lemma 17. *Let $\mathfrak{C}$ (resp. $\mathfrak{C}_0$) uniformly distributed in $[n, k]_q$ code (resp. $[n, k-1]_q$ codes) and $\mathfrak{c}$ uniformly distributed in $\mathfrak{C}$ of weight w. Then:*

1. $\mathbb{E}(\mathfrak{X}) = \mathbb{P}\left(\mathcal{H}\left(\mathfrak{c}^{-1} \star \mathfrak{C}\right) \neq \{0\}\right)$

2. $\mathbb{V}ar(\mathfrak{X}) \leqslant \mathbb{E}(\mathfrak{X})$
3. $\mathbb{E}(\mathfrak{X}) = \mathbb{P}\left(\mathcal{H}\left(\langle \mathbb{1}_w \rangle + \mathfrak{C}_0\right) \neq \{0\}\right)\left(1 + O\left(q^{-Mw}\right)\right)$ *where* $M = R/2$.

Proof. 1. comes from

$$\mathbb{P}\left(\mathcal{H}\left(\mathfrak{c}^{-1} \star \mathfrak{C}\right) \neq \{0\}\right) = \sum_{\mathcal{C}} \mathbb{P}\left(\mathcal{H}\left(\mathfrak{c}^{-1} \star \mathcal{C}\right) \neq \{0\}\right) \mathbb{P}\left(\mathfrak{C} = \mathcal{C}\right).$$

2. is trivial since $|\mathfrak{X}| < 1$.
3. Note $\Omega \triangleq \{\langle \mathbb{1}_w \rangle + \mathcal{C}_0 \mid \mathcal{C}_0 \text{ a } [n,k]_q\text{-code}\}$. Then, by Theorem 3,

$$\begin{aligned}
\mathbb{P}\left(\mathcal{H}\left(\mathfrak{c}^{-1} \star \mathfrak{C}\right) \neq \{0\}\right) &= \sum_{\mathcal{C} \in \Omega} \mathbb{P}\left(\mathfrak{c}^{-1} \star \mathfrak{C} = \mathcal{C}\right) \mathbb{P}\left(\mathcal{H}\left(\mathcal{C}\right) \neq \{0\}\right) \\
&= \sum_{\mathcal{C} \in \Omega} \mathbb{P}\left(\mathcal{H}\left(\mathcal{C}\right) \neq \{0\}\right) \mathbb{P}\left(\langle \mathbb{1}_w \rangle + \mathfrak{C}_0 = \mathcal{C}\right) \quad \left(1 + O(q^{-Mw})\right) \\
&= \mathbb{P}\left(\mathcal{H}\left(\langle \mathbb{1}_w \rangle + \mathfrak{C}_0\right) \neq \{0\}\right)\left(1 + O\left(q^{-Mw}\right)\right)
\end{aligned}$$

The probabilistic modeling for our attack assumes $\mathcal{C}$ is given in the public key and we draw uniformly $\mathfrak{c}$ in $\mathcal{C}$ of weight w. Yet we only have information whenever $\mathfrak{C}$ and $\mathfrak{c}$ are both random variables. In order to bridge the gap between the modeling and our results, we will use Bienaymé-Chebychev inequality and Lemma 16 17 to prove Theorem 4.

Proof (Theorem 4).

Note $U(\mathbb{E}(\mathfrak{X}))$ the maximum upper bound from Lemma 16. For $\mathfrak{C}_0$ a uniformly distributed $[w, k-1]_q$ code, $\mathbb{P}\left(\mathbb{1}_w \in \mathfrak{C}_0\right) = q^{-(w-k+1)}$ and thus by Lemma 1 provides:

$$U(\mathbb{E}(\mathfrak{X})) = 1 - H_0\ \left(1 + O\left(q^{-(\lfloor k/2 \rfloor)}\right)\right)$$

Applying to $\mathfrak{X}$ the Bienaymé-Chebychev inequality with a threshold set to $\mathbb{E}(\mathfrak{X})^{\frac{2}{3}}$ and use (ii) of Lemma 17, we get

$$\mathbb{P}\left(|\mathfrak{X} - \mathbb{E}(\mathfrak{X})| \geqslant \mathbb{E}(\mathfrak{X})^{\frac{2}{3}}\right) \leqslant \mathbb{E}(\mathfrak{X})^{\frac{5}{9}} \leqslant \mathbb{E}(\mathfrak{X})^{\frac{1}{2}} \leqslant U(\mathbb{E}(\mathfrak{X}))^{\frac{1}{2}}$$

We conclude by using $\mathbb{P}\left(|\mathfrak{X} - \mathbb{E}(\mathfrak{X})| \geqslant \mathbb{E}(\mathfrak{X})^{\frac{2}{3}}\right) \geqslant \mathbb{P}\left(|\mathfrak{X} - \mathbb{E}(\mathfrak{X})| \geqslant U(\mathbb{E}(\mathfrak{X}))^{\frac{2}{3}}\right)$.

A.5 False positive probabilities

Lemma 18. *Let* $\mathfrak{C}_1, \mathfrak{C}_2$ *two random* $[n,k]_q$ *code such that* $\mathfrak{C}_2 = \varphi(\mathfrak{C}_1)$ *with* φ *any monomial transformation and* $\mathfrak{c}, \mathfrak{c}' \in \mathfrak{C}_1$ *both of weight* $w > k$. *Assume* $\left|\mathrm{supp}(\mathfrak{c}) \cap \mathrm{supp}(\mathfrak{c}')\right| = \ell \leqslant k$ *and this intersection is included in an information set. Then* $\mathfrak{c}^{-1} \star \mathfrak{C}_1$ *and* $\varphi(\mathfrak{c}')^{-1} \star \mathfrak{C}_2$ *are independent.*

Proof. For any two independent $[n,k]_q$ code, we can find respective generator matrices having any $\ell \leqslant k$ independent columns in common. Indeed, in systematic form both has an identity matrix as submatrix.

Algorithm 2 deals with two levels. The first one produces some codewords for which we build a corresponding shorten code. The second level produces codewords in this shorten code and apply signature methods in order to find matches and a potential solution. Two different shorten codes might produce false positives but they would be even less probable than false positives from a single shorten:

$$\mathbb{P}\left(\mathbf{c^{(1)}}^{-1} \star \mathcal{C}^{(1)} \overset{\text{PCE}}{\sim} \mathbf{c^{(2)}}^{-1} \star \mathcal{C}^{(2)}\right) \leqslant \mathbb{P}\left(\mathbf{c^{(1)}}^{-1} \star \mathcal{C}^{(1)} \overset{\text{PCE}}{\sim} \mathbf{c^{(0)}}^{-1} \star \mathcal{C}^{(1)}\right)$$

where $\mathcal{C}^{(1)}, \mathcal{C}^{(2)}$ are two different shorten codes of a same code $\mathcal{C}$, $\mathbf{c^{(0)}}, \mathbf{c^{(1)}}$ are uniformly distributed in the codewords of weight w of $\mathcal{C}^{(1)}$ and $\mathbf{c^{(2)}}$ is uniformly distributed in the codewords of weight w of $\mathcal{C}^{(2)}$. Thus if we have an upper bound for false positives of Algorithm 1, we can deduce an upper bound for false positives of Algorithm 2. Hence, we restrict our study to false positives for Algorithm 1.

We call *relative weight* of a codeword the ratio between its weight and the length of its code. The two level technique decreases the rate of codes and increases the relative optimal weight w. The mean value and the variance of the length of the intersection between the support of two codewords increases with the relative weight of the codewords. Thus we need to enhance Lemma 18 to treat cases where the intersection of the support exceeds k.

Corollary 4. *Let $\mathfrak{C}_1, \mathfrak{C}_2$ be two random $[n,k]_q$ code such that $\mathfrak{C}_2 = \varphi(\mathfrak{C}_1)$ with φ any monomial transformation and $\mathbf{c}, \mathbf{c'} \in \mathfrak{C}_1$ be both of weight $w > k$. Assume $S \triangleq \text{supp}(\mathbf{c}) \cap \text{supp}(\mathbf{c'})$, $\left|S\right| = \ell \geqslant k$ and this intersection contains an information set I. Note $J_1 \triangleq S \backslash I$ and $J_2 \triangleq \pi(J_1)$ for π the underlying permutation defining φ. Then $\mathsf{Puncture}(\mathbf{c}^{-1} \star \mathfrak{C}_1, J_1)$ and $\mathsf{Puncture}(\varphi(\mathbf{c'})^{-1} \star \mathfrak{C}_2, J_2)$ are independent codes.*

Proof. It is a simple application of previous Lemma 18 on $\mathsf{Puncture}(\mathbf{c}^{-1} \star \mathfrak{C}_1, J_1)$ and $\mathsf{Puncture}(\varphi(\mathbf{c'})^{-1} \star \mathfrak{C}_2, J_2)$.

We prove here Proposition 9:

Proof. For a given code containing $\mathbb{1}_n$, there are at most $n!$ equivalent codes. We can remark that all of them contain $\mathbb{1}_n$. Moreover there are $\frac{q^{\lfloor Rn \rfloor}-1}{q^n-1}\begin{bmatrix} n \\ \lfloor Rn \rfloor \end{bmatrix}_q$ $[n,k]_q$ code containing $\mathbb{1}_w$. Thus the probability for two uniformly distributed $[n,k]_q$ code containing $\mathbb{1}_n$ to be permutation equivalent is upper bounded by $\frac{q^n-1}{q^{\lfloor Rn \rfloor}-1}\frac{n!}{\begin{bmatrix} n \\ \lfloor Rn \rfloor \end{bmatrix}_q}$.

We can now prove Proposition 8.

Proof. We note as in Corollary 4 S the intersection between the supports of cardinal ℓ. Assume I is an information set contained in S and we note $J_1 \triangleq S \backslash I$.

The probability for $\mathbf{c}^{-1} \star \mathfrak{C}_1$ and $\varphi(\mathbf{c'}^{-1}) \star \mathfrak{C}_2$ to be permutation equivalent is equal to the probability for $\mathbf{c}^{-1} \star \mathfrak{C}_1$ and $\varphi(\mathbf{c'}^{-1}) \star \mathfrak{C}_2$ to be permutation

equivalent whenever J_1 is sent to some J'. If $\mathfrak{c}^{-1} \star \mathfrak{C}_1$ and $\varphi(\mathfrak{c}'^{-1}) \star \mathfrak{C}_2$ are permutation equivalent then $\mathsf{Puncture}(\mathfrak{c}^{-1} \star \mathfrak{C}_1, J_1)$ and $\mathsf{Puncture}(\varphi(\mathfrak{c}')^{-1} \star \mathfrak{C}_2, J')$ are permutation equivalent. Those two last punctured codes are independent by Corollary 4 and uniformly distributed in the codes containing $\mathbb{1}_w$ since they are a uniformly distributed code containing $\mathbb{1}_w$ punctured on some columns. Then by Proposition 9 we get the desired upper bound.

A.6 Distribution of supports for codewords with prescribed weights

Let $\mathfrak{c}_1, \mathfrak{c}_2$ be two uniformly distributed codewords of weight w in a $[n,k]_q$ code $\mathcal{C}$ and note $\mathfrak{L}$ the length of their support intersection. We make the common assumption in coding theory that $\mathfrak{L}$ is distributed as the dimension of the intersection of the supports of two uniformly distributed codewords in $\mathbb{F}_q^n$ of weight w:

$$\mathbb{P}\left(\mathfrak{L} = \ell\right) = \frac{\binom{w}{\ell}\binom{n-w}{w-\ell}}{\binom{n}{w}}$$

of mean value known to be $\mathbb{E}[\mathfrak{L}] = \frac{w^2}{n}$ and variance $\mathbb{V}[\mathfrak{L}] = \frac{w^2(n-w)^2}{n^2(n-1)}$. Whenever $w = \lfloor \omega n \rfloor$ and $\omega \in]0,1[$, $\mathbb{E}[\mathfrak{L}] = \omega^2 n$ and $\mathbb{V}[\mathfrak{L}] = \omega^2(1-\omega)^2\frac{n^2}{n-1}$. Assume $\mathbb{E}[\mathfrak{L}] + \mathbb{V}[\mathfrak{L}] < w$ which is equivalence to $\omega + \omega(1-\omega)^2\frac{n}{n-1} < 1$. Then by using Bienaymé-Chebychev inequality we get:

$$\mathbb{P}\left(|\mathfrak{L} - \mathbb{E}[\mathfrak{L}]| \geqslant \mathbb{V}[\mathfrak{L}]\right) \leqslant \left[\frac{1}{\omega^2(1-\omega)^2}(1-\frac{1}{n})\right]\frac{1}{n}.$$

This implies that for a constant support size w, the probability for the two codewords to share more than $\mathbb{E}[\mathfrak{L}] + \mathbb{V}[\mathfrak{L}]$ positions in their support tends to 0 linearly in n.

Remark 7. This subsection precises the results of Proposition 8. Indeed, the results presented above ensure that as n tends to ∞, the proportion of false positive tends to 0 exponentially fast for a vast proportion of pair of codewords.

B Generic Decoding over $\mathbb{F}_q$

There are very few works exploring the hardness of decoding in the $\mathbb{F}_q$ case. A negative result [11] indicates that when q grows, it is difficult to improve on Prange's algorithm. The non asymptotic estimates we used for attack estimates are given in §8.2 are based on Peters' work [23].

The input of the decoding problem is a couple $(\boldsymbol{H}, \boldsymbol{s})$ uniformly distributed in $\mathbb{F}_q^{(n-k)\times n} \times \{\boldsymbol{y}\boldsymbol{H}^T \mid \boldsymbol{y} \in \mathbb{F}_q^n, |\boldsymbol{y}| = w\}$. If $w < d_{\mathrm{GV}}(n,k)$ there is exactly one solution, and when $w > d_{\mathrm{GV}}(n,k)$ there are $N_w = \binom{n}{w}(q-1)^w/q^{n-k}$ solutions. The variants of ISD we consider below are iterative, with independent iterations which repeat until a solution is obtained. The average running time writes as $T_{\mathrm{dec}} = C_{\mathrm{dec}}/P_{\mathrm{dec}}$, where C_{dec} is the cost of one iteration and P_{dec} its probability of success.

B.1 Prange Algorithm – Asymptotic

At each iteration, the Prange algorithm selects k random positions—more specifically an information set—and succeeds if there exists at least one solution of the problem with a support outside of those k positions. The success probability is, up to a constant factor,

$$P_{\text{prange}}(n,k,w) = \max(1, N_w)\frac{\binom{n-k}{w}}{\binom{n}{w}} = \max\left(\frac{\binom{n-k}{w}}{\binom{n}{w}}, \frac{\binom{n-k}{w}(q-1)^w}{q^{n-k}}\right)$$

and each iteration has a polynomial cost. Hence, up to a polynomial factor, the complexity is $T_{\text{prange}}(n,k,w) = 1/P_{\text{prange}}(n,k,w)$. With $\omega = w/n$ and $R = k/n$, the asymptotic exponent is

$$\tau_{\text{prange}}(R,\omega) = \min\Bigg(h_2(\omega) - (1-R)h_2\left(\frac{\omega}{1-R}\right), \\ (1-R)\log_2 q - (1-R)h_q\left(\frac{\omega}{1-R}\right)\Bigg),$$

where $T_{\text{prange}}(n,k,w) = 2^{n\tau_{\text{prange}}(R,\omega)(1+o(1))}$. In this work, we always have $w > d_{\text{GV}}(n,k)$ and the asymptotic exponent is the right-hand side of the minimum above.

B.2 Dumer Algorithm – Asymptotic

Dumer's algorithm features two additional parameters p and ℓ chosen to minimize the complexity ($p = \ell = 0$ corresponds to Prange). At each iteration, $k+\ell$ random positions are selected and the algorithm succeeds if there exists at least one solution of the problem with a support containing p of those $k+\ell$ positions. The success probability is, up to a constant factor,

$$\begin{aligned} P_{\text{dumer}}(n,k,w,p,\ell) &= \max(1, N_w)\frac{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}{\binom{n}{w}} \\ &= \max\left(\frac{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}{\binom{n}{w}}, \frac{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}(q-1)^w}{q^{n-k}}\right) \end{aligned}$$

and the cost of each iteration is, up to a polynomial factor,

$$C_{\text{dumer}}(n,k,w,p,\ell) = \sqrt{\binom{k+\ell}{p}(q-1)^p} + \binom{k+\ell}{p}(q-1)^p/q^\ell,$$

which is minimal, up to a constant factor, when $q^\ell = \sqrt{\binom{k+\ell}{p}(q-1)^p}$. The total complexity is, up to a polynomial factor,

$$T_{\text{dumer}}(n,k,w) = \min_p \frac{\min\left(\binom{n}{w}(q-1)^w, q^{n-k}\right)}{\binom{n-k-\ell}{w-p}(q-1)^{w-p}q^\ell}$$

with ℓ is implicitly defined as a function of p by $q^\ell = \sqrt{\binom{k+\ell}{p}(q-1)^p}$. Note that the range of admissible values for p must be limited to values such that the formula giving $P_{\text{dumer}}(n,k,w,p,\ell)$ do not exceed 1. Asymptotically, with $\omega = w/n$, $R = k/n$, $\rho = p/n$ and $\lambda = \ell/n$

$$\tau_{\text{dumer}}(R,\omega) = \min_{\rho}\Bigg(\min\left(h_q(\omega), (1-R)\log_2 q\right) \\ - \lambda\log_2 q - (1-R-\lambda)h_q\left(\tfrac{\omega-\rho}{1-R-\lambda}\right)\Bigg)$$

where λ is implicitly defined as a function of ρ by $\lambda = \frac{R+\lambda}{2\log_2 q} h_q\left(\frac{\rho}{R+\lambda}\right)$. When $w > d_{\text{GV}}(n,k)$ the formula simplifies.

B.3 MMT Algorithm – Asymptotic

MMT algorithm [21] is similar to Dumer's algorithm with an additional step using representations [19]. The formulas below correspond to a slightly modified version which includes an additional parameter e chosen such that the sum of two q-ary words of length $k+\ell$ and weight $p/2+e$ has a weight typically equal to p. This number is such that $2e(k+\ell)q = (p/2+e)^2$. This gives a marginally better algorithm, but its main interest is an easier formula for the complexity. The probability formula is the same as for Dumer, but the iteration cost differs and is, up to a polynomial factor,

$$C_{\text{mmt}}(n,k,w,p,\ell) = \frac{1}{R}\binom{k+\ell}{p/2+e}(q-1)^{p/2+e} + \binom{k+\ell}{p}(q-1)^p/q^\ell$$

where $R = \binom{k+\ell}{p/2+e}^2(q-1)^{p+2e}/\left(\binom{k+\ell}{p}(q-1)^p\right)$ is, up to a polynomial factor, the number of representations, in length $k+\ell$, of a word of weight p as a sum of two words of weight $p/2+e$. The iteration cost rewrites as

$$C_{\text{mmt}}(n,k,w,p,\ell) = \binom{k+\ell}{p}(q-1)^p\left(\frac{1}{\binom{k+\ell}{p/2+e}(q-1)^{p/2+e}} + \frac{1}{q^\ell}\right),$$

which is minimal, up to a constant factor, when $q^\ell = \binom{k+\ell}{p/2+e}(q-1)^{p/2+e}$. The total complexity is, up to a polynomial factor,

$$T_{\text{mmt}}(n,k,w) = \min_p \frac{\min\left(\binom{n}{w}(q-1)^w, q^{n-k}\right)}{\binom{n-k-\ell}{w-p}(q-1)^{w-p}q^\ell},$$

which is the same formula as Dumer, but with a different ℓ implicitly defined as a function of p by $q^\ell = \binom{k+\ell}{p/2+e}(q-1)^{p/2+e}$ and $2e(k+\ell)q = (p/2+e)^2$.

Asymptotically, with $\omega = w/n$, $R = k/n$, $\rho = p/n$, $\lambda = \ell/n$ and $\varepsilon = e/n$,

$$\tau_{\mathrm{mmt}}(R,\omega) = \min_{\rho}\Bigg(\min\left(h_q(\omega), (1-R)\log_2 q\right) \\ - \lambda\log_2 q - (1-R-\lambda)h_q\left(\tfrac{\omega-\rho}{1-R-\lambda}\right)\Bigg)$$

where λ is implicitly defined as a function of ρ by $\lambda = \frac{R+\lambda}{\log_2 q} h_q\left(\frac{\rho/2+\varepsilon}{R+\lambda}\right)$ and $\varepsilon = \frac{(\rho/2+\varepsilon)^2}{2q(R+\lambda)}$.

C Canonical Form for Permutation Equivalence

C.1 Complete Weight

The complete weight over $\mathbb{F}_q$ is defined for any length n and any $\boldsymbol{u} \in \mathbb{F}_q^n$ as

$$\mathrm{cw}(\boldsymbol{u}) = (c_\beta)_{\beta\in\mathbb{F}_q} \in \mathbb{N}^q \text{ with } c_\beta = |\{i \in [\![0,n[\![\mid u_i = \beta\}|.$$

The complete weight enumerator $\mathrm{CWE}(\mathcal{C})$ of a code $\mathcal{C}$ is the multiset of all $\mathrm{cw}(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{C}$. The complete weight enumerator is a code permutation-invariant, that is for all linear code $\mathcal{C}$ of length n and all permutation π of $[\![0,n[\![$, we have $\mathrm{CWE}(\mathcal{C}) = \mathrm{CWE}(\pi(\mathcal{C}))$.

Probability Distribution For a given $\mathrm{c} = (c_\beta)_{\beta\in\mathbb{F}_q} \in \mathbb{N}^q$, the number of vectors in $\mathbb{F}_q^n$ admitting c as complete weight is equal to the multinomial coefficient $n!/\prod_{\beta\in\mathbb{F}_q} c_\beta!$ and we have the following proposition.

Proposition 11. *For $\boldsymbol{u}$ uniformly distributed in $\mathbb{F}_q^n$, the probability that its complete weight is $\mathrm{c} = (c_\beta)_{\beta\in\mathbb{F}_q} \in \mathbb{N}^q$ is null if the c_β do not add to n and is equal to*

$$\mathbb{P}\left(\mathrm{cw}(\boldsymbol{u}) = \mathrm{c}\right) = \frac{n!}{\prod\limits_{\beta\in\mathbb{F}_q} c_\beta!}\frac{1}{q^n} \text{ if } \sum_{\beta\in\mathbb{F}_q} c_\beta = n,$$

and we have

$$\max_{\mathrm{c}\in\mathbb{N}^q} \mathbb{P}\left(\mathrm{cw}(\boldsymbol{u}) = \mathrm{c}\right) \underset{n\to\infty}{\sim} \frac{q^{\frac{q}{2}}}{(2\pi n)^{\frac{q-1}{2}}}.$$

The proof details are omitted, but essentially the multinomial coefficient is maximal when all the c_β are close to n/q and the asymptotic equivalent derives from the Stirling formula $n! \sim_{n\to\infty} \sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n$.

C.2 Scaled Hulls

We identify the projective line $\mathbb{P}^1(\mathbb{F}_q)$ with $\mathbb{F}_q \cup \{\infty\}$, *i.e.* $\beta = \langle(\beta, 1)\rangle$ for $\beta \in \mathbb{F}_q$ and $\infty = \langle(1, 0)\rangle$. The hull of a code $\mathcal{C}[n, k]_q$ is defined as its intersection with its dual code, $\mathcal{H}(\mathcal{C}) = \mathcal{C} \cap \mathcal{C}^\perp$. The scaled hull in position $i \in [\![0, n[\![$ for the scalar $\beta \in \mathbb{F}_q \cup \{\infty\}$ is obtained as the intersection of $\mathcal{C}^\perp$ with the code obtained by multiplying the i-th coordinate of $\mathcal{C}$ by β. Multiplication by $\beta = \infty$ essentially consists in shortening the code on that position.

Definition 9. *The* scaled hull *of $\mathcal{C}[n, k]_q$ for $i \in [\![0, n[\![$ and $\beta \in \mathbb{F}_q \cup \{\infty\}$ is defined as*

$$\mathcal{H}_{\beta \cdot i}(\mathcal{C}) = \{\boldsymbol{y} \in \mathcal{C}^\perp \mid \boldsymbol{x} \in \mathcal{C}, \mathrm{supp}(\boldsymbol{x} - \boldsymbol{y}) \subset \{i\}, (y_i, x_i) \in \beta\}. \tag{25}$$

The key feature of those scaled hulls, introduced in [28], is a form of invariance under permutation of the coordinates.

Proposition 12. *For any q-ary linear code $\mathcal{C}$ of length n, any permutation π of $[\![0, n[\![$, any $i \in [\![0, n[\![$, and any $\beta \in \mathbb{F}_q \cup \{\infty\}$, we have*

$$\pi(\mathcal{H}_{\beta \cdot i}(\mathcal{C})) = \mathcal{H}_{\beta \cdot \pi(i)}(\pi(\mathcal{C})).$$

This will allow the design of a code permutation-signature. Moreover, the discriminability of this signature is guarantied by properties given in [28] which we summarize below. For any code $\mathcal{C}[n, k]_q$ and all $i \in [\![0, n[\![$:

1. If $i \in \mathrm{supp}(\mathcal{H}(\mathcal{C}))$, we have $\mathcal{H}_{1 \cdot i}(\mathcal{C}) = \mathcal{H}(\mathcal{C})$ and for all $\beta \neq 1$, the codes $\mathcal{H}_{\beta \cdot i}(\mathcal{C})$ are all identical, obtained by shortening $\mathcal{H}(\mathcal{C})$ in position i.
2. If $i \notin \mathrm{supp}(\mathcal{H}(\mathcal{C}))$, there exists a unique $\beta \in \mathbb{F}_q \cup \{\infty\}$, $\beta \neq 1$, such that $\mathcal{H}_{\beta \cdot i}(\mathcal{C}) = \mathcal{H}(\mathcal{C}) \oplus \langle \boldsymbol{u} \rangle$ for some $\boldsymbol{u} \in \mathcal{C}^\perp \setminus \mathcal{C}$, and $\mathcal{H}_{\gamma \cdot i}(\mathcal{C}) = \mathcal{H}(\mathcal{C})$ for all $\gamma \neq \beta$.

In both case, there exists a unique element $\beta \in \mathbb{F}_q \cup \{\infty\}$ such that the code $\mathcal{H}_{\beta \cdot i}(\mathcal{C})$ differs from the other scaled hulls on the same position. For convenience, we refer to this β as the *distinguished hull scalar* and we denote $\mathrm{DHS}(\mathcal{C}, i) = \beta$. This quantity is invariant by permutation, in the sense that $\mathrm{DHS}(\mathcal{C}, i) = \mathrm{DHS}(\pi(\mathcal{C}), \pi(i))$.

A (Frequent) Particular Case: the Hull Is Reduced to {0} We explore here in detail the situation where $\mathcal{H}(\mathcal{C}) = \{0\}$. For $q = 5$ it is the case for 79% of the codes, and this proportion grows with q, close to $(q - 1)/q$.

Lemma 19. *Let $\mathcal{C}$ be an $[n, k]_q$ code such that $\mathcal{H}(\mathcal{C}) = \{0\}$, admitting a generator matrix of the form $\boldsymbol{G} = (\boldsymbol{I}_k \mid \boldsymbol{R})\boldsymbol{P} \in \mathbb{F}_q^{k \times n}$ with $\boldsymbol{P}$ an $n \times n$ permutation matrix. The following matrix, which we refer to as the* hull scalar matrix*, only depends of $\mathcal{C}$*

$$\boldsymbol{U} = \mathrm{HSM}(\mathcal{C}) \triangleq \boldsymbol{P}^T \left(\left(\begin{array}{c|c} \boldsymbol{I}_k & \boldsymbol{R} \\ \hline \boldsymbol{R}^T & -\boldsymbol{I}_{n-k} \end{array} \right)^{-1} - \left(\begin{array}{c|c} \boldsymbol{I}_k & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} \end{array} \right) \right) \boldsymbol{P}. \tag{23}$$

Moreover, for all i in $[\![0, n[\![$, we have $\mathcal{H}_{\beta\cdot i}(\mathcal{C}) = \langle \boldsymbol{u}_i \rangle$, where $\boldsymbol{u}_i$ denotes the i-th row of $\boldsymbol{U}$ and $\beta = \langle (\boldsymbol{u}_i(i), \boldsymbol{u}_i(i)+1) \rangle = \mathrm{DHS}(\mathcal{C}, i)$ is the distinguished hull scalar for $(\mathcal{C}, i)$.

Proof. If $\boldsymbol{G} = (\boldsymbol{I}_k \mid \boldsymbol{R})\,\boldsymbol{P}$ is a generator matrix of $\mathcal{C}$ then $\boldsymbol{H} = \left(\boldsymbol{R}^T \mid -\boldsymbol{I}_{n-k}\right)\boldsymbol{P}$ is a parity check matrix of $\mathcal{C}$ and the nullspace of

$$\boldsymbol{M} = \boldsymbol{P}^T \left(\begin{array}{c|c} \boldsymbol{I}_k & \boldsymbol{R} \\ \hline \boldsymbol{R}^T & -\boldsymbol{I}_{n-k} \end{array}\right) \boldsymbol{P} = \left(\boldsymbol{G}^T \middle| \boldsymbol{H}^T\right) \boldsymbol{P} = \boldsymbol{P}^T \left(\begin{array}{c} \boldsymbol{G} \\ \hline \boldsymbol{H} \end{array}\right) \tag{26}$$

is the hull $\mathcal{H}(\mathcal{C}) = \mathcal{C} \cap \mathcal{C}^\perp$ of $\mathcal{C}$. Hence, if $\mathcal{H}(\mathcal{C}) = \{0\}$, the matrix $\boldsymbol{M}$ is invertible and $\boldsymbol{U}$, as in (23), is well defined. From (26), we have

$$\boldsymbol{U}\boldsymbol{M}\boldsymbol{P}^T = (\boldsymbol{U}\boldsymbol{G}^T \mid \boldsymbol{U}\boldsymbol{H}^T),$$

and from (23)

$$\boldsymbol{U}\boldsymbol{M}\boldsymbol{P}^T = \boldsymbol{P}^T \left(\boldsymbol{I}_n - \left(\begin{array}{c|c} \boldsymbol{I}_k & \boldsymbol{R} \\ \hline \boldsymbol{0} & \boldsymbol{0} \end{array}\right)\right) = \boldsymbol{P}^T \left(\begin{array}{c|c} \boldsymbol{0} & -\boldsymbol{R} \\ \hline \boldsymbol{0} & \boldsymbol{I} \end{array}\right) = \left(\boldsymbol{0} \middle| -\boldsymbol{H}^T\right)$$

And thus, denoting $\boldsymbol{u}_i$ the i-th row of $\boldsymbol{U}$, we have, for all i, $\boldsymbol{u}_i \in \mathcal{C}^\perp$ because $\boldsymbol{U}\boldsymbol{G}^T = \boldsymbol{0}$ and $\boldsymbol{u}_i + \boldsymbol{e}^{(i)} \in \mathcal{C}$ because $(\boldsymbol{U} + \boldsymbol{I}_n)\boldsymbol{H}^T = \boldsymbol{0}$, where $\boldsymbol{e}^{(i)}$ is the word with a 1 in i-th coordinate and 0 everywhere else. It follows that $\beta = \langle (\boldsymbol{u}_i(i), \boldsymbol{u}_i(i)+1) \rangle$ is the distinguished hull scalar for $(\mathcal{C}, i)$, *i.e.* the only $\beta \in \mathbb{F}_q \cup \{\infty\}$ such that $\mathcal{H}_{\beta\cdot i}(\mathcal{C}) \neq \{0\}$, and we have

$$\mathcal{H}_{\beta\cdot i}(\mathcal{C}) = \{\boldsymbol{y} \in \mathcal{C}^\perp \mid \boldsymbol{x} \in \mathcal{C}, \mathrm{supp}(\boldsymbol{x} - \boldsymbol{y}) \subset \{i\}, (y_i, x_i) \in \beta\} = \langle \boldsymbol{u}_i \rangle.$$

Finally, for any given $i \in [\![0, n[\![$, let $\beta = \mathrm{DHS}(\mathcal{C}, i)$. There is a unique word $\boldsymbol{v}$ in the 1-dimensional space $\mathcal{H}_{\beta\cdot i}(\mathcal{C})$ such that $(v_i, v_i + 1) \in \beta$, and this vector is the i-th row of $\boldsymbol{U}$, regardless of the values of the matrices $\boldsymbol{R}$ and $\boldsymbol{P}$ and defining $\mathrm{HSM}(\mathcal{C}) \triangleq \boldsymbol{U}$ is sound.

Corollary 5. *Let $\mathcal{C}$ be an $[n, k]_q$ code, let π be a permutation of $[\![0, n[\![$ and let $\boldsymbol{P}$ denote the $n \times n$ permutation matrix such that $P_{i,\pi(i)} = 1$, $0 \leqslant i < n$. We have*

$$\mathrm{HSM}(\pi(\mathcal{C})) = \boldsymbol{P}^T \mathrm{HSM}(\mathcal{C}) \boldsymbol{P}.$$

Proof. Without loss of generality, we assume that $\mathcal{C}$ admits a generator matrix of the form $\boldsymbol{G} = (\boldsymbol{I}_k \mid \boldsymbol{R})$. With π and $\boldsymbol{P}$ defined as in the statement, the code $\pi(\mathcal{C})$ admits the generator matrix $\boldsymbol{G}\boldsymbol{P}$. We apply Lemma 19 for $\mathcal{C}$ with the generator matrix $(\boldsymbol{I}_k \mid \boldsymbol{R})$ and for $\pi(\mathcal{C})$ with the generator matrix $(\boldsymbol{I}_k \mid \boldsymbol{R})\,\boldsymbol{P}$ and we get

$$\mathrm{HSM}(\pi(\mathcal{C})) = \boldsymbol{P}^T \left(\left(\begin{array}{c|c} \boldsymbol{I}_k & \boldsymbol{R} \\ \hline \boldsymbol{R}^T & -\boldsymbol{I}_{n-k} \end{array}\right)^{-1} - \left(\begin{array}{c|c} \boldsymbol{I}_k & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} \end{array}\right) \right) \boldsymbol{P} = \boldsymbol{P}^T \mathrm{HSM}(\mathcal{C}) \boldsymbol{P}.$$

Those results prove that, if $\mathcal{H}(\mathcal{C}) = \{0\}$, the "hull scalar matrix" is identical up to the row/column order (the same permutation applies to rows and columns) for any code permutation equivalent to $\mathcal{C}$. Moreover, there is a one-to-one correspondence between the diagonal element and the corresponding distinguished hull scalar. Specifically, if $\beta = \text{DHS}(\mathcal{C}, i)$, then the i-th diagonal element of $\text{HSM}(\mathcal{C})$ is equal to $\beta/(1-\beta)$ if $\beta \in \mathbb{F}_q \setminus \{1\}$ and to -1 if $\beta = \infty$. Note that $\beta \neq 1$ when $\mathcal{H}(\mathcal{C}) = \{0\}$.

Discriminability. For a given $[n,k]_q$ code $\mathcal{C}$ with a hull reduced to $\{0\}$, as a consequence of Corollary 5, we may define a permutation-signature as $\Psi(\mathcal{C}, i) = (\boldsymbol{u}_i(i), \text{cw}(\boldsymbol{u}_i))$ where $\boldsymbol{u}_i$ is the i-th row of $\boldsymbol{U} = \text{HSM}(\mathcal{C})$. If all, or most, positions can be discriminated, then the conditions of Theorem 5 are met and we can obtain a canonical form of $\mathcal{C}$ and a canonical ordering of its coordinates. In the following statements, we show that all but a handful of rows could be discriminated if $\text{HSM}(\mathcal{C})$ was uniformly distributed in $\mathbb{F}_q^{n\times n}$.

Lemma 20. *Let $\mathcal{D}$ be a discrete distribution over a sample set $\mathcal{X}$, we denote p_x the probability for $x \in \mathcal{X}$ to occur, and $p = \max_x p_x$. We consider an experiment in which n elements are drawn independently in $\mathcal{X}$ according to $\mathcal{D}$.*

(i) *More than $n(1-np/2)$ elements of $\mathcal{X}$ are drawn at least once on average.*
(ii) *More than $n(1-np)$ elements of $\mathcal{X}$ are drawn exactly once on average.*

Proof. (i) An element $x \in \mathcal{X}$ is drawn at least once with probability $1-(1-p_x)^n$. The expected number of distinct elements after n draws is

$$\sum_{x\in\mathcal{X}}(1-(1-p_x)^n) \geqslant \sum_{x\in\mathcal{X}}(1-\mathrm{e}^{-p_x n}) \geqslant \sum_{x\in\mathcal{X}}(1-\mathrm{e}^{-pn})\frac{p_x}{p} = \frac{1-\mathrm{e}^{-pn}}{p} \geqslant n\left(1-\frac{pn}{2}\right)$$

We used the inequality $(1-x)^y \leqslant \mathrm{e}^{-xy}$ valid for all positive x and y, and the inequality $1-\mathrm{e}^{-\mu x} \geqslant (1-\mathrm{e}^{-\mu})x$ valid for all $0 \leqslant x \leqslant 1$ and all $\mu > 0$.

(ii) From above, at least $n - pn^2/2$ elements are drawn at least once and thus at most $pn^2/2 = n-(n-pn^2/2)$ are drawn more than once. It follows that at least $n - pn^2 = n - pn^2/2 - pn^2/2$ elements are drawn once exactly.

Proposition 13. *Define $\Phi(\boldsymbol{U}, i) = (U_{i,i}, \text{cw}(\text{row}(\boldsymbol{U}, i))) \in \mathbb{F}_q \times \mathbb{N}^q$ for $\boldsymbol{U} \in \mathbb{F}_q^{n\times n}$ and $i \in [\![0, n[\![$, and let $I_\Phi(\boldsymbol{U}) = \{i \in [\![0,n[\![\mid \Phi(\boldsymbol{U}, i) \notin \{\Phi(\boldsymbol{U}, j), 0 \leqslant j < n, j \neq i\}\}$ denote the positions i such that $\Phi(\boldsymbol{U}, i)$ is reached once only. For $\boldsymbol{U}$ uniformly distributed in $\mathbb{F}_q^{n\times n}$, we have for all $x \in \mathbb{F}_q \times \mathbb{N}^q$, all $i \in [\![0,n[\![$ and growing n*

$$\mathbb{P}[\Phi(\boldsymbol{U}, i) = x] \leqslant \frac{q^{\frac{q}{2}-1}}{(2\pi)^{\frac{q-1}{2}}}\frac{1+o(1)}{n^{\frac{q-1}{2}}},$$

and

$$\mathbb{E}[|I_\Phi(\boldsymbol{U})|] \geqslant n\left(1-\frac{q^{\frac{q}{2}-1}}{(2\pi)^{\frac{q-1}{2}}}\frac{1+o(1)}{n^{\frac{q-3}{2}}}\right).$$

Proof. If $\boldsymbol{U}$ is uniformly distributed, its i-th row $\boldsymbol{u}_i$ is uniformly distributed in $\mathbb{F}_q^n$. We have for $(\alpha, \mathrm{c}) \in \mathbb{F}_q \times \mathbb{N}^q$

$$\mathbb{P}[\Phi(\boldsymbol{U}, i) = (\alpha, \mathrm{c})] = \mathbb{P}[\boldsymbol{u}_i(i) = \alpha, \mathrm{cw}(\boldsymbol{u}_i) = \mathrm{c}]$$

but the events $\boldsymbol{u}_i(i) = \alpha$ and $\mathrm{cw}(\boldsymbol{u}_i) = \mathrm{c}$ are not independent. Instead, we may write

$$\mathbb{P}[\Phi(\boldsymbol{U}, i) = (\alpha, \mathrm{c})] = \mathbb{P}[\boldsymbol{u}_i(i) = \alpha, \mathrm{cw}(\boldsymbol{u}'_i) = \mathrm{c}']$$

where $\mathrm{c} = (c_\beta)_{\beta \in \mathbb{F}_q}$ and $\mathrm{c}' = (c'_\beta)_{\beta \in \mathbb{F}_q}$ with $c'_\beta = c_\beta$ for $\beta \neq \alpha$ and $c'_\alpha = c_\alpha - 1$ (remark that $c_\alpha > 0$ or the probability is null), and $\boldsymbol{u}'_i \in \mathbb{F}_q^{n-1}$ is equal to $\boldsymbol{u}_i$ punctured in its i-th coordinate. Now the events $\boldsymbol{u}_i(i) = \alpha$ and $\mathrm{cw}(\boldsymbol{u}'_i) = \mathrm{c}'$ are independent and, using Proposition 11, we get

$$\mathbb{P}[\Phi(\boldsymbol{U}, i) = (\alpha, \mathrm{c})] = \mathbb{P}[\boldsymbol{u}_i(i) = \alpha] \cdot \mathbb{P}[\mathrm{cw}(\boldsymbol{u}'_i) = \mathrm{c}'] \leqslant \frac{1}{q} \cdot \frac{q^{\frac{q}{2}}}{(2\pi)^{\frac{q-1}{2}}} \frac{1 + o(1)}{(n-1)^{\frac{q-1}{2}}}.$$

which easily simplifies to

$$\mathbb{P}[\Phi(\boldsymbol{U}, i) = (\alpha, \mathrm{c})] \leqslant \frac{q^{\frac{q}{2}-1}}{(2\pi)^{\frac{q-1}{2}}} \frac{1 + o(1)}{n^{\frac{q-1}{2}}} \triangleq p. \tag{27}$$

Now we apply Lemma 20 to the above discrete distribution, upper bounded by p, and the number of values (α, c) that are obtained exactly once among the n draws $\Phi(\boldsymbol{U}, i)$, $0 \leqslant i < n$, that is precisely the cardinality of $I_\Phi(\boldsymbol{U})$, is at least equal to $n(1 - np)$, average over all choice of $\boldsymbol{U}$. Using (27), we obtain

$$\mathbb{E}[|I_\Phi(\boldsymbol{U})|] \geqslant n(1 - np) = n\left(1 - \frac{q^{\frac{q}{2}-1}}{(2\pi)^{\frac{q-1}{2}}} \frac{1 + o(1)}{n^{\frac{q-3}{2}}}\right).$$

For $q > 3$ the expected size of $I_\Phi(\boldsymbol{U})$ is $n - o(n)$. This would be enough to meet the discriminability condition of Theorem 5, but obviously, and unfortunately, the hull scalar matrix as defined in (23) is not uniformly distributed in $\mathbb{F}_q^{n \times n}$ when $\boldsymbol{R}$ is uniformly distributed over $\mathbb{F}_q^{k \times (n-k)}$. Still, we empirically observed that for $q = 5$ and growing n, on average, we have $|I_\Phi(\boldsymbol{U})| \geqslant n - 0.4$ for $\boldsymbol{U}$ as in (23) with $\boldsymbol{R}$ uniformly distributed, and $|I_\Phi(\boldsymbol{U})| \geqslant n - 0.08$ when $\boldsymbol{U}$ is uniformly distributed, while the Proposition 13 guaranties $|I_\Phi(\boldsymbol{U})| \geqslant n - 0.29$.

Conjecture 1. For any R, $0 < R \leqslant 1/2$, any $q \geqslant 5$, for $\mathcal{C}$ a uniformly distributed q-ary linear $[n, \lfloor Rn \rfloor]$ code, we have (with I_Φ defined in Proposition 13)

$$\mathbb{E}[|I_\Phi(\mathrm{HSM}(\mathcal{C}))|] = n - O\left(\frac{1}{n^{\frac{q-5}{2}}}\right).$$

C.3 A Code Permutation-Signature Based on Hulls and Complete Weights

We define the following code permutation-signature

$$\Psi(\mathcal{C}, i) = \begin{cases} \Phi(\mathrm{HSM}(\mathcal{C}), i) & \text{if } \mathcal{H}(\mathcal{C}) = \{0\} \\ \perp & \text{else} \end{cases} \tag{28}$$

Corollary 6. *Let Ψ be the code permutation-signature defined in* (28)*. If Conjecture 1 holds, then for any R, $0 < R \leqslant 1/2$, any $q \geqslant 5$, and n large enough, the proportion of q-ary linear $[n, \lfloor Rn \rfloor]$ codes $\mathcal{C}$ such that Ψ discriminates an information set of $\mathcal{C}$ is $\geqslant 0.75$.*

Proof. For $q = 5$, a little more than 79% of the linear codes over $\mathbb{F}_5$ have a hull of dimension 0 [27]. The proportion is higher for larger q. When $\mathcal{H}(\mathcal{C}) = \{0\}$, we have $I_\Psi(\mathcal{C}) = I_\Phi(\text{HSM}(\mathcal{C}))$. Assuming Conjecture 1 holds, $I_\Psi(\mathcal{C})$ has size $n - O(1)$ for any $q \geqslant 5$ and thus, when n grows, contains an information set of $\mathcal{C}$ with probability as close to 1 as required.

D Full Optimization

The main purpose of this work was to identify the range of parameters for which the asymptotic lower bound could be met. This is done in Proposition 3 and Proposition 4. The Proposition 4 also gives indications on the asymptotic exponent when there are two levels of framework (Algorithm 2) which are optimized independently. A global optimization leads to a different, slightly suboptimal, weight for the first level

$$\omega_{\text{opt}}^{(0)}(R) = \underset{\omega \geqslant \delta_{\text{GV}}(R)}{\text{argmin}} \left(\max \left(\tau_{\text{dec}}(R, \omega), (1-\omega)\tau_{\text{LCE}}^{(1)} \left(\frac{R-\omega}{1-\omega} \right) \right) \right)$$

but achieves a lower global exponent for solving LCE, with $\omega = \omega_{\text{opt}}^{(0)}(R)$, let

$$\tau_{\text{LCE}}^{(0)}(R) = \frac{1}{2}\nu(R, \omega) + \max \left(\tau_{\text{dec}}(R, \omega), (1-\omega)\tau_{\text{LCE}}^{(1)} \left(\frac{R-\omega}{1-\omega} \right) \right). \tag{29}$$

The above exponent only applies when the optimal exponent $\tau_{\text{LCE}}^{(1)}(R)$ cannot be reached. The improvement is slight (see the Tables 5,6,7) except for the special case $q = 5$ (and $q = 7$ with the MMT decoder).

Sagemath code to reproduce the numerical values for complexity estimates and asymptotic exponents given in the paper is available at `https://github.com/sendrier/schurinv_lce`. It involves some optimization routines, including the one mentioned above, which can be applied to any parameter set.

q	$\tau^{(1)}_{\mathrm{LCE}}(R)$	this work, with optimization		$\tau^{(2)}_{\mathrm{LCE}}(R)$
		full (29)	per level	
5	0.23336107	0.23577236	0.26860701	0.23338885
7	0.26873065	= (2)	= (2)	0.26893758
8	0.28162410	= (2)	= (2)	0.28196000
9	0.29248125	= (2)	= (2)	0.29296226
11	0.30990606	= (**2**)	= (2)	**0.31070779**
13	0.32342300	0.32446529	= (2)	0.32456729
16	0.33903595	0.33980892	= (2)	0.34070739
17	0.34335639	0.34405738	= (2)	0.34520342
19	0.35101357	0.35159120	= (2)	0.35320897
23	0.36338266	0.36377602	= (2)	0.36625559
25	0.36848280	0.36880721	= (2)	0.37168309
27	0.37303556	0.37330241	= (2)	0.37655498
29	0.37713332	0.37735220	= (2)	0.38096332
31	0.38084789	0.38102642	= (2)	0.38498065
32	0.38257971	0.38274077	= (2)	0.38686103
37	0.39020726	0.39030023	= (**2**)	**0.39520057**
41	0.39532088	0.39537759	0.39962833	0.40085149
43	0.39761752	0.39766046	0.40134549	0.40340603
47	0.40178146	0.40180428	0.40445802	0.40806855
49	0.40367746	0.40369309	0.40587691	0.41020458
53	0.40715516	0.40716105	0.40848958	0.41414595
59	0.41171803	0.41171818	0.41192769	0.41936670
61	**0.41309288**	= (**1**)	= (**1**)	0.42095145
64	0.41503750	= (1)	= (1)	0.42320257
67	0.41685473	= (1)	= (1)	0.42531751
71	0.41910248	= (1)	= (1)	0.42794920
73	0.42015884	= (1)	= (1)	0.42919122
79	0.42309163	= (1)	= (1)	0.43266350
81	0.42399845	= (1)	= (1)	0.43374332
83	0.42487338	= (1)	= (1)	0.43478876
89	0.42732445	= (1)	= (1)	0.43773335
97	0.43024258	= (1)	= (1)	0.44127474
101	0.43157376	= (1)	= (1)	0.44290461
103	0.43221094	= (1)	= (1)	0.44368517
107	0.43343287	= (1)	= (1)	0.44519187
109	0.43401918	= (1)	= (1)	0.44591904
113	0.43514610	= (1)	= (1)	0.44731657
121	0.43723456	= (1)	= (1)	0.44993191
125	0.43820483	= (1)	= (1)	0.45115069
127	0.43867315	= (1)	= (1)	0.45174271

Table 5. Asymptotic Exponents for $R = 0.5$ with Prange Decoder (Threshold Values in Boldface) - Computed from `https://github.com/sendrier/schurinv_lce`. The indication "= (s)", $s \in \{1, 2\}$, means "= $\tau^{(s)}_{\mathrm{LCE}}(R)$"

q	$\tau^{(1)}_{\mathrm{LCE}}(R)$	this work, with optimization		$\tau^{(2)}_{\mathrm{LCE}}(R)$
		full (29)	per level	
5	0.22645894	0.22962293	0.26667036	0.22658397
7	0.26236866	= (2)	= (2)	0.26273803
8	0.27554146	= (2)	= (2)	0.27605897
9	0.28666436	= (2)	= (2)	0.28734033
11	0.30456828	= (2)	= (2)	0.30558055
13	0.31849596	= (**2**)	= (2)	**0.31985725**
16	0.33461821	0.33569585	= (2)	0.33650802
17	0.33908504	0.34006917	= (2)	0.34114887
19	0.34700640	0.34782969	= (2)	0.34941489
23	0.35981215	0.36039336	= (2)	0.36288908
25	0.36509480	0.36558300	= (2)	0.36849363
27	0.36981120	0.37022185	= (2)	0.37352405
29	0.37405654	0.37440224	= (2)	0.37807518
31	0.37790495	0.37819469	= (2)	0.38222110
32	0.37969912	0.37996432	= (2)	0.38416089
37	0.38760045	0.38776840	= (2)	0.39276218
41	0.39289626	0.39300992	= (**2**)	**0.39858608**
43	0.39527420	0.39536658	0.40077469	0.40121844
47	0.39958452	0.39964372	0.40391604	0.40601886
49	0.40154666	0.40159275	0.40535216	0.40821855
53	0.40514472	0.40517115	0.40798783	0.41227247
59	0.40986341	0.40987199	0.41145784	0.41763989
61	0.41128467	0.41128979	0.41250766	0.41926854
64	0.41329449	0.41329620	0.41399545	0.42158048
67	0.41517214	0.41517230	0.41538599	0.42375173
71	**0.41749389**	= (**1**)	= (**1**)	0.42645301
73	0.41858474	= (1)	= (1)	0.42772739
79	0.42161225	= (1)	= (1)	0.43128757
81	0.42254804	= (1)	= (1)	0.43239435
83	0.42345076	= (1)	= (1)	0.43346597
89	0.42597890	= (1)	= (1)	0.43648293
97	0.42898715	= (1)	= (1)	0.44011014
101	0.43035880	= (1)	= (1)	0.44177317
103	0.43101522	= (1)	= (1)	0.44257422
107	0.43227376	= (1)	= (1)	0.44411352
109	0.43287752	= (1)	= (1)	0.44485602
113	0.43403773	= (1)	= (1)	0.44628915
121	0.43618704	= (1)	= (1)	0.44895471
125	0.43718520	= (1)	= (1)	0.45020230
127	0.43766689	= (1)	= (1)	0.45080712

Table 6. Asymptotic Exponents for $R = 0.5$ with Dumer Decoder (Threshold Values in Boldface) - Computed from `https://github.com/sendrier/schurinv_lce`. The indication "= (s)", $s \in \{1, 2\}$, means "= $\tau^{(s)}_{\mathrm{LCE}}(R)$"

q	$\tau^{(1)}_{\mathrm{LCE}}(R)$	this work, with optimization		$\tau^{(2)}_{\mathrm{LCE}}(R)$
		full (29)	per level	
5	0.21897816	0.22267918	0.25399443	0.21897877
7	0.25539603	0.25861605	0.28948722	0.25570770
8	0.26885548	= (2)	= (2)	0.26944257
9	0.28026159	= (2)	= (2)	0.28109880
11	0.29869849	= (2)	= (2)	0.29997422
13	0.31310505	= (**2**)	= (2)	**0.31476452**
16	0.32984346	0.33132671	= (2)	0.33202101
17	0.33448541	0.33584446	= (2)	0.33683157
19	0.34271846	0.34386473	= (2)	0.34539797
23	0.35602853	0.35685356	= (2)	0.35935463
25	0.36151836	0.36222098	= (2)	0.36515681
27	0.36641883	0.36701835	= (2)	0.37036298
29	0.37082892	0.37134090	= (2)	0.37507009
31	0.37482571	0.37526293	= (2)	0.37935473
32	0.37668871	0.37709258	= (2)	0.38136008
37	0.38488996	0.38516068	= (2)	0.39024389
41	0.39038341	0.39057854	= (2)	0.39625334
43	0.39284906	0.39301290	= (**2**)	**0.39896801**
47	0.39731658	0.39743092	0.40338909	0.40391567
49	0.39934946	0.39944410	0.40483235	0.40618056
53	0.40307581	0.40313877	0.40749254	0.41035515
59	0.40795968	0.40799044	0.41099166	0.41587756
61	0.40942998	0.40945320	0.41205253	0.41755199
64	0.41150853	0.41152287	0.41354917	0.41992638
67	0.41344973	0.41345765	0.41495693	0.42215665
71	0.41584915	0.41585171	0.41670054	0.42492880
73	0.41697612	0.41697719	0.41752074	0.42623732
79	**0.42010264**	= (**1**)	= (**1**)	0.42988947
81	0.42106864	= (1)	= (1)	0.43102656
83	0.42200033	= (1)	= (1)	0.43212244
89	0.42460863	= (1)	= (1)	0.43521346
97	0.42771035	= (1)	= (1)	0.43892667
101	0.42912392	= (1)	= (1)	0.44062898
103	0.42980023	= (1)	= (1)	0.44144788
107	0.43109660	= (1)	= (1)	0.44302329
109	0.43171837	= (1)	= (1)	0.44378119
113	0.43291292	= (1)	= (1)	0.44524363
121	0.43512492	= (1)	= (1)	0.44796972
125	0.43615177	= (1)	= (1)	0.44924481
127	0.43664721	= (1)	= (1)	0.44986104

Table 7. Asymptotic Exponents for $R = 0.5$ with MMT Decoder (Threshold Values in Boldface) - Computed from `https://github.com/sendrier/schurinv_lce`. The indication "= (s)", $s \in \{1, 2\}$, means "$= \tau^{(s)}_{\mathrm{LCE}}(R)$"

E Comparison with [8,9]

An independent paper with similar results appeared [8] just before we released the first version of this preprint. It does not change our statements or results, though some of our comments about the state-of-the-art became obsolete.

As for our Algorithm 2, the proposal in [8] has two successive steps. The first step is a Type 1 instance of the framework, and produces multiple words of weight w in both codes. To check whether $(\boldsymbol{c}_1, \boldsymbol{c}_2) \in \mathcal{C}_1 \times \mathcal{C}_2$ are matching, *all* codewords of weight w_r are computed in the $[n-w, k]_q$ codes $\mathsf{Puncture}(\mathcal{C}_1, \boldsymbol{c}_1)$ and $\mathsf{Puncture}(\mathcal{C}_2, \boldsymbol{c}_2)$. From there the whole isometry can be recovered in polynomial time if w_r is chosen appropriately, that is just above the Gilbert-Varshamov distance, so that the number of weight-w_r codewords is a, possibly small, proportion of n. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be the initial $[n,k]_q$ codes, let $R = k/n$ and $\omega = w/n$, where w is the codeword weight in the first step. The asymptotic exponent is given in [8, Lemma 6] as

$$\tau'_{\mathrm{LCE}}(R) = \min_{\omega}\Bigg(\max\Bigg(h_2(\omega) - (1-R)h_2\left(\frac{\omega}{1-R}\right) - \gamma/2, \\ \gamma/2 + (1-\omega)h_2\left(\frac{\mu}{1-\omega}\right) - (1-R-\omega)h_2\left(\frac{\mu}{1-R-\omega}\right)\Bigg)\Bigg)$$

where $\gamma = h_2(\omega) + \omega\log_2(q-1) - (1-R)\log_2 q$ and $\mu = (1-\omega)h_q^{-1}\left(1 - \frac{R}{1-\omega}\right)$. With the notation of this paper, we have $\gamma = \nu(R,\omega)$ and $\delta \triangleq \frac{\mu}{1-\delta} = \delta_{\mathrm{GV}}(\frac{R}{1-\omega})$ and we rewrite

$$\tau'_{\mathrm{LCE}}(R) = \min_{\omega}\left(\frac{1}{2}\nu(R,\omega) + \max\left(\tau_{\mathrm{dec}}(R,\omega), (1-\omega)\tau_{\mathrm{dec}}\left(\tfrac{R}{1-\omega}, \delta\right)\right)\right), \quad (30)$$

with $\tau_{\mathrm{dec}} = \tau_{\mathrm{prange}}$. Note that the above form clearly shows that the second step is amortized if its asymptotic exponent (right-hand side of the max) is smaller than the decoding exponent of the first step (left-hand side of the max). The authors of [8] use Prange's asymptotic exponent but any other decoder can be plugged in (30). We remark in Table 8 that the algorithm proposed in [8] reaches the framework's optimal exponent (bold) for larger q, and otherwise is very close, though slightly above the exponent obtained with our approach. This can be understood by observing that the right-hand side term in the max of

$$\tau^{(0)}_{\mathrm{LCE}}(R) = \frac{1}{2}\nu(R,\omega) + \max\left(\tau_{\mathrm{dec}}(R,\omega), (1-\omega)\tau^{(1)}_{\mathrm{LCE}}\left(\frac{R-\omega}{1-\omega}\right)\right). \quad (29)$$

(minimized when $\omega = \omega^{(0)}_{\mathrm{opt}}(R)$) corresponds to an LCE instance of low rate for which our reduction applies and which can be solved optimally. In contrast, in [8] the LCE instance is solved with an algorithm which has the same, suboptimal, exponent as Leon's instance. Therefore, it takes a larger alphabet size q before it becomes lower than $\tau_{\mathrm{dec}}(R,\omega)$.

A function was included in `https://github.com/sendrier/schurinv_lce` to estimate (30).

	Prange		Dumer		MMT	
q	this work	[8]	this work	[8]	this work	[8]
5	0.23577236	0.23636954	0.22962293	0.23011120	0.22267918	0.22302621
7	0.26893758	0.27128752	0.26273803	0.26542511	0.25861605	0.25892366
8	0.28196000	0.28397277	0.27605897	0.27833397	0.26944257	0.27212853
9	0.29296226	0.29463998	0.28734033	0.28922088	0.28109880	0.28329448
11	0.31070779	0.31173856	0.30558055	0.30672376	0.29997422	0.30129818
13	0.32446529	0.32498990	0.31985725	0.32033109	0.31476452	0.31533073
16	0.33980892	0.34029155	0.33569585	0.33608230	0.33132671	0.33160611
17	0.34405738	0.34452648	0.34006917	0.34044802	0.33584446	0.33612166
19	0.35159120	0.35203323	0.34782969	0.34819250	0.34386473	0.34413630
23	0.36377602	0.36416762	0.36039336	0.36072442	0.35685356	0.35711283
25	0.36880721	0.36917531	0.36558300	0.36589964	0.36222098	0.36247345
27	0.37330241	0.37364868	0.37022185	0.37052471	0.36701835	0.36726343
29	0.37735220	0.37767647	0.37440224	0.37468876	0.37134090	0.37157851
31	0.38102642	0.38133049	0.37819469	0.37846723	0.37526293	0.37549206
32	0.38274077	0.38303484	0.37996432	0.38022981	0.37709258	0.37731781
37	0.39030023	0.39054841	0.38776840	0.38800009	0.38516068	0.38536474
41	0.39537759	0.39559248	0.39300992	0.39321619	0.39057854	0.39076479
43	0.39766046	0.39786058	0.39536658	0.39556099	0.39301290	0.39319121
47	0.40180428	0.40197505	0.39964372	0.39981415	0.39743092	0.39759240
49	0.40369309	0.40384964	0.40159275	0.40175239	0.39944410	0.39959718
53	0.40716105	0.40729160	0.40517115	0.40530913	0.40313877	0.40327541
59	0.41171818	0.41181316	0.40987199	0.40997998	0.40799044	0.40810333
61	**0.41309288**	0.41317691	0.41128979	0.41138842	0.40945320	0.40955840
64	**0.41503750**	0.41510739	0.41329620	0.41338136	0.41152287	0.41161688
67	**0.41685473**	0.41691149	0.41517230	0.41524400	0.41345765	0.41354064
71	**0.41910248**	0.41914533	**0.41749389**	0.41754932	0.41585171	0.41592057
73	**0.42015884**	0.42019584	**0.41858474**	0.41863311	0.41697719	0.41703917
79	**0.42309163**	0.42311444	**0.42161225**	0.42164361	**0.42010264**	0.42014500
81	**0.42399845**	0.42401752	**0.42254804**	0.42257477	**0.42106864**	0.42110541
83	**0.42487338**	0.42488916	**0.42345076**	0.42347339	**0.42200033**	0.42203204
89	**0.42732445**	0.42733262	**0.42597890**	0.42599246	**0.42460863**	0.42462818
97	**0.43024258**	0.43024489	**0.42898715**	0.42899202	**0.42771035**	0.42772006
101	**0.43157376**	0.43157458	**0.43035880**	0.43036127	**0.42912392**	0.42912923
103	**0.43221094**	0.43221132	**0.43101522**	0.43101681	**0.42980023**	0.42980416
107	**0.43343287**	**0.43343287**	**0.43227376**	0.43227421	**0.43109660**	0.43109848
109	**0.43401918**	**0.43401918**	**0.43287752**	0.43287767	**0.43171837**	0.43171952
113	**0.43514610**	**0.43514610**	**0.43403773**	**0.43403773**	**0.43291292**	0.43291318
121	**0.43723456**	**0.43723456**	**0.43618704**	**0.43618704**	**0.43512492**	**0.43512492**
125	**0.43820483**	**0.43820483**	**0.43718520**	**0.43718520**	**0.43615177**	**0.43615177**
127	**0.43867315**	**0.43867315**	**0.43766689**	**0.43766689**	**0.43664721**	**0.43664721**

Table 8. Asymptotic exponents for this work and [8] for rate $R = 0.5$ and various decoders. Values in boldface reach the lower bound.