Reductions Between Code Equivalence Problems

Mahdi Cheraghchi*, Nikhil Shagrithaya†, Alexandra Veliche‡

Abstract

In this paper we present two reductions between variants of the Code Equivalence problem. We give polynomial-time Karp reductions from Permutation Code Equivalence (PCE) to both Linear Code Equivalence (LCE) and Signed Permutation Code Equivalence (SPCE). Along with a Karp reduction from SPCE to the Lattice Isomorphism Problem (LIP) shown by Bennett and Win (2024), our second result implies a reduction from PCE to LIP.

1 Introduction

The *Code Equivalence* (CE) problem asks if two given codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are "equivalent" in some metric-preserving way; variants of the CE problem specify the type of equivalence. *Permutation Code Equivalence* (PCE) asks if the codes are the same up to permutation of the coordinates of codewords, while *Signed Permutation Code Equivalence* (SPCE) allows equivalence up to signed permutations. Still more generally, *Linear Code Equivalence* (LCE) allows an equivalence up to permutation and multiplication by a (non-zero) constant. The variants of CE belong to a larger class of isomorphism problems that ask the following question: Given two objects of the same kind, is there an isomorphism that transforms one object into the other? Other examples of such problems include Matrix Code Equivalence and Graph Isomorphism.

Besides being interesting problems in their own right, CE problems have many important applications. Perhaps most notably, the conditional hardness of CE variants has been used as a security assumption for several cryptographic schemes proposed to be post-quantum. These include the seminal McEliece public-key encryption scheme [McE78], a recent NIST post-quantum standardization submission called "Classic McEliece" [ABC+22], and the more recent LESS identification scheme [BBPS21, BBPS22].

Given the relevance of these problems to cryptography, there has been a considerable amount of work on designing efficient algorithms that solve these problems. Leon [Leo82] introduced an algorithm for the search version of PCE that works well for a large number of codes, but still requires exponential time in the worst case. The Support Splitting Algorithm developed by Sendrier in [Sen00] and extended by Sendrier and Simos in [SS13a] gives an algorithm for linear codes that is efficient for codes with small hull, where the hull of a code is defined by the intersection of the code and its dual. This algorithm, however, does not work for the case where the dimension of the hull is zero, but this case was later handled by Bardet, Otmani, and Saeed-Taha in [BOS19]. In the latter paper, the

*University of Michigan – Ann Arbor. Email: mahdich@umich.edu
†University of Michigan – Ann Arbor. Email: nshagri@umich.edu
‡University of Michigan – Ann Arbor. Email: aveliche@umich.edu

authors reduce the problem of deciding PCE to a weighted version of Graph Isomorphism, and then use a variant of Babai's algorithm for solving the latter problem [Bab16] to give a quasi-polynomial time algorithm that computes PCE for the zero hull case.

On the other side of the cryptographic coin, considerable attention has been devoted towards understanding the computational hardness of these problems. Petrank and Roth in [PR97] showed that Graph Isomorphsim reduces to PCE, and that PCE is not NP-complete unless the polynomial hierarchy collapses. This was used as evidence for the computational hardness of PCE until Babai introduced his quasi-polynomial time algorithm for deciding Graph Isomorphism [Bab16]. Even under the assumption that the polynomial hierarchy does not collapse, however, it is still possible that both Graph Isomorphism and PCE cannot be decided in polynomial time.

The Matrix Code Equivalence problem is concerned with code equivalence in the rank metric. Couvreur, Debris-Alazard and Gaborit in [CDG20] give a reduction from LCE to Matrix Code Equivalence, thus showing that determining code equivalence under the rank metric is at least as hard as determining equivalence under the Hamming metric.

In [SS13a], Sendrier and Simos give a reduction from LCE to PCE that runs in time polynomial in the blocklength n and alphabet size q of the codes. To do this, they use the *closure* of a code $\mathcal{C} \subseteq \mathbb{F}_q^n$, which is the set defined by taking every codeword and multiplying each of its n coordinates with all the non-zero field elements to produce a new vector of length $n(q-1)$. Then using the fact that multiplication by any non-zero field element induces a permutation on the elements in $\mathbb{F}_q^*$, they reframe scalar multiplication as a permutation over $\mathbb{F}_q^*$. In [DG23], Ducas and Gibbons use a specific form of this closure to prove a reduction from SPCE to PCE. Biasse and Micheli in [BM23] give a search-to-decision reduction for PCE, which implies that the decision version of PCE is at least as hard as the search version. Bennett and Win in [BW24] give several reductions between CE variants and other isomorphism problems. They extend the closure technique from [SS13a] to give a reduction from LCE to SPCE, and also give a reduction from SPCE to the *Lattice Isomorphism Problem* (LIP).

LIP is an analogous problem to CE for lattices. A *lattice* is an infinite discrete additive subgroup of Euclidean space generated by taking all integer linear combinations of a finite set of linearly independent vectors; these vectors form a *basis* for the lattice. Two lattices are said to be *isomorphic* if there exists an orthogonal transformation that transforms one lattice (basis) into the other. LIP asks if such an isomorphism exists. In their reduction from SPCE to LIP, Bennet and Win use a well-known construction (called Construction A) that lifts a linear code over a prime finite field $\mathbb{F}_p$ to a lattice in $\mathbb{R}^n$. Because this construction is only well-defined for prime fields, their reduction only works for prime fields. It remains an open problem to find a reduction from any CE variant to LIP for non-prime fields.

1.1 Results

Now we state our main results and place them into context of the prior work described above. In [BW24], the authors prove Karp reductions from LCE to PCE and from LCE to SPCE. In our work, we show a reverse Karp reduction from PCE to LCE.

Theorem 1.1 (PCE reduces to LCE, informal)**.** *For linear codes with blocklength n over a field of size q, there is a Karp reduction from* PCE *to* LCE *that runs in* $\mathrm{poly}(n, \log q)$ *time.*

The formal statement is given in Theorem 3.1. This result, along with the Karp reduction in [SS13b, BW24] from LCE to PCE running in $\mathrm{poly}(n, q)$ time, implies that the problems LCE and PCE are computationally equivalent up to factors in the runtime that are polynomial in n and q.

Additionally, the Karp reduction from LCE to SPCE detailed in [BW24, Theorem 4.4, Corollary 4.5] combines with our result to give a reduction from PCE to SPCE running in $\mathrm{poly}(n, q)$ time. In situations where q is much larger than n (as is the case for Reed-Solomon codes, for example), it is desirable to have the runtime depend only logarithmically on q. Note that at least $\log q$ time is required to express a single element of the field, so the dependence on q cannot be smaller than $\log q$. Our second result is a Karp reduction from PCE to SPCE whose runtime depends only logarithmically on q.

Theorem 1.2 (PCE reduces to SPCE, informal)**.** *For linear codes with blocklength n over a field of size q, there is a Karp reduction from* PCE *to* SPCE *that runs in* $\mathrm{poly}(n, \log q)$ *time.*

The formal statement is given in Theorem 3.2. This result, together with a Karp reduction from SPCE to PCE from [DG23, Lemma 10] running in $\mathrm{poly}(n, \log q)$ time, implies that PCE and SPCE are computationally equivalent problems up to factors in the runtime that are polynomial in n and $\log q$.

Finally, by combining the above result with the Karp reduction from SPCE over prime fields to LIP from [BW24, Theorem 5.1], we obtain the following corollary.

Corollary 1.3 (PCE reduces to LIP)**.** For any prime p, there is a Karp reduction from PCE over a field of order p to LIP that runs in $\mathrm{poly}(n, \log p)$ time.

2 Preliminaries

Let $\mathbb{N}$ denote the set of all positive integers. For any $n, m \in \mathbb{N}$ such that $n < m$, we use the notation $[n, m]$ to denote the set of integers $\{n, n+1, \ldots, m\}$. For $n \in \mathbb{N}$, we use the abbreviated notation $[n] := [1, n]$. For any prime power $q \in \mathbb{N}$, we use $\mathbb{F}_q$ to denote the field of size q. For any commutative ring R, we denote its multiplicative subgroup by R^*; for a field $\mathbb{F}$, we have $\mathbb{F}^* = \mathbb{F} \setminus \{0\}$.

We use boldface lowercase letters, such as $\mathbf{v}$, to denote vectors and boldface uppercase letters, such as $\mathbf{A}$, to denote matrices. We use $\mathbf{e}_i$ to denote the column vector containing a one in the i-th position and zeros everywhere else, and $\mathbf{0}$ will denote the all zeroes column vector. For any matrix $\mathbf{A}$ with n columns and $i \in [n]$, we use $\mathbf{A}[i]$ to denote the i-th column. We also use $\mathbf{A}[i, j]$ to denote the entry in the i-th row and j-th column. To denote the block submatrix of a matrix $\mathbf{A}$ spanned by rows r_i through r_j (inclusive) and columns c_k through c_ℓ (inclusive), we use the notation $\mathbf{A}[r_i : r_j, c_k : c_\ell]$.

2.1 Matrix Groups

For any field $\mathbb{F}$ and $n \in \mathbb{N}$, we use the following notation for special sets of $n \times n$ matrices over $\mathbb{F}$: $\mathcal{GL}_n(\mathbb{F})$ denotes the group of invertible matrices, $\mathcal{P}_n(\mathbb{F})$ denotes the set of permutation matrices, $\mathcal{SP}_n(\mathbb{F})$ denotes the set of signed permutation matrices, and $\mathcal{M}_n(\mathbb{F})$ denotes the set of monomial matrices. A *permutation* matrix $\mathbf{P} \in \mathcal{P}_n(\mathbb{F})$ contains exactly one 1 in each row and column and 0s everywhere else. A *signed permutation* matrix $\mathbf{P} \in \mathcal{SP}_n(\mathbb{F})$ contains exactly one non-zero entry in each row and column, but each of these can be either 1 or -1. A *monomial* matrix $\mathbf{M} \in \mathcal{M}_n(\mathbb{F})$ contains exactly one non-zero entry in each row and column, but these can take values in $\mathbb{F}^*$; any

monomial matrix $\mathbf{M}$ can be written as $\mathbf{M} = \mathbf{DP}$ for some diagonal matrix $\mathbf{D} = \text{diag}(d_1, ..., d_n)$, where $d_i \in \mathbb{F}^*$, and permutation matrix $\mathbf{P} \in \mathcal{P}_n(\mathbb{F})$. Each of these sets of matrices forms a group under matrix multiplication. Furthermore, these satisfy $\mathcal{P}_n(\mathbb{F}) \subseteq \mathcal{SP}_n(\mathbb{F}) \subseteq \mathcal{M}_n(\mathbb{F}) \subseteq \mathcal{GL}_n(\mathbb{F})$. If it is clear from context, we do not specify the field for these matrix groups. For any permutation matrix $\mathbf{P} \in \mathcal{SP}_n$, we denote the corresponding permutation map over the set of column indices by $\sigma_{\mathbf{P}} : [n] \to [n]$.

The following observation will be useful for our reduction in Section 3.

Observation 2.1. Any invertible matrix $\mathbf{S} \in \mathcal{GL}_k(\mathbb{F})$ induces a bijective map on $\mathbb{F}^k$. In particular, for any $\mathbf{x}, \mathbf{y} \in \mathbb{F}^k$, we have $\mathbf{Sx} = \mathbf{Sy}$ if and only if $\mathbf{x} = \mathbf{y}$. Then when $\mathbf{S}$ is multiplied by some matrix $\mathbf{A}$ of the appropriate dimension, it maps identical (distinct) columns in $\mathbf{A}$ to identical (distinct) columns in $\mathbf{SA}$.

2.2 Codes

A *linear code* is a finite-dimensional vector space over a finite field. Let $\mathbb{F}_q$ be a finite field of size q for some prime power q. Using the standard notation for linear codes, we say $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a $[n, k, d]_q$*-code* for some $q, n, k, d \in \mathbb{N}$, where q is the *alphabet size*, n is the *blocklength* given by the length of the codeword vectors, k is the *dimension* of $\mathcal{C}$ as a linear subspace of $\mathbb{F}_q^n$, and d is the *minimum distance* given by the minimum Hamming weight over all non-zero codewords of $\mathcal{C}$. Any linear code $\mathcal{C}$ can be expressed as the row span of a *generator* matrix $\mathbf{G} \in \mathbb{F}_q^{k \times n}$. Note this is not unique (as elementary row operations do not change the span of $\mathbf{G}$).

2.3 Equivalence Problems

Two codes $\mathcal{C}, \mathcal{C}' \subseteq \mathbb{F}_q^n$ are said to be *permutation equivalent* if there exists a permutation of the coordinates of $\mathcal{C}$ that gives $\mathcal{C}'$. We formalize this notion and define two variants of this problem below.

Definition 2.2 (PCE)**.** For $k, n \in \mathbb{N}$ and field $\mathbb{F}_q$ of size q, the *Permutation Code Equivalence* problem, denoted by PCE, is the following decision problem: Given a pair of generator matrices $\mathbf{G}, \mathbf{H} \in \mathbb{F}_q^{k \times n}$, decide whether there exist an invertible matrix $\mathbf{S} \in \mathcal{GL}_k(\mathbb{F}_q)$ and a permutation matrix $\mathbf{P} \in \mathcal{P}_n(\mathbb{F}_q)$ for which $\mathbf{SGP} = \mathbf{H}$.

If we allow code equivalence up to *signed* permutations, we can consider the following problem.

Definition 2.3 (SPCE)**.** For $k, n \in \mathbb{N}$ and field $\mathbb{F}_q$ of size q, the *Signed Permutation Code Equivalence* problem, denoted by SPCE, is the following decision problem: Given a pair of generator matrices $\mathbf{G}, \mathbf{H} \in \mathbb{F}_q^{k \times n}$, decide whether there exist an invertible matrix $\mathbf{S} \in \mathcal{GL}_k(\mathbb{F}_q)$ and a signed permutation matrix $\mathbf{P} \in \mathcal{SP}_n(\mathbb{F}_q)$ for which $\mathbf{SGP} = \mathbf{H}$.

More generally, we can consider code equivalence up to permutation and multiplication by any non-zero constant.

Definition 2.4 (LCE)**.** For $k, n \in \mathbb{N}$ and field $\mathbb{F}_q$ of size q, the *Linear Code Equivalence* problem, denoted by LCE, is the following decision problem: Given a pair of generator matrices $\mathbf{G}, \mathbf{H} \in \mathbb{F}_q^{k \times n}$, decide whether there exist an invertible matrix $\mathbf{S} \in \mathcal{GL}_k(\mathbb{F}_q)$ and a monomial matrix $\mathbf{M} \in \mathcal{M}_n(\mathbb{F}_q)$ for which $\mathbf{SGM} = \mathbf{H}$.

For brevity, we say that a pair of matrices $(\mathbf{G}, \mathbf{H})$ is in PCE (or SPCE, LCE) if $\mathbf{G}$ and $\mathbf{H}$ satisfy the conditions required for the pair to be a YES instance of the problem.

3 Reductions from PCE to LCE and SPCE

We formally state our results below and prove them in this section. All our reductions are deterministic Karp reductions, so we do not specify this in our statements hereafter.

Theorem 3.1 (PCE reduces to LCE)**.** *There is a reduction from* PCE *to* LCE *that runs in* $\mathrm{poly}(n, \log q)$ *time, where* n *is the blocklength and* q *is the field size of the input code pair.*

Theorem 3.2 (PCE reduces to SPCE)**.** *There is a reduction from* PCE *to* SPCE *that runs in* $\mathrm{poly}(n, \log q)$ *time, where* n *is the blocklength and* q *is the field size of the input code pair.*

We will use the following two lemmas to justify why certain assumptions about the input can be made without loss of generality.

Lemma 3.3. Let $k, n \in \mathbb{N}$ and q be a prime power. For any matrices $\mathbf{G}, \mathbf{H} \in \mathbb{F}_q^{k \times n}$, if $(\mathbf{G}, \mathbf{H})$ is in PCE, then no column of $\mathbf{G}$ appears in $\mathbf{G}$ more times than a column of $\mathbf{H}$ appears in $\mathbf{H}$.

Proof. By definition, there must be some $\mathbf{S} \in \mathcal{GL}_k$ and $\mathbf{P} \in \mathcal{P}_n$ such that $\mathbf{SGP} = \mathbf{H}$. By Observation 2.1, $\mathbf{S}$ takes identical columns in $\mathbf{G}$ to identical columns in $\mathbf{SG}$. When multiplied with $\mathbf{G}$ on the right, $\mathbf{P}$ only permutes the columns of $\mathbf{G}$ and so does not change the frequency with which any column appears in the matrix $\mathbf{SG}$. Therefore for every column $\mathbf{c}$ in $S\mathbf{G}$, its corresponding column in $\mathbf{H}$ must appear the same number of times as $\mathbf{c}$ appears in $\mathbf{SG}$. □

Lemma 3.4. For any $\mathbf{G}, \mathbf{H} \in \mathbb{F}_q^{k \times n}$, let $\overline{\mathbf{G}}$ and $\overline{\mathbf{H}}$ be the submatrices of $\mathbf{G}$ and $\mathbf{H}$, respectively, obtained by removing all columns equal to $\mathbf{0} \in \mathbb{F}_q^k$. Then $(\mathbf{G}, \mathbf{H})$ is in PCE if and only if $(\overline{\mathbf{G}}, \overline{\mathbf{H}})$ is in PCE.

This lemma follows from the following trivial reduction: If $\mathbf{G}$ and $\mathbf{H}$ satisfy the PCE condition, there exist matrices $\mathbf{S} \in \mathcal{GL}_k$ and $\mathbf{P} \in \mathcal{P}_n$ such that $\mathbf{SGP} = \mathbf{H}$. Because $\mathbf{S}$ is a linear map, it will always map $\mathbf{0} \in \mathbb{F}_q^k$ to itself, so there is an invertible submatrix of $\mathbf{S}$ and a permutation submatrix of $\mathbf{P}$ for which $\overline{\mathbf{G}}$ and $\overline{\mathbf{H}}$ satisfy the PCE condition.

As a result of Lemma 3.4 and Lemma 3.3, we can assume without loss of generality that any given input pair of matrices do not contain a zero column and have the same column frequency.

Now we define the construction that will be used to transform the input matrices in our reduction.

Construction 3.5. Given a $k \times n$ matrix $\mathbf{A}$ over $\mathbb{F}_q$ with column vectors $\mathbf{A}[1], \ldots, \mathbf{A}[n] \in \mathbb{F}_q^k$, let $m_{\mathbf{A}}$ denote the maximum number of times a column appears in $\mathbf{A}$. Denote $m := m_{\mathbf{A}} + 1$. We construct the $k \times nm$ matrix $\widehat{\mathbf{A}}$ by appending m copies of each column vector in order. More explicitly, for every $i \in [n]$ and $j \in [m]$, we set $\widehat{\mathbf{A}}[(i-1)m + j] := \mathbf{A}[i]$. Define the $(k+1) \times n$ matrix $\mathbf{A}'_1$ by appending a row of all ones at the bottom of matrix $\mathbf{A}$, define the $(k+1) \times nm$ matrix $\mathbf{A}'_2$ by appending a row of all zeros at the bottom of matrix $\widehat{\mathbf{A}}$, and define the $(k+1) \times (nm+1)$ matrix $\mathbf{A}'_3$ by placing ones in every entry of the last row and zeros everywhere else. Denoting $n' := n + 2nm + 1$, we obtain the final $(k+1) \times n'$ matrix $\mathbf{A}'$ by concatenating the block matrices $\mathbf{A}'_1$, $\mathbf{A}'_2$, and $\mathbf{A}'_3$:

$$\mathbf{A}' := \left[\begin{array}{c|c|c} \mathbf{A}'_1 & \mathbf{A}'_2 & \mathbf{A}'_3 \end{array} \right].$$

$\mathbf{A}$	$\widehat{\mathbf{A}}$	$\mathbf{0}$
1 1 ... 1	0 0 0	1 1 1

Figure 1: The matrix A' obtained by Construction 3.5.

Note that if the given matrix $\mathbf{A}$ has full row rank, then $\mathbf{A}'$ must have full row rank.

With these assumptions and definitions in place, we now prove Theorem 3.1.

Proof of Theorem 3.1: Denote the pair of input matrices by $\mathbf{G}, \mathbf{H} \in \mathbb{F}_q^{k \times n}$. By Lemma 3.4, we can assume without loss of generality that $\mathbf{G}$ and $\mathbf{H}$ do not contain any zero columns. Moreover, we can assume that $\mathbf{G}$ and $\mathbf{H}$ have the same row rank, because otherwise the codes represented by these matrices differ in size, and hence cannot be equivalent. We can also assume that their row rank is equal to k, as otherwise we can consider the submatrices consisting only of linearly independent rows, without loss of generality. Let $m_{\mathbf{G}}$ denote the maximum number of times a column appears in $\mathbf{G}$, and define $m_{\mathbf{H}}$ similarly. By Lemma 3.3, we must have $m_{\mathbf{G}} = m_{\mathbf{H}}$. Denoting $m := m_{\mathbf{G}} + 1 = m_{\mathbf{H}} + 1$ and $n' := n + 2nm + 1$, we obtain $(k+1) \times n'$ matrices $\mathbf{G}'$ and $\mathbf{H}'$, from matrices $\mathbf{G}$ and $\mathbf{H}$ respectively, by using Construction 3.5. Note that because $m \leq n + 1$, these matrices can be constructed deterministically in $\text{poly}(n, \log q)$ time. The claim in Theorem 3.1 then follows from Lemma 3.6 and Corollary 3.11, which are stated and proven below. □

The proof of Theorem 3.2 is nearly identical to the above proof, but with a restriction to the set of signs $\{-1, +1\}$ instead of all non-zero field elements in the proof of Corollary 3.11.

3.1 From PCE to LCE

First we show the forward direction. We prove that our construction preserves the permutation equivalence of the input pair, which gives the following stronger result.

Lemma 3.6. For any $\mathbf{G}, \mathbf{H} \in \mathbb{F}_q^{k \times n}$, if $(\mathbf{G}, \mathbf{H})$ is in PCE, then $(\mathbf{G}', \mathbf{H}')$ is in PCE (and therefore in LCE).

Proof. By definition, $\mathbf{G}, \mathbf{H}$ is in PCE if and only if there exist an invertible matrix $\mathbf{S} \in \mathcal{GL}_k$ and a permutation matrix $\mathbf{P} \in \mathcal{P}_n$ such that $\mathbf{SGP} = \mathbf{H}$. From these matrices we will construct $\mathbf{S}' \in \mathcal{GL}_{k+1}$ and $\mathbf{P}' \in \mathcal{P}_{n'}$ such that $\mathbf{S}'\mathbf{G}'\mathbf{P}' = \mathbf{H}'$. Define $\mathbf{S}' \in \mathcal{GL}_{k+1}$ such that the top-left block submatrix is $\mathbf{S}'[1:k, 1:k] = \mathbf{S}$, the bottom-right entry is $\mathbf{S}'[k+1, k+1] = 1$, and all other entries are zero.

Let $\sigma_{\mathbf{P}} \colon [n] \to [n]$ be the permutation map described by $\mathbf{P}$. Define a new permutation $\sigma_{\mathbf{P}'} \colon [n'] \to [n']$ that permutes the first n columns of $\mathbf{G}'$ among each other according to P, i.e. $\sigma_{\mathbf{P}'}(x) := \sigma_{\mathbf{P}}(x)$ for all $x \in [n]$. We also define $\sigma_{\mathbf{P}'}$ such that it permutes each of the nm columns in the second block according to how its corresponding column was permuted in the first block. Formally, for every $x \in [n+1, n+nm]$, we write $x = n + m(i-1) + j$ uniquely for $i \in [n]$ and $j \in [m]$ and set $\sigma_{\mathbf{P}'}(x) := n + m(\sigma_{\mathbf{P}}(i) - 1) + j$. Lastly, we define $\sigma_{\mathbf{P}'}$ such that it sends each of the last $nm + 1$

columns identically to itself, i.e. $\sigma_{\mathbf{P}'}(x) := x$ for every $x \in [n+nm+1, n+2nm+1]$. Finally we let $\mathbf{P}' \in \mathcal{P}_{n'}$ be the permutation matrix defined by $\sigma_{\mathbf{P}'}$.

We claim that $\mathbf{S}'\mathbf{G}'\mathbf{P}' = \mathbf{H}'$. This holds if and only if $\mathbf{S}'\mathbf{G}'\mathbf{P}'[x] = \mathbf{H}'[x]$ for every column $x \in [n']$. We show that this is the case for every column in each of the three blocks. For every $x \in [n]$, by the definition of $\sigma_{\mathbf{P}'}$, the fact that each column in the first block has 1 as the last entry, and the PCE assumption,

$$\mathbf{S}'\mathbf{G}'\mathbf{P}'[x] = \mathbf{S}'\mathbf{G}'[\sigma_{\mathbf{P}'}(x)] = \mathbf{S}\mathbf{G}[\sigma_{\mathbf{P}}(x)] \parallel 1 = \mathbf{S}\mathbf{G}[x]\mathbf{P} \parallel 1 = \mathbf{S}\mathbf{G}\mathbf{P}[x] \parallel 1 = \mathbf{H}[x] \parallel 1 = \mathbf{H}'[x].$$

Here "$\parallel$" denotes concatenation of an additional entry at the end of the column vector. For $x \in [n+1, n+nm]$, we can write $x = n + m(i-1) + j$ for some $i \in [n]$ and $j \in [m]$. Then by the definition of $\sigma_{\mathbf{P}'}$, the fact that each column in the second block has 0 as the last entry, and the PCE assumption,

$$\begin{aligned}\mathbf{S}'\mathbf{G}'\mathbf{P}'[x] &= \mathbf{S}\mathbf{G}[\sigma_{\mathbf{P}'}(x)] \parallel 0 = \mathbf{S}\mathbf{G}[n + m(\sigma_{\mathbf{P}}(i) - 1) + j] \parallel 0 = \mathbf{S}\mathbf{G}[\sigma_{\mathbf{P}}(i)] \parallel 0 \\ &= \mathbf{S}\mathbf{G}\mathbf{P}[i] \parallel 0 = \mathbf{H}[i] \parallel 0 = \mathbf{H}[n + m(i-1) + j] \parallel 0 = \mathbf{H}'[x].\end{aligned}$$

Lastly, for all $x \in [n+nm+1, n+2nm+1]$, by definition of $\sigma_{\mathbf{P}'}$, the fact that each column in the last block is identical to $\mathbf{e}_{k+1}$, and the PCE assumption,

$$\mathbf{S}'\mathbf{G}'\mathbf{P}'[x] = \mathbf{S}\mathbf{G}[\sigma_{\mathbf{P}'}(x)] \parallel 1 = \mathbf{S}\mathbf{G}[x] \parallel 1 = \mathbf{S}\mathbf{G}\mathbf{P}[x] \parallel 1 = \mathbf{H}[x] \parallel 1 = \mathbf{H}'[x].$$

Therefore, $(\mathbf{G}', \mathbf{H}')$ is in PCE. □

Because the matrix groups are nested $\mathcal{P}_n \subseteq \mathcal{SP}_n \subseteq \mathcal{M}_n$, Lemma 3.6 immediately implies the forward direction of both statements Theorem 3.1 and Theorem 3.2.

3.2 From LCE to PCE

Now we prove the other direction, and show that if the constructed matrix pair $(\mathbf{G}', \mathbf{H}')$ is in LCE, then the original matrix pair $(\mathbf{G}, \mathbf{H})$ must be in PCE. This requires some careful analysis of how the change-of-basis matrix $\mathbf{S}$ and monomial matrix $\mathbf{M}$ affect each block of the matrices $\mathbf{G}'$ and $\mathbf{H}'$.

First we show that for any monomial matrix $\mathbf{M} = \mathbf{D}\mathbf{P}$ for which $(\mathbf{G}', \mathbf{H}')$ is in LCE, the permutation matrix $\mathbf{P}$ must respect the "boundaries" between the block matrices of $\mathbf{G}'$.

Lemma 3.7. For any $\mathbf{G}, \mathbf{H} \in \mathbb{F}_q^{k \times n}$, if $\mathbf{S}'\mathbf{G}'\mathbf{M}' = \mathbf{H}'$ for some $\mathbf{S}' \in \mathcal{GL}_{k+1}$ and $\mathbf{M}' = \mathbf{D}'\mathbf{P}' \in \mathcal{M}_{n'}$, where $\mathbf{P}' \in \mathcal{P}_{n'}$, the corresponding permutation map $\sigma_{\mathbf{P}'} \colon [n'] \to [n']$ satisfies the following:

(i) For every $i \in [1, n]$, $\sigma_{\mathbf{P}'}(i) \in [1, n]$.

(ii) For every $i \in [n+1, n+mn]$, $\sigma_{\mathbf{P}'}(i) \in [n+1, n+mn]$.

(iii) For every $i \in [n+nm+1, n']$, $\sigma_{\mathbf{P}'}(i) \in [n+mn+1, n']$.

Proof. As a result of Lemma 3.4, we can assume without loss of generality that $\mathbf{G}$ and $\mathbf{H}$ do not contain a zero column. Suppose $\mathbf{S}' \in \mathcal{GL}_{k+1}$ and $\mathbf{M}' = \mathbf{D}'\mathbf{P}' \in \mathcal{M}_{n'}$ satisfy $\mathbf{S}'\mathbf{G}'\mathbf{M}' = \mathbf{H}'$. By definition, $\mathbf{S}'$ has an inverse $\mathbf{S}'^{-1} \in \mathcal{GL}_{k+1}$, so we can rewrite

$$\mathbf{G}'\mathbf{M}' = \mathbf{S}'^{-1}\mathbf{H}'. \tag{3.1}$$

By Observation 2.1, $\mathbf{S}'^{-1}$ maps identical columns in $\mathbf{H}'$ to identical columns in $\mathbf{S}'^{-1}\mathbf{H}'$. To prove each part of the claim, we analyze the effect of $\mathbf{S}'^{-1}$ on each block $\mathbf{H}'_1, \mathbf{H}'_2, \mathbf{H}'_3$ of matrix $\mathbf{H}'$.

First we prove part (iii). By construction, the last $nm+1$ columns of $\mathbf{H}'$, contained in $\mathbf{H}'_3$ are identical to $\mathbf{e}_{k+1}$. By assumption, $\mathbf{H}$ does not contain a zero column, so no other column in $\mathbf{H}'$ outside of the block $\mathbf{H}'_3$ is equal to $\mathbf{e}_{k+1}$. Then each of the $nm+1$ columns in $\mathbf{S}'^{-1}\mathbf{H}'_3$, and no other column of $\mathbf{S}'^{-1}\mathbf{H}'$, is equal to $\mathbf{S}'^{-1}\mathbf{e}_{k+1}$.

By Equation (3.1), the last $nm+1$ columns of $\mathbf{G}'\mathbf{M}'$ must be identical and equal to $\mathbf{S}'^{-1}\mathbf{e}_{k+1}$. Since $\mathbf{M}' = \mathbf{D}'\mathbf{P}'$ for some diagonal matrix $\mathbf{D}'$ and permutation $\mathbf{P}'$, it maps each column in $\mathbf{G}'$ to a (possibly) scaled permutation of that column in $\mathbf{G}'\mathbf{M}'$. Thus, the matrix $\mathbf{M}'$ must scale and permute the columns of $\mathbf{G}'$ to produce $nm+1$ identical columns in the last block of $\mathbf{G}'\mathbf{M}'$. We claim that multiplying by $\mathbf{M}'$ cannot cause the number of copies of a column in $\mathbf{G}'$ to increase in $\mathbf{G}'\mathbf{M}'$. In particular, no two columns from different blocks of $\mathbf{G}'$ can be scaled to produce identical columns in $\mathbf{G}'\mathbf{M}'$. By construction, the last row of $\mathbf{G}'$ ensures that no column of $\mathbf{G}'_1$ can be scaled to produce a column of $\mathbf{G}'_2$, so $\mathbf{M}'$ cannot map columns from $\mathbf{G}'_1$ and $\mathbf{G}'_2$ to identical columns in $\mathbf{G}'\mathbf{M}'$. Additionally, $\mathbf{M}'$ maps every $\mathbf{e}_{k+1}$ column in $\mathbf{G}'_3$ to a scaling of $\mathbf{e}_{k+1}$ in $\mathbf{G}'\mathbf{M}'$, and since no column in $\mathbf{G}'_1$ or $\mathbf{G}'_2$ can be scaled to produce a multiple of $\mathbf{e}_{k+1}$, all scalings of $\mathbf{e}_{k+1}$ in $\mathbf{G}'\mathbf{M}'$ can only come from $\mathbf{G}'_3$. The only column in $\mathbf{G}'$ that appears $nm+1$ times is $\mathbf{e}_{k+1}$ from $\mathbf{G}'_3$. In this way, $\mathbf{M}'$ can only produce $nm+1$ identical columns in the last block of $\mathbf{G}'\mathbf{M}'$ by mapping from the $nm+1$ columns of $\mathbf{G}'_3$. Therefore, $\sigma_{\mathbf{P}'}(i)$ maps every column with index in $[n+mn+1, n']$ to a column with index in $[n+mn+1, n']$.

Next we show part (ii). By part (iii), any column with index $i \in [n+1, n+mn]$ is permuted to a column with index $\sigma_{\mathbf{P}'}(i) \leq n+mn$, so it is enough to show that $\sigma_{\mathbf{P}'}(i) \geq n+1$. Suppose for the sake of contradiction that there is a column with index $i \in [n+1, n+nm]$ in $\mathbf{G}'$ that is mapped to a column with index $\sigma_{\mathbf{P}'}(i) \in [1, n]$. By design, the last entry of the columns in $\mathbf{G}'_1$ is 1 and differs from the last entry 0 of the columns in $\mathbf{G}'_2$, so each column in $\mathbf{G}'_1$ appears strictly less than m times. Then, if $\mathbf{M}'$ permutes any number of columns from $\mathbf{G}'_1$ with the same number of columns in $\mathbf{G}'_2$, there will be some column in the second block of $\mathbf{G}'\mathbf{M}'$ that appears less than m times. But by Observation 2.1, every column in $\mathbf{S}'^{-1}\mathbf{H}'_2$ appears at least m times, and since the second block of $\mathbf{G}'\mathbf{P}'$ must be equal to $\mathbf{S}'^{-1}\mathbf{H}'_2$, this gives a contradiction.

Finally, part (i) follows immediately from parts (ii) and (iii). □

This result gives the following corollary.

Corollary 3.8. For any $\mathbf{G}', \mathbf{H}', \mathbf{S}', \mathbf{M}'$ as in Lemma 3.7, the monomial matrix $\mathbf{M}'$ is comprised of three block matrices $\mathbf{M}_1 \in \mathcal{M}_n$, $\mathbf{M}_2 \in \mathcal{M}_{nm}$, $\mathbf{M}_3 \in \mathcal{M}_{nm+1}$, such that $\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{M}_3$ only act on the first n, next nm, and last $nm+1$ columns of $\mathbf{G}'$, respectively.

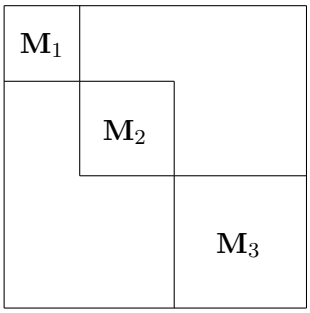

Figure 2: The structure of matrix $\mathbf{M}'$.

Now, we use this lemma to show that any invertible matrix for which $(\mathbf{G}', \mathbf{H}')$ is in LCE must contain only zeros in the last row and last column, except for the last entry.

Lemma 3.9. For any $\mathbf{G}', \mathbf{H}', \mathbf{S}', \mathbf{M}'$ as in Lemma 3.7, the change-of-basis matrix $\mathbf{S}'$ satisfies the following properties:

(i) The last column of $\mathbf{S}'$ contains zeros in the first k entries.

(ii) The last row of $\mathbf{S}'$ contains zeros in the first k entries.

(iii) The entry $\mathbf{S}'[k+1, k+1]$ is non-zero.

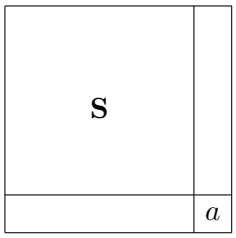

Figure 3: The structure of matrix $\mathbf{S}'$.

Proof. By definition, $\mathbf{S}'$ has an inverse $\mathbf{S}'^{-1} \in \mathcal{GL}_{k+1}$, so we can write $\mathbf{G}'\mathbf{M}' = \mathbf{S}'^{-1}\mathbf{H}'$. By Observation 2.1, $\mathbf{S}'^{-1}$ maps identical columns in $\mathbf{H}'$ to identical columns in $\mathbf{S}'^{-1}\mathbf{H}'$. To prove each part of the claim, we analyze the effect of $\mathbf{S}'^{-1}$ on each block $\mathbf{H}'_2$ and $\mathbf{H}'_3$ in the right-hand side of the equation.

By Corollary 3.8, we know that $\mathbf{M}'$ is comprised of three block matrices $\mathbf{M}_1 \in \mathcal{M}_n$, $\mathbf{M}_2 \in \mathcal{M}_{nm}$, and $\mathbf{M}_3 \in \mathcal{M}_{nm+1}$ that affect the first n, next nm, and last $nm+1$ columns of $\mathbf{G}'$, respectively. This allows us to write $\mathbf{G}'_2\mathbf{M}_2 = \mathbf{S}'^{-1}\mathbf{H}'_2$ and $\mathbf{G}'_3\mathbf{M}_3 = \mathbf{S}'^{-1}\mathbf{H}'_3$.

For part (i), consider the equation $\mathbf{G}'_3\mathbf{M}_3 = \mathbf{S}'^{-1}\mathbf{H}'_3$. By construction, all columns of $\mathbf{G}'_3$ and $\mathbf{H}'_3$ are identical and equal to $\mathbf{e}_{k+1}$. Since $\mathbf{M}_3$ permutes all columns of $\mathbf{G}'_3$ and multiplies them by a non-zero scalar, all columns of $\mathbf{G}'_3\mathbf{M}_3$ must be of the form $a \cdot \mathbf{e}_{k+1}$ for some non-zero a. By Observation 2.1, all columns of $\mathbf{S}'^{-1}\mathbf{H}'_3$ are identical. Then for any column $a \cdot \mathbf{e}_{k+1}$ of $\mathbf{G}'_3\mathbf{M}_3$, we have $\mathbf{S}'(a \cdot \mathbf{e}_{k+1}) = \mathbf{e}_{k+1}$. If any of the first k entries of the last column of $\mathbf{S}'$ is non-zero, then the corresponding entry in

$\mathbf{S}'(a \cdot \mathbf{e}_{k+1}) = \mathbf{e}_{k+1}$ must be non-zero, but this is not the case. Therefore, the last column of $\mathbf{S}'$ must contain zeros in the first k entries.

For part (ii), consider the equation $\mathbf{G}'_2\mathbf{M}_2 = \mathbf{S}'^{-1}\mathbf{H}'_2$. By construction, the last rows of $\mathbf{G}'_2$ and $\mathbf{H}'_2$ only contains zeros. By Lemma 3.7, no columns in $\mathbf{G}'_2\mathbf{M}_2$ could have been mapped from outside $\mathbf{G}'_2$, so the last row of $\mathbf{G}'_2\mathbf{M}_2$ must only contain zeros. Since $\mathbf{G}$ is a submatrix of $\mathbf{G}'_2$ with full row rank k, the first k rows of $\mathbf{G}'_2$, denoted by $\widehat{\mathbf{G}}$ in Construction 3.5, form a submatrix of rank k. Then $\widehat{\mathbf{G}}$ also has column rank k, and because the last row of $\mathbf{G}'_2$ only contains zeros, $\mathbf{G}'_2$ must contain k linearly independent columns. By Corollary 3.8, we know that $\mathbf{G}'_2\mathbf{M}_2$ is entirely comprised of columns of $\mathbf{G}'_2$ multiplied by a non-zero scalar, so $\mathbf{G}'_2\mathbf{M}_2$ has column rank k. Then the first k rows of $\mathbf{G}'_2\mathbf{M}_2$ form a submatrix of rank k. By definition, the last row of $\mathbf{S}'$ contains the coefficients that specify the linear combination of rows of $\mathbf{G}'_2\mathbf{M}_2$ which gives the last row of $\mathbf{H}_2$. But since the first k rows of $\mathbf{G}'_2\mathbf{M}_2$ are linearly independent, no linear combination of these can produce the all-zero last row of $\mathbf{H}_2$. Therefore, the first k entries in the last row of $\mathbf{S}'$ must be zero.

Part (iii) follows immediately from parts (i) and (ii) and the fact that $\mathbf{S}'$ is invertible. □

Finally, we show how Lemmas 3.7 and 3.9 combine to ensure the existence of an unsigned permutation for which $(\mathbf{G}, \mathbf{H})$ is in PCE.

Corollary 3.10. Let $\mathbf{M}' \in \mathcal{M}_n$ be a permutation with block submatrices $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$ as in Corollary 3.8. Then, $\mathbf{M}_1 = a \cdot \mathbf{P}$ for some permutation $\mathbf{P} \in \mathcal{P}_n$ and non-zero scalar a.

Proof. By Corollary 3.8, we can write $\mathbf{G}'_1\mathbf{M}_1 = \mathbf{S}'^{-1}\mathbf{H}'_1$. By Lemma 3.9 and Observation 2.1, the last row of $\mathbf{S}'^{-1}\mathbf{H}'_1$, and hence the last row of $\mathbf{G}'_1\mathbf{M}_1$, must be of the form $(a, a, \ldots, a)$ for some non-zero scalar a. By construction, the last row of $\mathbf{G}'_1$ contains only ones. Since $\mathbf{M}_1$ acts on $\mathbf{G}'_1$ by permuting and scaling the columns of $\mathbf{G}'_1$, and since the last row of $\mathbf{G}'_1\mathbf{M}_1$ contains identical entries a, we infer that $\mathbf{M}_1$ must multiply each column by the same scalar a. Therefore, $\mathbf{M}_1 = a \cdot \mathbf{P}$ for some unsigned permutation $\mathbf{P} \in \mathcal{P}_n$ and non-zero scalar a. □

Finally, we use the structure of the permutation and change-of-basis matrix for any LCE matrices $\mathbf{G}'$ and $\mathbf{H}'$ to show that the original matrices $\mathbf{G}$ and $\mathbf{H}$ must be in PCE.

Corollary 3.11. For any $\mathbf{G}, \mathbf{H} \in \mathbb{F}_q^{k \times n}$, if $(\mathbf{G}', \mathbf{H}')$ is in LCE, then $(\mathbf{G}, \mathbf{H})$ is in PCE.

Proof. If $(\mathbf{G}', \mathbf{H}')$ is in LCE, then there exists a monomial matrix $\mathbf{M}' \in \mathcal{M}_{n'}$ and an invertible matrix $\mathbf{S}' \in \mathcal{GL}_{k+1}$ such that $\mathbf{S}'\mathbf{G}'\mathbf{M}' = \mathbf{H}'$.

By Corollary 3.8, we know that $\mathbf{M}'$ is comprised of three block matrices $\mathbf{M}_1 \in \mathcal{M}_n$, $\mathbf{M}_2 \in \mathcal{M}_{nm}$, and $\mathbf{M}_3 \in \mathcal{M}_{nm+1}$ which act exclusively on the first n, next nm, and last $nm+1$ columns of $\mathbf{G}'$, respectively. By Corollary 3.10, we know that $\mathbf{M}_1 = a \cdot \mathbf{P}$ for some unsigned permutation $\mathbf{P} \in \mathcal{P}_n$ and non-zero scalar a.

By Lemma 3.9, the last row and last column of $\mathbf{S}'$ contain only zeros, except in the last entry. Let $\mathbf{S} \in \mathcal{GL}_k$ denote the top-left block submatrix of $\mathbf{S}'$ consisting of the intersection of the first k rows and k columns (see Figure 3). Since $\mathbf{S}'$ must have a non-zero determinant, this implies that $\mathbf{S}$ must be invertible. Then since a is non-zero, $a \cdot \mathbf{S}$ is also an invertible matrix.

By construction of $\mathbf{G}'$ and $\mathbf{H}'$, Lemma 3.9, and Corollary 3.8, we have the block matrix product $\mathbf{SGM}_1 = \mathbf{H}$. Then for the matrices $(a \cdot \mathbf{S})$ and $\mathbf{P}$ we obtain

$$(a \cdot \mathbf{S})\mathbf{GP} = \mathbf{S}\ \mathbf{G}\ (a \cdot \mathbf{P}) = \mathbf{SGM}_1 = \mathbf{H}.$$

Therefore, $(\mathbf{G}, \mathbf{H})$ is in PCE. □

For the reverse direction of Theorem 3.2, the proof is nearly identical to the one for LCE, but with the assumption that all non-zero scalars are restricted to the set of signs $\{-1, +1\}$.

4 Conclusion

Our results imply that the CE variants described above are nearly computationally equivalent, from the perspective of polynomial-time algorithms. In particular, we have shown that LCE and SPCE are at least as hard as PCE. So, in order to study the hardness of the former two problems, it suffices to focus on the hardness of PCE. We now have more reductions among the CE variants, and as a consequence of [BW24], from these variants to LIP, yet it remains an open problem to reduce LIP to any CE problem. While there has been some small progress in this direction, namely Ducas and Gibbons' Turing reduction from LIP for Construction A lattices to SPCE for codes with trivial hull [DG23], no other progress has yet been made.

Another interesting open problem is improving the runtime of the reduction from LCE to PCE given in [SS13a] from $\text{poly}(n, q)$ to $\text{poly}(n, \log q)$. This reduction requires time polynomial in the alphabet size q because the closure (as described in Section 1) increases the blocklength of the codes from n to $(q-1)n$. Any reduction running in $\log q$ time would either need to use a new type of closure that requires only a $\log q$ increase in blocklength, or a more efficient way to transform scalar multiplication operations into permutations.

5 Acknowledgments

The authors would like to thank Huck Bennett, Chris Peikert, and Yi Tang for helpful conversations during the course of this work. This research was partially supported by the National Science Foundation under Grant No. CCF-2236931.

References

[ABC+22] Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang. Classic McEliece. NIST PostQuantum Cryptography Standardization Project submission, 2022.

[Bab16] László Babai. Graph Isomorphism in Quasipolynomial Time [Extended Abstract]. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697. ACM, 2016.

[BBPS21] Alessandro Barenghi, Jean-Francois Biasse, Edoardo Persichetti, and Paolo Santini. LESS-FM: Fine-tuning Signatures from the Code Equivalence Problem. Cryptology ePrint Archive, Paper 2021/396, 2021.

[BBPS22] Alessandro Barenghi, Jean-Francois Biasse, Edoardo Persichetti, and Paolo Santini. On the Computational Hardness of the Code Equivalence Problem in Cryptography. Cryptology ePrint Archive, Paper 2022/967, 2022.

[BM23] Jean-François Biasse and Giacomo Micheli. A Search-to-Decision Reduction for the Permutation Code Equivalence Problem. In *IEEE International Symposium on Information Theory, ISIT 2023, Taipei, Taiwan, June 25-30, 2023*, pages 602–607. IEEE, 2023.

[BOS19] Magali Bardet, Ayoub Otmani, and Mohamed Saeed-Taha. Permutation Code Equivalence is not Harder than Graph Isomorphism when Hulls are Trivial. In *IEEE International Symposium on Information Theory, ISIT 2019, Paris, France, July 7-12, 2019*, pages 2464–2468. IEEE, 2019.

[BW24] Huck Bennett and Kaung Myat Htay Win. Relating Code Equivalence to Other Isomorphism Problems. Cryptology ePrint Archive, Paper 2024/782, 2024.

[CDG20] Alain Couvreur, Thomas Debris-Alazard, and Philippe Gaborit. On the Hardness of Code Equivalence Problems in Rank Metric. *CoRR*, abs/2011.04611, 2020.

[DG23] Léo Ducas and Shane Gibbons. Hull Attacks on the Lattice Isomorphism Problem. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *Public-Key Cryptography - PKC 2023 - 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7-10, 2023, Proceedings, Part I*, volume 13940 of *Lecture Notes in Computer Science*, pages 177–204. Springer, 2023.

[Leo82] Jeffrey S. Leon. Computing Automorphism Groups of Error-correcting Codes. *IEEE Trans. Inf. Theory*, 28(3):496–510, 1982.

[McE78] Robert J. McEliece. A Public-Key Cryptosystem based on Algebraic Coding Theory. DSN Progress Report, 1978.

[PR97] Erez Petrank and Ron M. Roth. Is code equivalence easy to decide? *IEEE Trans. Inf. Theory*, 43(5):1602–1604, 1997.

[Sen00] Nicolas Sendrier. Finding the Permutation between Equivalent Linear Codes: The Support Splitting Algorithm. *IEEE Trans. Inf. Theory*, 46(4):1193–1203, 2000.

[SS13a] Nicolas Sendrier and Dimitrios E. Simos. How easy is code equivalence over $\mathbb{F}_q$? In *International Workshop on Coding and Cryptography - WCC 2013*, Bergen, Norway, April 2013.

[SS13b] Nicolas Sendrier and Dimitris E. Simos. The Hardness of Code Equivalence over $\mathbb{F}_q$ and its Application to Code-based Cryptography. In Philippe Gaborit, editor, *Post-Quantum Cryptography*, pages 203–216, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.