Noname manuscript No.
(will be inserted by the editor)

How easy is code equivalence over $\mathbb{F}_q$?

Nicolas Sendrier · Dimitris E. Simos

Received: date / Accepted: date

Abstract The code equivalence problem is to decide whether two linear codes over $\mathbb{F}_q$ are equivalent, that is identical up to a linear isometry of the Hamming space. The support splitting algorithm [24] runs in polynomial time for all but a negligible proportion of all linear codes, and solves the latter problem by recovering the isometry when it is just a permutation of the code support. While for a binary alphabet isometries are exactly the permutations, this is not true for $q \geq 3$. We explore in this paper, a generalization of the support splitting algorithm where we aim to retrieve any isometry between equivalent codes. Our approach is twofold; first we reduce the problem of deciding the equivalence of linear codes to an instance of permutation equivalence. To this end, we introduce the notion of the closure of a code and give some of its properties. In the aftermath, we exhibit how this algorithm can be adapted for $q \in \{3, 4\}$, where its complexity is polynomial for almost all of its instances. Although the aforementioned reduction seems attractive, when $q \geq 5$ the closure reduces the instances of the code equivalence problem to exactly those few instances of permutation equivalence that were hard for the support splitting algorithm. Finally, we argue that for $q \geq 5$ the code equivalence problem might be hard for almost all instances.

Keywords Equivalence · Isometry · Closure of a Code · Linear Codes

Mathematics Subject Classification (2000) 94B05 · 05E20

Nicolas Sendrier
INRIA Paris-Rocquencourt
Project-Team SECRET
78153 Le Chesnay Cedex, France
E-mail: nicolas.sendrier@inria.fr

Dimitris E. Simos
INRIA Paris-Rocquencourt
Project-Team SECRET
78153 Le Chesnay Cedex, France
E-mail: dimitrios.simos@inria.fr

1 Introduction

The purpose of this work is to examine the worst-case and average-case hardness of the Code Equivalence problem. That is, given the generator matrices of two q-ary linear codes, how hard is it to decide whether or not these codes are identical up to an isometry of Hamming space?

The Permutation Code Equivalence problem is the restriction of the above problem when the isometries are limited to permutations of the code support[1]. Petrank and Roth proved [21] that the worst-case was not easier than for the Graph Isomorphism problem (unless the NP hierarchy collapses). On the other hand, the support splitting algorithm [24] solves the problem in time polynomial for all but an exponentially small proportion of the instances.

For a more general notion of code equivalence which includes all linear isometries, the situation seems to change drastically. Obviously, the Linear Code Equivalence problem is not easier in the worst-case than its Permutation Code Equivalence subproblem. In practice, the support splitting algorithm can be extended for $q \in \{3, 4\}$, and similarly solves all but an exponentially small proportion of the instances in polynomial time. However, for any fixed $q \geq 5$, the problem seems to be intractable for almost all instances.

The paper is structured as follows. In section 2, we present the different notions of code equivalence induced by isometries of Hamming space, while in section 3, we define in formal terms the Code Equivalence problem and mention the most significant contributions in terms of complexity and algorithms. In section 4, we illustrate a reduction of the Linear Code Equivalence problem as an instance of Permutation Code Equivalence, and its efficiency is analyzed in the following section. Finally, we elaborate on the hardness of the Code Equivalence problem and possible implications, in the concluding discussion.

2 Equivalence of linear codes

Code equivalence is a basic concept in coding theory. However, the equivalence of linear codes has met a few different definitions in the literature, often without motivation. We review the concept of what it means for codes to be "essentially different" by considering the metric Hamming space together with its isometries, which are the maps preserving the metric structure. This in turn will lead to a rigorous definition of equivalence of linear codes. In fact, we will call codes isometric if they are equivalent as subspaces of the Hamming space.

Let $\mathbb{F}_q$ be a finite field of cardinality $q = p^r$, where the prime number p is its characteristic, and r is a positive integer. As usual, a linear $[n, k]$ code C is a k-dimensional subspace of the finite vector space $\mathbb{F}_q^n$ and its elements are called codewords. We consider all vectors, as row vectors. Therefore, an element v of $\mathbb{F}_q^n$ is of the form $v := (v_1, \ldots, v_n)$. It can also be regarded as

[1] except for $q = 2$ the isometries are not limited to permutations.

the mapping υ from the set $\mathcal{I}_n = \{1, \ldots, n\}$ to $\mathbb{F}_q$ defined by $\upsilon(i) := \upsilon_i$. The Hamming distance (metric) on $\mathbb{F}_q^n$ is the following mapping,

$$d : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{N} : (x, y) \mapsto d(x, y) := \mid \{i \in \{1, 2, \ldots, n\} \mid x_i \neq y_i\} \mid .$$

The pair $(\mathbb{F}_q^n, d)$ is a metric space, called the Hamming space of dimension n over $\mathbb{F}_q$, denoted by $H(n, q)$. The Hamming weight $w(x)$ of a codeword $x \in C$ is simply the number of its non-zero coordinates, i.e. $w(x) := d(x, 0)$.

It is well-known due to a theorem of MacWilliams that any isometry between linear codes preserving the weight of the codewords induces an equivalence for codes [17]. Therefore, two codes C, C' are of the same quality if there exists a mapping $\iota : \mathbb{F}_q^n \mapsto \mathbb{F}_q^n$ with $\iota(C) = C'$ which preserves the Hamming distance, i.e. $d(\upsilon, \upsilon') = d(\iota(\upsilon), \iota(\upsilon'))$, for all $\upsilon, \upsilon' \in \mathbb{F}_q^n$. Mappings with the latter property are called the isometries of $H(n, q)$, and the two codes C and C' will be called isometric. Clearly, isometric codes have the same error-correction capabilities, and obvious permutations of the coordinates are isometries. We write $\mathcal{S}_n$ for the symmetric group acting on the set $\mathcal{I}_n$, equipped with the composition of permutations.

Definition 1 Two linear codes $C, C' \subseteq \mathbb{F}_q^n$ will be called permutationally equivalent[2], and will be denoted as $C \overset{\mathbf{PE}}{\sim} C'$, if there exists a permutation $\sigma \in \mathcal{S}_n$ that maps C onto C', i.e. $C' = \sigma(C) = \{\sigma(x) \mid x = (x_1, \ldots, x_n) \in C\}$ where $\sigma(x) = \sigma(x_1, \ldots, x_n) := (x_{\sigma^{-1}(1)}, \ldots, x_{\sigma^{-1}(n)})$.

Note also that the use of σ^{-1} in the index is consisted as we have $\sigma(\pi(C)) = \sigma \circ \pi(C)$. This can easily be seen by considering $x \in C$, and $\sigma, \pi \in \mathcal{S}_n$ such that $\sigma(\pi(x)) = \sigma((x_{\pi^{-1}(i)})_{i \in \mathcal{I}_n})$. Let $y_i = x_{\pi^{-1}(i)}$, $i \in \mathcal{I}_n$. Then $\sigma(\pi(x)) = \sigma((y_i)_{i \in \mathcal{I}_n}) = (y_{\sigma^{-1}(i)})_{i \in \mathcal{I}_n} = (x_{\pi^{-1}\sigma^{-1}(i)})_{i \in \mathcal{I}_n} = (x_{(\sigma\pi)^{-1}(i)})_{i \in \mathcal{I}_n} = \sigma \circ \pi(x)$.

Moreover, there is a particular subgroup of $\mathcal{S}_n$ that maps C onto itself, the permutation group of C defined as $\mathrm{PAut}(C) := \{C = \sigma(C) \mid \sigma \in \mathcal{S}_n\}$. $\mathrm{PAut}(C)$ always contains the identity permutation. If it does not contain any other element, we will say that it is trivial.

Recall, that we defined two codes to be isometric if there exists an isometry that maps one into another. Isometries that are linear[3], are called linear isometries. Therefore, we can obtain a more general notion of equivalence for codes induced by linear isometries of $\mathbb{F}_q$. Moreover, it can be shown that any linear isometry between two linear codes $C, C' \subseteq \mathbb{F}_q^n$ can always be extended to an isometry of $\mathbb{F}_q^n$ [3].

The group of all linear isometries of $H(n, q)$ corresponds to the semidirect product of $\mathbb{F}_q^{*n}$ and $\mathcal{S}_n$, $\mathbb{F}_q^{*n} \rtimes \mathcal{S}_n = \{(\upsilon; \pi) \mid \upsilon : \mathcal{I}_n \mapsto \mathbb{F}_q^*, \pi \in \mathcal{S}_n\}$, called the monomial group of degree n over $\mathbb{F}_q^*$, where the multiplication within this group is defined by

$$(\upsilon; \pi)(\upsilon'; \pi') = (\upsilon\upsilon'_\pi, \pi\pi') \quad \text{and} \quad (\upsilon\upsilon'_\pi)_i := \upsilon_i \upsilon'_{\pi^{-1}(i)} \tag{1}$$

[2] This definition can also met as permutationally isometric codes in the literature, see [3].

[3] For all $u, v \in \mathbb{F}_q^n$ we have $\iota(u + v) = \iota(u) + \iota(v)$ and $\iota(0) = 0$.

where $\mathbb{F}_q^*$ denotes the multiplicative group of $\mathbb{F}_q$. Hence, any linear isometry ι can be expressed as a pair of mappings $(\upsilon;\pi) \in \mathbb{F}_q^{*n} \rtimes \mathcal{S}_n$. Note that, some authors [3,8,10], describe this group as the wreath product $\mathbb{F}_q^* \wr_n \mathcal{S}_n$. The action of the latter group in an element of $\mathbb{F}_q^n$ is translated into an equivalence for linear codes.

Definition 2 Two linear codes $C, C' \subseteq \mathbb{F}_q^n$ will be called linearly or monomially equivalent, and will be denoted as $C \overset{\mathbf{LE}}{\sim} C'$, if there exists a linear isometry $\iota = (\upsilon;\sigma) \in \mathbb{F}_q^{*n} \rtimes \mathcal{S}_n$ that maps C onto C', i.e. $C' = (\upsilon;\sigma)(C) = \{(\upsilon;\sigma)(x) \mid (x_1,\ldots,x_n) \in C\}$ where $(\upsilon;\sigma)(x_1,\ldots,x_n) := (\upsilon_1 x_{\sigma^{-1}(1)},\ldots,\upsilon_n x_{\sigma^{-1}(n)})$.

If $q = p^r$ is not a prime, then the Frobenius automorphism $\tau : \mathbb{F}_q \to \mathbb{F}_q, x \mapsto x^p$ applied on each coordinate of $\mathbb{F}_q^n$ preserves the Hamming distance, too. Moreover, for $n \geq 3$, the isometries of $\mathbb{F}_q^n$ which map subspaces onto subspaces are exactly the semilinear mappings[4] of the form $(\upsilon;(\alpha,\pi))$, where $(\upsilon;\pi)$ is a linear isometry and α is a field automorphism, i.e. $\alpha \in \mathrm{Aut}(\mathbb{F}_q)$ (c.f. [3,14]). All these mappings form the group of semilinear isometries of $H(n,q)$ which is isomorphic to the semidirect product $\mathbb{F}_q^{*n} \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times \mathcal{S}_n)$, where the multiplication of elements is given by

$$(\upsilon;(\alpha,\pi))(\varphi;(\beta,\sigma)) := (\upsilon \cdot \alpha(\varphi_\pi);(\alpha\beta,\pi\sigma)) \tag{2}$$

Moreover, there is a description of $\mathbb{F}_q^{*n} \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times \mathcal{S}_n)$ as a generalized wreath product $\mathbb{F}_q^* \wr\wr_n (\mathrm{Aut}(\mathbb{F}_q) \times \mathcal{S}_n)$, see [3,9,14]. Clearly, the notion of semilinear isometry which can be expressed as a group action on the set of linear subspaces gives rise to the most general notion of equivalence for linear codes.

Definition 3 Two linear codes $C, C' \subseteq \mathbb{F}_q^n$ will be called semilinearly equivalent, and will be denoted as $C \overset{\mathbf{SLE}}{\sim} C'$, if there exists a semilinear isometry $(\upsilon;(\alpha,\sigma)) \in \mathbb{F}_q^{*n} \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times \mathcal{S}_n)$ that maps C onto C', i.e. $C' = (\upsilon;(\alpha,\sigma))(C) = \{(\upsilon;(\alpha,\sigma))(x) \mid (x_i)_{i\in\mathcal{I}_n} \in C\}$ where $(\upsilon;(\alpha,\sigma))(x_1,\ldots,x_n) = (\upsilon_1\alpha(x_{\sigma^{-1}(1)}),\ldots,\upsilon_n\alpha(x_{\sigma^{-1}(n)}))$.

Finally, we can define the monomial group of C as $\mathrm{MAut}(C) := \{C = (\upsilon;\sigma)(C) \mid (\upsilon;\sigma) \in \mathbb{F}_q^{*n} \rtimes \mathcal{S}_n\}$ and the automorphism group of C as $\mathrm{Aut}(C) := \{C = (\upsilon;(\alpha,\sigma))(C) \mid (\upsilon;(\alpha,\sigma)) \in \mathbb{F}_q^{*n} \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times \mathcal{S}_n)\}$ where their elements map each codeword of C to another codeword of C, under the respective actions of the involved groups. For more details, on automorphism groups of linear codes we refer to [13]. In addition, we remark the following:

1. When $\mathbb{F}_q = \mathbb{F}_2$ the group of linear isometries of $H(n,2)$ is isomorphic to $\mathcal{S}_n$, therefore all notions of equivalence are the same.
2. The group of semilinear isometries of $H(n,q)$ is the same as the group of linear isometries if and only if q is a prime (since $\mathrm{Aut}(\mathbb{F}_q)$ is trivial if and only if q is a prime). Therefore, semilinear equivalence reduces to linear equivalence for prime fields, and is different for all other cases.

[4] $\sigma : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is semilinear if there exists $\alpha \in \mathrm{Aut}(\mathbb{F}_q)$ such that for all $u, v \in \mathbb{F}_q^n$ and $k \in \mathbb{F}_q$ we have $\sigma(u+v) = \sigma(u) + \sigma(v)$ and $\sigma(ku) = \alpha(k)\sigma(u)$.

3 Previous work

For efficient computation of codes we represent them with generator matrices. A $k \times n$ matrix G over $\mathbb{F}_q$, is called a generator matrix for the $[n, k]$ linear code C if the rows of G form a basis for C, so that $C = \{xG \mid x \in \mathbb{F}_q^k\}$. In general, a linear code possess many different bases, and it is clear from linear algebra that the set of all generator matrices for C can be reached by $\{SG \mid S \in \mathrm{GL}_k(q)\}$, where $\mathrm{GL}_k(q)$ is the group of all $k \times k$ invertible matrices over $\mathbb{F}_q$.

For any $\sigma \in \mathcal{S}_n$ associate by $P_\sigma = [p_{i,j}]$ the $n \times n$ matrix such that $p_{i,j} = 1$ if $\sigma(i) = j$ and $p_{i,j} = 0$ otherwise, therefore P_σ is a permutation matrix. Note that, the action of $\sigma \in \mathcal{S}_n$ on $x \in \mathbb{F}_q^n$ agrees with the ordinary matrix multiplication. The permutation matrices form a subgroup of $M_n(q)$, the set of all $n \times n$ monomial matrices over $\mathbb{F}_q$, that is, matrices with exactly one nonzero entry per row and column from $\mathbb{F}_q$. If $M = [m_{i,j}] \in M_n(q)$, then $M = DP$, where P is a permutation matrix and $D = [d_{i,j}] = \mathrm{diag}(d_1, \ldots, d_n)$ is a diagonal matrix with $d_i = d_{i,i} = m_{i,j}$ if $m_{i,j} \neq 0$ and $d_{i,j} = 0$ if $i \neq j$. There is an isomorphism between diagonal matrices and $\mathbb{F}_q^{*n}$, therefore we associate $D_\upsilon = \mathrm{diag}(\upsilon_1, \ldots, \upsilon_n)$ for $\upsilon = (\upsilon_i)_{i \in \mathcal{I}_n} \in \mathbb{F}_q^{*n}$. Hence, we can map any linear isometry $(\upsilon; \sigma) \in \mathbb{F}_q^{*n} \rtimes \mathcal{S}_n$ to a monomial matrix $M_{(\upsilon;\sigma)} = D_\upsilon P_\sigma \in M_n(q)$, and this mapping is an isomorphism between $\mathbb{F}_q^{*n} \rtimes \mathcal{S}_n$ and $M_n(q)$. Therefore, we can express the equivalence between linear codes in terms of their generator matrices.

Problem 1 Given two $k \times n$ matrices G and G' over $\mathbb{F}_q$, whose rows span two $[n, k]$ linear codes C and C' over $\mathbb{F}_q$, does there exist $S \in \mathrm{GL}_k(q)$ and a monomial matrix $M_{(\upsilon;\sigma)} = D_\upsilon P_\sigma \in M_n(q)$ such that $G' = SGD_\upsilon P_\sigma$?

We will refer to the decidability of the previous problem, as the Linear Code Equivalence problem. One of our goals is to explore the hardness of this problem, therefore we deem necessary to briefly mention the most significant results in terms of complexity, for deciding it, and algorithms, for solving it.

3.1 Past complexity results

When the linear isometry $(\upsilon; \sigma)$ is just a permutation, i.e. D_υ is equal to I_n, we will call problem 1, as the Permutation Code Equivalence problem. The latter problem, was introduced in [21], who showed that if $\mathbb{F}_q = \mathbb{F}_2$ then it is harder than the Graph Isomorphism, there exists a polynomial time reduction, but not NP-complete unless P = NP. A different proof of this reduction is also given in [14]. Recently, the reduction of [21] was generalized in [12] over any field $\mathbb{F}_q$, hence Permutation Code Equivalence is harder than the Graph Isomorphism, for any field $\mathbb{F}_q$. The latter problem, has been extensively studied for decades, but until now there is no polynomial-time algorithm for solving all of its instances. Clearly, (Semi)-Linear Code

Equivalence for any $\mathbb{F}_q$ cannot be easier than the Graph Isomorphism, since it contains the Permutation Code Equivalence as a subproblem.

Last but not least, we would like to mention that the McEliece public-key cryptosystem [18] is related to the hardness of permutationally equivalent binary linear codes. Towards this direction, another important complexity result was shown in [6], that the Hidden Subgroup problem also reduces to Permutation Code Equivalence for any field $\mathbb{F}_q$.

3.2 Related algorithms for code equivalence

Due to its relation to Graph Isomorphism, some researchers have tried to solve the Permutation Code Equivalence problem by interpreting graph isomorphism algorithms to codes. This approach, was followed in [5] using the fact that ELC orbits of a bipartite graph correspond to equivalence classes of binary linear codes. Mapping codes to graphs and using the software Nauty by B. D. McKay has been used in [19], for binary, ternary and quaternary codes where the permutation, linear and semi-linear equivalence was considered, respectively. Moreover, an adaptation of Luks's algorithm for hypergraph isomorphism for solving the Permutation Code Equivalence over any $\mathbb{F}_q$ was presented in [2], whose complexity is simply-exponential in the length n of a code $C \subseteq \mathbb{F}_q^n$. Another approach using bipartite graphs for the Linear Code Equivalence problem over small fields was given in [4], where code equivalence is reduced to a decision problem regarding isomorphism of binary matrices. Note also, that in this work also the semilinear equivalence was considered for $\mathbb{F}_4$. Computation of canonical forms for generator matrices of linear codes for the Semilinear Code Equivalence problem[5] over $\mathbb{F}_q$, when q is small, by formulating the equivalence classes of codes as orbits of a group action from the left on the set of generator matrices was given in [7]. Finally, we would like to remark that, to the best of our knowledge there is no efficient algorithm for solving the Linear Code Equivalence problem for any field $\mathbb{F}_q$.

The support splitting algorithm can be used as an oracle to decide whether two binary codes are permutationally equivalent [24], as well as to retrieve the equivalence mapping. The main idea is to partition the support $\mathcal{I}_n$ of a code $C \subseteq \mathbb{F}_2^n$, into small sets that are fixed under operations of $\mathrm{PAut}(C)$. The algorithm employs the concept of invariants and signatures, defined below.

Let $\mathcal{L}_{n,k}$ denote the set of all linear codes of length n and dimension k, and let $\mathcal{L} = \bigcup_{n,k>0} \mathcal{L}_{n,k}$ be the set of all such codes.

Definition 4 An invariant $\mathcal{R}$ over a set E is defined to be a mapping $\mathcal{R} : \mathcal{L} \mapsto E$ such that any two permutation equivalent codes take the same value, i.e. if $C \overset{\mathbf{PE}}{\sim} C' \Longrightarrow \mathcal{R}(C) = \mathcal{R}(C')$.

[5] Same as the Linear Code Equivalence problem, with the additional application of a field automorphism in every column of the generator matrix.

For instance, the Hamming weight enumerator is an invariant over the polynomials with integer coefficients. Applying an invariant, for instance the weight enumerator, may help us to decide whether two codes are equivalent or not.

Definition 5 A signature S over a set F maps a code $C \subseteq \mathbb{F}_q^n$ and an element $i \in \mathcal{I}_n$ into an element of F and is such that for all $\sigma \in \mathcal{S}_n$, $S(C,i) = S(\sigma(C), \sigma(i))$. Moreover, S is called discriminant for C if there exist $i, j \in \mathcal{I}_n$ such that $S(C,i) \neq S(C,j)$ and fully discriminant if this holds $\forall\ i, j \in \mathcal{I}_n$.

If S is fully discriminant for C, and $C' = \sigma(C)$ for $\sigma \in \mathcal{S}_n$, we are able to retrieve σ. The support splitting algorithm ($\mathcal{SSA}$) takes as an argument a generator matrix G for a code C and returns a labeled partition $\Pi = \{(\Pi_j, j)\}_{j \in \mathcal{I}_n}$ of the code support. For any two linear codes C and C' with generator matrices G and G', let $\mathcal{SSA}(G) = \{(\Pi_j, j)\}_{j \in \mathcal{I}_n}$ and $\mathcal{SSA}(G') = \{(\Pi'_j, j)\}_{j \in \mathcal{I}_n}$. The fundamental property of $\mathcal{SSA}$ is that if

$$C' = \sigma(C) \quad \Longrightarrow \quad \forall j \in \mathcal{I}_n, \quad \Pi'_j = \sigma(\Pi_j) \tag{3}$$

and implies in particular that the output of $\mathcal{SSA}$ is independent of the choice of G. The converse of relation (3) is not necessarily true, but satisfied in practice under the assumption that the cells of the output of $\mathcal{SSA}$ achieve the orbits of the elements of the code support w.r.t. the action of $\mathrm{PAut}(C)$ and constitute its finest obtainable partition [16,24]. The main difficulty of the algorithm, is to obtain a fully discriminant signature, for as many codes as possible. In [24] it was shown that such a signature, can be built from the weight enumerator of the hull of a code C, denoted by $\mathcal{H}(C)$, and defined as the intersection of the code with its dual, $\mathcal{H}(C) = C \cap C^{\perp}$ [1], because the hull commutes with permutations[6], $\mathcal{H}(\sigma(C)) = \sigma(\mathcal{H}(C))$, and therefore it is an invariant for permutation equivalence. The (heuristic) complexity of $\mathcal{SSA}$ for an $[n,k]$ code C is $\mathcal{O}(n^3 + 2^h n^2 \log n)$ where h is the dimension of the hull [20,24]. In practice, for random codes, the hull has a small dimension with overwhelming probability [23] and the dominant cost for the average case is $\mathcal{O}(n^3)$. Note that, the worst case occurs when the hull dimension is maximal; weakly self-dual codes ($C \subset C^{\perp}$) are equal to their hulls. Then the algorithm becomes intractable with a complexity equal to $\mathcal{O}(2^k n^2 \log n)$.

4 Reduction of linear code equivalence to permutation code equivalence

Hence, we have in our disposal an algorithm, the support splitting algorithm, that solves the permutation equivalence in (almost) polynomial time. Therefore, it is natural to investigate a reduction of the Linear Code Equivalence problem as an instance of Permutation Code Equivalence. To this end, we introduce the *closure* of a linear code. We mention, that a similar approach was given in [25].

[6] No such property exists in general for linear codes when (semi)-linear equivalence is considered, see also lemma 1.

Definition 6 Let $\mathbb{F}_q = \{a_0, a_1, \ldots, a_{q-1}\}$, with $a_0 = 0$, and a linear code $C \subseteq \mathbb{F}_q^n$. Define $\mathcal{I}_{q-1}^{(n)}$ as the cartesian product of $\mathcal{I}_{q-1} \times \mathcal{I}_n$. The closure $\widetilde{C}$ of the code C is a code of length $(q-1)n$ over $\mathbb{F}_q$ where,

$$\widetilde{C} = \{(a_k x_i)_{(k,i)\in\mathcal{I}_{q-1}^{(n)}} \mid (x_i)_{i\in\mathcal{I}_n} \in C\}.$$

Clearly, we see that every coordinate of the closure $\widetilde{C}$, corresponds to a coordinate position of a codeword of C multiplied by a nonzero element of $\mathbb{F}_q$. Since, the index $(k,i) \in \mathcal{I}_{q-1}^{(n)}$ of a position of a codeword of the closure means that $k \in \mathcal{I}_{q-1}$ and $i \in \mathcal{I}_n$, we have taken into account every possible multiplication of x_i with nonzero elements of $\mathbb{F}_q$, and it is easy for someone to show[7] the following:

Theorem 1 *Let $C, C' \subseteq \mathbb{F}_q^n$. If C and C' are linearly equivalent, i.e. $C \overset{\mathbf{LE}}{\sim} C'$ then $\widetilde{C}$ and $\widetilde{C'}$ are permutationally equivalent, i.e. $\widetilde{C} \overset{\mathbf{PE}}{\sim} \widetilde{C'}$.*

Theorem 1 is of great importance, because it realizes a reduction from the Linear Code Equivalence problem to the Permutation Code Equivalence problem. Thus, we are able to decide if the codes C and C' are linearly equivalent by checking their closures for permutation equivalence. Moreover, if the closures are permutation equivalent there might be an algorithmic procedure that will allow us to recover the initial isometry between C and C'. However, as we shall see shortly after, the closure reduces an instance of the Linear Code Equivalence problem to exactly those instances that were hard for the support splitting algorithm for tackling the Permutation Code Equivalence problem over $\mathbb{F}_q$, $q \geq 5$.

We would also like to mention that this representation of the closure is not unique. In particular, it depends on a lexigographical ordering of $\mathbb{F}_q^*$.

For example, the ordering $(a_1, 1) < \ldots < (a_1, n) < (a_2, 1) < \ldots (a_2, n) < \ldots < (a_{q-1}, 1) < \ldots < (a_{q-1}, n)$ gives a total order for $\mathbb{F}_q^n$, and gives rise to the following closure,

$$\widetilde{C} = \{(a_1 x_1, \ldots, a_1 x_n, \ldots, a_{q-1} x_1, \ldots, a_{q-1} x_n) \mid (x_1, \ldots, x_n) \in C\}.$$

Note that, such an ordering can always be induced by a permutation of the symmetric group $\mathcal{S}_{\mathbb{F}_q^*}$ acting on $\mathbb{F}_q$ defined as $\mathcal{S}_{\mathbb{F}_q^*} := \{\rho \mid \rho : \mathbb{F}_q \to \mathbb{F}_q, \rho \text{ is a bijection and } \rho(0) = 0\}$.

Moreover, it is natural to ask which permutations can appear as permutations of the closures since $\mathcal{SSA}$ was designed exactly to retrieve the permutation between equivalent codes. If we assume that we are given a primitive element p of $\mathbb{F}_q$, it is well-known that all of its permissible powers generate the multiplicative group of $\mathbb{F}_q = \{p, p^2, \ldots, p^{q-2}, p^{q-1} = 1\}$. Then an ordering according to a cyclic shift of a power of p will produce a unique closure for

[7] The detailed proof of this theorem and all subsequent results will appear in an extended version of this paper.

the code C (consider the row echelon form on two generator matrices of the closures produced by such orderings).

Since, such a closure can always be reached by a composition of permutations of $\mathcal{S}_{\mathbb{F}_q^*}$, we define a canonical form for the closure as follows,

$$\widetilde{C}_{\mathbf{can}} = \{(x_1, px_1 \ldots, p^{q-2}x_1, \ldots, x_n, px_n \ldots, p^{q-2}x_n) \mid (x_1, \ldots, x_n) \in C\}.$$

If we consider the cyclic group $\mathcal{C}_{q-1}$ of order $q-1$ there is a natural isomorphism between $\mathbb{F}_q^{*n} \rtimes \mathcal{S}_n$ and $\mathcal{C}_{q-1} \wr_n \mathcal{S}_n$, the semidirect product of n copies of $\mathcal{C}_{q-1}$ and $\mathcal{S}_n$, called also the generalized symmetric group and denoted by $\mathcal{S}(q-1, n)$. Its order is $(q-1)^n n!$ and its elements are exactly those permutations that can appear as permutations of permutationally equivalent closures. This reasoning is sufficient for one to show that for $C \subseteq \mathbb{F}_q^n$ we have $\mathrm{MAut}(C)$ to be isomorphic to $\mathrm{PAut}(\widetilde{C})$. In particular, $\mathrm{MAut}(C) = \mathrm{PAut}(\widetilde{C}) \cap \mathcal{S}_{\mathbb{F}_q^*}^n$.

Moreover, it further implies tha the converse of theorem 1 also holds, and by involving the canonical forms of the closures as an intermediate step, after a non-trivial proof, we can show the following relation for equivalent codes and their closures.

Theorem 2 *Let $C, C' \subseteq \mathbb{F}_q^n$. Then C and C' are linearly equivalent, i.e. $C \overset{\mathbf{LE}}{\sim} C'$, if and only if $\widetilde{C}$ and $\widetilde{C'}$ are permutationally equivalent, i.e. $\widetilde{C} \overset{\mathbf{PE}}{\sim} \widetilde{C'}$.*

5 Efficiency of the Reduction

The $\mathcal{SSA}$ used as an invariant the hull $\mathcal{H}(C)$ of a code. In order to explore possible extensions of $\mathcal{SSA}$ we have to determine the quality of the hull of the closure $\mathcal{H}(\widetilde{C}) = \widetilde{C} \cap \widetilde{C}^{\perp}$, where the dual of the closure is defined according to some inner product. We are interested in two families of linear codes over $\mathbb{F}_q$ defined by the Euclidean and Hermitian inner product, respectively:

- (q^{E}) Linear codes over $\mathbb{F}_q$ with $\langle x, y\rangle_{\mathrm{E}} = \sum_{i=1}^n \langle x_i, y_i\rangle_{\mathrm{E}} = \sum_{i=1}^n x_i y_i = x_1 y_1 + \ldots + x_n y_n \in \mathbb{F}_q$. If q is a square, family q^{H} (below) is generally preferred to q^{E}.
- (q^{H}) Linear codes over $\mathbb{F}_q$, where q is an even power of an arbitrary prime ω, with $\overline{x} = x^{\sqrt{q}}$ for $x \in \mathbb{F}_q$ (c.f. [22]) and equipped with $\langle x, y\rangle_{\mathrm{H}} = \sum_{i=1}^n \langle x_i, y_i\rangle_{\mathrm{H}} = \sum_{i=1}^n x_i \overline{y}_i = x_1 \overline{y}_1 + \ldots + x_n \overline{y}_n \in \mathbb{F}_q$. Note that, for $x, y \in \mathbb{F}_q$,
$$(x+y)^{\sqrt{q}} = x^{\sqrt{q}} + y^{\sqrt{q}}, \quad x^q = x.$$

Now, consider two codewords $\widetilde{x}, \widetilde{y}$ of the closure $\widetilde{C}$ of $C \subseteq \mathbb{F}_q^n$. Then their inner product is given by $\langle \widetilde{x}, \widetilde{y}\rangle = \left(\sum_{i=1}^{q-1} a_i \overline{a}_i\right) \langle x, y\rangle$ where $\mathbb{F}_q = \{a_0 = 0, a_1, \ldots, a_{q-1}\}$. Using lemma 7.3. of [15] which states that $a_0, a_1, \ldots, a_{q-1}$ are distinct if and only if $\sum_{i=0}^{q-1} a_i^t = 0$ for $t = 0, 1, \ldots, q-2$ and $\sum_{i=0}^{q-1} a_i^t = -1$ for $t = q-1$, we can show that,

$$\langle \widetilde{x}, \widetilde{y} \rangle_{\mathrm{E}} = \begin{cases} 0 & \text{for} \quad q \geq 4 \\ -\langle x, y \rangle_{\mathrm{E}} & \text{for} \quad q = 3. \end{cases} \text{ and } \langle \widetilde{x}, \widetilde{y} \rangle_{\mathrm{H}} = \begin{cases} 0 & \text{for} \quad q > 4 \\ -\langle x, y \rangle_{\mathrm{H}} & \text{for} \quad q = 4. \end{cases}$$

This means, that the closure $\widetilde{C}$ is a weakly self-dual code for every $q \geq 5$, considering both Euclidean and Hermitian duals, which is exactly the hard instances of $\mathcal{SSA}$. Moreover, for $\mathbb{F}_3$ and $\mathbb{F}_4$ equipped with the Euclidean and Hermitian inner product, respectively, the distribution of the dimension of $\mathcal{H}(\widetilde{C})$ follows the distribution of the dimension $\mathcal{H}(C)$, since the closure has the same dimension as C, and will be on average a small constant, [23], except in the cases where C is also a weakly self-dual code.

It is worth mentioning that these are exactly the same cases where the hull of a code could be used as an invariant for (semi)-linear equivalence, because the duals of linear and semilinear codes remain equivalent with the same isometry of the original codes only in $\mathbb{F}_3$ and $\mathbb{F}_4$, due to the following relation (see [13,24]):

Lemma 1 *Let $C \subseteq \mathbb{F}_q^n$, and $(\upsilon; (\sigma, \alpha)) \in \mathbb{F}_q^{*n} \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times \mathcal{S}_n)$. Then*

(i) $(\upsilon; (\sigma, \alpha))(C)^\perp = (\upsilon^{-1}; (\sigma, \alpha))(C^\perp)$ *where* $C^\perp$ *is w.r.t.* $\langle , \rangle_{\mathrm{E}}$.
(ii) $(\upsilon; (\sigma, \alpha))(C)^\perp = (\overline{\upsilon}; (\sigma, \alpha))(C^\perp)$ *where* $C^\perp$ *is w.r.t.* $\langle , \rangle_{\mathrm{H}}$.

Then, a signature for an extension of $\mathcal{SSA}$ can be built from the weight enumerator of the $\mathcal{H}(\widetilde{C})$. The Linear Code Equivalence problem can be decided (and solved) in polyonomial time using $\mathcal{SSA}$ only in $\mathbb{F}_3$ and $\mathbb{F}_4$, as long as the hull of the given code is small (the worst-case being a weakly self-dual code). It does not seem possible to extend this result to larger alphabet. We conclude by posing the following conjecture.

Conjecture 1 For a given $q \geq 5$, the (Semi)-Linear Code Equivalence problem over $\mathbb{F}_q$ is hard for almost all instances.

Note that, there is a similar negative complexity result due to Dirk Vertigan [26]. The result is given for graphs, but, translated for codes, it states that evaluating the (homogeneous) weight enumerator polynomial of a linear code over $\mathbb{F}_q$ for $q \geq 5$ on any point of the complex unit circle is always difficult except for a constant number of trivial points. The evaluation of the weight enumerator in those points essentially provide the code cardinality. There is an additional point easy to evaluate for $q \in \{2, 3, 4\}$. The evaluation in this point essentially provides the cardinality of the hull of the code. For $q = 4$ the hull is defined according to the hermitian inner product. There is possibly more than just a coincidence here, but the connection with code equivalence is not obvious to establish. Doing so would certainly be enlightening.

6 Conclusion

In this paper, we explored the hardness of the Code Equivalence problem over $\mathbb{F}_q$. We showed that an extension of $\mathcal{SSA}$ for solving the latter problem when $q \in \{3, 4\}$ is possible, in (almost) polynomial time, however for $q \geq 5$ its complexity growth becomes exponential for all instances. Moreover, we conjectured that, for $q \geq 5$, Code Equivalence is hard for almost all instances. Our argument, is supported by some impossibility results on the Tutte polynomial of a graph which corresponds to the weight enumerator of a code. On the bright side, the negativity of our claim, might lead to some interesting features for applications. For example, in cryptography, zero-knowledge protocols have been designed in the past, based on the hardness of the Permutation Code Equivalence problem [11]. Moreover, the relation of the automorphism groups of the code and its closure might be of cryptographic interest. The context of the framework built in [6] suggests that codes with large automorphism groups resist quantum Fourier sampling as long as permutation equivalence is considered. It would thus be intriguing to investigate, if this result can also be extended for the linear and semilinear code equivalence.

References

1. Assmus, E.F.J., Key, J.D.: Designs and their Codes, *Cambridge Tracts in Mathematics*, vol. 103. Cambridge University Press (1992). Second printing with corrections, 1993
2. Babai, L., Codenotti, P., Grochow, J.A., Y.Qiao: Code equivalence and group isomorphism. In: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '11, pp. 1395–1408. SIAM (2011)
3. Betten, A., Braun, M., Fripertinger, H., Kerber, A., Kohnert, A., Wassermann, A.: Error-Correcting Linear Codes: Classification by Isometry and Applications, *Algorithms and Computation in Mathematics*, vol. 18. Springer, Berlin, Heidelberg (2006)
4. Bouyukliev, I.: About the code equivalence. Ser. Coding Theory Cryptol. **3**, 126–151 (2007)
5. Danielsen, L.E., Parker, M.G.: Edge local complementation and equivalence of binary linear codes. Des. Codes Cryptography **49**, 161–170 (2008)
6. Dinh, H., Moore, C., Russell, A.: Mceliece and niederreiter cryptosystems that resist quantum fourier sampling attacks. In: Proceedings of the 31st annual conference on Advances in cryptology, CRYPTO'11, pp. 761–779. Springer-Verlag, Berlin, Heidelberg (2011)
7. Feulner, T.: The automorphism groups of linear codes and canonical representatives of their semilinear isometry classes. Adv. Math. Commun. **3**, 363–383 (2009)
8. Fripertinger, H.: Enumeration of linear codes by applying methods from algebraic combinatorics. Grazer Math. Ber. **328**, 31–42 (1996)
9. Fripertinger, H.: Enumeration of the semilinear isometry classes of linear codes. Bayrether Mathematische Schriften **74**, 100–122 (2005)
10. Fripertinger, H., Kerber, A.: Isometry classes of indecomposable linear codes. In: Proceedings of the 11th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, AAECC-11, pp. 194–204. Springer-Verlag, London, UK (1995)
11. Girault, M.: A (non-practical) three-pass identification protocol using coding theory. In: J. Seberry, J. Pieprzyk (eds.) Advances in Cryptology AUSCRYPT '90, *Lecture Notes in Computer Science*, vol. 453, pp. 265–272. Springer Berlin Heidelberg (1990)
12. Grochow, J.A.: Matrix lie algebra isomorphism. Tech. Rep. TR11-168, Electronic Colloquium on Computational Complexity (2011). Also available as arXiv:1112.2012. To appear, IEEE Conference on Computational Complexity, 2012.

13. Huffman, W.C.: Codes and groups. In: V. Pless, W.C. Huffman (eds.) Handbook of Coding Theory, pp. 1345–1440. Elsevier, North-Holland, Amsterdam (1998)
14. Kaski, P., Östergård, P.R.J.: Classification Algorithms for Codes and Designs, *Algorithms and Computation in Mathematics*, vol. 15. Springer, Berlin, Heidelberg (2006)
15. Lidl, R., Niederreiter, H.: Finite Fields, *Encyclopedia of Mathematics and its Applications*, vol. 20, 2nd edn. Cambridge University Press (1997)
16. Loidreau, P., Sendrier, N.: Weak keys in McEliece public-key cryptosystem. IEEE Trans. Inform. Theory **47**, 1207–1212 (2001)
17. MacWilliams, F.J.: Error-correcting codes for multiple-level transmission. Bell. Syst. Tech. J. **40**, 281–308 (1961)
18. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. Tech. Rep. DSN Progress Report 42–44, California Institute of Technology, Jet Propulsion Laboratory, Pasadena, CA (1978)
19. Östergård, P.R.J.: Classifying subspaces of hamming spaces. Des. Codes Cryptography **27**, 297–305 (2002)
20. Overbeck, R., Sendrier, N.: Code-based cryptography. In: D. Bernstein, J. Buchmann, E. Dahmen (eds.) Post-Quantum Cryptography, pp. 95–145. Springer (2009)
21. Petrank, E., Roth, R.M.: Is code equivalence easy to decide? IEEE Trans. Inform. Theory **43**, 1602–1604 (1997)
22. Rains, E.M., Sloane, N.J.A.: Self-dual codes. In: V. Pless, W.C. Huffman (eds.) Handbook of Coding Theory, pp. 177–294. Elsevier, North-Holland, Amsterdam (1998)
23. Sendrier, N.: On the dimension of the hull. SIAM J. Discrete Math. **10**(2), 282–293 (1997)
24. Sendrier, N.: Finding the permutation between equivalent linear codes: The support splitting algorithm. IEEE Trans. Inform. Theory **26**, 1193–1203 (2000)
25. Skersys, G.: Calcul du groupe d'automorphisme des codes. détermination de l' equivalence des codes. Thèse de doctorat, Université de Limoges (1999)
26. Vertigan, D.: Bicycle dimension and special points of the Tutte polynomial. Journal of Combinatorial Theory, Series B **74**, 378–396 (1998)