Don't Use It Twice! Solving Relaxed Linear Code Equivalence Problems

Alessandro Budroni[1], Jesús-Javier Chi-Domínguez[1], Giuseppe D'Alconzo[2], Antonio J. Di Scala[2], and Mukul Kulkarni[1]

[1] Cryptography Research Center, Technology Innovation Institute, UAE
{alessandro.budroni,jesus.dominguez,mukul.kulkarni}@tii.ae
[2] Department of Mathematical Sciences, Polytechnic University of Turin, Italy
{giuseppe.dalconzo,antonio.discala}@polito.it

Abstract. The Linear Code Equivalence (LCE) Problem has received increased attention in recent years due to its applicability in constructing efficient digital signatures. Notably, the LESS signature scheme based on LCE is under consideration for the NIST post-quantum standardization process, along with the MEDS signature scheme that relies on an extension of LCE to the rank metric, namely Matrix Code Equivalence (MCE) Problem. Building upon these developments, a family of signatures with additional properties, including linkable ring, group, and threshold signatures, has been proposed. These novel constructions introduce relaxed versions of LCE (and MCE), wherein multiple samples share the same secret equivalence. Despite their significance, these variations have often lacked a thorough security analysis, being assumed to be as challenging as their original counterparts. Addressing this gap, our work delves into the sample complexity of LCE and MCE — precisely, the sufficient number of samples required for efficient recovery of the shared secret equivalence. Our findings reveal, for instance, that one shouldn't use the same secret twice in the LCE setting since this enables a polynomial time (and memory) algorithm to retrieve the secret. Consequently, our results unveil the insecurity of two advanced signatures based on variants of the LCE Problem.

Keywords: Algebraic Attack · Code Equivalence · Code-based Cryptography · Cryptanalysis · Post-quantum Cryptography

1 Introduction

There has been an increased interest in constructing new quantum-resistant digital signatures following the ongoing NIST post-quantum standardization process for additional digital signature schemes [24]. Moving beyond the proposals that appeared at the prior NIST post-quantum standardization process [23], the research community explored a wider spectrum of computational problems, conjectured to be hard, for building efficient signature schemes. Notably, two closely related problems, Linear Code Equivalence (LCE) and Matrix Code Equivalence

(MCE), served as the foundation for LESS [2] and MEDS [13], two efficient signature schemes submitted to the current NIST standardization process. Informally, we say that two linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ of length n and dimension k over a finite field $\mathbb{F}_q$ are equivalent if there exists a monomial matrix $\boldsymbol{Q} \in \mathbb{F}_q^{n\times n}$ such that $\mathcal{C}_2 = \mathcal{C}_1\boldsymbol{Q}$. LCE refers to the computational problem of finding $\boldsymbol{Q}$ given $\mathcal{C}_1$ and $\mathcal{C}_2$. When a permutation matrix $\boldsymbol{P}$ takes the place of the monomial matrix $\boldsymbol{Q}$, then the problem is referred to as Permutation Code Equivalence (PCE). On the other hand, MCE can be seen as a variant of LCE to matrix codes.

Multiple signature schemes have been proposed in the literature based on the one-way functions stemming from the computational hardness of equivalence problems. On the other hand, designing more advanced cryptographic schemes, such as linkable ring signatures, encryption schemes, key exchange mechanisms, etc., based on the hardness equivalence problems has been challenging without assuming additional features of the objects underlying these problems. One meaningful example comes from isogenies between elliptic curves, where more advanced protocols can be designed [1]. In general, one plausible reason for this designing difficulty is that while the equivalence problems can provide one-wayness, they may not provide stronger cryptographic properties such as unpredictability or pseudorandomness, which are necessary for constructing advanced primitives. In fact, recently [17] showed this to be the case for LCE, while [11] proved a similar result for a variant of the Lattice Isomorphism Problem. Both these works used the group action framework to argue that their respective underlying problems do not provide unpredictability or pseudorandomness.

Some researchers have also used the group actions framework to construct signature schemes with advanced functionalities from these equivalence problems [8,9,6,28]. However, to achieve such functionalities, they also introduced new relaxed versions of LCE and MCE, such as the Inverse Linear Code Equivalence (ILCE) problem [4], the Inverse Matrix Code Equivalence (IMCE) Problem [14], and the 2-Linear Code Equivalence Problem (2-LCE) [6].[3] These variants are often conjectured to exhibit a level of difficulty comparable to their original counterparts but without formal proof or comprehensive cryptanalytic investigation. In the case of ILCE, the authors took the conservative choice of not relying on such a problem at the cost of dropping the linkability property from their ring signature scheme, posing the open question of whether ILCE can be considered secure or not.

In this work, we fill this gap by studying these variants from an algebraic point of view and highlighting the scenarios among the proposed ones, for which, on the contrary, these do not provide an adequate level of security at all. Specifically, we show that, for the proposed parameters setting [4,6], both ILCE and 2-LCE can be solved in polynomial time and memory by exploiting the linear nature of these problems. As a consequence, the schemes that rely on the hardness of ILCE and 2-LCE are not secure.

[3] The authors in [6] gave a more general problem definition in terms of group actions, namely 2-Group Action Inverse Problem (2-GAIP). Here, we refer by 2-LCE to the 2-GAIP from [6] instantiated with LCE.

1.1 Contribution and organization

On the hardness of LCE given t samples (t-LCE)

After covering the necessary background in Section 2, we delves in Section 3 into the impact of having more than one LCE sample sharing the same secret monomial matrix $\boldsymbol{Q}$. We explore how this affects the hardness of recovering $\boldsymbol{Q}$ specifically, and without loss of generality, for when the generator matrix codes are in systematic form. In particular, we derive a concrete bound on the number of necessary samples allowing an efficient recovery of the secret with high probability. Additionally, a parallel analysis is conducted also for MCE. We report our result in the following lemma.

Lemma (Informal). *For (n,k)-linear codes over finite field $\mathbb{F}_q$, the secret monomial matrix $\boldsymbol{Q}$ can be recovered from $\left\lceil \frac{n^2}{k(n-k)} \right\rceil$ samples of* LCE *sharing the same $\boldsymbol{Q}$ in polynomial time, with overwhelming probability.*

The above result improves the work by D'Alconzo and Di Scala [17]. Their result provided a bound of nk samples, applicable solely to codes not in systematic form. In contrast, our result removes this limitation, extending to codes represented in systematic form as well. As a consequence, we derive the following corollary.

Corollary (Informal). *For $(2k,k)$-linear codes, the secret monomial matrix $\boldsymbol{Q}$ can be recovered from only 4 samples of* LCE *sharing the same $\boldsymbol{Q}$, with overwhelming probability.*

For the case of ILCE, we prove the following lemma.

Lemma (Informal). *For (n,k)-linear codes over finite field $\mathbb{F}_q$, the secret monomial matrix $\boldsymbol{Q}$ can be recovered from $\left\lceil \frac{n^2}{2k(n-k)} \right\rceil$ samples of* ILCE *using the same $\boldsymbol{Q}$ in polynomial time, with overwhelming probability.*

Solving 2-LCE and ILCE for $(2k,k)$-linear codes

In Section 4 we introduce a new polynomial-time algorithm for solving 2-LCE and ILCE when $k = n/2$. This algorithm is inspired by Saeed's for solving PCE [30] and, unlike the results mentioned above, it exploits the structure of the secret monomial matrix to recover. Our algorithm solves 2-LCE directly without converting it to a PCE instance.

Theorem (Informal). *For $k = \frac{n}{2}$, there exists a polynomial time algorithm that solves* 2-LCE *and* ILCE *in polynomial time and with overwhelming probability.*

As a consequence,

a. We show that the threshold signature scheme based on LCE and proposed in [6] is not secure since its distributed key generation relies on the conjectured hardness of 2-LCE,

b. We resolve the open question raised by Barenghi et al. [4] asking whether ILCE can be used to construct secure linkable ring signatures, and the answer is negative.

To highlight the diminished security of 2-LCE and ILCE, we report in Table 1 a comparison of our complexity estimations of these two problems against the one of LCE for the parameters sets proposed in [2].

n	k	q	LCE	2-LCE & ILCE
252	126	127	128	61
400	200	127	192	66
548	274	127	256	70

Table 1: The column corresponding to LCE is according to the security analysis from [2]. The column corresponding to 2-LCE & ILCE concerns the complexity of Algorithm 1 (detailed in Theorem 1) with $\omega = \log_2(7)$. The presented numbers are given in logarithm base two.

In addition, to support our findings, we report in Section 5 the results of extensive experiments performed on linear codes of length up to $n = 128$ and dimension $k = 64$. In accordance to the theory, our results show that, for $k = n/2$ two LCE samples sharing the same secret are enough to recover the secret equivalence in polynomial time and memory. These contributions collectively enhance the understanding of LCE (and MCE) and their aforementioned variants, offering a much clearer understanding of their sample complexity and preventing researchers from building new insecure cryptographic schemes.

1.2 Technical Overview

We begin by defining LCE in terms the number of samples available to the adversary. Let $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$ be a distribution which samples random (n,k)-linear code over finite field $\mathbb{F}_q$ for some prime q and for a fixed secret monomial matrix $\boldsymbol{Q} \in \mathsf{Mono}_n(\mathbb{F}_q)$, where $\mathsf{Mono}_n(\mathbb{F}_q)$ is the set of all $n \times n$ monomial matrices over $\mathbb{F}_q$. Let $\boldsymbol{G}$ be a generator matrix of the sampled (n,k)-linear code, and $\boldsymbol{G}'$ be a generator matrix of an equivalent code whose equivalence is determined by the secret $\boldsymbol{Q}$ as follows

$$\boldsymbol{G}' = \mathsf{SF}(\boldsymbol{G}\boldsymbol{Q}),$$

where $\mathsf{SF}()$ denote the systematic form operation. The distribution $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$ outputs $(\boldsymbol{G}, \boldsymbol{G}')$. Here (and throughout this paper) we consider that both $\boldsymbol{G}$ and $\boldsymbol{G}'$ are in the systematic form, that is $\boldsymbol{G} := (\,\boldsymbol{I}_k | \boldsymbol{M}\,)$ where $\boldsymbol{I}_k$ is $k \times k$ identity matrix over $\mathbb{F}_q$ and $\boldsymbol{M} \in \mathbb{F}_q^{k\times(n-k)}$. Similarly, $\boldsymbol{G}' := (\,\boldsymbol{I}_k | \boldsymbol{M}'\,)$. Let t-$\mathsf{LCE}_{n,k,q}$ be the problem of recovering $\boldsymbol{Q}$ from t samples from $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$ for a fixed secret

$\boldsymbol{Q} \in \mathsf{Mono}_n(\mathbb{F}_q)$ chosen uniform randomly. Note that, 1-$\mathsf{LCE}_{n,k,q}$ corresponds to the standard LCE problem.

Recall that if $\boldsymbol{G}$ is a generator matrix for an (n,k)-linear code and $\boldsymbol{H'}$ is a parity check matrix of an equivalent code with generator matrix $\boldsymbol{G'} = \mathsf{SF}(\boldsymbol{GQ})$, then $\boldsymbol{GQH'}^\top = \boldsymbol{0}$. Since $\boldsymbol{G}, \boldsymbol{G'}$ are in systematic form, as observed by Saeed in [30], we can create a linear system of equations given by

$$\left[\left(\boldsymbol{I}_k \middle| \boldsymbol{M} \right) \otimes \left(-\boldsymbol{M'}^\top \middle| \boldsymbol{I}_{n-k} \right) \right] \mathsf{vec}(\boldsymbol{Q}) = \boldsymbol{0}. \tag{1}$$

where $\otimes$ denotes the Kronecker product and $\mathsf{vec}(\boldsymbol{Q})$ denotes a column vector of length n^2 created by unrolling the entries of $\boldsymbol{Q}$.

Solving LCE with multiple instances. Note that the linear system of equations obtained from a single LCE instance in Equation (1) is underdetermined since we have n^2 unknown variables and only $k(n-k)$ equations given by the rows of the coefficient matrix in Equation (1). However, when we obtain t samples from $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$ we can combine them into a larger system using the additional equations, such a system can be written as

$$\boldsymbol{A} \coloneqq \left[\boldsymbol{A}_1{}^\top \ \boldsymbol{A}_2{}^\top \cdots \boldsymbol{A}_t{}^\top \right]^\top,$$

where each block $\boldsymbol{A}_i$ is constructed from an individual sample as per Equation (1). This is useful if the additional equations given by the extra samples are not linearly dependent on the previous equations. Since the equations correspond to the rows of the matrix, from here onwards we will consider the dependencies between these. Let $\boldsymbol{A}_i := \left(\boldsymbol{I}_k \middle| \boldsymbol{M} \right) \otimes \left(-\boldsymbol{M'}^\top \middle| \boldsymbol{I}_{n-k} \right)$ and $\boldsymbol{A}_j := \left(\boldsymbol{I}_k \middle| \boldsymbol{N} \right) \otimes \left(-\boldsymbol{N'}^\top \middle| \boldsymbol{I}_{n-k} \right)$ be two block matrices corresponding to the i^{th} and j^{th} sample respectively. We prove that the rows of $\boldsymbol{A}_i$ and $\boldsymbol{A}_j$ are linearly dependent if and only if the following conditions are simultaneously satisfied:

- the i'^{th} column of the matrices $\boldsymbol{M'}$ and $\boldsymbol{N'}$ are equal and,
- the j'^{th} row of the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ are equal,

for some $1 \leq i' \leq n-k$ and for some $1 \leq j' \leq k$. Assuming the samples from $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$ to be chosen randomly and independently of each other, we can show that the rows of matrix $\boldsymbol{A}$ are linearly independent with high probability. At this point, by setting $t = \left\lceil \frac{n^2}{k(n-k)} \right\rceil$ we obtain an overdetermined linear system of equations when $k \neq \frac{n}{2}$, and we recover the secret $\boldsymbol{Q}$ by solving this linear system. If $k = \frac{n}{2}$ and $t = \left\lceil \frac{n^2}{k(n-k)} \right\rceil$, the linear system defined by $\boldsymbol{A}$ is still underdetermined since we get only $n^2 - \frac{n}{2}$ linearly independent equations in this case. However, we can compute the generator set of the kernel of this system using Gaussian elimination. The cardinality of this set is less than or equal to $\frac{n}{2}$, and we show that (a multiple of) $\mathsf{vec}(\boldsymbol{Q})$ is in this set with high probability. Therefore, after computing the kernel we can simply search for $\boldsymbol{Q}$ efficiently. The overall complexity of recovering $\boldsymbol{Q}$ from t samples is $O(n^{2\omega})$, where ω is the complexity of performing matrix multiplication over $\mathbb{F}_q$.

Solving ILCE with half the samples. We also study the problem of inverse linear code equivalence (t-$\mathsf{ILCE}_{n,k,q}$), which is closely related to t-$\mathsf{LCE}_{n,k,q}$ problem. In t-$\mathsf{ILCE}_{n,k,q}$, the adversary gets t samples from the distribution $\widehat{\mathcal{L}}_{n,k,q,\boldsymbol{Q}}$, which is defined similarly to $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$ with the addition that $\widehat{\mathcal{L}}_{n,k,q,\boldsymbol{Q}}$ returns a tuple of three matrices $(\boldsymbol{G}, \boldsymbol{G}', \boldsymbol{G}'')$, where $\boldsymbol{G}, \boldsymbol{G}'$ are computed as in $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$ and $\boldsymbol{G}'' = \mathsf{SF}(\boldsymbol{G}\boldsymbol{Q}^{-1})$. The main idea in this case is that we can now create a system of equations as the one given by Equation (1) with $2k(n-k)$ rows instead of $k(n-k)$ from a single sample. Intuitively, this should result in halving the number of t-$\mathsf{ILCE}_{n,k,q}$ samples required to recover the secret $\boldsymbol{Q}$ when compared to t-$\mathsf{LCE}_{n,k,q}$. We can therefore retrieve $\boldsymbol{Q}$ as in the case of t-$\mathsf{LCE}_{n,k,q}$ by solving the system with only $t = \left\lceil \frac{n^2}{2k(n-k)} \right\rceil$ samples.

Extending to MCE and IMCE. Let $\boldsymbol{G}$ and $\boldsymbol{G}'$ be generators of two $(m \times r, k)$ matrix codes such that there exist $\boldsymbol{A} \in \mathsf{GL}_m(\mathbb{F}_q)$ and $\boldsymbol{B} \in \mathsf{GL}_r(\mathbb{F}_q)$ satisfying $\boldsymbol{G}' = \mathsf{SF}(\boldsymbol{G}(\boldsymbol{A}^\top \otimes \boldsymbol{B}))$. The Matrix Code Equivalence (MCE) problem is to find $\boldsymbol{A}$ and $\boldsymbol{B}$ given $\boldsymbol{G}$ and $\boldsymbol{G}'$. The IMCE problem is defined analogously as the ILCE for matrix codes. We extend the results obtained for LCE to the case of matrix codes. In particular, we have that MCE can be solved efficiently when $\left\lfloor \frac{(mr)^2}{k(mr-k)} \right\rfloor + 1$ samples are available. Similarly, $\left\lfloor \frac{(mr)^2}{2k(mr-k)} \right\rfloor + 1$ samples are enough to solve IMCE.

Solving 2-LCE and ILCE for $k = n/2$. The above approach, for $k = \frac{n}{2}$, gives that only 4 LCE or 2 ILCE samples are sufficient to recover the secret monomial matrix $\boldsymbol{Q}$ with high probability. We now show how to exploit the special structure of $\boldsymbol{Q}$ to reduce the number of samples even further. Note that $\boldsymbol{Q}$ contains exactly one non-zero entry in each of its rows as well as in each of its columns. Therefore, we construct a linear system of equations as in Equation (1) with 2 LCE samples. In total, we get $2k(n-k) = \frac{n^2}{2}$ equations with n^2 unknowns, hence an underdetermined linear system. Earlier, we resolved this issue by adding more equations (from extra samples) to make the system overdetermined. Here, we try to reduce the number of unknowns from n^2 to some number smaller or equal to $\frac{n^2}{2}$. Note that due to the monomial structure, if we know that $\boldsymbol{Q}$ has a non-zero entry at a certain position (i, j) then we can immediately set the rest of the unknowns in the corresponding row (i) and column (j) to zero. In short, the knowledge of a single non-zero entry allows us to guess further $2(n-1)$ entries for free. Moreover, since these $2(n-1)$ entries are equal to 0 we can modify the linear system by deleting the columns corresponding to these positions in $\mathsf{vec}(\boldsymbol{Q})$. Thus, we proceed by guessing the (i, j)-th entry of $\boldsymbol{Q}$ to be non-zero. Let S be the linear system obtained from two LCE samples with $\frac{n^2}{2}$ rows and n^2 columns. We guess that the (i, j)-th entry of $\boldsymbol{Q}$ is non-zero and drop the corresponding $2(n-1)$ columns from S. Let this modified system be $\mathsf{S}_{i,j}$. We know that by construction the system S accepts a solution since $\mathsf{vec}(\boldsymbol{Q})$ is a solution by construction. If our guess of the non-zero entry at (i, j)-th position is correct, then the modified system $\mathsf{S}_{i,j}$ must also accept a solution. Otherwise, our guess is incorrect. This allows us to efficiently check whether the guessed

positions of non-zero entries of $\boldsymbol{Q}$ are correct, and with a certain probability, to exclude incorrect guesses. We show that, with high probability, we can obtain an overdetermined system by performing guesses on each of the n^2 entries of $\boldsymbol{Q}$. Finally, we solve the resulting overdetermined system using Gaussian elimination to recover the secret monomial matrix $\boldsymbol{Q}$. The total cost of this procedure is $O(n^{2+2\omega})$.

Earlier we discussed that, in the case of t-$\mathsf{ILCE}_{n,k,q}$, it is possible to halve the number of necessary samples with respect to t-$\mathsf{LCE}_{n,k,q}$. Using similar arguments, we show that one only sample of ILCE is enough to efficiently recover the secret $\boldsymbol{Q}$ with high probability, again when $k = \frac{n}{2}$.

1.3 Related work

The cryptanalysis of equivalence problems on linear codes started with Leon's algorithm [21], which presented a way to compute the permutation between two equivalent codes using the information provided by codewords of minimal weight, but it is unpractical for cryptographic instances. Later, Petrank and Roth [27] showed that PCE is unlikely to be NP-complete.

In his seminal work [33], Sendrier introduced the Support Splitting Algorithm, which can recover the secret permutation underlying PCE in time $\widetilde{O}(q^h)$, where q is the cardinality of the field and h is the dimension of the *hull* of the code, namely the intersection between the code and its dual. While Sendrier's algorithm fails when the hull is trivial, the authors of [30,3] proposed two attacks on PCE with trivial hulls. All these results imply that PCE is not hard when the hull is small, and this happens with high probability when the code is randomly chosen ([32] showed that in this case, the hull dimension is a small constant). Hence, PCE must be instantiated with self-dual or weakly-dual codes to be suitable in cryptography.

Linear Code Equivalence and Matrix Code Equivalence. In [34] Sendrier and Simos showed that LCE can be reduced to PCE using the closure of the code. This implied that one should be able to solve LCE using the above techniques, but, for $q \geq 5$, the closure of a code is always weakly-self dual, and the Support Splitting Algorithm becomes unpractical. Contrary to PCE, random instances of LCE remain intractable, and hence, they can be used in the design of cryptosystems. After the publication of LESS [10], the effort for cryptanalyzing PCE and LCE increased [5,7], which led to a refinement of the conjectured practical complexity of solving these problems. In summary, the known techniques are practical for particular classes of codes, while finding the permutation or the linear map leading to the equivalence seems to be still intractable for carefully generated instances. In the case of matrix codes, the equivalence problem was first studied from a cryptographic point of view in [29] and it is further cryptanalyzed in the work that introduces MEDS [14], presenting an adaptation of Leon's algorithm in the setting of matrix codes and an algebraic modeling.

Code Equivalence Problems with multiple samples. All of the above works consider the classical statement of the equivalence problems, where just a

single pair of equivalent codes is given. The scenario, however, changes when we consider relaxed versions of LCE and MCE. For instance, suppose that the secret matrix $\boldsymbol{Q}$ is used more than once, exposing multiple pairs of codes $(\mathcal{C}_i, \mathcal{C}'_i)_i$ linked by the same matrix $\boldsymbol{Q}$. Recently, D'Alconzo and Di Scala [17] proved that the variants that do not use the systematic forms of MCE and PCE can be efficiently solved with a polynomial number of samples sharing the same secret.

2 Preliminaries

In this paper, we denote with $\mathbb{Z}$ and $\mathbb{R}$ the sets of integer and real numbers respectively. For a number $n \in \mathbb{N}$ we use $[n]$ for the set $\{1, 2, \ldots, n\}$. We denote matrices with upper-case bold letters (e.g. $\boldsymbol{A}$) and vectors with lower-case bold letters (e.g. $\boldsymbol{a}$). We treat vectors as columns unless otherwise specified. Let $\mathbb{F}_q$ denote a finite field of order q. The tensor product $(\boldsymbol{A} \otimes \boldsymbol{B}) \in \mathbb{F}_q^{mr \times ns}$ of two matrices $\boldsymbol{A} \in \mathbb{F}_q^{m \times n}$ and $\boldsymbol{B} \in \mathbb{F}_q^{r \times s}$ is defined as the Kronecker product of $\boldsymbol{A}$ and $\boldsymbol{B}$.

Denote with $\mathsf{GL}_n(\mathbb{F}_q)$ the set of invertible $n \times n$ matrices with elements in $\mathbb{F}_q$, with $\mathsf{Perm}_n(\mathbb{F}_q)$ the set of permutation matrices of dimension n, and with $\mathsf{Mono}_n(\mathbb{F}_q)$ the set of $n \times n$ *monomial* matrices, i.e., that can be written as $\boldsymbol{M} = \boldsymbol{D}\boldsymbol{P}$, where $\boldsymbol{D} \in \mathbb{F}_q^{n \times n}$ is full-rank diagonal, and $\boldsymbol{P} \in \mathsf{Perm}_n(\mathbb{F}_q)$. We also use $\boldsymbol{I}_n$ to denote $n \times n$ identity matrix over $\mathbb{F}_q$.

For any matrix $\boldsymbol{M} \in \mathbb{F}_q^{n \times n}$, we write $\mathsf{vec}(\boldsymbol{M})$ to denote the column vector of n^2 coefficients consisting of the concatenation of the rows of $\boldsymbol{M}$.

We assume that computing multiplication and inverse of matrices can be performed using $O(n^\omega)$ field operations for some $\omega \in [2, 3]$.[4] Consequently, we assume that solving a linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ with $\boldsymbol{A} \in \mathbb{F}_q^{n \times n}$ and $\boldsymbol{b} \in \mathbb{F}_q^n$ takes time $O(n^\omega)$ field operations, and that calculating the rank (and kernel) of $\boldsymbol{A} \in \mathbb{F}_q^{n \times n}$ costs $O(n^\omega)$ field operations.[5]

2.1 Linear Codes and Equivalence Problems

An (n, k)-linear code $\mathcal{C}$ over $\mathbb{F}_q$ is a k-dimensional vector subspace of $\mathbb{F}_q^n$. We say that $\mathcal{C}$ has length n and dimension k.

A matrix $\boldsymbol{G} \in \mathbb{F}_q^{k \times n}$ is called a *generator matrix* of $\mathcal{C}$ if its rows form a basis of $\mathcal{C}$, that is $\mathcal{C} = \{\boldsymbol{u}^T\boldsymbol{G}, \boldsymbol{u} \in \mathbb{F}_q^k\}$. We say that $\boldsymbol{G}$ is in *systematic form* if $\boldsymbol{G} = (\,\boldsymbol{I}_k \big| \boldsymbol{M}\,)$ for some $\boldsymbol{M} \in \mathbb{F}_q^{k \times (n-k)}$. The systematic form of a generator can be obtained in polynomial-time by computing its row-echelon form, and it gives a standard basis for the vector space. We denote this operation with $\mathsf{SF}(\cdot)$. For matrix $\boldsymbol{Q} \in \mathbb{F}_q^{m \times n}$, $\mathsf{vec}(\boldsymbol{Q})$ is the column vector over $\mathbb{F}_q$ of length mn created by unrolling the entries of $\boldsymbol{Q}$.

[4] For example, in case of the well-known Strassen's algorithm which is considered as the best algorithm for matrix multiplications for large n, one can set $\omega = \log_2(7)$.

[5] If the matrix $\boldsymbol{A} \in \mathbb{F}_q^{r \times s}$ is rectangular, we set $n = \max\{r, s\}$ in the complexity

A matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ is called *parity check matrix* of $\mathcal{C}$ if and only if $\forall\, \boldsymbol{c} \in \mathcal{C}$ it holds that $\boldsymbol{Hc} = \boldsymbol{0}$. Note that $\mathsf{SF}(\boldsymbol{G}) = (\,\boldsymbol{I}_k | \boldsymbol{M}\,) \iff \mathsf{SF}(\boldsymbol{H}) = (\,-\boldsymbol{M}^\top | \boldsymbol{I}_{n-k}\,)$ for a matrix $\boldsymbol{M} \in \mathbb{F}_q^{k\times(n-k)}$. The parity-check matrix generates the *dual code* of $\mathcal{C}$, denoted with $\mathcal{C}^\perp$. The *hull* of a code $\mathcal{C}$ is defined as the intersection of $\mathcal{C}$ with its dual. A code $\mathcal{C}$ is said *weakly self-dual* if $\mathcal{C} \subset \mathcal{C}^\perp$ and *self-dual* if $\mathcal{C} = \mathcal{C}^\perp$. In both these cases, the dimension of the hull is equal to the dimension of the code.

Due to the extended variety of namings to the Linear Code Equivalence Problem (see Table 2), and for consistency between notations in different articles, we use the acronyms from [34] and [14].

	Permutation Code Equivalence Problem	Linear Code Equivalence Problem	Matrix Code Equivalence Problem
[34,20,25]	PCE	LCE	—
[31]	PEP	—	—
[15,26,2]	PEP	LEP	—
[6]	PEP	LEP	MCE
[14,29]	—	—	MCE

Table 2: Notation naming for the Linear, Permutation, and Matrix Code Equivalence Problems through the state-of-the-art.

Linear Code Equivalence Problem: Let $\boldsymbol{G}, \boldsymbol{G'}$ be the generator matrices of two (n,k)-linear codes $\mathcal{C}, \mathcal{C}'$. We say that $\mathcal{C}$ and $\mathcal{C}'$ are *equivalent* if there exist $\boldsymbol{S} \in \mathsf{GL}_k(\mathbb{F}_q)$ and $\boldsymbol{Q} \in \mathsf{Mono}_n(\mathbb{F}_q)$ such that $\boldsymbol{G'} = \boldsymbol{SGQ}$.

Definition 1 (Linear Code Equivalence (LCE) Problem). *Let $\boldsymbol{G}, \boldsymbol{G'} \in \mathbb{F}_q^{k\times n}$ be the generator matrices of two equivalent (n,k)-linear codes $\mathcal{C}, \mathcal{C}'$, respectively. The Code Equivalence Problem is to find matrices $\boldsymbol{S} \in \mathsf{GL}_k(\mathbb{F}_q)$ and $\boldsymbol{Q} \in \mathsf{Mono}_n(\mathbb{F}_q)$ such that $\boldsymbol{G'} = \boldsymbol{SGQ}$.*

Sometimes, in the literature, LCE is stated as in Definition 1 but without the assurance that such matrices $\boldsymbol{S}$ and $\boldsymbol{Q}$ establishing the equivalence between the two codes actually exist. Nevertheless, cryptographic schemes inherently guarantee the equivalence by construction. Consequently, this work explicitly addresses and incorporates this scenario.

Permutation Code Equivalence Problem: The following is a simpler version of Definition 1 where a permutation matrix is used instead of the monomial matrix.

Definition 2 (Permutation Code Equivalence (PCE) Problem). *Let $\boldsymbol{G}$, $\boldsymbol{G'} \in \mathbb{F}_q^{k\times n}$ be the generator matrices of two equivalent (n,k)-linear codes $\mathcal{C}, \mathcal{C}'$,*

respectively. The Permutation Code Equivalence Problem is to find $\boldsymbol{S} \in \mathsf{GL}_k(\mathbb{F}_q)$ *and* $\boldsymbol{P} \in \mathsf{Perm}_n(\mathbb{F}_q)$ *such that* $\boldsymbol{G'} = \boldsymbol{SGP}$.

Matrix Code Equivalence Problem: A $(m \times r, k)$ *matrix code* is a subspace $\mathcal{D}$ of the space of $m \times r$ matrices. The following problem was introduced in [29,14]. Two matrix codes $\mathcal{D}, \mathcal{D}'$ are *equivalent* if there exists two matrices $\boldsymbol{A} \in \mathsf{GL}_m(\mathbb{F}_q)$ and $\boldsymbol{B} \in \mathsf{GL}_r(\mathbb{F}_q)$ such that $\mathcal{D}' = \boldsymbol{A}\mathcal{D}\boldsymbol{B}$. In fact, [14, Lemma 1] proved that the MCE problem can be redefined in terms of the tensor product $A^{\mathsf{T}} \otimes B$ as described below.

Definition 3 (Matrix Code Equivalence (MCE) Problem). *Let* $\boldsymbol{G}, \boldsymbol{G}' \in \mathbb{F}_q^{k \times mr}$ *be generators of two equivalent* $(m \times r, k)$*-matrix codes* $\mathcal{D}, \mathcal{D}'$ *respectively. The Matrix Code Equivalence problem is to find* $\boldsymbol{S} \in \mathsf{GL}_k(\mathbb{F}_q)$, $\boldsymbol{A} \in \mathsf{GL}_m(\mathbb{F}_q)$ *and* $\boldsymbol{B} \in \mathsf{GL}_r(\mathbb{F}_q)$ *such that* $\boldsymbol{G'} = \boldsymbol{SG}(\boldsymbol{A}^\top \otimes \boldsymbol{B})$.

Inverse Linear Code Equivalence Problem: In the context of linkable ring signatures, the following problem was initially introduced in [4].

Definition 4 (Inverse Linear Code Equivalence (ILCE) Problem). *Let* $\boldsymbol{G}, \boldsymbol{G'}, \boldsymbol{G''} \in \mathbb{F}_q^{k \times n}$ *be the generator matrices of three equivalent* (n, k)*-linear codes* $\mathcal{C}$, $\mathcal{C}'$ *and* $\mathcal{C}''$ *respectively. The Inverse Linear Code Equivalence Problem is to find* $\boldsymbol{S} \in \mathsf{GL}_k(\mathbb{F}_q)$ *and* $\boldsymbol{Q} \in \mathsf{Mono}_n(\mathbb{F}_q)$ *such that* $\boldsymbol{G'} = \boldsymbol{SGQ}$ *and* $\boldsymbol{G''} = \boldsymbol{S}^{-1}\boldsymbol{GQ}^{-1}$.

There is also an Inverse Matrix Code Equivalence Problem variant, named IMCE and introduced in [14], that essentially replaces $\boldsymbol{Q} \in \mathsf{Mono}_r(\mathbb{F}_q)$ with $\boldsymbol{Q} \in \mathsf{GL}_{mr}(\mathbb{F}_q)$.

Definition 5 (Inverse Matrix Code Equivalence (IMCE) Problem). *Let* $\boldsymbol{G}, \boldsymbol{G'}, \boldsymbol{G''} \in \mathbb{F}_q^{k \times mr}$ *be generators of three equivalent* $(m \times r, k)$*-matrix codes* $\mathcal{D}, \mathcal{D}'$ *and* $\mathcal{D}''$ *respectively. The Inverse Matrix Code Equivalence problem is to find* $\boldsymbol{S} \in \mathsf{GL}_k(\mathbb{F}_q)$, $\boldsymbol{A} \in \mathsf{GL}_m(\mathbb{F}_q)$ *and* $\boldsymbol{B} \in \mathsf{GL}_r(\mathbb{F}_q)$ *such that* $\boldsymbol{G'} = \boldsymbol{SGQ}$ *and* $\boldsymbol{G''} = \boldsymbol{S}^{-1}\boldsymbol{GQ}^{-1}$ *with* $\boldsymbol{Q} = (\boldsymbol{A}^\top \otimes \boldsymbol{B}) \in \mathsf{GL}_{mr}(\mathbb{F}_q)$.

We introduce below the permutation variant of ILCE for completeness.

Definition 6 (Inverse Permutation Code Equivalence (IPCE) Problem). *Let* $\boldsymbol{G}, \boldsymbol{G'}, \boldsymbol{G''} \in \mathbb{F}_q^{k \times n}$ *be the generator matrices of three equivalent* (n, k)*-linear codes* $\mathcal{C}, \mathcal{C}'$ *and* $\mathcal{C}''$ *respectively. The Inverse Permutation Code Equivalence Problem is to find* $\boldsymbol{S} \in \mathsf{GL}_k(\mathbb{F}_q)$ *and* $\boldsymbol{P} \in \mathsf{Perm}_n(\mathbb{F}_q)$ *such that* $\boldsymbol{G'} = \boldsymbol{SGP}$ *and* $\boldsymbol{G''} = \boldsymbol{S}^{-1}\boldsymbol{GP}^{-1}$.

Remark 1. In practice, one often works with generator matrices in systematic forms. Hence, when $\boldsymbol{G}, \boldsymbol{G'}$ are in systematic form, we say that $\mathcal{C}, \mathcal{C}'$ are equivalent if there exists $\boldsymbol{Q} \in \mathsf{Mono}_n(\mathbb{F}_q)$ such that $\boldsymbol{G'} = \mathsf{SF}(\boldsymbol{GQ})$. Definitions 1 to 6 can all be equivalently restated with the generators in systematic form without changing the hardness of the problems.

2.2 Code equivalence problems with multiple samples

In order to study the stronger cryptographic properties of the equivalence problems, we introduce some new definitions allowing an interaction with stronger adversaries. We give in Definition 7 a relaxed versions of LCE where the adversary has access to multiple LCE samples for the same secret monomial $\boldsymbol{Q}$.

Definition 7 (t-$\mathsf{LCE}_{n,k,q}$)**.** *Let n, k, q be integers such that $k < n$ and q is prime. Let $\boldsymbol{Q} \in \mathsf{Mono}_n(\mathbb{F}_q)$ be a* secret *monomial matrix. We denote by $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$ the probability distribution on $\mathbb{F}_q^{k\times n} \times \mathbb{F}_q^{k\times n}$ obtained by sampling $\boldsymbol{M} \in \mathbb{F}_q^{k\times(n-k)}$ uniformly at random, setting $\boldsymbol{G} = (\boldsymbol{I}_k | \boldsymbol{M}) \in \mathbb{F}_q^{k\times n}$, and returning*

$$(\boldsymbol{G}, \boldsymbol{G}' = \mathsf{SF}(\boldsymbol{GQ})).$$

Given t independent samples from $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$, the t-samples LCE problem*, denoted as t-$\mathsf{LCE}_{n,k,q}, is to find $\boldsymbol{Q}$.*

Informally, the distribution $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$ samples a generator matrix $\boldsymbol{G}$ (in systematic form) of a random (n,k)-linear code over $\mathbb{F}_q$ and outputs the pair $(\boldsymbol{G}, \boldsymbol{G}')$, where $\boldsymbol{G}'$ is the generator matrix (in systematic form) of another equivalent linear code, and the equivalence is established via a secret monomial matrix $\boldsymbol{Q}$. When the parameters n, k, q are clear by the context, we simplify the notation and drop the indices from the shortening of the problem, i.e., we simply write t-LCE. Also, notice that 1-LCE corresponds to LCE, so in this case only write LCE.

We give in Definition 8 and Definition 9 the corresponding t-samples problems for PCE (Definition 2) and MCE (Definition 3).

Definition 8 (t-$\mathsf{PCE}_{n,k,q}$)**.** *Let n, k, q be integers such that $k < n$ and q is prime. Let $\boldsymbol{Q} \in \mathsf{Perm}_n(\mathbb{F}_q)$ be a* secret *permutation matrix. We denote by $\mathcal{P}_{n,k,q,\boldsymbol{Q}}$ the probability distribution on $\mathbb{F}_q^{k\times n} \times \mathbb{F}_q^{k\times n}$ obtained by sampling $\boldsymbol{M} \in \mathbb{F}_q^{k\times(n-k)}$ uniformly at random, setting $\boldsymbol{G} = (\boldsymbol{I}_k | \boldsymbol{M}) \in \mathbb{F}_q^{k\times n}$, and returning*

$$(\boldsymbol{G}, \boldsymbol{G}' = \mathsf{SF}(\boldsymbol{GP})).$$

Given t independent samples from $\mathcal{P}_{n,k,q,\boldsymbol{P}}$, the t-samples PCE problem*, denoted as t-$\mathsf{PCE}_{n,k,q}, is to find $\boldsymbol{P}$.*

Definition 9 (t-$\mathsf{MCE}_{m,r,k,q}$)**.** *Let k, m, r, q be integers such that $k < mr$ and q is prime. Let $\boldsymbol{A} \in \mathsf{GL}_m(\mathbb{F}_q)$ and $\boldsymbol{B} \in \mathsf{GL}_r(\mathbb{F}_q)$ be* secret *unimodular matrices. We denote by $\mathcal{M}_{m,r,k,q,\boldsymbol{A}^\top \otimes \boldsymbol{B}}$ the probability distribution on $\mathbb{F}_q^{k\times mr} \times \mathbb{F}_q^{k\times mr}$ obtained by sampling $\boldsymbol{M} \in \mathbb{F}_q^{k\times(mr-k)}$ uniformly at random, setting $\boldsymbol{G} = (\boldsymbol{I}_k | \boldsymbol{M}) \in \mathbb{F}_q^{k\times mr}$, and returning*

$$(\boldsymbol{G}, \boldsymbol{G}' = \mathsf{SF}(\boldsymbol{G}(\boldsymbol{A}^\top \otimes \boldsymbol{B}))).$$

Given t independent samples from $\mathcal{M}_{m,r,k,q,\boldsymbol{A}^\top \otimes \boldsymbol{B}}$, the t-samples MCE problem*, denoted as t-$\mathsf{MCE}_{m,r,k,q}, is to find $\boldsymbol{A}$ and $\boldsymbol{B}$.*

The *t-samples* version problem for ILCE is as follows.

Definition 10 (t-$\mathsf{ILCE}_{n,k,q}$)**.** *Let n, k, q be integers such that $k < n$ and q is prime. Let $\boldsymbol{Q} \in \mathsf{Mono}_n(\mathbb{F}_q)$ be a* secret *monomial matrix. We denote by $\widehat{\mathcal{L}}_{n,k,q,\boldsymbol{Q}}$ the probability distribution on $\mathbb{F}_q^{k\times n} \times \mathbb{F}_q^{k\times n} \times \mathbb{F}_q^{k\times n}$ obtained by sampling $\boldsymbol{M} \in \mathbb{F}_q^{k\times(n-k)}$ uniformly at random, setting $\boldsymbol{G} = (\boldsymbol{I}_k | \boldsymbol{M}) \in \mathbb{F}_q^{k\times n}$, and returning*

$$(\boldsymbol{G}, \boldsymbol{G}' = \mathsf{SF}(\boldsymbol{GQ}), \boldsymbol{G}'' = \mathsf{SF}(\boldsymbol{GQ}^{-1})).$$

Given t independent samples from $\widehat{\mathcal{L}}_{n,k,q,\boldsymbol{Q}}$, the t-samples ILCE problem*, denoted as t-$\mathsf{ILCE}_{n,k,q}$, is to find $\boldsymbol{Q}$.*

Analogously, one can give the definitions for t-$\mathsf{IPCE}_{n,k,q}$ and t-$\mathsf{IMCE}_{m,r,k,q}$, and the corresponding distributions $\widehat{\mathcal{P}}_{n,k,q,\boldsymbol{Q}}$ and $\widehat{\mathcal{M}}_{m,r,k,q,\boldsymbol{A}\otimes\boldsymbol{B}}$. Similarly to t-LCE, we simplify, whenever possible, the shortening of all these problems. Observe that for t-$\mathsf{PCE}_{n,k,q}$ and t-$\mathsf{IPCE}_{n,k,q}$ the relevant case is when the distributions $\mathcal{P}_{n,k,q,\boldsymbol{P}}$ and $\widehat{\mathcal{P}}_{n,k,q,\boldsymbol{Q}}$ sample (weakly) self-dual codes, since for general random codes the problem is solvable in polynomial time as early as $t = 1$.

2.3 Code equivalences modeled as group actions

A group action is a mapping of the form $\star : G \times X \to X$, where G is a group and X is a set, such that for any $g_1, g_2 \in G$ and any $x \in X$, we have $g_1 \star (g_2 \star x) = (g_1 g_2) \star x$. Cryptographic group actions are endowed with the certain hardness properties, such as *one-wayness*, *weak-unpredictability* and *weak-pseudorandomness* [1].

The Linear Code Equivalence problem (Definition 1) can be modeled as a group action as follows. Define the group $G = \mathsf{GL}_k(\mathbb{F}_q) \times \mathsf{Mono}_n(\mathbb{F}_q)$ and the set $X = \mathbb{F}_q^{k\times n}$. Then the group action is defined as

$$\star\colon G \times X \to X, \quad ((\boldsymbol{S}, \boldsymbol{Q}), \boldsymbol{G}) \mapsto (\boldsymbol{S}, \boldsymbol{Q}) \star \boldsymbol{G} := \boldsymbol{SGQ}.$$

Similarly, PCE and MCE are modeled as group actions following the same framework. Consequently, it follows that LCE, PCE, and MCE are instances of the so-called Vectorization Problem [16].

Similarly, 2-LCE, 2-PCE, 2-MCE are special cases of the 2-GAIP defined in [6, Problem 3]. Additionally, Definition 11 describes a useful property required for building secure threshold signatures as analyzed in [6].

Definition 11. (2-weakly pseudorandom group action [6, Def. 3]) *A group action $\star\colon G \times X \to X$ is 2-weakly pseudorandom if there is no probabilistic polynomial time algorithm that given $(x, g \star x)$ can distinguish with non negligible probability between (x', y') and $(x', g \star x')$ with. $x', y' \in X$ sampled uniformly at random from X.*

3 Solving Code Equivalence with Multiple Instances

Recently, D'Alconzo and Di Scala [17] showed that, using representation theory, for certain group actions $(G, X, \star)$ it is possible to recover the secret $g \in G$ from a polynomial number of samples of the form $(x_i, g \star x_i)$ for random $x_i \in X$. In the case of the group action defined in Section 2.3, this can be viewed as variants of the problems t-LCE, t-PCE, and t-MCE that do not use the systematic form SF. They show that these variants can be solved efficiently (with high probability) when $t \in \mathsf{poly}(\lambda)$. In the case of t-LCE they showed that $t \geq nk$ samples are sufficient to recover the secret matrices $\boldsymbol{S}$ and $\boldsymbol{Q}$ (with high probability).

In this section, we show that the use of the systematic form leads to a significantly smaller number of samples from $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$ (resp. $\mathcal{P}_{n,k,q,\boldsymbol{Q}}$ and $\mathcal{M}_{m,r,k,q,\boldsymbol{Q}}$)) needed to solve the corresponding computational problems. The key ingredient of our results relies on [30, Corollary 3.2.13].

In what follows, we focus our analysis on t-LCE. Since we don't take advantage of the structure of the secret matrix, we obtain simultaneously results for t-MCE too, as it falls in the same setting. Moreover, unless differently specified, we do not restrict our linear codes to have any specific structure or properties, e.g. self-dual codes or non-self-dual codes.

Lemma 1. *Given two generator matrices* $\boldsymbol{G} = (\boldsymbol{I}_k | \boldsymbol{M}) \in \mathbb{F}_q^{k \times n}$ *and* $\boldsymbol{G}' = \mathsf{SF}(\boldsymbol{GQ}) = (\boldsymbol{I}_k | \boldsymbol{M}') \in \mathbb{F}_q^{k \times n}$ *of two equivalent codes for some* $\boldsymbol{Q} \in \mathsf{GL}_n(\mathbb{F}_q)$*, we have that*

$$\left[(\boldsymbol{I}_k | \boldsymbol{M}) \otimes (-\boldsymbol{M}'^{\top} | \boldsymbol{I}_{n-k}) \right] \mathsf{vec}(\boldsymbol{Q}) = \boldsymbol{0}. \tag{2}$$

Proof. This is a straightforward application of [30, Definition 1.1.3 and Corollary 3.2.13] without assuming the matrix $\boldsymbol{Q}$ to be a permutation, where $\boldsymbol{G} = (\boldsymbol{I}_k | \boldsymbol{M})$ and the parity-check matrix of the code generated by $\boldsymbol{G}' = (\boldsymbol{I}_k | \boldsymbol{M}')$ is given by $(-\boldsymbol{M}'^{\top} | \boldsymbol{I}_{n-k})$. □

Notice that Lemma 1 gives $k(n-k)$ linear equations in the n^2 variables determining the entries of $\boldsymbol{Q}$. [6] Such a linear system has the following particular structure. Let us denote the (i,j)-th entry of $\boldsymbol{M}$ by $\boldsymbol{M}_{i,j}$, then the homogeneous linear system of equations derived from Equation (2) can be written as: $\boldsymbol{A} \cdot \mathsf{vec}(\boldsymbol{Q}) = \boldsymbol{0}$ where $\boldsymbol{A}$ is equal to

$$\begin{bmatrix}
-\boldsymbol{M}'^{\top} & \boldsymbol{I}_c & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & -\boldsymbol{M}_{1,1}\boldsymbol{M}'^{\top} & \boldsymbol{M}_{1,1}\boldsymbol{I}_c & \cdots & -\boldsymbol{M}_{1,c}\boldsymbol{M}'^{\top} & \boldsymbol{M}_{1,c}\boldsymbol{I}_c \\
\boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{M}'^{\top} & \boldsymbol{I}_c & \ddots & \vdots & -\boldsymbol{M}_{2,1}\boldsymbol{M}'^{\top} & \boldsymbol{M}_{2,1}\boldsymbol{I}_c & \cdots & -\boldsymbol{M}_{2,c}\boldsymbol{M}'^{\top} & \boldsymbol{M}_{2,c}\boldsymbol{I}_c \\
\vdots & \ddots & \ddots & \ddots & \ddots & \boldsymbol{0} & \vdots & \vdots & \cdots & \vdots & \vdots \\
\boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{M}'^{\top} & \boldsymbol{I}_c & -\boldsymbol{M}_{k,1}\boldsymbol{M}'^{\top} & \boldsymbol{M}_{k,1}\boldsymbol{I}_c & \cdots & -\boldsymbol{M}_{k,c}\boldsymbol{M}'^{\top} & \boldsymbol{M}_{k,c}\boldsymbol{I}_c
\end{bmatrix}$$

[6] In case of LCE we restrict $\boldsymbol{Q}$ to be in $\mathsf{Mono}_n(\mathbb{F}_q)$, while for MCE we assume that $n = mr$ and $\boldsymbol{Q} = \boldsymbol{A}^{\top} \otimes \boldsymbol{B}$ for some $\boldsymbol{A} \in \mathsf{GL}_m(\mathbb{F}_q)$ and $\boldsymbol{B} \in \mathsf{GL}_r(\mathbb{F}_q)$.

with $c = (n-k)$. In particular, the matrix $\boldsymbol{A}$ has full (row) rank due to the presence of k identity blocks $\boldsymbol{I}_{n-k}$.

Lemma 2 gives the necessary number of samples to build a determined linear system that allows an efficient recovery of $\boldsymbol{Q}$.

Proposition 1. *For a prime $q \geq 2$ and integers $n \geq 2$, $k \in [n-1]$, for any matrices $\boldsymbol{M}, \boldsymbol{M}', \boldsymbol{N}, \boldsymbol{N}' \in \mathbb{F}_q^{k \times (n-k)}$, let*

$$\boldsymbol{A} := \begin{bmatrix} (\,\boldsymbol{I}_k \,|\, \boldsymbol{M}\,) \otimes (\,-\boldsymbol{M}'^{\top} \,|\, \boldsymbol{I}_{n-k}\,) \\ (\,\boldsymbol{I}_k \,|\, \boldsymbol{N}\,) \otimes (\,-\boldsymbol{N}'^{\top} \,|\, \boldsymbol{I}_{n-k}\,) \end{bmatrix} \in \mathbb{F}_q^{2k(n-k) \times n^2}.$$

Then rows of $\boldsymbol{A}$ are linearly dependent if and only if for some $i \in [n-k]$ and for some $j \in [k]$

- *the i^{th} column of the matrices $\boldsymbol{M}'$ and $\boldsymbol{N}'$ are equal, and*
- *the j^{th} row of the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ are equal.*

Proof. Let us consider the matrix $\boldsymbol{A} := \left[\begin{array}{c} \boldsymbol{A}_{\boldsymbol{M},\boldsymbol{M}'} \\ \hline \boldsymbol{A}_{\boldsymbol{N},\boldsymbol{N}'} \end{array}\right]$ in blocks $\boldsymbol{B}_{1,1}, \boldsymbol{B}_{1,2}, \boldsymbol{B}_{2,1}, \boldsymbol{B}_{2,2}$ as written below. In the following we use $c := (n-k)$ for ease of notation.

$$\left[\begin{array}{c|c} \boldsymbol{B}_{1,1} & \boldsymbol{B}_{1,2} \\ \hline \boldsymbol{B}_{2,1} & \boldsymbol{B}_{2,2} \end{array}\right] = \left[\begin{array}{cccccc|ccccc} -\boldsymbol{M}'^{\top} & \boldsymbol{I}_c & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & -\boldsymbol{M}_{1,1}\boldsymbol{M}'^{\top} & \boldsymbol{M}_{1,1}\boldsymbol{I}_c & \cdots & -\boldsymbol{M}_{1,c}\boldsymbol{M}'^{\top} & \boldsymbol{M}_{1,c}\boldsymbol{I}_c \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{M}'^{\top} & \boldsymbol{I}_c & \ddots & \vdots & -\boldsymbol{M}_{2,1}\boldsymbol{M}'^{\top} & \boldsymbol{M}_{2,1}\boldsymbol{I}_c & \cdots & -\boldsymbol{M}_{2,c}\boldsymbol{M}'^{\top} & \boldsymbol{M}_{2,c}\boldsymbol{I}_c \\ \vdots & \ddots & \ddots & \ddots & \ddots & \boldsymbol{0} & \vdots & \vdots & \cdots & \vdots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{M}'^{\top} & \boldsymbol{I}_c & -\boldsymbol{M}_{k,1}\boldsymbol{M}'^{\top} & \boldsymbol{M}_{k,1}\boldsymbol{I}_c & \cdots & -\boldsymbol{M}_{k,c}\boldsymbol{M}'^{\top} & \boldsymbol{M}_{k,c}\boldsymbol{I}_c \\ \hline -\boldsymbol{N}'^{\top} & \boldsymbol{I}_c & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & -\boldsymbol{N}_{1,1}\boldsymbol{N}'^{\top} & \boldsymbol{N}_{1,1}\boldsymbol{I}_c & \cdots & -\boldsymbol{N}_{1,c}\boldsymbol{N}'^{\top} & \boldsymbol{N}_{1,c}\boldsymbol{I}_c \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{N}'^{\top} & \boldsymbol{I}_c & \ddots & \vdots & -\boldsymbol{N}_{2,1}\boldsymbol{N}'^{\top} & \boldsymbol{N}_{2,1}\boldsymbol{I}_c & \cdots & -\boldsymbol{N}_{2,c}\boldsymbol{N}'^{\top} & \boldsymbol{N}_{2,c}\boldsymbol{I}_c \\ \vdots & \ddots & \ddots & \ddots & \ddots & \boldsymbol{0} & \vdots & \vdots & \cdots & \vdots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{N}'^{\top} & \boldsymbol{I}_c & -\boldsymbol{N}_{k,1}\boldsymbol{N}'^{\top} & \boldsymbol{N}_{k,1}\boldsymbol{I}_c & \cdots & -\boldsymbol{N}_{k,c}\boldsymbol{N}'^{\top} & \boldsymbol{N}_{k,c}\boldsymbol{I}_c \end{array}\right]$$

Let us determine when the matrix $\boldsymbol{A}$ is not full rank. Because of the presence of the sub-blocks $\boldsymbol{I}_c$, we can obtain linear dependence on the rows of $\boldsymbol{A}$ only by subtracting two rows. Let us look at the block $\begin{bmatrix} \boldsymbol{B}_{1,1} \\ \boldsymbol{B}_{2,1} \end{bmatrix}$. Here the sub-matrices $\boldsymbol{I}_c$ are never in the same row. Hence, the only possibility of having a linear dependency is when the j^{th} column of $\boldsymbol{M}'$ is equal to the j^{th} column of $\boldsymbol{N}'$, for some $1 \leq j \leq n-k$ (note that $\boldsymbol{M}', \boldsymbol{N}'$ are transposed). When such an event happens, then we have that the block $\begin{bmatrix} \boldsymbol{B}_{1,2} \\ \boldsymbol{B}_{2,2} \end{bmatrix}$ has two rows that are linearly dependent if and only if the i^{th} row of $\boldsymbol{M}$ is equal to the i^{th} row of $\boldsymbol{N}$, for some $1 \leq i \leq k$. □

Proposition 1 gives the conditions for which the matrix $\boldsymbol{A}$ has maximum rank. Notice that the probability that the j^{th} row of $\boldsymbol{M}$ and $\boldsymbol{N}$ are equal, for some $j \in [k]$ is $p < \frac{k}{q^{n-k}}$, thus negligible in n. This is because both $\boldsymbol{M}$ and $\boldsymbol{N}$

are sampled uniformly at random over $\mathbb{F}_q^{k\times(n-k)}$. On the other hand, $\boldsymbol{M}'$ and $\boldsymbol{N}'$ are not sampled uniformly at random, but they are output of the equivalence transformation for a fixed $\boldsymbol{Q}$. Hence, in this case, we consider the following assumption.

Assumption 1 *Let $(\boldsymbol{G}_\ell, \boldsymbol{G}'_\ell = \mathsf{SF}(\boldsymbol{G}_\ell\boldsymbol{Q}))$ be two samples from $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$, for $\ell = 1, 2$ and for a random $\boldsymbol{Q} \in \mathsf{Mono}_n(\mathbb{F}_q)$. Let $\boldsymbol{M}', \boldsymbol{N}' \in \mathbb{F}_q^{k\times(n-k)}$ be such that $\boldsymbol{G}'_1 = (\boldsymbol{I}_k|\boldsymbol{M}')$ and $\boldsymbol{G}'_2 = (\boldsymbol{I}_k|\boldsymbol{N}')$. Then, the probability that $\boldsymbol{M}'$ and $\boldsymbol{N}'$ have their i^{th} column equal, for some $i \in [n-k]$ is negligible in n.*

Under assumption Assumption 1, we have that the probability of the conditions in Proposition 1 to be simultaneously satisfied is negligible in n. We use Assumption 1 in Lemma 2 and Theorem 1 (in Section 4). In Corollary 1, we implicitly take an analogous assumption for MCE.

Lemma 2. *Under Assumption 1, for $t \geq \left\lceil \frac{n^2}{k(n-k)} \right\rceil$, the t-$\mathsf{LCE}_{n,k,q}$ is solvable with overwhelming probability in time $O(n^{2\omega})$.*

Proof. We show how to recover the secret monomial matrix $\boldsymbol{Q}$ below. For each LCE sample, we use Lemma 1 to construct a system of $k(n-k)$ linear equations with n^2 variables representing the entries of $\boldsymbol{Q}$. So, we only need to show that t samples obtained from the challenger are sufficient to construct a system of linear equations in the n^2 variables from which we can recover $\boldsymbol{Q}$. Below we show how to construct such a system of equation which has row-rank $(n^2 - \xi)$ where $\xi \in \left\{1, \frac{n}{2}\right\}$, with overwhelming probability.

We can then solve this system of equations efficiently to recover $\boldsymbol{Q}$.

Let $\{(\boldsymbol{G}_\ell, \boldsymbol{G}'_\ell = \mathsf{SF}(\boldsymbol{G}_\ell\boldsymbol{Q}))\}_{\ell=1,\ldots,t}$ be t independent samples from $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$. Note that $\boldsymbol{G}_\ell \coloneqq (\boldsymbol{I}_k|\boldsymbol{M}_\ell)$ and $\boldsymbol{G}'_\ell \coloneqq (\boldsymbol{I}_k|\boldsymbol{M}'_\ell)$ where the matrices $\boldsymbol{M}_\ell \in \mathbb{F}_q^{k\times(n-k)}$ are sampled uniformly at random.

Let us construct the following matrix

$$\boldsymbol{A} \coloneqq \left[\boldsymbol{A}_1{}^\top \; \boldsymbol{A}_2{}^\top \cdots \boldsymbol{A}_t{}^\top\right]^\top,$$

where $\boldsymbol{A}_\ell \coloneqq (\boldsymbol{I}_k|\boldsymbol{M}_\ell) \otimes (-\boldsymbol{M}'^\top_\ell|\boldsymbol{I}_{n-k})$ is the system given by Lemma 1, for $\ell \in [t]$. Our desired system of equations is given by $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$. Note that, the kernel of $\boldsymbol{A}$ is not trivial, since we know that there exists at least one non-zero vector (namely $\mathsf{vec}(\boldsymbol{Q})$) in $\mathrm{Ker}(\boldsymbol{A})$. We have that $(t-1)k(n-k) < n^2 \leq tk(n-k)$, where the equality holds only if $k = \frac{n}{2}$.

Let

$$\boldsymbol{A}_{-t} \coloneqq \left[\boldsymbol{A}_1{}^\top \; \boldsymbol{A}_2{}^\top \cdots \boldsymbol{A}_{t-1}{}^\top\right]^\top.$$

Thanks to Proposition 1 and Assumption 1, we know that rows of $\boldsymbol{A}_{-t}$ are linearly dependent with probability $p = \binom{t-1}{2}p'$, where $p' \in \mathsf{negl}(n)$. This implies that, with overwhelming probability $1 - p$,

$$\mathrm{rank}(\boldsymbol{A}_{-t}) = (t-1)k(n-k) = (n^2 - k(n-k)).$$

Now let us see what happens when we add the block $\boldsymbol{A}_t$ to $\boldsymbol{A}_{-t}$ and obtain the full matrix $\boldsymbol{A}$. Again under Assumption 1, $\boldsymbol{A}_t$ shall add $k(n-k)$ rows which are pairwise linearly independent with all the blocks in $\boldsymbol{A}_{-t}$ with overwhelming probability. However, we know that $\text{rank}(\boldsymbol{A}) \leq (n^2-1)$, therefore $(n^2 - k(n-k)) < \text{rank}(\boldsymbol{A}) \leq (n^2-1)$, where the first inequality is strict because of presence of k blocks $\boldsymbol{I}_{n-k}$ in each block matrix $\boldsymbol{A}_\ell$.

Case 1 ($k \neq n/2$ - $\boldsymbol{A}$ is overdetermined): the matrix $\boldsymbol{A}$ has $tk(n-k) > n^2$ rows. Therefore, with overwhelming probability, the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ is overdetermined, since $\boldsymbol{A}$ contains more rows than columns (i.e. we have more equations than the unknown variables). The kernel of $\boldsymbol{A}$ has dimension one as one non-zero solution exists by construction, and thus it coincides with $\langle \mathsf{vec}(\boldsymbol{Q}) \rangle$.

Case 2 ($k = n/2$ - $\boldsymbol{A}$ is underdetermined): the matrix $\boldsymbol{A}$ is square and accepts at least one non-zero solution by construction ($\mathsf{vec}(\boldsymbol{Q})$). Hence, there is at least one row of $\boldsymbol{A}$ that is linearly dependant on the others. But because of the repetitive structure of $\boldsymbol{A}$, every dependent row leads to k dependent rows. Therefore, we have that $\text{rank}(\boldsymbol{A}) = n^2 - n/2$ with overwhelming probability. The kernel of $\boldsymbol{A}$ has dimension $n/2$, and we compute a basis $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n/2} \in \mathbb{F}_q^{n^2}\}$ for it using Gaussian elimination. We prove by contradiction that $\mathsf{vec}(\boldsymbol{Q}) \in \langle \boldsymbol{v}_i \rangle$, for some $i \in [n/2]$. Let us assume that $\mathsf{vec}(\boldsymbol{Q}) \notin \langle \boldsymbol{v}_i \rangle$, for every $i \in [n/2]$. Then, because of how standard Gaussian elimination for computing the kernel generators of a matrix works, we have that, with overwhelming probability, $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n/2}$ stacked in columns as a matrix, will have the lowest $n/2 \times n/2$ block as the identity $\boldsymbol{I}_{n/2}$. Also, $\mathsf{vec}(\boldsymbol{Q}) = \sum_{i=0}^{n/2} \alpha_i \boldsymbol{v}_i$, for some $\alpha_1, \ldots, \alpha_{n/2} \in \mathbb{F}_q$ such that at least two are different from zero. Because of the monomial structure of $\boldsymbol{Q}$, we have that $\mathsf{vec}(\boldsymbol{Q})$ can have at most one non-zero entry among its last $n/2$ entries. Let us consider the case for which $\mathsf{vec}(\boldsymbol{Q})$ has one non-zero entry among its last $n/2$ entries, then we have that $\alpha_i \neq 0$, for some i, and $\alpha_j = 0$, for every $j \in [n/2] \setminus \{i\}$ (because $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n/2}$ are row-Echelon reduced). On the other hand, if $\mathsf{vec}(\boldsymbol{Q})$ has only zero entries in its last $n/2$ entries, then there exist no linear combination of them that results in $\mathsf{vec}(\boldsymbol{Q})$ (again, because $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n/2}$ are row-Echelon reduced). Hence we can conclude that $\mathsf{vec}(\boldsymbol{Q}) \in \langle \boldsymbol{v}_i \rangle$, for some $i \in [n/2]$, and so it can be efficiently recovered.

In summary, finding $\mathsf{vec}(\boldsymbol{Q})$ reduces to calculating the kernel of $\boldsymbol{A}$, giving a polynomial time complexity of $O(n^{2\omega})$ field operations.

□

Notice that, under an analogous Assumption 1 for PCE, Lemma 2 applies naturally to t-$\mathsf{PCE}_{n,k,q}$, for any dimension of the hull. We extend Lemma 2 to MCE in Corollary 1.

Corollary 1. *For $t \geq \left\lfloor \frac{m^2 r^2}{k(mr-k)} \right\rfloor + 1$, the t-$\mathsf{MCE}_{m,r,k,q}$ is solvable with overwhelming probability in time $O((mr)^{2\omega})$.*

Proof. Since in the proof of Lemma 2 the monomial structure of the secret matrix is never used, except for the case of $k = n/2$, the proof follows with analogous arguments by setting $n := mr$. When $k = n/2$, one must still ensure that the obtained linear system is overdetermined. However, in this case, this is guaranteed by the lower bound on the number of samples t. □

Corollary 2 shows the sufficient number of samples required to recover the secret for t-$\mathsf{LCE}_{n,k,q}$ and t-$\mathsf{MCE}_{m,r,k,q}$ for the parameters used in signature schemes LESS [2] and MEDS [13].

Corollary 2. *The following statements hold.*

a) If $k = n/2$, then $t = 4$ samples from $\mathcal{L}_{n,k,q,\boldsymbol{Q}}$ are sufficient to recover $\boldsymbol{Q}$ with overwhelming probability in time $O(n^{2\omega})$.

b) If $k = r = m$, then $t = \left\lceil \frac{k^2}{k-1} \right\rceil$ samples from $\mathcal{M}_{m,r,k,q,\boldsymbol{A}^\top \otimes \boldsymbol{B}}$ are sufficient to recover $\boldsymbol{Q} = \boldsymbol{A}^\top \otimes \boldsymbol{B}$ with overwhelming probability in time $O(k^{4\omega})$.

3.1 Implications to ILCE and IMCE

In this section we apply the results from Lemma 1 and Lemma 2 to t-$\mathsf{ILCE}_{n,k,q}$. Specifically, we show that t-$\mathsf{ILCE}_{n,k,q}$ allows a recovery of the secret with half the number of samples with respect to t-$\mathsf{LCE}_{n,k,q}$. Similarly as for Lemma 2, we extend our results also to t-$\mathsf{IMCE}_{m,r,k,q}$.

The main difference compared to the LCE case lies in the fact that we cannot use Assumption 1 here. In fact, we need to consider the following.

Assumption 2 *Let $(\boldsymbol{G}, \boldsymbol{G}', \boldsymbol{G}'')$ be a sample from $\widehat{\mathcal{L}}_{n,k,q,\boldsymbol{Q}}$, for a random $\boldsymbol{Q} \in \mathsf{Mono}_n(\mathbb{F}_q)$, and let $\boldsymbol{M}, \boldsymbol{M}', \boldsymbol{M}'' \in \mathbb{F}_q^{k\times(n-k)}$ be such that $\boldsymbol{G} = (\boldsymbol{I}_k | \boldsymbol{M})$, $\boldsymbol{G}' = (\boldsymbol{I}_k | \boldsymbol{M}')$ and $\boldsymbol{G}'' = (\boldsymbol{I}_k | \boldsymbol{M}'')$. Then, the following event*

- *the i^{th} column of the matrices $\boldsymbol{M}'$ and $\boldsymbol{M}$ are equal, and*
- *the j^{th} row of the matrices $\boldsymbol{M}$ and $\boldsymbol{M}''$ are equal,*

for some $i \in [n-k]$ and $j \in [n]$, happens with probability negligible in n.

Note that this assumption is stronger than Assumption 1. However, we observed experimentally that both of them hold in practice. We consider Assumption 2 in Lemma 3 and Theorem 2 (Section 4), and we take an analogous assumption for IMCE in Corollary 3.

Lemma 3. *Under Assumption 2, for $t \geq \left\lceil \frac{n^2}{2k(n-k)} \right\rceil$, the t-$\mathsf{ILCE}_{n,k,q}$ is solvable with overwhelming probability in time $O(n^{2\omega})$.*

Proof. Consider the samples from $\widehat{\mathcal{L}}_{n,k,q,\boldsymbol{Q}}$ (see Definition 10) as two LCE pairs $(\boldsymbol{G}, \boldsymbol{G}')$ and $(\boldsymbol{G}'', \boldsymbol{G})$. Then, we apply Lemma 1 on both of them to get the following system of $2k(n-k)$ linear equations in the n^2 variables $\boldsymbol{Q}_{i,j}$.

$$\begin{bmatrix} (\boldsymbol{I}_k | \boldsymbol{M}) \otimes (-\boldsymbol{M}'^\top | \boldsymbol{I}_{n-k}) \\ (\boldsymbol{I}_k | \boldsymbol{M}'') \otimes (-\boldsymbol{M}^\top | \boldsymbol{I}_{n-k}) \end{bmatrix} \mathsf{vec}(\boldsymbol{Q}) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}. \tag{3}$$

Where, $\boldsymbol{G}' = (\boldsymbol{I}_k | \boldsymbol{M}')$ and $\boldsymbol{G}'' = (\boldsymbol{I}_k | \boldsymbol{M}'')$ for some matrices $\boldsymbol{M}', \boldsymbol{M}'' \in \mathbb{F}_q^{k\times(n-k)}$ and $\boldsymbol{G} = \mathsf{SF}(\boldsymbol{G}''\boldsymbol{Q})$. Proposition 1 together with Assumption 2 ensure that the rows of the matrix in Equation (3) are linearly independent with overwhelming probability, and so the proof follows as in Lemma 2. Since one sample from $\widehat{\mathcal{L}}_{n,k,q,\boldsymbol{Q}}$ gives us two blocks at time (as in Equation (3)), the sufficient number of samples t for recovering $\boldsymbol{Q}$ is halved compared to t-$\mathsf{LCE}_{n,k,q}$. □

Corollary 3. *For $t \geq \left\lfloor \frac{m^2r^2}{2k(mr-k)} \right\rfloor + 1$, the t-$\mathsf{IMCE}_{m,r,k,q}$ is solvable with overwhelming probability in time $O((mr)^{2\omega})$.*

Proof. Follows the same arguments as Lemma 3 and Corollary 1. □

Corollary 4. *The following statements hold.*

a) If $k = n/2$, then $t = 2$ samples from $\widehat{\mathcal{L}}_{n,k,q,\boldsymbol{Q}}$ are sufficient to recover $\boldsymbol{Q}$ with overwhelming probability in time $O(n^{2\omega})$.

b) If $k = r = m$, then $t = \left\lceil \frac{k^2}{2(k-1)} \right\rceil$ samples from $\widehat{\mathcal{M}}_{m,r,k,q,\boldsymbol{A}^\top\otimes\boldsymbol{B}}$ are sufficient to recover $\boldsymbol{Q} = \boldsymbol{A}^\top \otimes \boldsymbol{B}$ with overwhelming probability in time $O(k^{4\omega})$.

4 Further Improvements by Exploiting the Monomial Matrix Structure

In this section, we exploit the structure of the secret matrix in LCE and ILCE for $k = n/2$ to further reduce, for certain parameters range, the number of samples necessary to retrieve the secret. Specifically, we show how to solve, in polynomial time, 2-LCE and ILCE. The approach presented below builds upon the algorithm by Saeed for PCE [30, Sec. 3.7].

4.1 A polynomial-time algorithm for solving 2-LCE for $k = n/2$

Lemma 2 shows that, for $k = n/2$, 4-LCE can be solved in polynomial time. In this section, we give the conditions on q to solve 2-LCE in polynomial time by a new algorithm introduced here. Consider a monomial matrix $\boldsymbol{Q}$ and the following two LCE instances,

$$\begin{aligned} &\Big(\boldsymbol{G}_1 = (\boldsymbol{I}_k | \boldsymbol{M}),\ \boldsymbol{G}_1' = \mathsf{SF}(\boldsymbol{G}_1\boldsymbol{Q}) = (\boldsymbol{I}_k | \boldsymbol{M'})\Big), \\ &\Big(\boldsymbol{G}_2 = (\boldsymbol{I}_k | \boldsymbol{N}),\ \boldsymbol{G}_2' = \mathsf{SF}(\boldsymbol{G}_2\boldsymbol{Q}) = (\boldsymbol{I}_k | \boldsymbol{N'})\Big). \end{aligned} \tag{4}$$

From these, we apply Lemma 1 to each instance to write the following linear system S:

$$\begin{bmatrix} (\boldsymbol{I}_k | \boldsymbol{M}) \otimes (-\boldsymbol{M'}^\top | \boldsymbol{I}_{n-k}) \\ (\boldsymbol{I}_k | \boldsymbol{N}) \otimes (-\boldsymbol{N'}^\top | \boldsymbol{I}_{n-k}) \end{bmatrix} \mathsf{vec}(\boldsymbol{Q}) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \tag{5}$$

for $k = n/2$. Notice that the $2k(n-k) \times n^2$ system in Equation (5) is likely to have full rank when the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ are uniformly random over $\mathbb{F}_q^{k\times(n-k)}$. The idea of our algorithm is to guess, for each row of the secret monomial $\boldsymbol{Q}$, the position of the non-zero entry, and check whether the system admits acceptable solutions or not to infer information related to $\boldsymbol{Q}$.

The following observation holds regarding the monomial matrix $\boldsymbol{Q}$: if a certain entry takes a non-zero value, its structure tells us that all the other entries corresponding to the same row and column in take zero values. Recall that $\boldsymbol{Q} = \boldsymbol{P}\boldsymbol{D}$, for some permutation matrix $\boldsymbol{P} \in \mathsf{Perm}_n(\mathbb{F}_q)$ and diagonal matrix $\boldsymbol{D} \in \mathsf{GL}_n(\mathbb{F}_q)$. Let $d_i \in \mathbb{F}_q^*$ be the i^{th} diagonal entry of $\boldsymbol{D}$. Then $\boldsymbol{R}_i = d_i^{-1}\boldsymbol{Q}$ satisfies $\boldsymbol{G}_1' = \mathsf{SF}(\boldsymbol{G}_1\boldsymbol{R}_i)$ and $\boldsymbol{G}_2' = \mathsf{SF}(\boldsymbol{G}_2\boldsymbol{R}_i))$, for each $i \coloneqq 1, \ldots, n$. In this case, $\boldsymbol{R}_i$ is also monomial, but its i^{th} non-zero entry takes value of 1. Hence, instead of guessing i^{th} non-zero entry of $\boldsymbol{Q}$, we guess i^{th} non-zero entry of $\boldsymbol{R}_i$ which is equal to 1. Given that $\boldsymbol{R}_i$ has the same permutation structure of $\boldsymbol{Q}$, we can infer that $2n-2$ additional entries are automatically zero, resulting in a total of $2n-1$ guessed entries.

For each guessing, we perform the following test.

Test 1 *For the guess on the entry (i,j)-th entry of $\boldsymbol{R}_i$ to be equal to 1 construct from S (Equation (5)) a reduced system $\mathsf{S}_{i,j}$ with n^2-2n+1 variables by setting $\boldsymbol{R}_i(i,j) = 1$ and $\boldsymbol{R}_i(i,\mu), \boldsymbol{R}_i(\eta,j) = 0$, for $\mu \in \{1\ldots n\} \setminus \{j\}$ and $\eta \in \{1\ldots n\} \setminus \{i\}$. Accept the guess if the system $\mathsf{S}_{i,j}$ accepts at least one solution.*

The main idea of our algorithm is to use Test 1 in order to eliminate variables from the system in Equation (5) that are zero, hoping to be able to exclude enough so that the system becomes (over)determined. Notice that Test 1 has no guarantees of rejecting all incorrect guesses. Indeed, an incorrect guess for which the corresponding system $\mathsf{S}_{i,j}$ admits a solution (and this happens with a certain probability depending on the parameters of the problem) will pass Test 1. On the other hand, all correct guesses are such that $\mathsf{S}_{i,j}$ admits at least one solution, thus they always pass Test 1.

One way of checking whether $\mathsf{S}_{i,j}$ accepts solutions is to use Rouché–Capelli Theorem. Indeed, the system $\mathsf{S}_{i,j}$ is of the form $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, and one could simply check whether $\mathrm{rank}(\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{A}|\boldsymbol{b})$. Alternatively, and similarly to what observed by Saeed in [30, Sec. 3.7] for PCE, one could note that making a correct guess to the monomial matrix $\boldsymbol{Q}$ is equivalent to performing a puncturing on the corresponding column of the codes whose equivalence is determined by $\boldsymbol{Q}$. Hence, we have that the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, determined by the punctured codes, accepts a solution (by construction) and must have rank smaller than the original linear system S. Given that the rank of the coefficients matrix of S is $2k(n-k)$, then we have that $\mathrm{rank}((\boldsymbol{A}|\boldsymbol{b})) < 2k(n-k)$. This gives a practical speed-up as it allows to save one rank computation.

The whole method is outlined in Algorithm 1.

Algorithm 1 Solving 2-LCE

Input: A 2-LCE instance as in Equation (4)
Output: A monomial matrix $\boldsymbol{R}$ solution to Equation (4) or $\perp$

Construct the linear system S given by Equation (5)
Set $g = [g_1, \ldots, g_n]$ such that g_i is an empty list
for $i \coloneqq 1$ to n **do** ▷ loop over rows
    for $j \coloneqq 1$ to n **do** ▷ loop over columns
        if Test 1 passes **then**
            Append j to the list g_i
        end if
    end for
end for
Construct the linear system $\mathsf{S}_{\mathsf{red}}$ obtained by substituting $\boldsymbol{Q}_{i,j} = 0$ in S for each $i \coloneqq 1, \ldots, n$ and $j \notin \mathsf{g}_i$
if $\mathsf{S}_{\mathsf{red}}$ is underdetermined **then**
    Return $\perp$
end if
Compute a solution matrix $\boldsymbol{R}$ of the linear system $\mathsf{S}_{\mathsf{red}}$
Return $\boldsymbol{R}$

Notice that, when Algorithm 1 succeeds, it returns an equivalent solution (a multiple) to the original secret matrix $\boldsymbol{Q}$.

Analysis of Algorithm 1 Let g_i be the list of guesses that pass Test 1 in Algorithm 1, for each i^{th} row of $\boldsymbol{R}_i$. Let us fix i and j, and consider the linear system $\mathsf{S}_{i,j}$ from Test 1. Let

$$\mathcal{S}_{i,j} = \{\mathsf{S}_{i,j} \mid \text{for every } \boldsymbol{M}, \boldsymbol{N} \in \mathbb{F}_q^{k\times(n-k)}\}.$$

Since $\boldsymbol{Q}$ is fixed, $\boldsymbol{M}'$ and $\boldsymbol{N}'$ are completely determined by $\boldsymbol{M}$ and $\boldsymbol{N}$. Then, the size of $\mathcal{S}_{i,j}$ is bounded by all possibilities for $\boldsymbol{M}$ and $\boldsymbol{N}$, that is $\#\mathcal{S}_{i,j} \leq q^{2k(n-k)}$.

For our analysis, we consider the following.

Assumption 3 *Given the system S, for any indexes i and j, the reduced system $\mathsf{S}_{i,j}$ is distributed uniformly at random over $\mathcal{S}_{i,j}$.*

We give the success condition and the time and memory complexities of Algorithm 1 in Theorem 1.

Theorem 1. *Let $n, k \in \mathbb{N}$ be such that $k = n/2$ and $n > 2$, and let q be a prime such that*

$$q \geq \frac{3n-4}{n-2}.$$

Under Assumption 1 and Assumption 3, we have that Algorithm 1 solves $2\text{-LCE}_{n,k,q}$ in polynomial time and memory with overwhelming probability.

Proof. We determine the complexity of Algorithm 1 with the given constraints on the parameters. Because the pair of LCE samples are random, the constructed system S has maximum rank with overwhelming probability (see Section 3). The cost for calculating the rank of the coefficient matrix of the reduced system $\mathsf{S}_{i,j}$ and of its augmented matrix is $O(n^{2\omega})$ field operations for every i, j, resulting in a total of $O(n^{2+2\omega})$. We compute the probability that $\mathsf{S}_{i,j}$ has non-maximum rank, i.e., that Test 1 is accepted, as follows.

$$\begin{aligned}
\mathsf{Pr}\left(\text{Test 1 is accepted}\right) &= \mathsf{Pr}\left(\mathsf{S}_{i,j} \xleftarrow{\$} \mathcal{S}_{i,j} \text{ has rank smaller than } 2k(n-k)\right) \\
&= \frac{q^{2k(n-k)-1} + q^{2k(n-k)-2} + \cdots + q^2 + q}{q^{2k(n-k)}} \\
&= \frac{1}{q} + \frac{1}{q^2} + \cdots + \frac{1}{q^{2k(n-k)-2}} + \frac{1}{q^{2k(n-k)-1}} \\
&= \frac{q^{2k(n-k)} - q}{(q-1)q^{2k(n-k)}}.
\end{aligned}$$

For $k = \frac{n}{2}$, we have that

$$\mathsf{Pr}\left(\text{Test 1 is accepted}\right) = \frac{q^{\frac{n^2}{2}} - q}{(q-1)q^{\frac{n^2}{2}}} = (1 - \epsilon(n))\frac{1}{q-1},$$

for some negligible $\epsilon(n)$. Test 1 always accepts the guess corresponding to the right solution which reduces the rank of $\mathsf{S}_{i,j}$ by construction. The list g_i in Algorithm 1 contains the accepted guesses, and its expected size $\#g_i$ is

$$1 + (n-1)\mathsf{Pr}(\text{Test 1 is accepted}) = 1 + (1 - \epsilon(n))\frac{n-1}{q-1}.$$

Thus, for every row i, we reduce the amount of variables $\boldsymbol{Q}_{i,1}, \ldots, \boldsymbol{Q}_{i,n}$ from n to $\#g_i$. The expected number of variables $\sum_{i=1}^{n} \#g_i$ of $\mathsf{S}_{\mathsf{red}}$ is

$$\left(1 + (1 - \epsilon(n))\frac{n-1}{q-1}\right) n < \left(1 + \frac{n-1}{q-1}\right) n \leq \left(1 + \frac{(n-1)(n-2)}{2(n-1)}\right) n = n^2/2.$$

Hence, we have that the system $\mathsf{S}_{\mathsf{red}}$ with $n^2/2$ equations is overdetermined with overwhelming probability, and so it has a unique solution by construction. The cost of solving such a system is $O(n^{2\omega})$. Therefore, Algorithm 1 runs in polynomial time complexity

$$O(n^{2+2\omega} + n^{2\omega}) = O(n^{2+2\omega}),$$

and has a memory complexity of $O(n^4)$ field elements. □

Remark 2. Algorithm 1 is highly parallelizable since all n^2 guesses can be evaluated independently.

Remark 3. The success condition $q \geq \frac{3n-4}{n-2}$ in Theorem 1 translates to $q > 3$, for $n \geq 4$, covering all cryptographic relevant values of q and n.

4.2 A polynomial-time algorithm for solving ILCE for $k = n/2$

In this section, we show how to solve ILCE in polynomial time, for certain parameters sets. Consider a monomial matrix $\boldsymbol{Q}$ and the ILCE instance $(\boldsymbol{G} = (\boldsymbol{I}_k|\boldsymbol{M}), \boldsymbol{G}' = \mathsf{SF}(\boldsymbol{GQ}), \boldsymbol{G}'' = \mathsf{SF}(\boldsymbol{GQ}^{-1}))$, where

$$\Big(\boldsymbol{G} = (\boldsymbol{I}_k|\boldsymbol{M}), \boldsymbol{G}' = (\boldsymbol{I}_k|\boldsymbol{M}'), \ \boldsymbol{G}'' = (\boldsymbol{I}_k|\boldsymbol{M}'')\Big) \tag{6}$$

for some $\boldsymbol{M}, \boldsymbol{M}', \boldsymbol{M}'' \in \mathbb{F}_q^{k\times(n-k)}$. From this, we apply Lemma 3 to the above ILCE instance to get the linear system S given by Equation (3). Notice that the $2k(n-k)\times n^2$ system in Equation (3) is likely to have full rank when the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ are uniformly random over $\mathbb{F}_q^{k\times(n-k)}$ (see Proposition 1). The main idea here is to adapt Algorithm 1 for finding a matrix $\boldsymbol{R} \in \mathsf{Mono}_n(\mathbb{F}_q)$ such that $\boldsymbol{G}' = \mathsf{SF}(\boldsymbol{GR})$ and $\boldsymbol{G}'' = \mathsf{SF}(\boldsymbol{GR}^{-1})$.

An ILCE instance takes the form of a 2-LCE (see Equation (4)) instance as $(\boldsymbol{G}_1 = \boldsymbol{G}, \boldsymbol{G}_1' = \boldsymbol{G}')$ and $(\boldsymbol{G}_2 = \boldsymbol{G}'', \boldsymbol{G}_2' = \boldsymbol{G})$. However, such a 2-LCE instance is not expected to be uniformly distributed in the set $\mathcal{S}_{i,j}$ (this is because $\boldsymbol{N}$ is determined by $\boldsymbol{M}$ in this scenario). Nevertheless, we can still model the ILCE instance similarly to the 2-LCE scenario, with the difference that the set $\mathcal{S}_{i,j}$ is replaced by another set. Consider the following set

$$\mathcal{S}'_{i,j} = \{\text{system given by Equation (3)} \mid \boldsymbol{M}', \boldsymbol{M}'' \in \mathbb{F}_q^{k\times(n-k)}\},$$

whose size is also upper bounded by $q^{2k(n-k)}$, and the following assumption.

Assumption 4 *Given the system S, for any indexes i and j, the reduced system $\mathsf{S}_{i,j}$ is distributed uniformly at random over $\mathcal{S}'_{i,j}$.*

Hence, we have the following theorem that gives conditions for ILCE to be solved in polynomial time.

Theorem 2. *Let $n, k \in \mathbb{N}$ be such that $k = n/2$ and $n > 2$, and let q be a prime such that*

$$q \geq \frac{3n-4}{n-2}.$$

Under Assumption 2 and Assumption 4, we have that there exists an algorithm that solves $\mathsf{ILCE}_{n,k,q}$ in polynomial time and memory with overwhelming probability.

Proof. It follows analogous arguments as in the proof of *Theorem* 1. □

4.3 Implications for self-dual codes, 2-PCE, and IPCE instantiations

Since PCE is a particular case of LCE, we obtain that the above results can be carried for IPCE and 2-PCE. One should note that, for random instances, PCE can be solved in polynomial time using the Support Splitting Algorithm [33].

Moreover, if we restrict to the case of trivial hulls, the algebraic approach proposed by Saeed [30] gives a practical algorithm to find the permutation between the two codes. However, cryptographic relevant instantiations of PCE concerning self-dual codes remain secure since both algorithms from [33,30] have an exponential running time concerning the hull dimension (for example the use of PCE is suggested in order to reduce signature sizes in [4,26]). Nevertheless, Theorems 1 and 2 are unaffected by the dimension of the hull.

Comparison with Saeed's algorithm: Due to the structure of the secret permutation, the strategy presented for LCE can be refined adding some linear equations leading to an alternative algorithm solving PCE in the case of trivial hulls. For a fixed permutation matrix $\boldsymbol{P}$, let $\boldsymbol{G}$ and $\boldsymbol{G}'$ such that $\boldsymbol{M} \in \mathbb{F}_q^{k\times(n-k)}$, $\boldsymbol{G} = (\boldsymbol{I}_k | \boldsymbol{M}) \in \mathbb{F}_q^{k\times n}$ and $\boldsymbol{G}' = \mathsf{SF}(\boldsymbol{GP})$. Notice that $\boldsymbol{P}^{-1} = \boldsymbol{P}^\top$ since $\boldsymbol{P} \in \mathsf{Perm}_n(\mathbb{F}_q)$, and thus $\boldsymbol{G} = \mathsf{SF}(\boldsymbol{G}'\boldsymbol{P}^{-1}) = \mathsf{SF}(\boldsymbol{G}'\boldsymbol{P}^\top)$ also holds.

Now, from [30, Corollary 3.2.13] follows that we can build the following system of $2k(n-k)$ linear equations in the n^2 variables $\boldsymbol{P}_{i,j}$:

$$\begin{bmatrix} (\boldsymbol{I}_k | \boldsymbol{M}) \otimes (-\boldsymbol{M}'^\top | \boldsymbol{I}_{n-k}) \\ (-\boldsymbol{M}^\top | \boldsymbol{I}_{n-k}) \otimes (\boldsymbol{I}_k | \boldsymbol{M}') \end{bmatrix} \mathsf{vec}(\boldsymbol{P}) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \tag{7}$$

One can observe that the system in Equation (7) has rank $2k(n-k)$ for codes with trivial hull. Additionally, one can add some extra linear equations to the systems Equation (7). The permutation $\boldsymbol{P}$ satisfies that the sum of each row (resp. column) must give one. That observation gives $2n$ sparse linear equations in the n^2 variables $\boldsymbol{P}_{i,j}$; however, we can only select $2n-1$ of them to ensure linear independence. In other words, we get $2n-1$ sparse linear equations represented in Equation (8) (see [30, Proposition 3.2.21]),

$$\begin{bmatrix} \boldsymbol{I}_n \otimes \boldsymbol{1}_n^\top \\ \boldsymbol{1}_n^\top \otimes \boldsymbol{I}_n \end{bmatrix} \mathsf{vec}(\boldsymbol{P}) = \begin{bmatrix} \boldsymbol{1}_n \\ \boldsymbol{1}_n \end{bmatrix}, \tag{8}$$

where $\boldsymbol{1}_n$ denotes the $1 \times n$ matrix with each entry equals to one.

In this way, we have enough independent linear equations with just one sample using the same strategy from Section 4.1. Like in [30], we get an algorithm that solves PCE in polynomial time for codes with trivial hull. More precisely, in this setting, our algorithm coincides with the algebraic algorithm proposed in [30, Sec. 3.7] for solving the PCE problem. In other words, our algorithm extends Saeed's algebraic method from solving PCE to solving 2-LCE and ILCE.

5 Experiments and Cryptographic Implications

5.1 Experiments

We support our results presented in Lemmas 2 and 3, Corollaries 1 and 3 and Theorems 1 and 2 with extensive experiments and simulations performed by

means of a SageMath [35] proof-of-concept implementation. All scripts are available in [12].

To better illustrate the impact of the results from Section 4, we start by giving a comparison between the estimated asymptotic complexities of LCE according to [2], and 2-LCE, and ILCE cording to Theorems 1 and 2. We follow the parameter sets from [2], ensuring 128, 192, and 256 security bits for LCE under the current most efficient algorithms for solving it. On the other hand, the estimations from Theorems 1 and 2 imply a security of 2-LCE and ILCE of around 60-70 security bits for the same parameter sets (see Table 1).

We perform intensive experiments to corroborate the lesser security bits for 2-LCE and ILCE compared to LCE. We take into consideration the following observation on the parameter set from [2]:

- 128 bits: $n = 252$ and $q = 127$ satisfies $q \approx n/2$,
- 192 bits: $n = 400$ and $q = 127$ satisfies $q \approx n/3$, and
- 256 bits: $n = 548$ and $q = 127$ satisfies $q \approx n/4$.

Since prior to this work it was assumed that the best method solving 2-LCE was directly solving LCE, we compare our results on 2-LCE with the security for LCE. On that basis, we center our experiments on the following parameter set: $n \in [32, 40, 48, 64, 72, 80, 96, 128]$, $k = n/2$, and $q \in [n/2, n/3, n/4, 127]$. Essentially, we tackle cases which are believed to provide security equivalent to $20-70$ bits in the case of LCE; such a complexity estimation is based on the analysis presented in [2]. Table 3 presents the time and memory measurements of our experiments performed on a 2.45 GHz AMD EPYC 7763 64-core Processor machine with 1T of RAM running Ubuntu 22.04.2 LTS. Our implementation employs parallelization per row; more precisely, it runs n processors in parallel, and the j^{th} processor has the task of computing the rank of $\mathsf{S}_{i,j}$. Consequently, that parallelization approach gives a factor of n times faster, but the memory increases by the same factor (i.e., it is n times bigger). We use the multiprocessing Python package for the parallelization and the tracemalloc Python module to measure the memory usage. In addition, for each parameters set considered, Table 3 reports a comparison of the expected number of variables in $\mathsf{S}_{\mathsf{red}}$ against the average obtained in our experiments. This comparison serves to illustrate that our experimental findings align with the analysis presented in the proof of Theorem 1.

To highlight the polynomial time (and polynomial memory) complexity, we interpolate the first and the last measurements for the memory and runtime columns from Table 3, with the corresponding asymptotic estimations $O(n^5)$ and $O(n^{1+2\omega})$ for $\omega = \log_2(7)$. Figure 1 illustrates that the experimental measurements fit well with the theoretical estimations.

It is worth highlighting that our analysis does not assume anything about the hull dimension of the codes. In fact, our results hold for any codes (even for self-dual codes), assuming the conditions from Theorem 1 hold. In particular,

n	q	Estimated LCE bit security	Expected vars in $\mathsf{S}_{\mathsf{red}}$	Measured vars in $\mathsf{S}_{\mathsf{red}}$	Memory	Runtime
32	7	20	198	170	1.00 GB	19.00 s
	11	22	132	116	1.00 GB	19.43 s
	17	23	94	91	1.00 GB	19.27 s
	127	29	40	39	1.03 GB	19.22 s
40	11	25	196	185	2.54 GB	46.89 s
	13	25	170	166	2.53 GB	47.16 s
	19	27	127	121	2.53 GB	50.37 s
	127	33	53	50	2.55 GB	44.97 s
48	13	28	236	224	5.31 GB	1 m 41 s
	17	29	189	184	5.31 GB	1 m 44 s
	23	31	151	151	5.32 GB	1 m 40 s
	127	37	66	64	5.33 GB	1 m 45 s
64	17	35	316	292	16.93 GB	7 m 10 s
	23	37	248	255	16.94 GB	7 m 6 s
	31	38	199	203	16.94 GB	7 m 15 s
	127	44	96	99	16.95 GB	6 m 38 s
72	19	39	356	348	27.18 GB	11 m 56 s
	23	40	305	298	27.18 GB	12 m 11 s
	37	42	214	224	27.18 GB	12 m 38 s
	127	47	113	115	27.18 GB	12 m 36 s
80	19	41	432	397	41.47 GB	20 m 46 s
	29	44	306	295	41.45 GB	20 m 59 s
	41	46	238	235	41.48 GB	21 m 4 s
	127	51	131	129	41.48 GB	20 m 41 s
96	23	48	511	486	86.07 GB	1 h 10 m
	31	51	400	368	86.09 GB	1 h 11 m
	47	54	295	294	86.07 GB	1 h 11 m
	127	58	169	172	86.07 GB	1 h 11 m
128	31	63	670	604	272.06 GB	6 h 8 m
	43	66	516	522	272.05 GB	5 h 2 m
	61	69	399	380	272.06 GB	4 h 40 m
	127	73	258	246	272.07 GB	5 h 3 m

Table 3: The data corresponds to the average of solving ten random 2-LCE instances. The fourth and the fifth columns present the theoretically expected number of variables in $\mathsf{S}_{\mathsf{red}}$ and the average of the observed values, respectively.

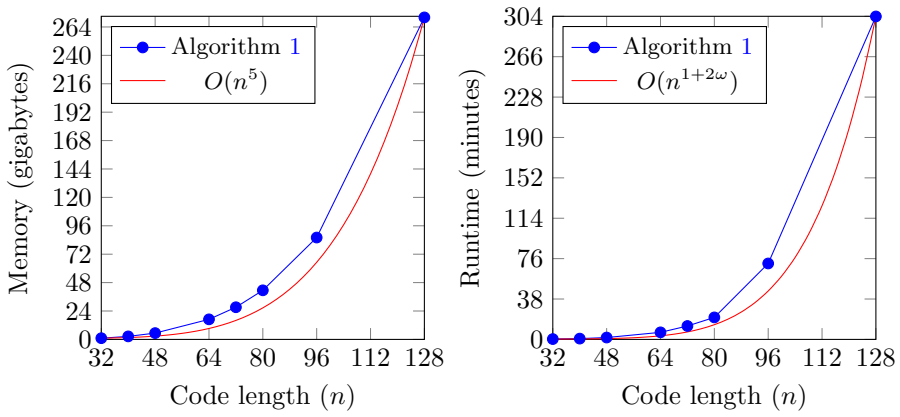

Fig. 1: The data corresponds with the parameter set with $q = 127$. The experiments employ parallelization per row, which increases the memory (and decreases the runtime) by a factor of n. In the above two plots, we interpolate the first and the last measurements for the memory and runtime columns from Table 3, with the corresponding asymptotic estimations $O(n^5)$ and $O(n^{1+2\omega})$ for $\omega = \log_2(7)$. In particular, the curves in red ink color correspond to $\mathsf{memory}(x) = ax^5 + b$ and $\mathsf{time}(x) = a'x^{1+2\log_2(7)} + b'$ for some real (positive) numbers $a, a', b, b' \in \mathbb{R}$.

we add some small examples (with $n \in \{16, 24, 28\}$, $k = 8$, and $q = 7$) in the provided implementation, showing that our results also work for self-dual codes.[7]

5.2 Cryptographic implications

On the impact on ILCE*-based linkable signatures:* In [4], the authors stated that if the ILCE problem were proved to be safe, all the necessary linkable properties would be satisfied, thus building a secure linkable ring signature scheme. Nevertheless, as a direct consequence of Section 4.2, we have that any linkable signature relying on the hardness of the ILCE problem is insecure, when the conditions from Theorem 2 are satisfied.

On the impact on 2-LCE*-based threshold signatures:* The authors of [6] introduced the 2-LCE problem in the group action framework [6, Problem 3] and emphasized constructions for 2-weakly pseudorandom scenarios. Specifically, they proposed a threshold signature whose distributed key generation algorithm is based on the conjectured 2-weakly pseudorandom group actions built on top of the LCE and MCE problems. Nevertheless, as another consequence of Theorem 1, we show that Definition 11 when instantiated with group action based on LCE does not achieve the pseudorandomness property as we can use Algorithm 1 to

[7] Such experiments test Algorithm 1 with the self-dual codes from [19,18]

recover the secret, which breaks the unpredictability as well as the pseudorandomness of the group action. Therefore, the threshold signature instantiations with LESS from [4, Sec. 5.3] become insecure when $k = n/2$.

Possible other implications: It is true that the analysis from Sections 3 and 4 centers on the systematic form, but all the analysis easily extends to any specific code form. As we take advantage of the use of the systematic form, our results could further benefit from the canonical forms analyzed in [26] and [15], and they could be applied even in those cases. In fact, the canonical forms allow the transmission of less information about the secret monomial $\boldsymbol{Q}$, encoding it in how the code is represented. Hence, fewer unknowns could be used to represent $\boldsymbol{Q}$. Further analysis is required, and we leave this as a future work.

Acknowledgments

Giuseppe D'Alconzo and Antonio J. Di Scala are members of GNSAGA of INdAM and of CrypTO, the group of Cryptography and Number Theory of the Politecnico di Torino.

The work of Antonio J. Di Scala was partially supported by the QUBIP project (https://www.qubip.eu), funded by the European Union under the Horizon Europe framework programme [grant agreement no. 101119746].

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union – NextGenerationEU.

We would also like to thank Ricardo Pontaza for his helpful insights and discussions which helped us improve the analysis of our techniques.

References

1. Alamati, N., De Feo, L., Montgomery, H., Patranabis, S.: Cryptographic group actions and applications. In: Moriai and Wang [22], pp. 411–439. https://doi.org/10.1007/978-3-030-64834-3_14
2. Baldi, M., Beckwith, A.B.L., Biasse, J.F., Esser, A., Gaj, K., Mohajerani, K., Pelosi, G., Persichetti, E., Saarinen, M.J.O., Santini, P., Wallace, R.: LESS (version 1.1). Tech. rep., National Institute of Standards and Technology (2023), https://www.less-project.com/
3. Bardet, M., Otmani, A., Saeed-Taha, M.: Permutation Code Equivalence is Not Harder Than Graph Isomorphism When Hulls Are Trivial. In: 2019 IEEE International Symposium on Information Theory (ISIT). pp. 2464–2468 (2019). https://doi.org/10.1109/ISIT.2019.8849855
4. Barenghi, A., Biasse, J., Ngo, T., Persichetti, E., Santini, P.: Advanced signature functionalities from the code equivalence problem. International Journal of Computer Mathematics: Computer Systems Theory **7**(2), 112–128 (2022), https://doi.org/10.1080/23799927.2022.2048206
5. Barenghi, A., Biasse, J.F., Persichetti, E., Santini, P.: On the computational hardness of the code equivalence problem in cryptography. Advances in Mathematics of Communications **17**(1), 23–55 (2023), https://doi.org/10.3934/amc.2022064

6. Battagliola, M., Borin, G., Meneghetti, A., Persichetti, E.: Cutting the GRASS: Threshold GRoup Action Signature Schemes. Cryptology ePrint Archive, Paper 2023/859 (2023), https://eprint.iacr.org/2023/859
7. Beullens, W.: Not enough LESS: An improved algorithm for solving code equivalence problems over $\mathbb{F}_q$. In: International Conference on Selected Areas in Cryptography. pp. 387–403. Springer (2020), https://doi.org/10.1007/978-3-030-81652-0_15
8. Beullens, W., Dobson, S., Katsumata, S., Lai, Y.F., Pintore, F.: Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 95–126. Springer, Heidelberg (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3_4
9. Beullens, W., Katsumata, S., Pintore, F.: Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In: Moriai and Wang [22], pp. 464–492. https://doi.org/10.1007/978-3-030-64834-3_16
10. Biasse, J.F., Micheli, G., Persichetti, E., Santini, P.: LESS is more: Code-based signatures without syndromes. In: Nitaj, A., Youssef, A.M. (eds.) AFRICACRYPT 20. LNCS, vol. 12174, pp. 45–65. Springer, Heidelberg (Jul 2020). https://doi.org/10.1007/978-3-030-51938-4_3
11. Budroni, A., Benčina, B., Chi-Domínguez, J.J., Kulkarni, M.: Properties of lattice isomorphism as a cryptographic group action. Cryptology ePrint Archive, Paper 2023/1093 (2023), https://eprint.iacr.org/2023/1093, https://eprint.iacr.org/2023/1093
12. Budroni, A., Chi-Domínguez, J.J., D'Alconzo, G., Di Scala, A.J., Kulkarni, M.: `relaxed-lce-algorithms`, available at https://github.com/JJChiDguez/relaxed-lce-algorithms.git
13. Chou, T., Niederhagen, R., Persichetti, E., Ran, L., Hajatiana, T., Reijnders, K., Samardjiska, S., Trimoska, M.: MEDS (version 1.1). Tech. rep., National Institute of Standards and Technology (2023), https://www.meds-pqc.org/
14. Chou, T., Niederhagen, R., Persichetti, E., Randrianarisoa, T.H., Reijnders, K., Samardjiska, S., Trimoska, M.: Take your MEDS: digital signatures from matrix code equivalence. In: Mrabet, N.E., Feo, L.D., Duquesne, S. (eds.) Progress in Cryptology - AFRICACRYPT 2023 - 14th International Conference on Cryptology in Africa, Sousse, Tunisia, July 19-21, 2023, Proceedings. Lecture Notes in Computer Science, vol. 14064, pp. 28–52. Springer (2023). https://doi.org/10.1007/978-3-031-37679-5_2
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. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291 (2006), https://eprint.iacr.org/2006/291
17. D'Alconzo, G., Di Scala, A.J.: Representations of Group Actions and their Applications in Cryptography. Cryptology ePrint Archive, Paper 2023/1247 (2023), https://eprint.iacr.org/2023/1247
18. Gaborit, P., Otmani, A.: `TABLES OF SELF-DUAL CODES`, available at https://www.unilim.fr/pages_perso/philippe.gaborit/SD/
19. Gaborit, P., Otmani, A.: Experimental constructions of self-dual codes. Finite Fields and Their Applications **9**(3), 372–394 (2003). https://doi.org/https://doi.org/10.1016/S1071-5797(03)00011-X
20. Kazmi, R.A.: Cryptography from post-quantum assumptions. Cryptology ePrint Archive, Report 2015/376 (2015), https://eprint.iacr.org/2015/376

21. Leon, J.: Computing automorphism groups of error-correcting codes. IEEE Transactions on Information Theory **28**(3), 496–511 (1982), https://doi.org/10.1109/TIT.1982.1056498
22. Moriai, S., Wang, H. (eds.): ASIACRYPT 2020, Part II, LNCS, vol. 12492. Springer, Heidelberg (Dec 2020)
23. National Institute of Standards and Technology: Post-Quantum Cryptography Standardization. https://csrc.nist.gov/projects/post-quantum-cryptography (2017)
24. National Institute of Standards and Technology: Post-quantum cryptography: Digital signature schemes. Round 1 Additional Signatures (2023), https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures
25. Persichetti, E., Randrianariso, T.H., Santini, P.: An attack on a non-interactive key exchange from code equivalence. Tatra Mountains Mathematical Publications **82**(2), 53–64 (2023), https://doi.org/10.2478/tmmp-2022-0018
26. Persichetti, E., Santini, P.: A New Formulation of the Linear Equivalence Problem and Shorter LESS Signatures. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023. pp. 351–378. Springer Nature Singapore, Singapore (2023), https://doi.org/10.1007/978-981-99-8739-9_12
27. Petrank, E., Roth, R.M.: Is code equivalence easy to decide? IEEE Transactions on Information Theory **43**(5), 1602–1604 (1997), https://doi.org/10.1109/18.623157
28. Pham, M.T.T., Duong, D.H., Li, Y., Susilo, W.: Threshold ring signature scheme from cryptographic group action. In: Zhang, M., Au, M.H., Zhang, Y. (eds.) Provable and Practical Security. pp. 207–227. Springer Nature Switzerland, Cham (2023), https://doi.org/10.1007/978-3-031-45513-1_12
29. Reijnders, K., Samardjiska, S., Trimoska, M.: Hardness Estimates of the Code Equivalence Problem in the Rank Metric. Designs, Codes and Cryptography pp. 1–30 (01 2024). https://doi.org/10.1007/s10623-023-01338-x
30. Saeed, M.A.: Algebraic Approach for Code Equivalence. Ph.D. thesis, Normandie Université, University of Khartoum, (2017), Available at https://theses.hal.science/tel-01678829v2
31. Santini, P., Baldi, M., Chiaraluce, F.: Computational hardness of the permuted kernel and subcode equivalence problems. Cryptology ePrint Archive, Report 2022/1749 (2022), https://eprint.iacr.org/2022/1749
32. Sendrier, N.: On the dimension of the hull. SIAM Journal on Discrete Mathematics **10**(2), 282–293 (1997), https://doi.org/10.1137/S0895480195294027
33. Sendrier, N.: Finding the permutation between equivalent linear codes: the support splitting algorithm. IEEE Transactions on Information Theory **46**(4), 1193–1203 (2000). https://doi.org/10.1109/18.850662
34. Sendrier, N., Simos, D.E.: The hardness of code equivalence over $\mathbb{F}_q$ and its application to code-based cryptography. In: Gaborit, P. (ed.) Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013. pp. 203–216. Springer Heidelberg (June 2013), https://doi.org/10.1007/978-3-642-38616-9_14
35. The Sage Developers: SageMath, the Sage Mathematics Software System (Version 9.8) (2023), `https://www.sagemath.org`