Rare structures in tensor graphs

Bermuda triangles for cryptosystems based on the Tensor Isomorphism problem

Lars Ran and Simona Samardjiska

Radboud University, Nijmegen, Netherlands
{lran,simonas}@cs.ru.nl

Abstract. Recently, there has been a lot of interest in improving the understanding of the practical hardness of the 3-Tensor Isomorphism (3-TI) problem, which, given two 3-tensors, asks for an isometry between the two. The current state-of-the-art for solving this problem is the algebraic algorithm of Ran et al. '23 and the graph-theoretic algorithm of Narayanan et al. '24 that have both slightly reduced the security of the signature schemes MEDS and ALTEQ, based on variants of the 3-TI problem (Matrix Code Equivalence (MCE) and Alternating Trilinear Form Equivalence (ATFE) respectively).
In this paper, we propose a new combined technique for solving the 3-TI problem. Our algorithm, as typically done in graph-based algorithms, looks for an invariant in the graphs of the isomorphic tensors that can be used to recover the secret isometry. However, contrary to usual combinatorial approaches, our approach is purely algebraic. We model the invariant as a system of non-linear equations and solve it. Using this modelling we are able to find very rare invariant objects in the graphs of the tensors — cycles of length 3 (triangles) — that exist with probability approximately $1/q$. For solving the system of non-linear equations we use Gröbner-basis techniques adapted to tri-graded polynomial rings. We analyze the algorithm theoretically, and we provide lower and upper bounds on its complexity. We further provide experimental support for our complexity claims. Finally, we describe two dedicated versions of our algorithm tailored to the specifics of the MCE and the ATFE problems.
The implications of our algorithm are improved cryptanalysis of both MEDS and ALTEQ for the cases when a triangle exists, i.e. in approximately $1/q$ of the cases. While for MEDS, we only marginally reduce the security compared to previous work, for ALTEQ our results are much more significant with at least 60 bits improvement compared to previous work for all security levels. For Level I parameters, our attack is practical, and we are able to recover the secret key in only 1501 seconds. The code is available for testing and verification of our results.

Keywords: matrix codes · trilinear form · algebraic cryptanalysis

This research has been supported by the Dutch government through the NWO grant OCNW.M.21.193 (ALPaQCa).

1 Introduction

In recent years, all main standardization bodies around the world (NIST [31], ISO [25], IETF [24]) have initiated processes for standardization of post-quantum cryptography as the most solid solution for securing our digital world against the quantum computer menace. Post-quantum cryptography is the common name for cryptography based on hard mathematical problems believed to be hard even for quantum computers. NIST's standardization process already produced drafts for standards — the winners Kyber [39], Dilithium [29], and SPHINCS$^+$ [23] are well on the way to be standardized under the names of ML-KEM, ML-DSA, and SLH-DSA. The situation for Falcon [36] is not even close to this phase, and we have only recently seen some activity towards a draft standard. On top of this slow progress, NIST reopened the call for digital signatures in search for alternatives not based on structured lattices that additionally have short signatures and fast verification [1].

In the additional round that has been running for a year, we see an abundance of UOV variants, MPC-in-the-head Fiat-Shamir signatures and a few Fiat-Shamir signatures based on equivalence problems. In particular, MEDS [13, 12] and ALTEQ [40, 8] are based on problems equivalent to the 3-tensor isomorphism problem (TI) and LESS [3] on a problem at most as difficult as 3-TI [15]. Informally speaking, given two 3-tensors, the 3-TI problem asks for the isomorphism, i.e. isometry, between them. The shape of the isometry depends on the specific types of tensors in question. In the case of MEDS, these are general 3-tensors, and the isometry is given as a triple of the general linear group. The objects in MEDS can also be seen as matrix codes and the corresponding problem as Matrix Code Equivalence (MCE), which is actually the definition used in the original description of MEDS. In the case of ALTEQ, the objects can be seen as alternating trilinear forms (they can also be represented as matrix codes of skew-symmetric matrices with additional structure) and the isometry can be described using a single element of the general linear group. In this form, the problem is known as Alternating Trilinear Form Equivalence (ATFE).

Both of these schemes follow the well-known construction of GMW [20] first defined for graph isomorphism. Interestingly, already in '96, it was instantiated by Patarin [33] for isomorphism of polynomials (Cubic-IP) whose version for homogeneous polynomials QMLE is also polynomial time equivalent to 3-TI, and thus also to MCE and ATFE [38]. Initially, the scheme did not get too much attention because of signature size inefficiency, but this changed due to an array of optimization techniques [16, 6, 7, 3] that were developed before the additional round submission deadline and resulted in the proposals of MEDS and ALTEQ. The question of practical hardness of these problems became interesting again, and the understanding of it significantly advanced as a result of these two submissions.

Related work. The first works analyzing problems from the TI class, considered the Isomorphism of polynomials. As one of the problems intrinsically

related to the security of ad-hoc multivariate schemes, it was analyzed in several works including [34, 19, 10]. An important observation from [19] is that the inhomogeneous version of the problem can be solved heuristically in polynomial time. Another one is that knowing a single (point, image) of the isomorphism is enough to solve the homogeneous version of the problem. Indeed, this pair can be used to transform the instance of the problem to an inhomogeneous one that can then be solved efficiently. Bouillaguet et al. [10] gave a nice graph-theoretic interpretation of this observation as a matching point between the graphs of the two sets of polynomials which was utilized in a collision based algorithm.

The algorithm of [10] remained the best known for the TI class for a long time (note that the term 'TI class' was coined only recently in [21]). With the involvement of MEDS and ALTEQ in the NIST standardization process a significant advancement in the understanding of the asymptotic and practical hardness of the related underlying problems was made. We have witnessed basically the development of two types of algorithms — graph-based and purely algebraic. We say 'purely algebraic' because even the graph-based algorithms can employ an algebraic step for collecting low-rank points. This is especially beneficial for the case of large fields where the enumeration of points is expensive.

The graph-based algorithms build upon the earlier mentioned work of Bouillaguet et al. [10]. The first improvement was given by Beullens [5] for ATFE and by Chou et al. [13] for MCE. In both cases, the improvement follows the Leon's algorithm for the Hamming metric [28, 4]. Currently, the state-of-the-art in graph-based algorithms is the work of Narayanan, Qiao and Tang [30], which presents two different algorithms for MCE and ATFE. The algorithm for ATFE is the one used for choosing the ALTEQ parameters from the specs. The one for MCE uses graph-walking techniques and notably, breaks the MEDS parameters submitted to NIST. As a result of this attack, very recently, at the NIST standardization conference [32], the MEDS team announced new parameters for all security levels.

The basic algebraic modeling for the MCE problem is somewhat of a folklore modeling known also from QMLE. The first non-trivial model was developed in [13] where coding theory relations were exploited. Later, in the MEDS specs [12] the modeling was improved by exploiting that 3-tensors give rise to 3 different matrix codes. The same approach was used in [37] but adapted to ATFE effectively reducing the security of the ALTEQ parameters. At the time of writing, the ALTEQ team has not announced new parameters as a reaction to the attack.

Our contribution. In this work, we propose a new graph-based algebraic technique for solving Tensor Isomorphism (TI) problems. In particular, we algebraically find a rare invariant in the graph of the two isomorphic tensors and use it to find the isometry between the tensors. By an invariant, we mean a property or an object that is a characteristic of the graph of the tensor, and that is not destroyed by an isometric transformation, i.e. it is an *invariant* with respect to

isometries. Previous graph-theoretic algorithms use invariants like vertex degree or long paths to recover the isometry, and they typically consist of two parts:

1. Searching for occurrences of the invariant property within the two graphs of the two isomorphic tensors and forming two lists L_1 and L_2 containing corresponding points.
2. Testing each pair $(a, b) \in L_1 \times L_2$ whether b is an image of a for some isometry. If this is the case, we have found the isometry.

Typically, for this to work, the testing needs to be efficient, often, polynomial of low degree. Since the graphs of the tensors are large, enumerating the entire graphs is typically not feasible, so the algorithms rely on invariants that are abundant and easy to find. Using a birthday argument, it can then be estimated how big the lists need to be in order to have a high likelihood of finding a collision in the lists.

The first contribution of our work is that we take a different approach to finding invariants — namely, an algebraic one. We model the invariant as a system of non-linear equations — once we solve the system, we get the invariant. Using this modelling, we are able to find very rare, (almost) unique objects in the entire graph that would otherwise take an exponential amount of time to find by enumeration, which is particularly prohibitive in big fields. In contrast, an algebraic approach is oblivious to the field size up to the cost of the arithmetic which scales only logarithmically.

An important step in the approach is determining the right invariant object to look for. Indeed, we need one that is rare and can be described using a relatively small number of variables, but at the same time, we can impose enough restrictions in the form of algebraic relations. We show that with probability of approximately $1/q$, where q is the field size, there exist in the graphs of the tensors (almost) unique invariant objects. These objects are small cycles of length 3, i.e. triangles. This is reminiscent of Beullens' attack [5] that exploits the fact that with probability $\approx 1/q$, there exists a unique point of rank 4 in the graph of an alternating trilinear form in dimension $n = 10$.

Our second contribution is developing an algorithm for exploiting the existence of triangles. Our algorithm consists of two parts:

- First, we find the triangles in the graphs of the tensors. Once we have the algebraic model of the triangles, we solve the system using Gröbner basis techniques. The system we obtain has a specific structure and in order to solve it we develop a solving algorithm and a machinery for estimating the costs by extending known techniques for bigraded polynomial rings to trigraded polynomial rings. We identify the syzygies characteristic of the modeling and conjecture the Hilbert series. We are able to precisely estimate the first degree fall of the system which we take as an indicator of the solving costs or lower bound of our algorithm. As usual, we take the solving degree as an upper bound.
- Using the found pair of matching triangles we find the secret isometry. For the second part of the algorithm, we show that from the pair of triangles

we can obtain linear relations in the algebraic modelling of [37, 12] which are enough to heuristically find the isometry in polynomial time. Thus, this second part is significantly cheaper than the first part.

We have fully implemented the algorithm in MAGMA [9] which is publicly available at:

https://github.com/LarsMath/tensor-triangles

For the ATFE problem, our code can be used to demonstrate practicality. The git further contains the MAGMA source code for all experiments which we performed to verify our theoretical claims and confirm our conjectures. All the numbers in this paper can be reproduced using the code in the git.

Our third contribution is applying our new algorithm to the two digital signature schemes MEDS and ALTEQ whose security relies on problems equivalent to 3-TI. For the submitted parameter of MEDS we only marginally improve upon [30] for keys exhibiting a triangle. Recently, the MEDS team increased the parameters as a result of the attack from [30], which, due to the small difference, are also secure from our attack. The results are shown in Table 6 and Table 7.

The impact for ALTEQ is much more dramatic as can be seen from Table 1. We improve the best previous attack by at least ≈ 60 bits. Even more, we are able to practically break Level I parameters (provided a triangle exists, which happens with a probability of $\approx 1/q$). The attack takes merely 1501 seconds. The implementation of the full attack is also available in the git repository above.

Table 1. The $\log_2$ complexity for solving ATFE (with probability $1/q$) in field operations. The parameters are taken from the ALTEQ specifications [8]

		[8]	[37]	**This work**		
	n	Specs	Best previous	Upper bound	First degree fall	practical
Level I	13	143	120	62	51	**1501 s**
Level III	20	219	165	108	96	
Level V	25	276	203	141	128	

Organization. The paper is organized as follows. Section 2 introduces the necessary preliminaries including an analysis of the tri-graded XL that we will use in our analysis. In Section 3 we recall the state-of-the-art algorithms for solving the TI problem. Our new algorithm is developed in Section 4 in which we also directly estimate the impact on MCE and MEDS. We apply our approach to ATFE and ALTEQ in Section 5. We conclude with a discussion on potential generalizations and future work in Section 6.

2 Preliminaries

Let us first establish some notation. We denote by $\mathbb{F}_q$ the finite field of q elements. By $\mathrm{GL}(V)$ we denote the general linear group on the vector space V. The space of $n \times m$ matrices over $\mathbb{F}_q$ are denoted by $\mathcal{M}_{n,m}(\mathbb{F}_q)$. We use bold letters to denote vectors $\mathbf{a}, \mathbf{c}, \mathbf{x}, \ldots$, and capital bold letters to denote matrices $\mathbf{A}, \mathbf{B}, \ldots$. The entries of a vector $\mathbf{a}$ are denoted by a_i and the entries of a matrix $\mathbf{A}$ are denoted by a_{ij}. We denote by $\mathbf{e}_1, \ldots, \mathbf{e}_n$ the vectors of the canonical basis of $\mathbb{F}_q^n$. By $\mathbb{P}(V)$ we denote the projective space associated to a vector space V. To distinguish between vectors from V and elements from $\mathbb{P}(V)$, we denote the latter by $\hat{\mathbf{v}}$. All the vector spaces that we consider are finite-dimensional.

2.1 The Tensor Isomorphism Problem (TI) and related problems.

Among cryptographic hardness assumptions based on equivalence, quite a few of them can be stated as a TI problem and especially as a 3-TI problem. For example, Matrix Code Equivalence (MCE), Alternating Trilinear Form Equivalence (ATFE), Quadratic Maps Linear Equivalence (QMLE) and Cubic Isomorphism of Polynomials (Cubic-IP) are all some form of 3-TI in disguise. In order to define 3-TI, we first need to define tensor isomorphisms.

Definition 1. *Given vector spaces U, V, and W over a field $\mathbb{F}_q$, a 3-tensor $\mathcal{C}$ over U, V, W is a trilinear map:*

$$\mathcal{C} : U \times V \times W \to \mathbb{F}_q.$$

We denote the space of tensors over U, V, W as $(U \otimes V \otimes W)^*$.

Note that, by tri-linearity, a 3-tensor $\mathcal{C}$ is completely determined by its values on the basis vectors of U, V, W, i.e. by $\mathcal{C}(\mathbf{e}_i^U, \mathbf{e}_j^V, \mathbf{e}_k^W)$ where $\mathbf{e}_i^U, 1 \leq i \leq \dim(U)$, $\mathbf{e}_j^V, 1 \leq j \leq \dim(V)$, and $\mathbf{e}_k^W, 1 \leq k \leq \dim(W)$ are basis vectors of U, V and W respectively. With this definition, we are ready to define the 3-TI problem.

Problem 1 (3-TI). Let U, V, and W be vector spaces over a field $\mathbb{F}_q$ and let $\mathcal{C}, \mathcal{D} \in (U \otimes V \otimes W)^*$ be two given 3-tensors. The 3-TI problem asks to find, if any exists, a triplet of matrices $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathrm{GL}(U) \times \mathrm{GL}(V) \times \mathrm{GL}(W)$ such that:

$$\mathcal{C}(\mathbf{Au}, \mathbf{Bv}, \mathbf{Cw}) = \mathcal{D}(\mathbf{u}, \mathbf{v}, \mathbf{w}) \quad \forall\ \mathbf{u} \in U, \mathbf{v} \in V, \mathbf{w} \in W.$$

Let us compare this to MCE.

Problem 2 (MCE). Let $n, m, k \geq 2$. Given matrices $C_1, \ldots, C_k, D_1, \ldots, D_k \in \mathcal{M}_{n,m}(\mathbb{F}_q)$ the MCE problem asks to find, if any exists, a pair of matrices $\mathbf{A}, \mathbf{B} \in \mathrm{GL}(\mathbb{F}_q^n) \times \mathrm{GL}(\mathbb{F}_q^m)$ such that:

$$\langle \mathbf{A}^\top C_1 \mathbf{B}, \ldots, \mathbf{A}^\top C_k \mathbf{B} \rangle = \langle D_1, \ldots, D_k \rangle.$$

Note the similarity between 3-TI and MCE. In fact, these problems are equivalent [22] and $\mathbf{C}$ in 3-TI is exactly the change of basis from $\langle \mathbf{A}^\top C_1 \mathbf{B}, \ldots, \mathbf{A}^\top C_k \mathbf{B} \rangle$ to $\langle D_1, \ldots, D_k \rangle$. Other variants of 3-TI can be obtained by limiting the space of tensors $(U \otimes V \otimes W)^*$. For example, taking $U = V = W$, we can impose that the tensors from $(V \otimes V \otimes V)^*$ must be alternating, i.e. we ask:

$$\mathcal{C}(\mathbf{v}, \mathbf{v}, \mathbf{w}) = \mathcal{C}(\mathbf{v}, \mathbf{w}, \mathbf{v}) = \mathcal{C}(\mathbf{w}, \mathbf{v}, \mathbf{v}) = 0 \text{ for all } \mathbf{v}, \mathbf{w} \in V.$$

We denote the space of tensors that satisfy this constraint by $(\bigwedge^3 V)^*$. Now we can state the ATFE problem similarly as 3-TI:

*Problem 3 (*ATFE*).* Let V be a vector space over a field $\mathbb{F}_q$ and let $\phi, \psi \in (\bigwedge^3 V)^*$ be two given alternating 3-tensors. The ATFE problem asks to find, if it exists, a matrix $\mathbf{A} \in \mathrm{GL}(V)$ such that:

$$\phi(\mathbf{Au}, \mathbf{Av}, \mathbf{Aw}) = \psi(\mathbf{u}, \mathbf{v}, \mathbf{w}) \quad \forall\ \mathbf{u}, \mathbf{v}, \mathbf{w} \in V.$$

An immediate consequence of the alternating property is that compared to 3-TI we now need $\mathbf{A} = \mathbf{B} = \mathbf{C}$. The other two problems, QMLE and Cubic-IP, come from considering partly symmetric and fully symmetric 3-tensors.

Remark 1. The two signature schemes MEDS and ALTEQ do not assume any further structure on the codes (alternating forms) and isometries they use. Hence, we too, will be interested in unstructured (random) variants of these problems.

2.2 Graphs associated with tensors

A strong invariant of a 3-tensor is the graph associated with it. The points of this graph are given by elements in the disjoint union of the projective vector spaces. Its edges consist of the pairs of points on which the tensor completely vanishes. To define this we will use the following shorthand notation. Given a tensor $\mathcal{C} : U \times V \times W \to \mathbb{F}_q$, the statement $\mathcal{C}(\mathbf{u}, \mathbf{v}, -) = 0$ denotes that $\mathcal{C}(\mathbf{u}, \mathbf{v}, \mathbf{x}) = 0$ for all $\mathbf{x} \in W$. We define $\mathcal{C}(\mathbf{u}, -, \mathbf{w}) = 0$ and $\mathcal{C}(-, \mathbf{v}, \mathbf{w}) = 0$ similarly. Note that the statement $\mathcal{C}(\mathbf{u}, \mathbf{v}, -) = 0$ is independent of scaling of $\mathbf{u}$ and $\mathbf{v}$, so we will use the notation $\mathcal{C}(\hat{\mathbf{u}}, \hat{\mathbf{v}}, -) = 0$ as well.

Definition 2. *Let $\mathcal{C} : U \times V \times W \to \mathbb{F}_q$ be a tensor. The graph $\mathcal{G}(\mathcal{C}) = (\mathcal{V}_\mathcal{C}, \mathcal{E}_\mathcal{C})$ is defined as follows:*

$$\begin{aligned}
\mathcal{V}_\mathcal{C} &= \mathbb{P}(U) \uplus \mathbb{P}(V) \uplus \mathbb{P}(W) \\
\mathcal{E}_\mathcal{C} &= \{(\hat{\mathbf{u}}, \hat{\mathbf{v}}) \in \mathbb{P}(U) \times \mathbb{P}(V) \mid \mathcal{C}(\hat{\mathbf{u}}, \hat{\mathbf{v}}, -) = 0\} \\
&\quad \cup \{(\hat{\mathbf{u}}, \hat{\mathbf{w}}) \in \mathbb{P}(U) \times \mathbb{P}(W) \mid \mathcal{C}(\hat{\mathbf{u}}, -, \hat{\mathbf{w}}) = 0\} \\
&\quad \cup \{(\hat{\mathbf{v}}, \hat{\mathbf{w}}) \in \mathbb{P}(V) \times \mathbb{P}(W) \mid \mathcal{C}(-, \hat{\mathbf{v}}, \hat{\mathbf{w}}) = 0\}
\end{aligned}$$

From the definition, we can immediately see that these graphs are tripartite with vertex partition $(\mathbb{P}(U), \mathbb{P}(V), \mathbb{P}(W))$. A particularly useful feature about these associated graphs is that this construction is functorial. In other words, suppose

we have two 3-tensors $\mathcal{C} : U \times V \times W \to \mathbb{F}_q$ and $\mathcal{D} : U' \times V' \times W' \to \mathbb{F}_q$ and a transformation $\mathbf{A} : U' \to U$, $\mathbf{B} : V' \to V$, $\mathbf{C} : W' \to W$ between them, such that:

$$\mathcal{D} = \mathcal{C} \circ (\mathbf{A}, \mathbf{B}, \mathbf{C})$$

Then we obtain a map on graphs by applying the matrices $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ on the vertices. It is then easy to check that this map indeed maps edges to edges. The map $(\mathbf{A}, \mathbf{B}, \mathbf{C}) : \mathcal{G}(\mathcal{D}) \to \mathcal{G}(\mathcal{C})$ is given by:

$$(\mathbf{A}, \mathbf{B}, \mathbf{C}) : v \mapsto \begin{cases} \mathbf{A}v & \text{if } v \in \mathbb{P}(V') \\ \mathbf{B}v & \text{if } v \in \mathbb{P}(W') \\ \mathbf{C}v & \text{if } v \in \mathbb{P}(U'). \end{cases}$$

The important takeaway here is that isomorphisms of 3-tensors, elements of $\mathrm{GL}(U) \times \mathrm{GL}(V) \times \mathrm{GL}(W)$, induce isomorphisms on the associated graphs.

Graphs associated with structured tensors When there is a structure present in the tensor, for example, (anti-)symmetry, then the above construction has a lot of superfluous points and edges. For example, consider the alternating trilinear form $\phi : V \times V \times V \to \mathbb{F}_q$, and two elements $\mathbf{v}, \mathbf{v}' \in \mathbb{P}(V)$ such that $\phi(\mathbf{v}, \mathbf{v}', -) = 0$. Then by anti-symmetry we also have $\phi(\mathbf{v}, -, \mathbf{v}') = 0$, and similarly for any other permutation of $\mathbf{v}, \mathbf{v}'$, and $-$. In other words, without labels, it is impossible to distinguish whether $\mathbf{v}$ lies in the first, second or third term of $\mathbb{P}(V) \uplus \mathbb{P}(V) \uplus \mathbb{P}(V)$. More precisely, there are 6 graph automorphisms permuting the terms in the disjoint union. Therefore, instead, we consider the vertex set of the graph to be $\mathcal{V}_\phi = \mathbb{P}(V)$. For other types of symmetries, we consider similar quotient graphs.

2.3 Gröbner basis algorithms

To obtain solutions to the polynomial systems we encounter below, we will use Gröbner basis algorithms [11, 27, 18, 14, 2, 26]. These algorithms come in all kinds of variants, but they all share the same underlying idea. The goal is to gather enough algebraic combinations of the initial polynomials such that linear combinations of these reduce the problem to a linear system. Generally, it is hard to say how many algebraic combinations one needs to do this. However, heuristically, we can make some approximations based on assumptions about (structured) random systems.

Macaulay matrices Let us consider a polynomial system $\mathcal{F} = (f_1, \ldots, f_m) \subset \mathbb{F}_q[x_1, \ldots, x_n] = \mathcal{R}$. We are interested in the spaces of algebraic combinations of $\mathcal{F}$ up to a certain degree:

$$I_{\leq d} := \mathrm{span}_{\mathbb{F}_q}\{u \cdot f \mid u \in \mathcal{R},\ f \in \mathcal{F},\ \deg(uf) \leq d\}.$$

If $I_{\leq d}$ contains n linear independent linear elements, we reduced our problem to a linear system. If it contains the element $1_{\mathcal{R}}$ then we know that our system does not have a solution.

Algorithmically, to find the solution, we build the Macaulay matrix $\mathcal{M}_d$ of degree d. This matrix has its columns labeled by the monomials up to degree d in $\mathcal{R}$. The rows are given by the products uf_i where $\deg(u) \leq d - \deg(f)$. The entry at (uf_i, m) is then the coefficient of the monomial m in the product uf_i.

Note that this is generally a sparse matrix, as multiplying with a monomial does not change the amount of terms. Also note that the rowspace of $\mathcal{M}_d$ is isomorphic to $I_{\leq d}$. Therefore, to check that $I_{\leq d}$ contains linear relations we can echolonize $\mathcal{M}_d$ and see if we end up with rows having only linear monomials.

The question now is how high d needs to be for this to happen. To estimate this we need to assume that we can exactly predict the amount of linear dependencies, called syzygies, among the rows of $\mathcal{M}_d$. Even though this assumption might look strong, in practice, for random systems, the rank of $\mathcal{M}_d$ neatly follows a pattern for different n, m, d. Then, we will obtain a solution exactly when we predict $\mathcal{M}_d$ to be of corank 1. We will call the degree in which this happens the solving degree d_{solv}. To then extract the linear system we need to echelonize this matrix which has $\binom{n+d_{solv}}{d_{solv}}$ columns. Since this matrix is sparse we can use the Block-Wiedemann algorithm to obtain a complexity of:

$$\rho \cdot \binom{n + d_{solv}}{d_{solv}}^2$$

where ρ is the density of the matrix and is equal to the maximum number of terms among the polynomials in $\mathcal{F}$.

Degree falls Instead of choosing d such that we can completely linearize, some algorithms, like F4 and MutantXL, take a different strategy. In these cases we are looking for *degree falls* which happen when the vector space

$$I_d := \text{span}_{\mathbb{F}_q}\{u \cdot f \mid u \in \mathcal{R},\ f \in \mathcal{F},\ \deg(uf) = d\}$$

has elements of degree smaller than d. The degree at which this happens is called the first fall degree d_{ff}. Note that this can only happen in inhomogeneous systems. Then, we can extend the Macaulay matrix in degree d_{ff} by adding these elements multiplied by linear monomials. In this way, the system might be solved in lower degree.

The complexity analysis of these algorithms and how they behave after finding degree falls is much more complex. Still, it is not uncommon to take the complexity of finding degree falls, $\rho \cdot \binom{n+d_{ff}-1}{d_{ff}}^2$, as an estimator for the complexity of solving the system.

Tri-graded XL Due to the structure of the polynomial systems that we will consider, a tri-graded polynomial ring is often a better fit. These are rings

$$\mathcal{R} = \mathbb{F}_q[x_1, \ldots, x_n, y_1, \ldots, y_m, z_1, \ldots, z_k]$$

where the grading is determined by the degree in each of the three variables sets $\{x_1, \ldots, x_n\}, \{y_1, \ldots, y_m\}, \{z_1, \ldots, z_k\}$. In this ring, we index monomials by $\alpha \in \mathbb{Z}^n_{\geq 0}, \beta \in \mathbb{Z}^m_{\geq 0}, \gamma \in \mathbb{Z}^k_{\geq 0}$ and we use the following notation:

$$\mathbf{x}^\alpha \mathbf{y}^\beta \mathbf{z}^\gamma = \prod_{i=1}^{n} x_i^{\alpha_i} \prod_{i=1}^{m} y_i^{\beta_i} \prod_{i=1}^{k} z_i^{\gamma_i}$$

for a monomial of tri-degree $(\sum_i \alpha_i, \sum_i \beta_i, \sum_i \gamma_i)$.

A polynomial is tri-homogeneous of tri-degree (d_x, d_y, d_z) if all its terms share the same tri-degree. As an example, a tri-homogeneous polynomial in degree $(1, 1, 1)$ is exactly a trilinear form! We sometimes drop the tri prefix and use homogeneous and degree if it is clear from the context what is meant.

Given two tri-degrees $\mathbf{d} = (d_x, d_y, d_z)$ and $\mathbf{d}' = (d'_x, d'_y, d'_z)$, we define the partial order $\mathbf{d} \succeq \mathbf{d}'$ if $(d_x \geq d_x) \wedge (d_y \geq d'_y) \wedge (d_z \geq d'_z)$. If, among the degrees of the monomials of a polynomial f, there is a greatest tri-degree, then we call this degree the top degree of f.

Just as for singly graded systems, we can define

$$I_{\preceq \mathbf{d}} := \mathrm{span}_{\mathbb{F}_q}\{u \cdot f \mid u \in \mathcal{R},\ f \in \mathcal{F},\ \deg(uf) \preceq \mathbf{d}\}$$

and corresponding Macaulay matrices. Then, instead of a single lowest degree for which this is linearizable, we may have multiple "lowest" tri-degrees. Let us call a tri-degree *admissible* if $I_{\preceq \mathbf{d}}$ contains $n + m + k$ linear relations or the element 1. Let us denote by $\mathfrak{D}_{solv}$ the set of all admissible tri-degrees for $\mathcal{F}$. Then the complexity of linearizing can be given by:

$$\rho \cdot \min_{\mathbf{d} \in \mathfrak{D}_{solv}} \left(\binom{n + d_x}{d_x} \binom{m + d_y}{d_y} \binom{k + d_z}{d_z} \right)^2.$$

Just as before we can define the first fall degree. However, since, again, there might be multiple tri-degrees for which this is the case, we write the complexity as:

$$\rho \cdot \min_{\mathbf{d} \in \mathfrak{D}_{ff}} \left(\binom{n + d_x - 1}{d_x} \binom{m + d_y - 1}{d_y} \binom{k + d_z - 1}{d_z} \right)^2.$$

Remark 2. Estimating the impact of degree falls in tri-graded systems is even more complex than in singly-graded systems. The resulting polynomials do not have a unique top degree anymore hence usual counting strategies fail. However, it is still clear that degree falls can only speed up computation.

3 Algorithms for solving TI

The algorithms against TI build upon relatively old algorithms against the Isomorphism of polynomials [35, 10]. Here we review the state-of-the-art.

3.1 Graph-theoretic algorithm of Narayanan et al. [30]

The work of Narayanan, Qiao and Tang [30] builds on top of the works of [10, 5] and presents two different algorithms for MCE and ATFE. On a high level, both algorithms follow the structure of the graph-theoretic algorithm of [10] and can be described as follows:

- Form the graphs of the two isomorphic tensors $\mathcal{C}$ and $\mathcal{D}$ (matrix codes or alternating trilinear forms) as described in Section 2.2.
- Collect points from the two graphs into two lists $L_{\mathcal{C}}$ and $L_{\mathcal{D}}$ such that the bilinear forms obtained by fixing the tensor at the given point is of specific rank R. Due to the birthday paradox, the size of the lists needs to be only a square root of all points satisfying the rank R property.
- For each element in $L_{\mathcal{C}}$ and $L_{\mathcal{D}}$ apply some sort of distinguishing function f constructed in such a way that $f(a) = f(b)$ only if (a, b) is a collision pair for the unknown isometry
- Use the collision pair (a, b) for which $f(a) = f(b)$ to recover the isometry

The difference from [10] which is also the main novelty of this approach, is the formulation of the distinguishing function. Previously this function was just solving an inhomogeneous QMLE for every possible pair in $L_{\mathcal{C}} \times L_{\mathcal{D}}$. With this approach, there is no need to test each pair, but one can make full use of the birthday paradox and just look for a collision in the lists.

Besides this novel global invariant (as called in [30]), there are also some new interesting techniques introduced. For example in the algorithm for MCE, in order to construct the distinguishing function, the authors use a graph walking technique to efficiently find a path of length $3n$. Until this work, it was an open question of how to use graph walking techniques to solve MCE. Previously, graph walking has been efficiently used against ATFE by Beullens in [5].

In some sense, looking for distinguishing cycles in the two graphs was an inspiration for our work. However, we take one more shortcut, and look for a unique cycle in the two graphs.

3.2 Purely algebraic algorithms for solving TI

The described graph-theoretic algorithms in the previous subsection crucially rely on algebraic techniques. For example, the inhomogeneous efficient solver is purely algebraic, and most likely, can't be replaced by an equally efficient combinatorial procedure. Also, the enumeration part of low-rank codewords can be done much more efficiently by solving the MinRank problem algebraically.

There are, however, also purely algebraic approaches developed in [13, 12] for MCE and [40, 8, 37] for the related ATFE. Here the solution is modelled as part of a solution to a system of equations.

Taking the coefficients of the matrices $\mathbf{A}, \mathbf{A}'$, $\mathbf{B}, \mathbf{B}'$, and $\mathbf{C}, \mathbf{C}'$ as unknowns, we can build the following system of equations:

$$\begin{cases} \mathbf{A}\mathbf{A}' = \mathbf{I}_n = \mathbf{A}'\mathbf{A} \\ \mathbf{B}\mathbf{B}' = \mathbf{I}_m = \mathbf{B}'\mathbf{B} \\ \mathbf{C}\mathbf{C}' = \mathbf{I}_k = \mathbf{C}'\mathbf{C} \end{cases} \tag{1}$$

$$\begin{cases} \mathcal{C}(\mathbf{A}\mathbf{x}, \mathbf{B}\mathbf{y}, \mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{y}, \mathbf{C}'\mathbf{z}) \\ \mathcal{C}(\mathbf{A}\mathbf{x}, \mathbf{y}, \mathbf{C}\mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{B}'\mathbf{y}, \mathbf{z}) \\ \mathcal{C}(\mathbf{x}, \mathbf{B}\mathbf{y}, \mathbf{C}\mathbf{z}) = \mathcal{D}(\mathbf{A}'\mathbf{x}, \mathbf{y}, \mathbf{z}) \end{cases} \quad \forall\, \mathbf{x} \in \mathbb{F}_q^n, \mathbf{y} \in \mathbb{F}_q^m, \mathbf{z} \in \mathbb{F}_q^k \tag{2}$$

$$\begin{cases} \mathcal{C}(\mathbf{A}\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{B}'\mathbf{y}, \mathbf{C}'\mathbf{z}) \\ \mathcal{C}(\mathbf{x}, \mathbf{B}\mathbf{y}, \mathbf{z}) = \mathcal{D}(\mathbf{A}'\mathbf{x}, \mathbf{y}, \mathbf{C}'\mathbf{z}) \\ \mathcal{C}(\mathbf{x}, \mathbf{y}, \mathbf{C}\mathbf{z}) = \mathcal{D}(\mathbf{A}'\mathbf{x}, \mathbf{B}'\mathbf{y}, \mathbf{z}) \end{cases} \quad \forall\, \mathbf{x} \in \mathbb{F}_q^n, \mathbf{y} \in \mathbb{F}_q^m, \mathbf{z} \in \mathbb{F}_q^k \tag{3}$$

This is an immense quadratic system of $6nmk + 2(n^2 + m^2 + k^2)$ equations in $2(n^2 + m^2 + k^2)$ variables. However, there is a lot of superfluity in this system. By construction, the syzygy module in degree 3 is quite big.

Solving the system "as is" has a high complexity, the number one reason being that it has a huge amount of variables. So instead, the current best purely algebraic algorithm from [12] looks at a subsystem generated as follows.

Consider the equations in (2). Here the left-hand side is quadratic in $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and the right-hand side is linear in $\mathbf{A}', \mathbf{B}', \mathbf{C}'$. This means we can take linear combinations to eliminate $\mathbf{A}', \mathbf{B}', \mathbf{C}'$. Then we end up with a system of $3nmk - (n^2 + m^2 + k^2)$ equations, quadratic in $n^2 + m^2 + k^2$ variables, $\mathbf{A}, \mathbf{B}, \mathbf{C}$. These equations are tri-homogeneous and hence the following Hilbert series is conjectured:

$$\mathcal{H}(r, s, t) = \frac{(1 - rs)^{nmk - k^2}(1 - rt)^{nmk - m^2}(1 - st)^{nmk - n^2}(1 - rst)^{-\alpha}}{(1 - r)^{n^2}(1 - s)^{m^2}(1 - t)^{k^2}}.$$

Here $\alpha = 2nmk - n^2 - m^2 - k^2 + 1$ is the dimension of the syzygy module in tri-degree $(1, 1, 1)$.

Specializing to ATFE For ATFE a similar system is obtained in the unknown $\mathbf{A}$ and $\mathbf{A}'$:

$$\begin{cases} \mathbf{A}\mathbf{A}' = \mathbf{I}_n = \mathbf{A}'\mathbf{A} \\ \phi(\mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{y}, \mathbf{z}) = \psi(\mathbf{x}, \mathbf{y}, \mathbf{A}'\mathbf{z}) \\ \phi(\mathbf{A}\mathbf{x}, \mathbf{y}, \mathbf{z}) = \psi(\mathbf{x}, \mathbf{A}'\mathbf{y}, \mathbf{A}'\mathbf{z}) \end{cases} \quad \forall\, \mathbf{x} \in \mathbb{F}_q^n$$

This is a system of $2n\binom{n}{2} + 2n^2$ quadratic equations in $2n^2$ variables. However, the same trick of eliminating $\mathbf{A}'$ can be applied to obtain a system of $n\left(\binom{n}{2} - n\right)$ quadratic equations in n^2 variables. The following Hilbert series is conjectured for this system:

$$\mathcal{H}(t) = \frac{(1 - t^2)^{n\left(\binom{n}{2} - n\right)}(1 - t^3)^{-\beta}}{(1 - t)^{n^2}}.$$

Here $\beta = n\left(\binom{n}{2} - n\right) - \binom{n}{3} + 1$ is the dimension of the syzygy module in degree 3. This Hilbert series was experimentally verified for small n and d in [37] and seems to hold for n not too small.

4 A hybrid algorithm for solving TI

As described in Section 2 and seen in Section 3 a strong invariant of a tensor is its graph. Most notably, any isometry between two tensors maps substructures in one graph to substructures in the second that have some common property. These can be, for example, points of a certain degree or an edge between two points of certain degrees. Identifying the substructures that are mapped one onto another (we will call them "collisions") can be turned into an algorithm for finding the isometry.

However, two costly factors arise in such algorithms. The first is finding all (or a fraction of) such substructures in the graphs and the second is testing for collisions in some enumerative way. This generally involves a balancing act. Often, substructures of which there are few are hard to find and substructures that are easy to find are far from unique. The latter case becomes particularly prohibitive for big fields, since usually, the amount of substructures is highly dependent on the field size. So even though finding is easy, enumerating all candidates becomes the bottleneck. In contrast, when a substructure appears only once in a graph, finding it in both graphs immediately leads to a collision.

Our attack follows this general approach, but instead of exploiting abundant substructures, we strive to find such that are rare or unique in a graph. While we were not able to find a substructure that occurs exactly once, we found a substructure that occurs exactly once with sufficiently large probability of $\approx 1/q$ — we call this substructure a triangle. Then, if a triangle appears in the graph of a specific tensor, by invariance, it occurs in every graph of tensors in its orbit. Also, given that a triangle consists of three points, whenever we find such triangles, we get a three-point collision. As we will see, this three-point collision is enough to solve the corresponding TI problem efficiently.

4.1 Triangles

The rare structure that we consider is a triplet of points that form a triangle in the graph associated to the tensor. To define it formally, denote the set of all triplets of points in the graph by $\mathbb{T}(U, V, W) = \mathbb{P}(U) \times \mathbb{P}(V) \times \mathbb{P}(W)$. We have:

Definition 3. *Let $\mathcal{C} : U \times V \times W \to \mathbb{F}_q$ be a 3-tensor. We call a triplet of points $(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\mathbf{w}}) \in \mathbb{T}(U, V, W)$ a triangle for $\mathcal{C}$ if it holds that:*

$$\mathcal{C}(\hat{\mathbf{u}}, \hat{\mathbf{v}}, -) = 0, \qquad \mathcal{C}(\hat{\mathbf{u}}, -, \hat{\mathbf{w}}) = 0, \qquad \mathcal{C}(-, \hat{\mathbf{v}}, \hat{\mathbf{w}}) = 0.$$

Initially, it might seem that such a simple structure should be abundant in the graphs. However, it turns out that this is not the case at all.

Lemma 1. *Let U, V and W be vector spaces over $\mathbb{F}_q$. Then the expectation value for the amount of triangles is equal to:*

$$\mathbb{E}_{\mathcal{C}\in(U\otimes V\otimes W)^*}(|\{T \in \mathbb{T}(U,V,W) \mid T \text{ is a triangle for } \mathcal{C}\}|) = q^{-1}.$$

Proof. We follow a proof technique similar to [5]. We first find the probability that an arbitrary triplet $(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\mathbf{w}}) \in \mathbb{T}(U,V,W)$ is a triangle, then, we use linearity of the expectation value to get the desired result. Given a point $T = (\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\mathbf{w}})$ we pick a non-zero representative $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ and extend it to a base. We obtain $U = \langle \mathbf{u}, \mathbf{u}_2, \ldots, \mathbf{u}_n \rangle$, $V = \langle \mathbf{v}, \mathbf{v}_2, \ldots, \mathbf{v}_m \rangle$, and $W = \langle \mathbf{w}, \mathbf{w}_2, \ldots, \mathbf{w}_k \rangle$.

When we consider a tensor $\mathcal{C}$ with respect to this basis we obtain a tensor $\mathcal{C}'$. If we pick $\mathcal{C}$ uniformly at random then $\mathcal{C}'$ will be uniformly random as well. Then, the condition that T is a triangle for $\mathcal{C}$ is equivalent to the condition that $(\mathbf{e}_1^U, \mathbf{e}_1^V, \mathbf{e}_1^W)$ is a triangle for $\mathcal{C}'$. This latter condition can be reformulated as

$$\mathcal{C}(\mathbf{e}_{i_1}^U, \mathbf{e}_1^V, \mathbf{e}_1^W) = \mathcal{C}(\mathbf{e}_1^U, \mathbf{e}_{i_2}^V, \mathbf{e}_1^W) = \mathcal{C}(\mathbf{e}_1^U, \mathbf{e}_1^V, \mathbf{e}_{i_3}^W) = 0$$

for all $1 \leq i_1 \leq n, 1 \leq i_2 \leq m, 1 \leq i_3 \leq k$. See Figure 1 for a depiction. Since

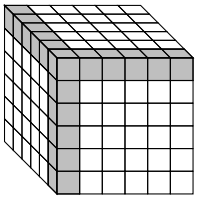

Fig. 1. For a 3-tensor $\mathcal{C}$, the coefficients in the light gray positions should be zero in order for $(\mathbf{e}_1^U, \mathbf{e}_1^V, \mathbf{e}_1^W)$ to be a triangle.

these values are all independently uniform in $\mathbb{F}_q$ we obtain that:

$$\mathbb{P}_{\mathcal{C}\in(U\otimes V\otimes W)^*}(T \text{ is a triangle for } \mathcal{C}) = q^{-(n+m+k-2)}.$$

Now by simple linearity of the expectation value we obtain:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{C}\in(U\otimes V\otimes W)^*}(|\{T \in \mathbb{T}(U,V,W) \mid T \text{ is a triangle for } \mathcal{C}\}|) \\
&\qquad = \mathbb{P}_{\mathcal{C}\in(U\otimes V\otimes W)^*, T\in\mathbb{T}(U,V,W)}(T \text{ is a triangle}) \cdot |\mathbb{T}(U,V,W)| \\
&\qquad = q^{-(n+m+k-2)} \cdot q^{n-1} \cdot q^{m-1} \cdot q^{k-1} \\
&\qquad = q^{-1}.
\end{aligned}$$

Of course, the expectation value does not directly give us the probability that a triangle occurs, let alone a unique one. Therefore, we also show a lower bound for this probability.

Corollary 1. *Given vector spaces U, V, and W over $\mathbb{F}_q$ of dimensions n, m, and k such that $n \geq m \geq k \geq 3$ and $m + k - 2 \geq n$. Then:*

$$\mathbb{P}_{\mathcal{C}\in(U\otimes V\otimes W)^*}(\mathcal{C} \text{ has a unique triangle}) \geq q^{-1} - \mathcal{O}(q^{-2}).$$

Proof. See Appendix A.1.

To support this result we did a Monte-Carlo simulation of the amount of tensors with (unique) triangles using MAGMA's `VarietySize`, the results can be found in Table 2. For practical reasons, triangles of which the first $\mathbf{u}, \mathbf{v}$, or $\mathbf{w}$ coordinate is 0 were not found, so actual numbers might be higher.

Table 2. Fraction of tensors where a triangle was found on 10000 experiments. We also report the times the triangle was unique.

q	(n, m, k)	Predicted	Found	Unique
13	(5,5,5)	769	739	713
	(6,5,5)	769	755	725
	(6,6,6)	769	692	676
31	(5,5,5)	323	301	196
	(6,5,5)	323	306	303
	(6,6,6)	323	298	290
251	(5,5,5)	40	46	46
	(6,5,5)	40	33	33
	(6,6,6)	40	43	43

Remark 3. The constraint $m + k - 2 \geq n$ is not limiting. The probability that a tensors graph has any (V, W) edge, which is necessary for a triangle, is upper bounded by $q^{(m-1)+(k-1)-n}$. So if instead $m + k - 1 \leq n$, this probability is upper bounded by q^{-1}. If such an edge would exist it would serve as a better invariant of the tensor than a triangle. Even more, it would be easier to find.

4.2 Finding triangles

A major reason why previous algorithms do not exploit rare or unique graph structures is the difficulty of finding them. Searching the whole graph is prohibitively expensive and birthday-based techniques can not be used since the structures are extremely rare.

To overcome this difficulty, we propose to look for these algebraically. For the triangles we defined above, the problem can be modeled as a system of quadratic equations in $R = \mathbb{F}_q[x_2, \ldots, x_n, y_2, \ldots, y_m, z_2, \ldots, z_k]$. Let us denote $\mathbf{x} = [1, x_2, \ldots, x_n]$, $\mathbf{y} = [1, y_2, \ldots, y_m]$ and $\mathbf{z} = [1, z_2, \ldots, z_k]$. Indeed, we can consider the following system:

$$\begin{cases} \mathcal{C}(\mathbf{x}, \mathbf{y}, \mathbf{e}_i^W) = 0 & \text{for } 1 \leq i \leq k \\ \mathcal{C}(\mathbf{x}, \mathbf{e}_i^V, \mathbf{z}) = 0 & \text{for } 1 \leq i \leq m \\ \mathcal{C}(\mathbf{e}_i^U, \mathbf{y}, \mathbf{z}) = 0 & \text{for } 1 \leq i \leq n. \end{cases}$$

Table 3. These indicate the experimental nullities in each tri-degree. All values that were predicted wrong are formatted (predicted/actual).

Parameters	**Tri-degree**						
(n, m, k)	(1, 1, 1)	(2, 1, 1)	(3, 1, 1)	(2, 2, 1)	(4, 1, 1)	(3, 2, 1)	(2, 2, 2)
(7,7,7)	2	61	336	772	1141	—	—
(8,8,8)	2	78	504	1174	1960	6356	11601
(9,9,9)	2	97	720	1694	3156	10512	DNF
(8,7,7)	2	63	399	882	1540	4599/4620	7969/8064
(9,8,8)	2	80	584	1316	2544	7896	13907

Then, if we find a solution to the system, the point $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is a representative of the triangle. The attentive reader might find that we are not able to find all triangles in this way. Whenever the first coordinate of $\mathbf{u}, \mathbf{v}$, or $\mathbf{w}$ is zero. However, we simply ignore this issue, since one can rerandomize by applying a random transformation to the given tensor.

Looking at our system we see that we found $n + m + k$ quadratic equations in $n + m + k - 3$ variables. We can now go and use our favorite system-solving technique to find all such solutions. The technique that we will use to compute the complexity is tri-homogeneous XL, see Section 2.3.

We consider the system in the following three sets of variables $\{x_2, \ldots, x_n\}$, $\{y_2, \ldots, y_m\}$, and $\{z_2, \ldots, z_k\}$. Then the system described above consists of n equations in each of the tri-degrees $(1, 1, 0)$, $(1, 0, 1)$, and $(0, 1, 1)$. Furthermore, if we set $x_1 = y_1 = z_1 = 1$ we have the following 2 syzygies in tri-degree $(1, 1, 1)$:

$$\sum_{i=1}^{k} \mathcal{C}(\mathbf{x}, \mathbf{y}, e_i^U) \cdot z_i = \sum_{i=1}^{m} \mathcal{C}(\mathbf{x}, e_i^W, \mathbf{z}) \cdot y_i = \sum_{i=1}^{n} \mathcal{C}(e_i^V, \mathbf{y}, \mathbf{z}) \cdot x_i.$$

Therefore, we conjecture the following Hilbert series:

$$\mathcal{H}(r, s, t) = \frac{(1 - rs)^n (1 - rt)^n (1 - st)^n (1 - rst)^{-2}}{(1 - r)^{n-1} (1 - s)^{m-1} (1 - t)^{k-1}}.$$

We confirmed the predicted number of syzygies in all tri-degrees (d_x, d_y, d_z) for several n, m, k. For simplicity, we only checked parameters that are in use in cryptography, so $n = m = k$ and $n = m + 1 = k + 1$. For the $(8, 7, 7)$ case we find some wrong predictions. This could be due to extra structure for small n. The results are summarized in Table 3. In Table 4, one can find the results of running MAGMA's `GroebnerBasis` on the described system for some small values of n, m, and k. As we can see the solving degree neatly matches the predicted first fall degree, indicating that this is a valid estimator for the complexity. Note that `GroebnerBasis` does not take the specialized tri-graded structure into account. Hence, specialized implementations, like some form of Tri-XL, could speed up computation even more.

Table 4. Finding triangles in 3-TI.

		Actual		Predicted	
(n, m, k)	Time	Memory	d_{solv}	$\mathbf{d}_{solv}$	$\mathbf{d}_{ff}$
(6,6,6)	4 s	32 MB	5	(4,3,1)	(3,1,1)
(7,7,7)	163 s	288 MB	6	(6,3,1)	(3,2,1)
(8,8,8)	7 h	10 GB	7	(6,4,1)	(3,3,1)
(7,6,6)	7 s	64 MB	5	(4,3,1)	(3,1,1)
(8,7,7)	450 s	626 MB	6	(6,3,1)	(3,2,1)

4.3 From matching triangles to isometry

Now let us see what we can do with such a triangle. Recall our problem statement, we need to find the isometry $(\mathbf{A}, \mathbf{B}, \mathbf{C}) : \mathcal{C} \to \mathcal{D}$. As stated before, if $\mathcal{C}$ (and thus $\mathcal{D}$) has no triangle, then this attack is impossible. If the triangles do exist but are not unique, then we iterate over the combinations. So let us assume that $\mathcal{C}$ and $\mathcal{D}$ have unique triangles $T_{\mathcal{C}}$ and $T_{\mathcal{D}}$. By applying suitable basis transformations we can transform these triangles to lie in any position. Therefore, without loss of generality we assume $T_{\mathcal{C}} = (\mathbf{e}_1^U, \mathbf{e}_1^V, \mathbf{e}_1^W) = T_{\mathcal{D}}$. Afterwards we just compose the solution with the inverse of the chosen basis transformations.

Since the isometry maps $T_{\mathcal{C}}$ to $T_{\mathcal{D}}$, we have that $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ have the following form:

$$\mathbf{A} = \begin{bmatrix} \lambda & \mathbf{A}_{12} \\ \mathbf{0}_{(n-1)\times 1} & \mathbf{A}_{22} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mu & \mathbf{B}_{12} \\ \mathbf{0}_{(m-1)\times 1} & \mathbf{B}_{22} \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} \nu & \mathbf{C}_{12} \\ \mathbf{0}_{(k-1)\times 1} & \mathbf{C}_{22} \end{bmatrix} \tag{4}$$

For the inverses of the matrices, we have the form

$$\mathbf{A}' = \lambda^{-1} \begin{bmatrix} 1 & \mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ \mathbf{0} & \lambda\mathbf{A}_{22}^{-1} \end{bmatrix} \quad \mathbf{B}' = \mu^{-1} \begin{bmatrix} 1 & \mathbf{B}_{12}\mathbf{B}_{22}^{-1} \\ \mathbf{0} & \mu\mathbf{B}_{22}^{-1} \end{bmatrix} \quad \mathbf{C}' = \nu^{-1} \begin{bmatrix} 1 & \mathbf{C}_{12}\mathbf{C}_{22}^{-1} \\ \mathbf{0} & \nu\mathbf{C}_{22}^{-1} \end{bmatrix}. \tag{5}$$

Here λ, μ, ν are some scalars in $\mathbb{F}_q$ that come from the fact that our triangles live in projective space. However, we can freely rescale $\mathbf{B}$ and $\mathbf{C}$ by scaling $\mathbf{A}$ accordingly, so without loss of generality we assume $\mu = \nu = 1$. Under these assumptions, we get that the triangle vectors are eigenvectors of the isometry matrices, more concretely, $\mathbf{A}\mathbf{e}_1 = \lambda\mathbf{e}_1$, $\mathbf{B}\mathbf{e}_1 = \mathbf{e}_1$, $\mathbf{C}\mathbf{e}_1 = \mathbf{e}_1$. Plugging these relations into the equations in (2) and (3), part of them become linear, i.e. we obtain:

$$\begin{cases} \mathcal{C}(\mathbf{A}\mathbf{x}, \mathbf{e}_1^V, \mathbf{z}) = \mathcal{D}(\mathbf{x}, \mathbf{e}_1^V, \mathbf{C}'\mathbf{z}) \\ \mathcal{C}(\mathbf{x}, \mathbf{e}_1^V, \mathbf{C}\mathbf{z}) = \mathcal{D}(\mathbf{A}'\mathbf{x}, \mathbf{e}_1^V, \mathbf{z}) \end{cases} \quad \forall\, \mathbf{x} \in \mathbb{F}_q^n, \mathbf{z} \in \mathbb{F}_q^k \tag{6}$$

$$\begin{cases} \mathcal{C}(\mathbf{A}\mathbf{x}, \mathbf{y}, \mathbf{e}_1^W) = \mathcal{D}(\mathbf{x}, \mathbf{B}'\mathbf{y}, \mathbf{e}_1^W) \\ \mathcal{C}(\mathbf{x}, \mathbf{B}\mathbf{y}, \mathbf{e}_1^W) = \mathcal{D}(\mathbf{A}'\mathbf{x}, \mathbf{y}, \mathbf{e}_1^W) \end{cases} \quad \forall\, \mathbf{x} \in \mathbb{F}_q^n, \mathbf{y} \in \mathbb{F}_q^m \tag{7}$$

Table 5. Running times for the post-collision algorithm. We also report the linear independent equations in the system presented in Equations (6), (7) and (8).

(n, m, k)	Variables	Linear equations	Quadratic equations	Time	d_{solv}
(8,8,8)	238	196	2802	2 s	2
(14,14,14)	1094	674	15962	245 s	2
(20,20,20)	2282	1444	47042	3 h	2
(9,8,8)	370	224	3156	68 s	2
(16,15,15)	1322	840	13566	536 s	2
(21,20,20)	2362	1520	49364	3 h	2

Notice that these correspond exactly to the top and front slices of the tensor. Furthermore, we also get the following, almost linear, equations (recall that λ is unknown):

$$\begin{cases} \mathcal{C}(\lambda \mathbf{e}_1^U, \mathbf{B}\mathbf{y}, \mathbf{z}) = \mathcal{D}(\mathbf{e}_1^U, \mathbf{y}, \mathbf{C}'\mathbf{z}) \\ \mathcal{C}(\lambda \mathbf{e}_1^U, \mathbf{y}, \mathbf{C}\mathbf{z}) = \mathcal{D}(\mathbf{e}_1^U, \mathbf{B}'\mathbf{y}, \mathbf{z}) \end{cases} \quad \forall\ \mathbf{y} \in \mathbb{F}_q^m, \mathbf{z} \in \mathbb{F}_q^k \tag{8}$$

Since the matrices $\mathcal{C}(\mathbf{x}, \mathbf{e}_1^V, \mathbf{z})$ are not full rank by construction, we know that not all the linear relations in Equation (6) are independent. Similarly, for the Equation (7) and Equation (8). As we can see, a big chunk of the variables can already be eliminated by linear equations. On top of that, there is a substantial amount of quadratic equations that the solution has to satisfy. Instead of analyzing whether there are any algebraic dependencies between those linear and quadratic equations, we went with a more practical approach and ran a Gröbner basis algorithm on the resulting system. For all experiments, the system terminated in degree 2 so we conjecture that this is the solving degree for such systems. This would lead to a complexity of $\mathcal{O}(\binom{n^2+1}{2}^\omega) = \mathcal{O}(n^{4\omega})$. Here ω is the linear algebra constant. In any case, as can be seen from the experimental results in Table 5, this part of the algorithm is practical for all values of n for which we can hope to find a triangle.

4.4 Putting it all together

As we saw above, the cost of finding the triangle is the most dominating. If we take into account the search for an attack then we get the following complexity for the original MEDS parameters (Table 6):

Recently, as a result of the attack from [30], the MEDS team updated the parameters [32]. In Table 7, we show the complexity results for the new parameters. As we can see, the new parameters are secure against this attack, even up to the first fall degree.

Table 6. The $\log_2$ complexity estimates for the original MEDS parameters. All complexities are in field operations. We use the solving degree as an estimator here.

			[12]	[30]	**This work**	
	n	q	Specs	Best previous	incl. search	prob. $1/q$
Level I	14	4093	147	95	102	90
Level III	22	4093	217	145	155	143
Level V	30	2039	276	180	208	197

Table 7. Complexity estimations $\log_2$ for newest MEDS parameter set. As [30] was only shown to work for balanced parameters we estimate its complexity for the nearest balanced parameters.

					First fall degree		Solving degree		[30]
	n	m	k	q	$\mathbf{d}_{ff}$	compl.	$\mathbf{d}_{solv}$	compl.	compl.
Level I	26	25	25	4093	(11, 18, 1)	151	(13, 20, 1)	165	164*
Level III	35	34	34	4093	(16, 25, 1)	210	(18, 27, 1)	224	219*
Level V	45	44	44	4093	(23, 31, 1)	275	(25, 33, 1)	289	280*

5 Application to ATFE

Having seen the above algorithm, a natural question is to ask whether it can also be applied to more structured tensors, like alternating tensors. It turns out, that indeed it does, and that after taking care of some technicalities, we can even use the extra structure in our advantage.

First, let us try the same as before, but now in the reduced graph. We look at triplets of points $(\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3) \in \mathbb{T}(V, V, V)$ for which

$$\phi(\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, -) = \phi(\hat{\mathbf{v}}_1, -, \hat{\mathbf{v}}_3) = \phi(-, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3) = 0.$$

Now by anti-symmetry and tri-linearity this means that $\phi(\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2 + \hat{\mathbf{v}}_3, -) = 0$ and also $\phi(\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_1, -) = 0$. In other words, let $\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, \hat{\mathbf{w}}_3$ be any linear combinations of $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3$, then $(\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, \hat{\mathbf{w}}_3)$ is a triangle as well. Even more, $(\hat{\mathbf{v}}, \hat{\mathbf{v}}, \hat{\mathbf{v}})$ is a triangle for any $\hat{\mathbf{v}} \in \mathbb{P}(V)$.

This structure points us to look at a modified definition of a "triangle". Let $\mathbb{T}(V) = \{W \subset V \mid \dim(W) = 3\}$ be the set of all 3-dimensional subspaces of V.

Definition 4. *Let $\phi : \bigwedge^3 V \to \mathbb{F}_q$ be an alternating trilinear form. We call a 3-dimensional subspace $T \in \mathbb{T}(V)$ a triangle for ϕ if*

$$\phi(\mathbf{v}, \mathbf{w}, -) = 0 \quad \forall\ \mathbf{v}, \mathbf{w} \in T.$$

This definition has been considered before in [17] in the pursuit of classifying alternating trilinear forms. There they were called 3-dimensional 2-singular subspaces. Just as in the unstructured case, we have the following:

Lemma 2. *The expectation value for the number of such constructions in a random alternating trilinear form is equal to q^{-1}, i.e:*

$$\mathbb{E}_{\phi\in(\bigwedge^3 V)^*}(|\{T \in \mathbb{T}(V) \mid T \text{ is a triangle for } \phi\}|) = q^{-1}.$$

Proof. We follow a similar structure as in Lemma 1. We extend $\langle \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\rangle = T \in \mathbb{T}(V)$ to a basis $V = \langle \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_n\rangle$. Applying a basis transformation keeps the coefficients of ϕ uniformly random. So we need to compute the probability that $(\mathbf{e}_1^V, \mathbf{e}_2^V, \mathbf{e}_3^V)$ is a triangle for a random ϕ. This latter condition can be reformulated as $\phi_{123} = \phi_{12i} = \phi_{13i} = \phi_{23i} = 0$ for all $4 \leq i \leq n$. (See Figure 2.)

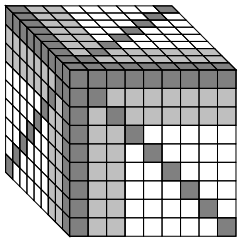

Fig. 2. For an ATF ϕ, the coefficients in the light gray positions should be zero for $\langle \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\rangle$ to be a triangle. The coefficients corresponding to the dark gray positions are zero by alternatingness.

Taking symmetries into account these are exactly $3(n-3)+1$ relations. Since the coefficients are all uniform in $\mathbb{F}_q$ and using simple linearity of the expectation value, we obtain:

$$\begin{aligned}
&\mathbb{E}_{\phi\in(\bigwedge^3 V)^*}(|\{T \in \mathbb{T}(V) \mid T \text{ is a triangle for } \phi\}|) \\
&\qquad = \mathbb{P}_{\phi\in(\bigwedge^3 V)^*, T\in\mathbb{T}(V)}(T \text{ is a triangle for } \phi) \cdot |\mathbb{T}(V)| \\
&\qquad = q^{-(3(n-3)+1)} \cdot q^{3(n-3)} \\
&\qquad = q^{-1}.
\end{aligned}$$

To say something about how reliably our new algorithm can be used, we need a stronger result, i.e., not only the expectation, but the probability of having a unique triangle. Fortunately, we have the following corollary:

Corollary 2. *Given a vector space V over $\mathbb{F}_q$ of dimension $n = \dim(V) \geq 9$:*

$$\mathbb{P}_{\phi\in(\bigwedge^3 V)^*}(\phi \text{ has a unique triangle}) \geq q^{-1} - \mathcal{O}(q^{-2}).$$

Proof. See Appendix A.2.

We verify this experimentally using the polynomial system described below. The results, which are in line with our corollary, are given in Table 8. For practical reasons, we assume that the triangle does not intersect with the $x_1 = x_2 = x_3 = 0$ hyperplane. This allows us to fix variables to make solving computable. Again, MAGMA's `VarietySize` is used on the system described in the next section.

Table 8. Fraction of tensors where a triangle was found on 10000 experiments. We also report the times the triangle was unique.

q	n	Predicted	Found	Unique
13	9	769	720	665
	10	769	716	694
	11	769	759	740
31	9	323	266	259
	10	323	311	308
	11	323	319	314
251	9	40	33	33
	10	40	33	33
	11	40	40	40

5.1 Finding triangles

To find these triangles we are going to use algebraic methods again. In this case we will be working over the ring $\mathcal{R} = \mathbb{F}_q[x_4, \ldots, x_n, y_4, \ldots, y_n, z_4, \ldots, z_n]$. We use a shorthand notation for the following vectors in $\mathcal{R}^n$:

$$\mathbf{x} = [1, 0, 0, x_4, \ldots, x_n],\ \mathbf{y} = [0, 1, 0, y_4, \ldots, y_n],\ \mathbf{z} = [0, 0, 1, z_4, \ldots, z_n].$$

Now, given an alternating trilinear form ϕ, our system for finding a triangle looks as follows:

$$\begin{cases} f^{(i)}_{\mathbf{xy}} := \phi(\mathbf{x}, \mathbf{y}, \mathbf{e}_i) = 0 \\ f^{(i)}_{\mathbf{yz}} := \phi(\mathbf{y}, \mathbf{z}, \mathbf{e}_i) = 0 \\ f^{(i)}_{\mathbf{zx}} := \phi(\mathbf{z}, \mathbf{x}, \mathbf{e}_i) = 0 \end{cases} \qquad \forall 1 \leq i \leq n \tag{9}$$

It is clear that when we have found a solution $(x_4, \ldots, x_n, y_4, \ldots, y_n, z_4, \ldots, z_n)$ we indeed have also found a triangle.

We consider solving this system using Gröbner basis methods. To make the analysis precise we are going to conjecture a Hilbert series, but first, we find a class of structural syzygies. In tri-degree $(1, 1, 1)$, just as in the MCE case, we have the following 2 linear independent syzygies appearing:

$$\sum_{i=1}^{n} \phi(\mathbf{x}, \mathbf{y}, e_i) \cdot z_i = \sum_{i=1}^{n} \phi(\mathbf{y}, \mathbf{z}, e_i) \cdot x_i = \sum_{i=1}^{n} \phi(\mathbf{z}, \mathbf{x}, e_i) \cdot y_i.$$

However, in the ATFE case, there are 6 more syzygies, in tri-degrees that are a permutation of $(2, 1, 0)$. They are due to the alternating properties:

$$\sum_{i=1}^{n} \phi(\mathbf{x}, \mathbf{y}, e_i) \cdot x_i = \phi(\mathbf{x}, \mathbf{y}, \mathbf{x}) = 0$$

Taking everything together we conjecture the Hilbert series. For reasons that become apparent in Section 5.3 we consider the tri-homogenized version instead:

$$\mathcal{S}(r,s,t) = (1-r^2s)(1-rs^2)(1-s^2t)(1-st^2)(1-t^2r)(1-tr^2)(1-rst)^2$$
$$\mathcal{H}(r,s,t) = \frac{(1-rs)^n(1-rt)^n(1-st)^n}{(1-r)^{n-2}(1-s)^{n-2}(1-t)^{n-2}} \cdot \mathcal{S}^{-1}.$$

We created the Macaulay matrix for different values of n, d_x, d_y, and d_z and computed their rank. With this, we verified the amount of linear independent syzygies predicted by the Hilbert series. The results are summarized in Table 9.

Table 9. Experimental syzygies in each tri-degree ($q = 29$). All values that were predicted wrong are formatted (predicted/actual).

	Tri-degree								
n	(2, 2, 0)	(2, 1, 1)	(4, 1, 0)	(3, 2, 0)	(3, 1, 1)	(2, 2, 1)	(5, 1, 0)	(4, 2, 0)	(3, 3, 0)
12	86	184	55	803	1726	4002	220/221	4281/4282	7241/7242
13	100	213	66	1032	2207	5140	286/287	6018/6019	10319/10320
14	115	244	78	1300	2768	6472	364	8231	14277
15	131	277	91	1610	3415	8013	455	10999	
16	148	312	105	1965	4154	9778	560		
17	166	349	120	2368	4991				
18	185	388	136	2822	5932				
19	205	429	153	3330					
20	226	472	171	3895					

Now, let us see what this means in practice. In Table 10, we report the running time of MAGMA's `GroebnerBasis` on this system. As we can see, both $\mathbf{d}_{solv}$ and $\mathbf{d}_{ff}$ largely overestimate the actual solving degree. Given that the experimental results on the Hilbert series are so affirmative, this might seem odd. What happens in practice is that there are quite some structural degree falls. We will see more on these in Section 5.3. For now, the main takeaway is that for $n = 13$ and $n = 14$, finding triangles is practical.

5.2 Post-collision

Now, following the same line of argumentation as in Section 4.3, assume that ϕ and ψ both have a unique triangle in general position $\langle \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \rangle$. This places the following limit on any transformation $\mathbf{A} : \phi \to \psi$ between them:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{0}_{(n-3)\times 3} & \mathbf{A}_{22} \end{pmatrix} \qquad \mathbf{A}^{-1} = \begin{pmatrix} \mathbf{A}_{11}^{-1} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ \mathbf{0}_{(n-3)\times 3} & \mathbf{A}_{22}^{-1} \end{pmatrix}$$

These restrictions unfortunately do not generate linear equations. However, instead, we consider this system as a tri-graded system in variable sets

$$\{\mathbf{A}_{11}, \mathbf{A}'_{11}\}, \quad \{\mathbf{A}_{12}, \mathbf{A}'_{12}\}, \quad \{\mathbf{A}_{22}, \mathbf{A}'_{22}\}.$$

Table 10. Finding triangles in ATFE, experimental results ($q = 29$).

	Actual			Predicted	
n	Time	Memory	d_{solv}	$\mathbf{d}_{solv}$	$\mathbf{d}_{ff}$
12	29 s	96 MB	4	(7, 3, 1)	(5, 2, 1)
13	490 s	850 MB	4	(7, 4, 1)	(5, 3, 1)
14	30 h	29 GB	5	(7, 5, 1)	(5, 4, 1)
15	$\geq$ 14 d	$\geq$ 260 GB	≥ 6	(7, 6, 1)	(5, 5, 1)

Table 11. Practical runtimes of the post-collision algorithm, $q = 2^{32} - 5$

n	Time (s)	Memory (MB)	$(wt_{11}, wt_{12}, wt_{22})$	weighted d_{solv}
13	102	864	(1, 3, 3)	6
20	2317	8287	(1, 3, 3)	6
25	8736	21750	(1, 3, 3)	6

Inspired by an unverified Hilbert series (not shown here) we use degree weights to put more emphasis on the $\{\mathbf{A}_{11}, \mathbf{A}'_{11}\}$ variables. Now running `GroebnerBasis` yields the desired results, these can be found in Table 11 below. Given the efficiency of solving $n = 25$, we do not optimize beyond this.

Given that the complexity of the post-collision is negligible we provide the complexities for the complete algorithm in Table 12. These hold for the $1/q$ amount of ATFE problems that have a triangle.

5.3 A peculiar set of degree falls

The results in Table 10 indicate that there is something more going on than our Hilbert series predicts. This has to do with the fact that there are additional degree falls present. In F4-like algorithms, these degree falls immediately speed up computation. In this section, we show some structural degree falls that happen for all values of n. First, we need to extend the definition of alternating multilinear forms to multivariate polynomials:

Definition 5. *Let $f \in \mathbb{F}_q[\mathbf{x}, \mathbf{y}, \mathbf{z}]$, where $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_n)$, $\mathbf{z} = (z_1, \ldots, z_n)$. We call f alternating if*

$$f(\mathbf{x}, \mathbf{y}, \mathbf{z}) = -f(\mathbf{y}, \mathbf{x}, \mathbf{z}) = -f(\mathbf{z}, \mathbf{y}, \mathbf{x}).$$

Remark 4. Just as in the case of alternating trilinear forms, one can show that, in finite fields, this is equivalent to putting constraints on the coefficients of such polynomials. The polynomial $f = \sum_{\alpha,\beta,\gamma \in \mathbb{Z}^n_{\geq 0}} c_{\alpha\beta\gamma} \mathbf{x}^\alpha \mathbf{y}^\beta \mathbf{z}^\gamma$ is alternating if and only if

$$c_{\alpha\beta\gamma} = -c_{\beta\alpha\gamma} = -c_{\gamma\beta\alpha} \quad \forall \alpha, \beta, \gamma \in \mathbb{Z}^n_{\geq 0}.$$

Table 12. The $\log_2$ complexity for solving ATFE (with probability $1/q$) in field operations. The parameters are taken from the ALTEQ specifications [8].

		[8]	[37]	**This work**		
	n	Specs	Best previous	$\mathbf{d}_{solv}$ compl.	$\mathbf{d}_{ff}$ compl.	Time
Level I	13	143	120	62	51	1501 s
Level III	20	219	165	108	96	
Level V	25	276	203	141	128	

As an example, the alternating tri-homogeneous polynomials living in tri-degree $(1,1,1)$ are exactly the alternating trilinear forms. A simple counting argument tells us that the dimension of tri-homogeneous alternating elements in tri-degree (d,d,d) is exactly $\binom{M_{n,d}}{3}$. Here $M_{n,d} = \binom{n+d-1}{d}$ is the amount of degree d monomials in n variables. For $d = 1$ we then indeed get $\binom{n}{3}$. Furthermore, we find that there are no tri-homogeneous alternating elements in tri-degree (d_x, d_y, d_z) if $d_x \neq d_y$ or $d_x \neq d_z$.

Now back to the degree falls. A degree fall happens when there is a syzygy among the homogeneous top parts of elements which is not a full syzygy. Recall our system Equation (9). We can write the top-degree part of our system using the following substitutes

$$\overline{\mathbf{x}} = [0,0,0,x_4,\ldots,x_n],\ \overline{\mathbf{y}} = [0,0,0,y_4,\ldots,y_n],\ \overline{\mathbf{z}} = [0,0,0,z_4,\ldots,z_n].$$

Then we can easily write down the top-degree parts of our system:

$$\overline{f^{(i)}_{\mathbf{xy}}} = f^{(i)}_{\mathbf{xy}}(\overline{\mathbf{x}},\overline{\mathbf{y}},\overline{\mathbf{z}})$$

To construct syzygies among these, we will build alternating functions using symmetry. We will use the following lemma.

Lemma 3. *Let $f \in \mathbb{F}_q[x_1,\ldots,x_n,y_1,\ldots,y_n,z_1,\ldots,z_n]$ such that f is alternating in its first two arguments, i.e. $f(\mathbf{x},\mathbf{y},\mathbf{z}) = -f(\mathbf{y},\mathbf{x},\mathbf{z})$. Then,*

$$\mathcal{S}(f)(\mathbf{x},\mathbf{y},\mathbf{z}) := f(\mathbf{x},\mathbf{y},\mathbf{z}) + f(\mathbf{z},\mathbf{x},\mathbf{y}) + f(\mathbf{y},\mathbf{z},\mathbf{x})$$

is alternating.

In our system, the polynomials $f^{(i)}_{\mathbf{xy}}$ are indeed alternating in the first two arguments and hence the polynomials $\overline{f^{(i)}_{\mathbf{xy}}}$ are too. Furthermore, for the top-degree parts we have $\overline{f^{(i)}_{\mathbf{xy}}}(\mathbf{y},\mathbf{z},\mathbf{x}) = \overline{f^{(i)}_{\mathbf{yz}}}(\mathbf{x},\mathbf{y},\mathbf{z})$. Therefore, the polynomial function

$$\mathcal{S}(\overline{f^{(i)}_{\mathbf{xy}}}) = \overline{f^{(i)}_{\mathbf{xy}}} + \overline{f^{(i)}_{\mathbf{yz}}} + \overline{f^{(i)}_{\mathbf{zx}}}$$

is alternating and a linear combination of the top degrees of polynomials in our system. We can take this a step further and consider $\mathcal{S}(\overline{f^{(i)}_{\mathbf{xy}}} \cdot g(\mathbf{z}))$ for a linear

Table 13. Structural degree falls in tri-degree $(1, 1, 1)$, experimentally verified.

n	degree falls
10	35
11	32
12	24
13	10

Table 14. Structural degree falls in tri-degree $(2, 2, 2)$ and $(3, 3, 3)$.

degree falls $(2, 2, 2)$		
n	predicted	actual
14	15224	18941
15	15184	22337
16	11011	

degree falls $(3, 3, 3)$	
n	predicted
17	23866080
20	77142736
25	245964576

polynomial g. The result lies in degree $(1, 1, 1)$ and is, in fact, an alternating trilinear form. More generally, we can look at the subspace of the Macaulay space in $(1, 1, 1)$ generated by $\mathcal{S}(\overline{f^{(i)}_{\mathbf{xy}}} \cdot \mathbf{z}_j)$ for $1 \leq i \leq n$ and $1 \leq j \leq n-3$. We know that this space is contained in $(\bigwedge^3 \mathbb{F}_q^{n-3})^*$. Hence, by the rank nullity theorem, we can conclude that there are at least $n(n-3) - \binom{n-3}{3}$ syzygies in the top-degree parts. For $n \leq 13$ this turns out to be more than 0. This explains why $n = 13$ runs so much faster than anticipated. With experiments, we were able to verify that these are indeed degree falls and not complete syzygies and that the number is correct. The results are given in Table 13.

For $n \geq 14$ these degree falls do not appear anymore, but there are others. Let $g \in \mathcal{R}$ be a homogeneous polynomial in tri-degree $(d-1, d-1, d)$ such that it is symmetric in the first two arguments, i.e. $g(\mathbf{x}, \mathbf{y}, \mathbf{z}) = g(\mathbf{y}, \mathbf{x}, \mathbf{z})$. Now, due to the symmetry of g, $\overline{f^{(i)}_{\mathbf{xy}}} \cdot g$ is still alternating in its first two elements. Therefore, $\mathcal{S}(\overline{f^{(i)}_{\mathbf{xy}}} \cdot g)$ is again alternating and in the degree (d, d, d) Macaulay space. We can apply rank-nullity again for the subspace generated by such g and find that the amount of top-degree syzygies appearing is

$$\binom{M_{n-3,d-1}+1}{2} \cdot M_{n-3,d} \cdot n - \binom{M_{n-3,d}}{3}.$$

We did experiments to verify these numbers but found that there are even more degree falls appearing than predicted by the formula above. All of those were indeed degree falls and not full syzygies. Due to the size of the matrices involved only a part of the predictions could be verified, see Table 14.

Even after identifying these $(2, 2, 2)$ degree falls for $n = 15$ and adding those to the system, unfortunately, the system was still not solvable in total degree 5. So while these syzygies likely contribute to a lower solving degree than $11 = 5 + 5 + 1$ (as in Table 10), we do not (yet) know by how much.

6 Potential generalizations and future work

Given the success of these small graph invariants on 3-TI and its structured variant, the question arises as to how this generalizes to other structures on tensors. A quick back-of-the-envelope calculation tells us the expectation value for similar structures in QMLE and Cubic-IP should be $1/q$ too. However, as we saw already in Remark 3, experimental evidence is needed before drawing conclusions.

Another interesting generalization might be to see how well this applies for k-tensors with $k \geq 4$. While these tensors are not (yet) cryptographically relevant, this might still provide insights into general invariants of tensors. One obvious hurdle in this case is that simple graphs might not be sufficient anymore.

Next to these generalizations, there are also two other open questions remaining after this work. It would be interesting to know how extra degree falls that we found contribute to the solving degree. Being able to make better predictions for the solving degree would give us more precise security estimates. Secondly, those degree falls also seem to exist, albeit fewer, in general systems of skew-symmetric bilinear equations. These might be used to speed up solving such systems as well.

A Lower bounds on probabilities of triangles

The purpose of this section is to prove Corollary 1 and Corollary 2. To lower bound the probability of triangles we are going to use the following lemma.

Lemma 4. *Let X be a discrete distribution with values in $\mathbb{Z}_{\geq 0}$. Then:*

$$\mathbb{P}(X = 1) \geq 2\mathbb{E}(X) - \mathbb{E}(X^2).$$

Proof. Using that probabilities are non-negative:

$$\mathbb{P}(X = 1) \geq \sum_{i \geq 0} (2i - i^2)\mathbb{P}(X = i) = 2\mathbb{E}(X) - \mathbb{E}(X^2).$$

Now, using this lemma, the task at hand is to compute $\mathbb{E}(X^2)$ for MCE and ATFE. Let us denote $T \triangle \mathcal{C}$ (resp. $T \triangle \phi$) if T is a triangle for $\mathcal{C}$ (resp. ϕ).

A.1 MCE

Lemma 5. *Suppose we have vector spaces U, V, and W over $\mathbb{F}_q$ of dimensions n, m, and k, then:*

$$\begin{aligned}&\mathbb{E}_{\mathcal{C} \in (U \otimes V \otimes W)^*}(|\{T \in \mathbb{T}(U, V, W) \mid T \triangle \mathcal{C}\}|^2) \\ &\approx q^{-1} + q^{-2} + q^{-(n-1)} + q^{-(m-1)} + q^{-(k-1)} \\ &+ q^{-(n+m-k)} + q^{-(m+k-n)} + q^{-(k+n-m)}.\end{aligned}$$

Table 15. Sizes and indices that should be zero for the different sets in the partition.

Set	Size ($\log_q$)	Zeroes	Set of indices
$\mathbb{T}_\emptyset$	$2n+2m+2k-6$	$2n+2m+2k-4$	$(1,1,\star),(1,\star,1),(\star,1,1),(2,2,\star),(2,\star,2),(\star,2,2)$
$\mathbb{T}_u$	$n+2m+2k-5$	$2n+2m+2k-6$	$(1,1,\star),(1,\star,1),(\star,1,1),(1,2,\star),(1,\star,2),(\star,2,2)$
$\mathbb{T}_{uv}$	$n+m+2k-4$	$2n+2m+k-4$	$(1,1,\star),(1,\star,1),(\star,1,1),(1,\star,2),(\star,1,2)$
$\mathbb{T}_{uvw}$	$n+m+k-3$	$n+m+k-2$	$(1,1,\star),(1,\star,1),(\star,1,1)$

Proof. We start with the observation that, for a given tensor $\mathcal{C}$, we have:

$$|\{T \in \mathbb{T} \mid T \triangle \mathcal{C}\}|^2 = |\{(T_1, T_2) \in \mathbb{T}^2 \mid T_1 \triangle \mathcal{C}, T_2 \triangle \mathcal{C}\}|.$$

We are going to calculate the probability that both $T_1 = (u, v, w)$ and $T_2 = (u', v', w')$ are triangles for a random $\mathcal{C}$. To do this, we need to distinguish some cases. We define the following partition of $\mathbb{T}^2$:

$$\begin{aligned}
\mathbb{T}_\emptyset &= \{(T_1, T_2) \in \mathbb{T}^2 | u \neq u', v \neq v', w \neq w'\} \\
\mathbb{T}_u &= \{(T_1, T_2) \in \mathbb{T}^2 | u = u', v \neq v', w \neq w'\} \quad (\text{resp. } \mathbb{T}_v, \mathbb{T}_w) \\
\mathbb{T}_{uv} &= \{(T_1, T_2) \in \mathbb{T}^2 | u = u', v = v', w \neq w'\} \quad (\text{resp. } \mathbb{T}_{vw}, \mathbb{T}_{wu}) \\
\mathbb{T}_{uvw} &= \{(T_1, T_2) \in \mathbb{T}^2 | u = u', v = v', w = w'\}.
\end{aligned}$$

For each of these sets, the probability that an element is a pair of triangles for a random $\mathcal{C}$ is constant. To see this we are going to consider the tensor in the extended basis $\langle u, u_2, \ldots, u_n \rangle$ if $u = u'$ and $\langle u, u', u_3 \ldots, u_n \rangle$ if $u \neq u'$ (similarly for v and w). Then, the coefficients of these tensors are again uniformly random. We count the amount of indices (i, j, l) which should have a 0 so that T_1 and T_2 are triangles in Table 15.

Then, by linearity we can compute the expectation value:

$$\begin{aligned}
\mathbb{E}(|\{T_1, T_2 \in \mathbb{T} \mid T_1 \triangle \mathcal{C}, T_2 \triangle \mathcal{C}\}|) &= q^{-(2n+2m+2k-4)} |\mathbb{T}_\emptyset| \\
&+ q^{-(2n+2m+2k-6)} \left(|\mathbb{T}_u| + |\mathbb{T}_v| + |\mathbb{T}_w|\right) \\
&+ q^{-(2n+2m+k-4)} |\mathbb{T}_{uv}| + q^{-(n+2m+2k-4)} |\mathbb{T}_{vw}| \\
&+ q^{-(2n+m+2k-4)} |\mathbb{T}_{wu}| + q^{-(n+m+k-2)} |\mathbb{T}_{uvw}| \\
&= q^{-1} + q^{-2} + q^{-(n-1)} + q^{-(m-1)} + q^{-(k-1)} \\
&+ q^{-(n+m-k)} + q^{-(m+k-n)} + q^{-(k+n-m)}.
\end{aligned}$$

Corollary 3. *Given vector spaces U, V, and W over $\mathbb{F}_q$ of dimensions n, m, and k, then:*

$$\begin{aligned}
\mathbb{P}_{\mathcal{C} \in (U \otimes V \otimes W)^*}(\mathcal{C} \text{ has unique triangle}) &\geq q^{-1} - q^{-2} - q^{-(n-1)} - q^{-(m-1)} - q^{-(k-1)} \\
&- q^{-(n+m-k)} - q^{-(m+k-n)} - q^{-(k+n-m)}.
\end{aligned}$$

Now Corollary 1 follows immediately.

Table 16. Sizes and indices that should be zero for the different sets in the partition.

Set	Size ($\log_q$)	Zeroes	Set of index sets
$\mathbb{T}_0$	$6n-18$	$6n-16$	$\{1,2,\star\},\{1,3,\star\},\{2,3,\star\},\{4,5,\star\},\{4,6,\star\},\{5,6,\star\}$
$\mathbb{T}_1$	$5n-13$	$6n-20$	$\{1,2,\star\},\{1,3,\star\},\{2,3,\star\},\{1,4,\star\},\{1,5,\star\},\{4,5,\star\}$
$\mathbb{T}_2$	$4n-10$	$5n-14$	$\{1,2,\star\},\{1,3,\star\},\{2,3,\star\},\{1,4,\star\},\{2,4,\star\}$
$\mathbb{T}_3$	$3n-9$	$3n-8$	$\{1,2,\star\},\{1,3,\star\},\{2,3,\star\}$

A.2 ATFE

Lemma 6. *Suppose we have a vector space V over $\mathbb{F}_q$ of dimension n then:*

$$\mathbb{E}_{\phi\in(\bigwedge^3 V)^*}(|\{T\in\mathbb{T}(V)\mid T\triangle\phi\}|^2)\approx q^{-1}+q^{-2}+q^{-(n-4)}+q^{-(n-7)}.$$

Proof. The proof is structured similarly. This time, we partition $\mathbb{T}^2$ in the following sets:

$$\mathbb{T}_i=\{(T_1,T_2)\in\mathbb{T}^2|\dim(T_1\cap T_2)=i\}\quad\text{for } i=0,1,2,3$$

The probability is again constant on these sets for random ϕ. Given a pair of triangles $(T_1,T_2)\in\mathbb{T}_i$ we pick a basis $\langle u_1,\dots,u_n\rangle$ such that, $T_1=\langle u_1,u_2,u_3\rangle$ and

$$T_2=\begin{cases}\langle u_4,u_5,u_6\rangle \text{ if } i=0,\\ \langle u_1,u_4,u_5\rangle \text{ if } i=1,\\ \langle u_1,u_2,u_4\rangle \text{ if } i=2,\\ \langle u_1,u_2,u_3\rangle \text{ if } i=3.\end{cases}$$

Note that this is possible by first picking a basis of $T_1\cap T_2$ by definition of $\mathbb{T}_i$. Then, the coefficients of these ATFs are again uniformly random. Now we count the number of index sets $\{i,j,k\}$ such that ϕ_{ijk} must be zero in order for T_1 and T_2 to be a triangle. These results are in table Table 16

Then, again by linearity, we can compute the expectation value:

$$\begin{aligned}&\mathbb{E}(|\{T_1,T_2\in\mathbb{T}\mid T_1\triangle\phi,T_2\triangle\phi\}|)\\ &\quad=q^{-(6n-16)}|\mathbb{T}_0|+q^{-(6n-20)}|\mathbb{T}_1|+q^{-(5n-14)}|\mathbb{T}_2|+q^{-(3n-8)}|\mathbb{T}_3|\\ &\quad=q^{-1}+q^{-2}+q^{-(n-7)}+q^{-(n-4)}.\end{aligned}$$

Corollary 4. *Given a vector space V over $\mathbb{F}_q$ of dimension $n=\dim(V)$ then:*

$$\mathbb{P}_{\phi\in(\bigwedge^3 V)^*}(\phi \textit{ has a unique triangle})\geq q^{-1}-q^{-2}-q^{-(n-7)}-q^{-(n-4)}.$$

Now Corollary 2 follows immediately.

References

[1] NIST fourth round announcement. NIST Official Website (2021), https://csrc.nist.gov/projects/post-quantum-cryptography/round-4-submissions

[2] Bardet, M., Faugère, J., Salvy, B., Spaenlehauer, P.: On the complexity of solving quadratic Boolean systems. Journal of Complexity **29**(1), 53–75 (2013)

[3] Barenghi, A., Biasse, J., Persichetti, E., Santini, P.: LESS-FM: fine-tuning signatures from the code equivalence problem. In: Cheon, J.H., Tillich, J. (eds.) PQCrypto 2021. LNCS, vol. 12841, pp. 23–43. Springer (2021)

[4] Beullens, W.: Not enough LESS: an improved algorithm for solving code equivalence problems over $\mathbb{F}_q$. In: Dunkelman, O., Jacobson, M.J., O'Flynn, C. (eds.) SAC 2020. LNCS, vol. 12804, pp. 387–403. Springer (2020)

[5] Beullens, W.: Graph-theoretic algorithms for the alternating trilinear form equivalence problem. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. pp. 101–126. Springer Nature Switzerland, Cham (2023)

[6] Beullens, W., Katsumata, S., Pintore, F.: Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12492, pp. 464–492. Springer (2020)

[7] Biasse, J.F., Micheli, G., Persichetti, E., Santini, P.: LESS is More: Code-Based Signatures Without Syndromes. In: Nitaj, A., Youssef, A. (eds.) AFRICACRYPT 2020. LNCS, vol. 12174, pp. 45–65. Springer (2020)

[8] Bläser, M., Duong, D.H., Narayanan, A.K., Plantard, T., Qiao, Y., Sipasseuth, A., , Tang, G.: The ALTEQ Signature Scheme: Algorithm Specifications and Supporting Documentation. NIST PQC Submission (2023)

[9] Bosma, W., Cannon, J., Playoust, C.: The Magma Algebra System. I. The User Language. J. Symbolic Comput. **24**(3-4), 235–265 (1997)

[10] Bouillaguet, C., Fouque, P., Véber, A.: Graph-theoretic algorithms for the "isomorphism of polynomials" problem. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 211–227. Springer (2013)

[11] Buchberger, B.: Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. Ph.D. thesis, University of Innsbruck (1965)

[12] Chou, T., Niederhagen, R., Persichetti, E., Ran, L., Randrianarisoa, T.H., Reijnders, K., Samardjiska, S., Trimoska, M.: MEDS – Matrix Equivalence Digital Signature (2023), https://meds-pqc.org/spec/MEDS-2023-05-31.pdf, submission to the NIST Digital Signature Scheme standardization process.

[13] 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: El Mrabet, N., De Feo, L., Duquesne, S. (eds.) Progress in Cryptology - AFRICACRYPT 2023. pp. 28–52. Springer Nature Switzerland, Cham (2023)

[14] Courtois, N.T., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 392–407. Springer (2000)
[15] Couvreur, A., Debris-Alazard, T., Gaborit, P.: On the hardness of code equivalence problems in rank metric. arXiv (2021)
[16] De Feo, L., Galbraith, S.D.: SeaSign: Compact isogeny signatures from class group actions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11478, pp. 759–789. Springer (2019)
[17] Draisma, J., Shaw, R.: Singular lines of trilinear forms. Linear algebra and its applications **433**(3), 690–697 (2010)
[18] Faugère, J.C.: A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra **139**, 61–88 (June 1999)
[19] Faugère, J.C., Perret, L.: Polynomial equivalence problems: Algorithmic and theoretical aspects. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 30–47. Springer (2006)
[20] Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. J. ACM **38**(3), 690–728 (jul 1991). https://doi.org/10.1145/116825.116852, https://doi.org/10.1145/116825.116852
[21] Grochow, J.A., Qiao, Y.: Isomorphism problems for tensors, groups, and cubic forms: completeness and reductions (2019)
[22] Grochow, J.A., Qiao, Y., Tang, G.: Average-case algorithms for testing isomorphism of polynomials, algebras, and multilinear forms. Journal of Groups, Complexity, Cryptology **Volume 14, Issue 1** (Aug 2022). https://doi.org/10.46298/jgcc.2022.14.1.9431, https://gcc.episciences.org/9836, preliminary version appeared in STACS '21, doi:10.4230/LIPIcs.STACS.2021.38. Preprint available at arXiv:2012.01085
[23] Hulsing, A., Bernstein, D.J., Dobraunig, C., Eichlseder, M., Fluhrer, S., Gazdag, S.L., Kampanakis, P., Kolbl, S., Lange, T., Lauridsen, M.M., Mendel, F., Niederhagen, R., Rechberger, C., Rijneveld, J., Schwabe, P., Aumasson, J.P., Westerbaan, B., Beullens, W.: SPHINCS+. NIST PQC Submission (2020)
[24] Hülsing, A., Butin, D., Gazdag, S.L., Rijneveld, J., Mohaisen, A.: XMSS: extended hash-based signatures. RFC 8391 (2018)
[25] ISO (International Organization for Standardization): Information security, cybersecurity and privacy protection: Iso/iec wd 14888-4 information technology — security techniques — digital signatures with appendix — part 4: Stateful hash-based mechanisms, https://www.iso.org/standard/80492.html
[26] Joux, A., Vitse, V.: A Crossbred Algorithm for Solving Boolean Polynomial Systems. In: Kaczorowski, J., Pieprzyk, J., Pomykała, J. (eds.) Number-Theoretic Methods in Cryptology. pp. 3–21. Springer International Publishing, Cham (2018)
[27] Lazard, D.: Gröbner-Bases, Gaussian elimination and resolution of systems of algebraic equations. In: van Hulzen, J.A. (ed.) EUROCAL. Lecture Notes in Computer Science, vol. 162, pp. 146–156. Springer (1983)

[28] Leon, J.S.: Computing automorphism groups of error-correcting codes. IEEE Trans. Inf. Theory **28**(3), 496–510 (1982)
[29] Lyubashevsky, V., Ducas, L., Kiltz, E., Lepoint, T., Schwabe, P., Seiler, G., Stehlé, D., Bai, S.: CRYSTALS-DILITHIUM. NIST PQC Submission (2020)
[30] Narayanan, A.K., Qiao, Y., Tang, G.: Algorithms for matrix code and alternating trilinear form equivalences via new isomorphism invariants. Springer-Verlag (2024)
[31] NIST (National Institute for Standards and Technology): Post-Quantum Cryptography Standardization (2017), uRL: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography
[32] NIST (National Institute for Standards and Technology): Fifth PQC Standardization Conference (2024), uRL: https://csrc.nist.gov/Events/2024/fifth-pqc-standardization-conference
[33] Patarin, J.: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of asymmetric algorithms. In: EUROCRYPT '96. LNCS, vol. 1070, pp. 33–48. Springer (1996)
[34] Perret, L.: On the computational complexity of some equivalence problems of polynomial systems of equations over finite fields. Electronic Colloquium on Computational Complexity (ECCC) (116) (2004)
[35] Perret, L.: A Fast Cryptanalysis of the Isomorphism of Polynomials with One Secret Problem. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 3494, pp. 354–370. Springer (2005)
[36] Prest, T., Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: FALCON. NIST PQC Submission (2020)
[37] Ran, L., Samardjiska, S., Trimoska, M.: Algebraic algorithm for the alternating trilinear form equivalence problem. In: Esser, A., Santini, P. (eds.) Code-Based Cryptography. pp. 84–103. Springer Nature Switzerland, Cham (2023)
[38] Reijnders, K., Samardjiska, S., Trimoska, M.: Hardness estimates of the code equivalence problem in the rank metric. Designs, Codes and Cryptography **92**, 1–30 (01 2024). https://doi.org/10.1007/s10623-023-01338-x
[39] Schwabe, P., Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Seiler, G., Stehlé, D.: CRYSTALS-KYBER. NIST PQC Submission (2020)
[40] Tang, G., Duong, D.H., Joux, A., Plantard, T., Qiao, Y., Susilo, W.: Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In: EUROCRYPT 2022. LNCS, vol. 13277, pp. 582–612. Springer (2022)