Algebraic Algorithm for the Alternating Trilinear Form Equivalence Problem

Lars Ran[1], Simona Samardjiska[1], and Monika Trimoska[2]

[1] Radboud University, Nijmegen, Netherlands
{lran,simonas}@cs.ru.nl
[2] Eindhoven University of Technology, Eindhoven, Netherlands
m.trimoska@tue.nl

Abstract. The Alternating Trilinear Form Equivalence (ATFE) problem was recently used by Tang et al. as a hardness assumption in the design of a Fiat-Shamir digital signature scheme ALTEQ. The scheme was submitted to the additional round for digital signatures of the NIST standardization process for post-quantum cryptography.

ATFE is a hard equivalence problem known to be in the class of equivalence problems that includes, for instance, the Tensor Isomorphism (TI), Quadratic Maps Linear Equivalence (QMLE) and the Matrix Code Equivalence (MCE) problems. Due to the increased cryptographic interest, the understanding of its practical hardness has also increased in the last couple of years. Currently, there are several combinatorial and algebraic algorithms for solving it, the best of which is a graph-theoretic algorithm that also includes an algebraic subroutine.

In this paper, we take a purely algebraic approach to the ATFE problem, but we use a coding theory perspective to model the problem. This modelling was introduced earlier for the MCE problem. Using it, we improve the cost of algebraic attacks against ATFE compared to previously known ones.

Taking into account the algebraic structure of alternating trilinear forms, we show that the obtained system has less variables but also less equations than for MCE and gives rise to structural degree-3 syzygies. Under the assumption that outside of these syzygies the system behaves semi-regularly, we provide a concrete, non-asymptotic complexity estimate of the performance of our algebraic attack. Our results show that the complexity of our attack is below the estimated security levels of ALTEQ by more than 20 bits for NIST level I (and more for the others), thus the scheme requires reparametrization for all three NIST security levels.

Keywords: trilinear form · matrix codes · algebraic cryptanalysis

This research has been supported by the Dutch government through the NWO grant OCNW.M.21.193 (ALPaQCa) and by the European Commission through the ERC Starting Grant 805031 (EPOQUE). The work was carried out while Monika Trimoska was affiliated with Radboud University.

1 Introduction

NIST's announcement of reopening the call for post-quantum digital signature proposals specifies the need for shorter signatures whose security is based on problems outside the realm of structured lattices. One family of problems that has recently been brought to focus by the quest for alternative signatures is the family of equivalence problems. The reason behind the rising interest in these problems is that they typically can be used to construct a cryptographic group action. Once we have a cryptographic group action, the vectorization problem is used to build a Sigma protocol that, through the Fiat-Shamir transform, can be transformed into a digital signature scheme. For instance, if we take the set comprised of k-tuples of multivariate polynomials together with the group of isomorphisms acting on this set, then we obtain the cryptographic group action underlying the IP signature scheme proposed by Patarin [Pat96]. The original proposition of this scheme is based on the inhomogenous quadratic variant of the isomorphism of polynomials (IP) problem, that is, the case where the polynomials have quadratic, linear and constant terms. This subclass of IP turned out to be easy to solve in practice and hence the IP signature scheme is considered to be broken [FP06]. However, this would not be case if it were instantiated with another subclass of IP, such as the homogenous quadratic variant, also referred to as the Quadratic Maps Linear Equivalence (QMLE) problem.

Recently, as a result of several optimization techniques [DFG19, BKP20, BMPS20, BBPS21] Patarin's construction became attractive again. It was revived through two new signature schemes based on the hardness of two problems closely related to QMLE. A signature scheme based on the hardness of the alternating trilinear form equivalence (ATFE) problem was introduced at Eurocrypt 2022 [TDJ+22], whereas matrix code equivalence (MCE) was used in the more recently proposed construction called MEDS [CNP+22]. Both of these schemes, the first under the name of ALTEQ, were latter submitted to the additional round for digital signatures of the NIST standardization process for post-quantum cryptography [BDN+23, CNP+23].

Because of this attention, the understanding of the practical hardness of both ATFE and MCE also significantly improved. An adaptation of Bouillaguet et al. graph-theoretic algorithm for the IP problem [BFV13, Bou11] to the case of ATFE provides an upper bound of $\tilde{\mathcal{O}}(q^{2n/3})$ and this one was used to choose parameters for the scheme from [TDJ+22]. The authors of [TDJ+22] also analyzed the problem purely algebraically, but their model and assumptions on the obtained algebraic system gave worse estimates of $\mathcal{O}(2^{6\omega n \log_2(n)})$. They further provided a basic collision based approach similar in nature to the one in [BFV13] but looking at low rank codewords as in Leon's algorithm for the Hamming metric [Leo82, Beu20]. This basic attack was subsequently improved by Beullens [Beu22] to $\tilde{\mathcal{O}}(q^{\max(n-5)/2,n-7})$ for odd n and $\tilde{\mathcal{O}}(q^{\max(n-4)/2,n-4})$ for even n. For some special cases of weak keys, even better results were presented leading to practical polynomial time attacks. If such weak keys are avoided, the attack performs better in the odd n case. Later, Beullens' attack [Beu22] was used as the basis for setting parameters for ALTEQ [BDN+23].

Similar approaches were taken into account for the MCE problem which was analyzed in [RST22, CNP+22]. In [RST22], Bouillaguet et al.'s algorithm was transformed into an algorithm of complexity $\tilde{\mathcal{O}}(q^{4n/3})$. Using a different property for building the graph, the authors proposed an improvement resulting in a complexity of $\tilde{\mathcal{O}}(q^n)$. Currently the best algorithms against MCE were developed in [CNP+22, CNP+23] and they take nontrivial approaches in adapting Leon's algorithm to the rank metric and modeling the problem algebraically but from a coding theory viewpoint. The focus of this work is related to this improved algebraic modeling.

Our contribution. In this work, we take advantage of the relation between the two equivalence problems – ATFE and MCE to improve the cryptanalysis of ATFE. Namely, an alternating trilinear form can easily be represented as a matrix code. A reduction from ATFE to MCE directly follows from the reduction results in [TDJ+22] and [RST22]. Theorem 2 in [TDJ+22] states that ATFE is tensor isomorphism complete and thus equivalent to QMLE and Theorem 11 in [RST22] shows a reduction from QMLE (the bilinear case) to MCE. Specifically, an MCE instance with a pair of matrix codes derived from a positive ATFE instance, has a solution of the form $(\mathbf{A}^\top, \mathbf{A})$, where the matrix $\mathbf{A}$ is a solution to the original ATFE instance.

Viewing ATFE as a problem on matrix codes enables us to model the problem using coding theory techniques. In particular, we model ATFE algebraically in a nontrivial way using the approach from [CNP+22] for MCE. This model improves the cost of an algebraic attack compared to previously known models as for example described in [TDJ+22] and the ALTEQ specifications [BDN+23].

Taking into account the algebraic structure of alternating trilinear forms, we show that the obtained system has less variables but also less equations than for MCE. In particular we can model ATFE as a system of $n(\binom{n}{2} - n)$ equations in n^2 variables. For our complexity analysis, we first show the existence of $\frac{(n+1)(n-1)(n-3)}{3}$ structural degree-3 syzygies in such systems. Then, under the assumption that outside of these syzygies the system behaves semi-regularly, we show that the complexity is below the estimated security levels of the signature scheme in [TDJ+22] and the proposed parameter sets of ALTEQ [BDN+23] for all three NIST security levels. Our results for the parameters proposed in [TDJ+22] and [BDN+23] are given in Table 1. Hence in order to attain the claimed NIST security levels, the parameter n of ALTEQ needs to be increased.

Organization. The paper is organized as follows. Section 2 introduces the necessary preliminaries and Section 3 reviews state-of-the-art algorithms for solving ATFE. In Section 4, we show how a positive ATFE instance is transformed into a positive MCE instance and we explore the structure of the matrix codes obtained from this transformation. In Section 5 we show algebraic modellings for the ATFE problem and in Section 6 we give a complexity analysis for solving the systems from our proposed variant. Finally, in Section 7 we show our experimental results.

Table 1. Comparison of the concrete complexities (in $\log_2$ scale) of different algorithms for solving ATFE. The superscript ^{a} means that the estimate was obtained from an algebraic model, and ^{b} from a graph-based birthday model. The column 'Our work' includes the cost of field operations estimated as $O(\lceil \log_2 q \rceil^2)$ bit operations.

NIST Sec. level	n	q	Tang et al. [TDJ+22]	Beullens [Beu22]	ALTEQ specs. [BDN+23]	**Our work**
—	9	$2^{18}-1$	133	38	—	99
—	10	$2^{17}-1$	133	122	—	105
—	11	$2^{16}-5$	138	85	—	109
I	13	$2^{32}-5$	—	—	143^a	**120**
III	20	$2^{32}-5$	—	—	219^a	**165**
V	25	$2^{32}-5$	—	—	252^b	**203**

2 Preliminaries

Let $\mathbb{F}_q$ be the finite field of q elements. $\mathrm{GL}_n(q)$ and $\mathrm{AGL}_n(q)$ denote respectively the general linear group and the general affine group of degree n over $\mathbb{F}_q$. We use bold letters to denote vectors $\mathbf{a}, \mathbf{c}, \mathbf{x}, \ldots$, and matrices $\mathbf{A}, \mathbf{B}, \ldots$. The entries of a vector $\mathbf{a}$ are denoted by a_i, and we write $\mathbf{a} = (a_1, \ldots, a_n)$ for a (column) vector of dimension n over some field. Similarly, 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 $\mathbb{F}_q^n$. If $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is a basis for a vector space, we denote by $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ the corresponding dual basis. We denote by $\mathcal{S}_n$ the symmetric group of degree n. Finally, we denote the set of all $m \times n$ matrices over $\mathbb{F}_q$ by $\mathcal{M}_{m,n}(\mathbb{F}_q)$.

Cryptographic group actions.

Definition 1. Let X be a set and $(G, \cdot)$ be a group. A group action is a mapping

$$\begin{aligned} \star : G \times X &\to X \\ (g, x) &\mapsto g \star x \end{aligned}$$

such that the following conditions hold for all $x \in X$:

- $e \star x = x$, where e is the identity element of G.
- $g_2 \star (g_1 \star x) = (g_2 \cdot g_1) \star x$, for all $g_1, g_2 \in G$.

A *cryptographic* group action commonly refers to a group action that has some additional properties that are useful for cryptographic applications. To begin with, there are some desirable properties of computational nature. Namely, the following procedures should be efficient:

- *Evaluation*: given x and g, compute $g \star x$.
- *Sampling*: sample uniformly at random from G.
- *Membership testing*: verify that $x \in X$.

The crucial property that distinguishes cryptographic group actions is that the corresponding *vectorization problem* should be hard:

Problem 1. $\mathsf{GroupActionVectorization}(X, x_1, x_2)$:
Input: The pair $x_1, x_2 \in X$.
Question: Find – if any – $g \in G$ such that $g \star x_1 = x_2$.

Early constructions using this paradigm are based on the action of finite groups of prime order, for which the vectorization problem is the discrete logarithm problem. Notable isogeny-based constructions can be found, for instance, in the work of Couveignes in [Cou06] and later by Rostovtsev and Stolbunov [RS06]. Recently, a general framework based on group actions was explored in more detail by [ADFMP20], allowing for the design of several primitives.

The Alternating Trilinear Form Equivalence problem. A k-linear form is a function $\phi : \mathbb{F}_q^n \times \cdots \times \mathbb{F}_q^n \to \mathbb{F}_q$ that is linear in each argument: if we fix $k-1$ arguments, it is linear in the remaining argument. A k-linear form is called

- *symmetric*: if $\phi(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \phi(\mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(k)})$ for any permutation $\pi \in \mathcal{S}_k$;
- *skew-symmetric*: if $\phi(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \phi(\mathbf{x}_{\tau(1)}, \ldots, \mathbf{x}_{\tau(k)})$ for any transposition $\tau \in \mathcal{S}_k$;
- *alternating* if $\phi(\mathbf{x}_1, \ldots, \mathbf{x}_k) = 0$ whenever $\mathbf{x}_i = \mathbf{x}_j$ for some $i \neq j$.

Every alternating form is skew-symmetric, and if $q \geq 3$, every skew-symmetric form is alternating. In the following, we will focus on the $k = 2$ and $k = 3$ cases: bilinear and trilinear forms.

An alternating trilinear form can be represented as $\sum_{1 \leqslant i < j < s \leqslant n} c_{ijs}(\mathbf{e}_i^* \wedge \mathbf{e}_j^* \wedge \mathbf{e}_s^*)$, where $c_{ijs} \in \mathbb{F}_q$, $\mathbf{e}_i$ is the ith canonical basis vector, $\mathbf{e}_i^*$ is the linear form sending $u = (u_1, ..., u_n) \in \mathbb{F}_q^n$ to u_i and $\wedge$ denotes the wedge product. Hence, $\mathbf{e}_i^* \wedge \mathbf{e}_j^* \wedge \mathbf{e}_s^*$ is an alternating form sending $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ to the determinant $\begin{vmatrix} x_i & y_i & z_i \\ x_j & y_j & z_j \\ x_s & y_s & z_s \end{vmatrix}$. From this representation it is clear that an alternating trilinear form can be stored using $\binom{n}{3}$ entries: one for each coefficient c_{ijs}.

The alternating trilinear form equivalence problem is formally defined as follows:

Problem 2. $\mathsf{ATFE}(n, \phi, \psi)$:
Input: Two alternating trilinear forms ϕ, ψ.
Question: Find – if any – $\mathbf{A} \in \mathrm{GL}_n(q)$ such that $\psi(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \phi(\mathbf{Ax}, \mathbf{Ay}, \mathbf{Az})$.

The ATFE-based group action is defined by the action of the general linear group $\mathrm{GL}_n(q)$ on the set of all alternating trilinear forms defined over $\mathbb{F}_q^n$. The vectorization problem is the ATFE problem defined above. Since ATFE is a hard problem, we obtain a cryptographic group action.

Array representation of bilinear and trilinear forms. It is common to represent a bilinear form as $\mathbf{x}^\top \mathbf{My}$, where $\mathbf{M}$ is a matrix where the (i, j) entry holds the coefficient of the term $x_i y_j$. Similarly, trilinear forms can be represented

with a 3-way array where the (i, j, s) entry holds the coefficient of $x_i y_j z_s$. In this representation, we implicitly choose $\mathbf{e}_1, \ldots, \mathbf{e}_n$ as a basis for $\mathbb{F}_q^n$. Alternating bilinear and trilinear forms can be represented in such a way, although it is not the most efficient representation. The array representation of an alternating bilinear form is a skew-symmetric matrix with zeros on the main diagonal. The array representation of a trilinear form has even more redundancy. Notice from the 'determinant representation' above that for all permutations of the index triple (i, j, s), the terms $x_i y_j z_s$ have the same coefficient, up to sign. Specifically, if we denote by M_{ijs} the (i, j, s) entry of the 3-way array, then $M_{ijs} = -M_{isj} = M_{sij} = -M_{jis} = M_{jsi} = -M_{sji}$. This is the key property that makes all of the terms cancel out (and hence the form evaluate to zero) whenever two arguments are the same.

The Matrix Code Equivalence problem. A *matrix code* is a subspace $\mathcal{C}$ of $m \times n$ matrices over $\mathbb{F}_q$ endowed with the rank metric defined as $d(\mathbf{A}, \mathbf{B}) = \mathrm{Rank}(\mathbf{A} - \mathbf{B})$. We denote by k the dimension of $\mathcal{C}$ as a subspace of $\mathbb{F}_q^{m \times n}$ and its basis by $(\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(k)})$ where $\mathbf{C}^{(i)} \in \mathbb{F}_q^{m \times n}$ are linearly independent.

The matrix code equivalence problem is formally defined as follows:

Problem 3. $\mathsf{MCE}(k, n, m, \mathcal{C}, \mathcal{D})$:
Input: Two k-dimensional matrix codes $\mathcal{C}, \mathcal{D} \subset \mathcal{M}_{m,n}(\mathbb{F}_q)$.
Question: Find – if any – $\mathbf{A} \in \mathrm{GL}_m(q), \mathbf{B} \in \mathrm{GL}_n(q)$ such that for all $\mathbf{C} \in \mathcal{C}$, it holds that $\mathbf{ACB} \in \mathcal{D}$.

Algebraically, the MCE problem corresponds to the problem of finding the unknown entries of matrices $\mathbf{A}, \mathbf{B}, \mathbf{T}$ such that

$$\mathbf{D}^{(i)} = \sum_{1 \leqslant j \leqslant n} t_{ji} \mathbf{A} \mathbf{C}^{(j)} \mathbf{B}, \quad \forall i, 1 \leqslant i \leqslant n$$

is satisfied. The matrix $\mathbf{T} \in \mathrm{GL}_k(q)$ corresponds to a change of basis of $\mathbf{A}\mathcal{C}\mathbf{B}$.

The MCE problem also gives rise to a group action: the group $\mathrm{GL}_m(q) \times \mathrm{GL}_n(q)$ acts on the set formed by the k-dimensional matrix codes of size $m \times n$ over the base field $\mathbb{F}_q$. The vectorization problem is MCE, and since this is a hard problem, we obtain a cryptographic group action.

Exterior powers and extending trilinear forms. For combinatorial analysis it can be useful to work with linear maps instead of trilinear maps. To this end we introduce, for every k, the exterior powers of a vector space. These are vector spaces generated by wedge products:

$$\bigwedge^k \mathbb{F}_q^n := \{\sum_i (\mathbf{x}_1)_i \wedge \ldots \wedge (\mathbf{x}_k)_i \mid (\mathbf{x}_j)_i \in \mathbb{F}_q^n\}.$$

These vector spaces have dimension $\binom{n}{k}$. Furthermore, linear transformations $\mathbf{A} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ also act on $\bigwedge^k \mathbb{F}_q^n$ by

$$\mathbf{A}(\mathbf{x}_1 \wedge \ldots \wedge \mathbf{x}_k) = \mathbf{A}\mathbf{x}_1 \wedge \ldots \wedge \mathbf{A}\mathbf{x}_k.$$

Now each alternating k-linear form $\phi : \mathbb{F}_q^n \times \ldots \times \mathbb{F}_q^n \to \mathbb{F}_q$ can be extended to a linear form $\hat{\phi} : \bigwedge^k \mathbb{F}_q^n \to \mathbb{F}_q$ where the map is given by:

$$\hat{\phi}\left(\sum_i (\mathbf{x}_1)_i \wedge \ldots \wedge (\mathbf{x}_k)_i\right) = \sum_i \phi\left((\mathbf{x}_1)_i, \ldots, (\mathbf{x}_k)_i\right).$$

This extension is unique and is in fact a natural bijection between k-linear forms and linear forms on the kth exterior power. Therefore we will abuse notation and write ϕ for both maps. The number of arguments will indicate what is meant.

This can also be used to partly linearize a k-linear form in the first l arguments. In this case, an alternating k-linear form $\phi : \mathbb{F}_q^n \times \ldots \times \mathbb{F}_q^n \to \mathbb{F}_q$ can be extended to a $(k-l+1)$-linear form

$$\begin{aligned} \hat{\phi} : \bigwedge^l \mathbb{F}_q^n \times \overbrace{\mathbb{F}_q^n \times \ldots \times \mathbb{F}_q^n}^{k-l \text{ times}} &\to \mathbb{F}_q \\ (\mathbf{x}_1 \wedge \ldots \wedge \mathbf{x}_l, \mathbf{x}_{l+1}, \ldots \mathbf{x}_k) &\mapsto \phi(\mathbf{x}_1, \ldots \mathbf{x}_k). \end{aligned}$$

This extension is again unique. Note that this extension has arguments from different spaces so it is not alternating any more. We will again denote both forms by ϕ, the number and type of arguments should indicate what is meant. For our use case, $k = 3$, this implies the following equations:

$$\phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \phi(\mathbf{x} \wedge \mathbf{y}, \mathbf{z}) = \phi(\mathbf{x} \wedge \mathbf{y} \wedge \mathbf{z}).$$

For a more thorough treatment on exterior powers, alternating forms and multilinear algebra in general we refer the reader to [Gre12].

3 Previous algorithms for solving ATFE

The state-of-the-art algorithms against ATFE build upon relatively old algorithms against the Isomorphism of polynomials (IP) [Per05, BFFP11, BFV13]. We present the two most relevant below.

3.1 Graph-theoretic algorithm of Bouillaguet et al. [BFV13]

More than 10 years ago, Bouillaguet et al.[BFV13] proposed a birthday-based graph-theoretic algorithm for solving the Quadratic Maps Linear Equivalence (QMLE) problem. It is now known that the ATFE problem is polynomial-time equivalent to the homogeneous version of QMLE [GQ21] implying that this algorithm can be adapted for ATFE.

Specifically, two isomorphic alternating trilinear forms ϕ and ψ over $\mathbb{F}_q^n$ can be seen as two equivalent homogeneous quadratic maps $\mathcal{F}$ and $\mathcal{P}$ of n multivariate polynomials in n variables over $\mathbb{F}_q$. Furthermore, these quadratic maps are alternating and bilinear, so they have a skew-symmetric matrix representation. The main observation of the algorithm is that once a pair of vectors $\mathbf{u}, \mathbf{v} \in \mathbb{F}_q^n$

is known such that $\mathbf{u} = \mathbf{A}\mathbf{v}$, this information is enough to find the isomorphism with low complexity [3]. Hence, the goal of the algorithm is to find this collision of points, and different invariants under isomorphism can be used to achieve this.

For the case of ATFE, a useful invariant is the rank of the corresponding bilinear form $\phi_{\mathbf{v}}(\mathbf{w}, \mathbf{z}) = \phi(\mathbf{v}, \mathbf{w}, \mathbf{z})$ which is preserved under the isomorphism defined by $\mathbf{A}$. The algorithm now proceeds as a standard collision-search algorithm in two steps: First, create lists L_ϕ and L_ψ of size $\mathcal{O}(q^{n/3})$ elements in $\mathbb{F}_q^n$ of the same rank. Then, find a collision between these lists by calling the efficient algorithm described above. The total complexity amounts to $\tilde{\mathcal{O}}(q^{2n/3})$ where we neglect the estimated $\mathcal{O}(n^9)$ cost of finding the isomorphism once one collision is known.

3.2 Graph-theoretic algorithm of Beullens [Beu22]

Beullens [Beu22] improves generically upon the previous approach by further using clever graph-walking techniques. The basic idea is to populate the lists faster by exploiting the structure of a particular invariant graph for alternating trilinear forms. This graph had been studied before and was used for complete classification of trilinear forms of dimensions $n = 8, 9$. Namely, the structure of the graph allows to find points of the same or lower rank in the neighborhood of an identified point of a specified rank in polynomial time. Thus, one can first find using brute force a point of higher rank (which is easier than finding one of lower rank), and then by exploring the neighborhood can find points of lower rank faster. In total, this costs $\tilde{\mathcal{O}}(q^{(n-5)/2})$ for odd n and $\tilde{\mathcal{O}}(q^{(n-4)/2})$ for even n. The second part of the algorithm is as previous and consists of matching each pair in the lists and checking whether it leads to the unknown isomorphism. This part has a complexity of $\tilde{\mathcal{O}}(q^{n-7})$ for odd n and $\tilde{\mathcal{O}}(q^{n-4})$ for even n, and for larger n, it becomes the dominating part of the algorithm.

4 A coding theory perspective of ATFE

A trilinear form can be seen as a matrix code and the other way around.

For an informal argument for the equivalence between these two objects, we refer to their algorithmic representation. A matrix code is usually represented by an array of the matrices forming its basis. This is a 3-way array, no different than a 3-way array representing a trilinear form as described in Section 2. It is then evident that we can obtain a matrix code from an (alternating) trilinear form simply by choosing a basis for the code.

Indeed, let $\phi^{(i)}(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}, \mathbf{y}, \mathbf{e}_i)$ be the bilinear form obtained by fixing the third argument of a trilinear form ϕ to $\mathbf{e}_i$, where $\mathbf{e}_i$ denotes the ith vector of the

[3] In [BFV13] it was conjectured that this complexity is $\mathcal{O}(n^9)$ i.e. polynomial. Later in [Bou11, RST22] this was reevaluated and shown that the conclusion was made based on some false assumptions. Nevertheless, even though there is no proof of the polynomial behavior of this step, in practice it does finish in an expected polynomial time.

canonical basis $(\mathbf{e}_1, \ldots, \mathbf{e}_n)$. With respect to this basis, a vector $\mathbf{a} = \sum \alpha_i \mathbf{e}_i$ can be written as $\mathbf{a} = (\alpha_1, \ldots, \alpha_n)$. If ϕ is alternating, then $\phi^{(i)}$ is also alternating and it holds that $\phi^{(i)}(-, \mathbf{e}_i) = \phi^{(i)}(\mathbf{e}_i, -) = 0$. Let $\mathbf{C}^{(i)}$ be the matrix representation of $\phi^{(i)}$. Then, $(\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(n)})$ is a basis of an n-dimensional matrix code. The only piece left is to show the relation between the solutions of two such related instances. Specifically, we show the following.

Lemma 1. *Finding a solution of the form* $(\mathbf{A}^\top, \mathbf{A})$ *to an* MCE *instance derived from an* ATFE *instance is equivalent to finding a solution* $\mathbf{A}$ *to the original* ATFE *instance.*

Proof. Let (n, ϕ, ψ) be an instance of ATFE and let $\mathcal{C}$ and $\mathcal{D}$ be matrix codes obtained by applying the above transformation to ϕ and ψ respectively. If (n, ϕ, ψ) is a positive instance of ATFE, then there exists $\mathbf{A} \in \mathrm{GL}_n(q)$ such that $\psi^{(i)}(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{x}, \mathbf{y}, \mathbf{e}_i) = \phi(\mathbf{Ax}, \mathbf{Ay}, \mathbf{Ae}_i)$ for all $i \in \{1, \ldots, n\}$. Since $\mathbf{Ae}_i = (a_{1i}, \ldots, a_{ni})$, we have that $\psi^{(i)}(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{Ax}, \mathbf{Ay}, a_{1i}\mathbf{e}_1 + \cdots + a_{ni}\mathbf{e}_n)$. By linearity, we infer that $\psi^{(i)}(\mathbf{x}, \mathbf{y}) = \sum_{1 \leqslant j \leqslant n} a_{ji}\phi(\mathbf{Ax}, \mathbf{Ay}, \mathbf{e}_j) = \sum_{1 \leqslant j \leqslant n} a_{ji}\phi^{(j)}(\mathbf{Ax}, \mathbf{Ay})$. This can be rewritten in matrix form as $\mathbf{x}^\top \mathbf{D}^{(i)} \mathbf{y} = \sum_{1 \leqslant j \leqslant n} a_{ji}(\mathbf{Ax})^\top \mathbf{C}^{(j)}(\mathbf{Ay}), \ \ \forall i, 1 \leqslant i \leqslant n.$ Since this holds for any $(\mathbf{x}, \mathbf{y})$, we have that

$$\mathbf{D}^{(i)} = \sum_{1 \leqslant j \leqslant n} a_{ji}\mathbf{A}^\top \mathbf{C}^{(j)} \mathbf{A}, \ \ \forall i, 1 \leqslant i \leqslant n. \tag{1}$$

Taking $(\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(n)})$ as a basis of a matrix code $\mathcal{C}$ and $(\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(n)})$ as a basis of a matrix code $\mathcal{D}$, from Equation (1) we infer that

- The codes $\mathcal{C}$ and $\mathcal{D}$ are equivalent up to a change of basis represented by the matrix $\mathbf{A}$.
- $(\mathbf{A}^\top, \mathbf{A})$ is a solution to the MCE instance $(n, n, n, \mathcal{C}, \mathcal{D})$. □

Example 1. Let

$$\begin{aligned}\phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) = {} & x_2y_3z_1 + 3x_2y_4z_1 + 6x_3y_2z_1 + 6x_3y_4z_1 + 4x_4y_2z_1 + x_3y_4z_1 + \\ & + 6x_1y_3z_2 + 4x_1y_4z_2 + x_3y_1z_2 + 6x_3y_4z_2 + 3x_4y_1z_2 + x_4y_3z_2 + x_1y_2z_3 + \\ & + x_1y_4z_3 + 6x_2y_1z_3 + x_2y_4z_3 + 6x_4y_1z_3 + 6x_4y_2z_3 + 3x_1y_2z_4 + 6x_1y_3z_4 + \\ & + 4x_2y_1z_4 + 6x_2y_3z_4 + x_3y_1z_4 + x_3y_2z_4\end{aligned}$$

and

$$\begin{aligned}\psi(\mathbf{x}, \mathbf{y}, \mathbf{z}) = {} & 6x_2y_3z_1 + 6x_2y_4z_1 + x_3y_2z_1 + x_4y_2z_1 + x_1y_3z_2 + x_1y_4z_2 + \\ & + 6x_3y_1z_2 + 6x_3y_4z_2 + 6x_4y_1z_2 + x_4y_3z_2 + 6x_1y_2z_3 + x_2y_1z_3 + x_2y_4z_3 + \\ & + 6x_4y_2z_3 + 6x_1y_2z_4 + x_2y_1z_4 + 6x_2y_3z_4 + x_3y_2z_4\end{aligned}$$

be two equivalent alternating trilinear forms over $\mathbb{F}_7$. The terms that are redundant in a compact representation are written in green. An isomorphism between

these two forms is, for instance,

$$\mathbf{A} = \begin{pmatrix} 6\,4\,5\,1 \\ 2\,0\,2\,0 \\ 1\,2\,6\,2 \\ 5\,6\,6\,1 \end{pmatrix}.$$

The corresponding codes are

$$\mathcal{C} = \left(\begin{pmatrix} 0\,0\,0\,0 \\ 0\,0\,1\,3 \\ 0\,6\,0\,6 \\ 0\,4\,1\,0 \end{pmatrix}, \begin{pmatrix} 0\,0\,6\,4 \\ 0\,0\,0\,0 \\ 1\,0\,0\,6 \\ 3\,0\,1\,0 \end{pmatrix}, \begin{pmatrix} 0\,1\,0\,1 \\ 6\,0\,0\,1 \\ 0\,0\,0\,0 \\ 6\,6\,0\,0 \end{pmatrix}, \begin{pmatrix} 0\,3\,6\,0 \\ 4\,0\,6\,0 \\ 1\,1\,0\,0 \\ 0\,0\,0\,0 \end{pmatrix} \right)$$

and

$$\mathcal{D} = \left(\begin{pmatrix} 0\,0\,0\,0 \\ 0\,0\,6\,6 \\ 0\,1\,0\,0 \\ 0\,1\,0\,0 \end{pmatrix}, \begin{pmatrix} 0\,0\,1\,1 \\ 0\,0\,0\,0 \\ 6\,0\,0\,6 \\ 6\,0\,1\,0 \end{pmatrix}, \begin{pmatrix} 0\,6\,0\,0 \\ 1\,0\,0\,1 \\ 0\,0\,0\,0 \\ 0\,6\,0\,0 \end{pmatrix}, \begin{pmatrix} 0\,6\,0\,0 \\ 1\,0\,6\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,0 \end{pmatrix} \right).$$

We can check that $(\mathbf{A}^\top, \mathbf{A})$ is an isometry from $\mathcal{C}$ to $\mathcal{D}$. Note that for such small parameters ($n = 4$), there are probably many isometries from $\mathcal{C}$ to $\mathcal{D}$.

The codes $\mathcal{C}$ and $\mathcal{D}$ have several properties intrinsic to their derivation from alternating trilinear forms. For simplicity, we discuss all of them assuming the choice of basis specified in the beginning of this section. All of the matrices forming the basis of $\mathcal{C}$ are skew-symmetric with zeros on the main diagonal, hence they are all of even rank. More generally, we have the following relations between their entries: $C^{(s)}_{ij} = -C^{(j)}_{is} = C^{(j)}_{si} = -C^{(s)}_{ji} = C^{(i)}_{js} = -C^{(i)}_{sj}$. The same holds for the basis of $\mathcal{D}$. The ith column and the ith row is zero in the ith matrix of the basis, that is, the matrix corresponding to the bilinear form $\phi^{(i)}(\mathbf{x}, \mathbf{y})$ (resp. $\psi^{(i)}(\mathbf{x}, \mathbf{y})$ for $\mathcal{D}$). These zero column and row vectors, as well as the zeros on the diagonal, result from the property that in an alternating trilinear form, the coefficient of a term $x_i y_j z_s$ is zero if any two of the three indices (i, j, s) are the same. Finally, positive MCE instances derived from positive ATFE instances have a specific solution. Instead of a pair of unrelated matrices, we have a solution $(\mathbf{A}, \mathbf{B})$ such that $\mathbf{A} = \mathbf{B}^\top$. Hence ATFE can be reduced to a subclass of MCE.

5 Algebraic algorithms for solving ATFE

In view of the connection of ATFE to MCE we continue to use the matrix code representation introduced in the previous section.

5.1 Direct modelling

A straightforward way to model this problem algebraically is to describe Equation (1) as a system of $n \cdot \binom{n}{2}$ equations in n^2 variables, corresponding to the

coefficients of $\mathbf{A}$. The resulting system is of degree three. Alternatively, we can move one linear transformation to the other side of the equality and obtain

$$\sum_{1 \leqslant j \leqslant n} \tilde{a}_{ji} \mathbf{D}^{(i)} = \mathbf{A}^\top \mathbf{C}^{(j)} \mathbf{A}, \quad \forall i, 1 \leqslant i \leqslant n, \tag{2}$$

where $\tilde{a}_{ji}$ is the (j, i) entry of $\mathbf{A}^{-1}$. When we rewrite the system like this, the number of equations does not change and we double the number of variables, but we obtain an inhomogenous quadratic system instead of a cubic one. Specifically, the system is quadratic in the $\mathbf{A}$-variables and linear in the $\mathbf{A}^{-1}$-variables. We add to this the constraint $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$, which yields n^2 equations that are bilinear in $\mathbf{A}$-variables and $\mathbf{A}^{-1}$-variables. We will refer to this approach as the *direct* modelling. The direct modelling dates back to the work in [FP06] for solving the MQMLE problem, with further analysis in [BFV13, Bou11]. Recently, it was analysed as a modelling for MCE in [CNP+22], before a more advanced approach was introduced. The work in this paper shows that the improved modelling introduced in [CNP+22] is even more relevant in the ATFE case. We describe this approach in the following subsection.

A similar modelling was used in [TDJ+22] for the analysis of an algebraic attack on ATFE. In fact, with the algebraic modelling in [TDJ+22] we obtain a subset of the equations in the system arising from Equation (2). Due to the compact representation of ATFE, the number of equations is $\binom{n}{3} + n^2$, which is less than the $n \cdot \binom{n}{2} + n^2$ equations that we obtain from the corresponding matrix representation. The complexity of this approach is analysed under the assumption that the polynomials in the system form a semi-regular sequence. Using the analysis techniques from [Bar04, BFSY05], the degree of regularity is estimated to be $3n$ asymptotically, and the complexity is upper-bounded by $\mathcal{O}(N^{3n\omega})$, where $N = 2n^2$ is the number of variables and ω is the linear algebra constant.

In [BDN+23], the direct modelling is improved by adding the equations arising from

$$\sum_{1 \leqslant j \leqslant n} a_{ji} \mathbf{C}^{(i)} = (\mathbf{A}^{-1})^\top \mathbf{D}^{(j)} \mathbf{A}^{-1}, \quad \forall i, 1 \leqslant i \leqslant n,$$

and also $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$. This is called the *quadratic with inverse* modelling and results in a system of $2n(\binom{n}{2} + n)$ equations in $2n^2$ variables. In [BDN+23], it is used as reference for calculating the complexity of an algebraic attack on the ATFE problem.

5.2 Improved matrix-code modelling

The improved modelling uses ideas from coding theory and its greatest advantage is that all variables that occur linearly in the direct modelling are not present in the improved system. In this description of the modelling, we will focus on MCE instances derived from ATFE instances. For these instances, we obtain a polynomial system in n^2 variables, which is a significant improvement over the system with $2n^2$ variables obtained from the direct modelling.

Let $\mathbf{G}$ and $\mathbf{G}'$ be the $n \times n^2$ generator matrices of $\mathcal{C}$ and $\mathcal{D}$ respectively. These generator matrices are obtained by *flattening* the matrix code, in the following manner. For a matrix $\mathbf{C} \in \mathcal{M}_{n,n}(\mathbb{F}_q)$, let vec be a mapping that sends a matrix $\mathbf{C}$ to the vector $\mathsf{vec}(\mathbf{C}) \in \mathbb{F}_q^{n^2}$ obtained by:

$$\mathsf{vec}: \ \mathbf{C} = \begin{pmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,n} \end{pmatrix} \mapsto \mathsf{vec}(\mathbf{C}) = (a_{1,1}, \dots, a_{1,n}, \dots, a_{n,1}, \dots, a_{n,n}).$$

Then $\mathbf{G}$ is constructed as follows

$$\mathbf{G} \coloneqq \begin{pmatrix} \mathsf{vec}(\mathbf{C}_1) \\ \vdots \\ \mathsf{vec}(\mathbf{C}_n) \end{pmatrix}.$$

The representation using generator matrices constructed as above allows us to view a matrix code as an $\mathbb{F}_q$-subspace of $\mathbb{F}_q^{n^2}$. We can now describe the improved modelling in three steps:

- Compute $\mathbf{G}'^{\perp}$, that is, the generator matrix of the dual code of $\mathcal{D}$. This is an $(n^2-n) \times n^2$ matrix containing only constant values, and it can be computed directly from $\mathbf{G}'$.
- Compute $\tilde{\mathbf{G}}$, that is, a generator matrix of $\mathcal{D}$ represented as $\mathbf{A}^\top \mathcal{C} \mathbf{A}$ for $\mathbf{A}$ with unknown coefficients. This is an $n \times n^2$ matrix whose entries are quadratic equations in the $\mathbf{A}$-variables. It can be obtained either by computing matrices $\mathbf{A}^\top \mathbf{C}_i \mathbf{A}$ and flattening them to obtain the rows of $\tilde{\mathbf{G}}$, or by computing $\tilde{\mathbf{G}} = \mathbf{G}(\mathbf{A} \otimes \mathbf{A})$.
- Construct the system
$$\mathbf{G}'^{\perp} \cdot \tilde{\mathbf{G}}^\top = \mathbf{0}_{(n^2-n) \times n}. \tag{3}$$

Note that the system obtained from Equation (3) has $n(n^2 - n)$ equations, but only $n(\binom{n}{2} - n)$ of them are linearly independent because of the specific structure of matrix codes obtained from alternating trilinear forms. Recall from Section 4 that we have the following relations between the entries of the matrices from the basis: $C_{ij}^{(s)} = -C_{is}^{(j)} = C_{si}^{(j)} = -C_{ji}^{(s)} = C_{js}^{(i)} = -C_{sj}^{(i)}$. This shows that any generator matrix $\mathbf{G}$ of a matrix code derived from an alternating trilinear form has $\binom{n}{2}$ linearly independent columns. For an alternative view of this modelling that is in the spirit of the minors modellings of MinRank [FdVP08, BBC+20], we refer the reader to [CNP+22].

5.3 Removing invalid solutions

One drawback of the improved modelling is that it does not contain the constraint that the solution $\mathbf{A}$ has to be an invertible matrix. As a consequence, the polynomial system can have solutions that do not correspond to solutions to the ATFE instance, and this effect can significantly slow down the resolution of

the system. Note that the direct modelling does not have this problem because there are equations describing $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$.

As an example for invalid solutions we show that all rank-1 matrices $\mathbf{A}$ are a solution to the improved modelling as is. Let $\mathbf{A} = \mathbf{a}\mathbf{b}^\top$, then $\mathbf{A}^\top\mathbf{C}_i\mathbf{A} = \mathbf{b}\mathbf{a}^\top\mathbf{C}_i\mathbf{a}\mathbf{b}^\top$. But we know that $\mathbf{C}_i$ is skew-symmetric, hence $\mathbf{a}^\top\mathbf{C}_i\mathbf{a} = 0$. After flattening, $\tilde{G} = \mathbf{0}$ and our system is trivially satisfied.

In the following, we show how we can add the constraint that $\mathbf{A}$ has to be invertible to the improved modelling and remove the *invalid* solutions without introducing new variables.

First, we take some equations from the system in Equation (2) and use them to express $\mathbf{A}^{-1}$ in terms of $\mathbf{A}$. This is possible because the variables of $\mathbf{A}^{-1}$ appear only linearly and there are more that n^2 equations in the system. Specifically, we build the Macaulay matrix of the system, choosing an ordering such that the linear $\mathbf{A}^{-1}$-variables correspond to the leading columns. Then, we find the reduced row echelon form and take the first n^2 equations. They all contain only one linear $\mathbf{A}^{-1}$-variable, so the variable can be expressed as a quadratic equation in $\mathbf{A}$-variables. We use these terms to substitute the $\mathbf{A}^{-1}$-variables in the system corresponding to $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$. This approach yields $n^2 - n$ homogeneous and n inhomogeneous cubic equations in the $\mathbf{A}$-variables, that we add to the system derived from Equation (3).

Since the new equations are all cubic, they do not influence greatly the asymptotic complexity of solving the system using a Gröbner basis algorithm like F4 [Fau99]. However, they are useful for eliminating the invalid solutions and they improve the running times for practical sizes. Hence, we use these equations in our experimental work, but we do not consider them in the complexity analysis in Section 6, or rather, we assume that they can only improve the solving complexity. It is commonly known that adding equations improves the solving time of Gröbner basis algorithms, and our experiments (in Section 7) show that this holds true for our case. In conclusion, we consider the following complexity analysis to be an upper bound, and, asymptotically, we do not expect it to differ a lot from the complexity analysis that includes the added cubic equations.

6 Complexity analysis

The system obtained from Equation (3) is a quadratic system of $n \cdot (\binom{n}{2} - n) = n^2 \cdot \frac{n-3}{2}$ equations in n^2 variables. With the assumption that this system is semi-regular, the asymptotic behavior of the degree of regularity can be estimated using [BFSY05]. Then, with $\alpha = \frac{n-3}{2}$, the resulting degree of regularity would grow as $d_{reg} \sim \frac{n}{4}$. However, as we will shortly see, the system is not semi-regular.

6.1 Non-trivial syzygies

The exterior powers described in Section 2 hold a lot of extra structure. These will allow us to find extra syzygies in our system. Consider the following vector

space:

$$L(\phi) := \{\omega \in \bigwedge^2 \mathbb{F}_q^n \mid \phi(\omega, z) = 0,\ \forall z \in \mathbb{F}_q^n\}.$$

This vector space can also be realized as the kernel of the following map:

$$\bigwedge^2 \mathbb{F}_q^n \to \mathbb{F}_q^n, \quad \omega \mapsto \begin{bmatrix} \phi(\omega, e_1) \\ \vdots \\ \phi(\omega, e_n) \end{bmatrix}.$$

This vector space enables a different perspective on the improved matrix-code modelling. The system described in Equation (3) is also generated by

$$\{\phi_i(\mathbf{A}\omega) \mid 1 \leq i \leq n,\ \forall \omega \in L(\psi)\}. \tag{4}$$

Let us now consider the degree-3 elements of the ideal generated by the system above. This is a vector space generated by elements $\{a_{jk} \cdot \phi_i(\mathbf{A}\omega)\}$. For any combination (ω, i) there is a specific linear combination given by

$$\sum_j a_{ji}\phi_j(\mathbf{A}\omega) = \phi(\mathbf{A}\omega, \mathbf{A}\mathbf{e}_i) = \phi(\mathbf{A}(\omega \wedge \mathbf{e}_i)).$$

These linear combinations are all of the form $\phi(\mathbf{A}\theta)$ where $\theta \in \bigwedge^3 \mathbb{F}_q^n$. More specifically, $0 = \phi(\mathbf{A}\theta) = \psi(\theta)$, must hold for every θ, therefore $\theta \in \ker(\psi)$. With this structure in consideration let us look at the map

$$\xi_\psi : L(\psi) \otimes \mathbb{F}_q^n \to \ker(\psi), \quad (\omega, \mathbf{x}) \mapsto \omega \wedge \mathbf{x} \tag{5}$$

Of special interest are elements in the kernel of ξ_ψ. Let $\sum_k \omega_k \otimes \mathbf{e}_{i_k} \in \ker \xi_\psi$ then

$$\begin{aligned} \sum_k \sum_j a_{ji_k}\phi_j(\mathbf{A}\omega) &= \sum_k \phi(\mathbf{A}\omega_k, \mathbf{A}\mathbf{e}_{i_k}) \\ &= \phi\left(\mathbf{A}\left(\sum_k \omega_k \wedge \mathbf{e}_{i_k}\right)\right) \\ &= \phi(\mathbf{A}(\mathbf{0})) \\ &\equiv 0. \end{aligned}$$

Thus, we get a syzygy for each vector in the kernel of ξ_ψ. Let us call these *wedge syzygies*.

Remark 1. Empirical analysis for n up to 25 shows that this map is surjective for $n \in \{4, 5\} \cup \{7, \ldots, 25\}$ for random alternating trilinear forms. In the case $n = 6$, the image consistently has dimension one lower than $\ker(\psi)$. This might be interesting to look at from a mathematical point of view. However, for practical considerations we treat this as just a curiosity.

Now using the rank-nullity theorem we obtain the dimension for the module in degree 3 generated by wedge syzygies:

$$\left(\binom{n}{2} - n\right) \cdot n - \left(\binom{n}{3} - 1\right) = \frac{(n+1)(n-1)(n-3)}{3}.$$

6.2 Hilbert series and the solving degree

We analyze how the system behaves under the block Wiedemann XL algorithm [Cop94]. For this we need the Hilbert series, the generating function for the monomials, and the density of our equations. In order to state the Hilbert series we have to make an assumption about the syzygies appearing in our system.

Assumption 1. *The syzygy module of the ideal in the system in Equation (4) is generated by the trivial syzygies and the wedge syzygies.*

Using this assumption we can state the Hilbert series for the ideal generated by our system. To sum-up, we have a system of $\frac{n^2(n-3)}{2}$ quadratic equations in n^2 variables with $\frac{(n+1)(n-1)(n-3)}{3}$ syzygies in degree 3. First let us give the generating function for the amount of monomials in each degree as:

$$\mathcal{M}(t) = \frac{1}{(1-t)^{n^2}}.$$

Here we denote by $[t^\alpha]\mathcal{M}$ the coefficient of t^α in the series. Now we can state the Hilbert series:

$$\mathcal{H}(t) = (1-t^2)^{\frac{n^2(n-3)}{2}}(1-t^3)^{-\frac{(n+1)(n-1)(n-3)}{3}}\mathcal{M}(t).$$

Next let us look at the density of the equations in our system. In the modelling in Equation (3) we take the product of the matrices $\mathbf{G}'^{\perp}$ and $\mathbf{G}(\mathbf{A}\otimes\mathbf{A})$. The dual code of $\mathcal{D}$ is of dimension $\binom{n}{2}-n$ in a vector space of dimension $\binom{n}{2}$. Therefore it can be represented by a basis of skew-symmetric matrices with $n+1$ non-zero entries in the upper-half triangle. Then taking the systematic form of $\mathbf{G}'^{\perp}$, we obtain $2(n+1)$ nonzero entries per row. On the other hand $\mathbf{G}(\mathbf{A}\otimes\mathbf{A})$ has a linear combination of $\binom{n-1}{2}$ terms $a_{ij}a_{i'j'}$ in every cell. Therefore, the density per equation is at most $2(n+1)\binom{n-1}{2}$.

The complexity for using the block Wiedemann XL algorithm is given by:

$$\mathcal{O}\left(\min_{\alpha,[t^\alpha]\mathcal{H}\leq 0} 3\cdot(n-2)(n-1)(n+1)\cdot([t^\alpha]\mathcal{M})^2\right).$$

Here the factor $(n-2)(n-1)(n+1)$ is the density and 3 is a hidden constant of the algorithm itself. Now a simple computation will give us the witness degree and complexities for solving ATFE systems. These are summarized in Table 2. In comparison, the ALTEQ specs provide an algebraic analysis resulting in the numbers from Table 3.

Now, our results show that the parameters for ALTEQ need to be increased to attain the claimed NIST security levels. In particular, Table 2 indicates that for NIST level I, the size of the matrices needs to be increased to at least $n = 17$, for NIST level III to at least $n = 27$, and for NIST level V to at least $n = 37$. Note that we are not taking into account the possibility of existence of weak keys, which may further increase parameters.

Table 2. Solving degrees and complexities for ATFE instances using the improved matrix-code modelling. The cost is given in terms of field operations.

n	d_{wit}	$\log_2$ cost attack
8	9	83
9	9	90
10	9	95
11	9	101
12	9	105
13	9	110
14	10	123
15	10	127
17	10	135
18	11	148
20	11	155
25	13	193
27	13	199
28	13	202
30	14	219
35	15	245
37	16	263
38	16	265
40	17	283

Table 3. Comparison to the algebraic analysis from the ALTEQ specifications [BDN+23]. The cost is given in terms of field operations.

n	ALTEQ specs [BDN+23]		This paper	
	d	$\log_2$ cost	d	$\log_2$ cost
13	11	143	9	110
20	15	219	11	155
25	18	276	13	193

7 Experimental results

To confirm our theoretical findings, we implemented both the direct modelling described in Section 5.1 and the improved modelling with our proposed variant described in Section 5.3. Using this implementation, we perform experiments to confirm the estimates in our complexity analysis. In addition, we solve random instances of both modellings to compare the running times.

7.1 Computing syzygies

In order to find the structure of the system of equations, we ran experiments to look for syzygies. This was done in two ways. In the first setting, we ran experiments by computing the entire Macaulay matrix up to certain degrees.

However, since these experiments are computationally heavy we considered also another approach to be able to tell something for higher n. In the second setting, we looked at the kernel of ξ_ψ as in Equation (5), as these generate syzygies.

Using Macaulay matrices. We ran experiments on computing the Macaulay matrices for several degrees and several values for n. For this, we first generate the system of equations from our modelling. Next, we multiply all equations by all monomials of the corresponding degrees. Then, we construct the Macaulay matrix from this and finally, we row reduce in order to find the left nullity. The left nullity will tell us the amount of syzygies in the corresponding degree. The predicted amount of syzygies in each degree can be calculated from the Hilbert series and correspond to the coefficients of the following series:

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

For $d = 3, 4$ this corresponds to

$$[t^3]\mathcal{S} = \frac{(n+1)(n-1)(n-3)}{3}$$

and

$$[t^4]\mathcal{S} = n^2\frac{(n+1)(n-1)(n-3)}{3} + \binom{n\left(\binom{n}{2} - n\right)}{2}.$$

Note that the resources required to run these calculations are high and this limits the size of n and d in our setup. The results can be found in Table 4.

Table 4. Experimental syzygies.

n	$d=3$		$d=4$	
	experiment	prediction	experiment	prediction
5	16	16	906	700
6	72	35	4149	2691
7	64	64	7889	7889
8	105	105	20386	19440
9	160	160	42363	42363
10	231	231		
11	320	320		

From the results, we conclude that we correctly predict the amount of syzygies in degree 3 for the n values that we tested (except for $n = 6$) and that we correctly predict $n \in \{7, 9\}, d = 4$. As we can see, for $n \in \{5, 6, 8\}$, the predictions for $d = 4$ are off and extra syzygies appear. For $n \in \{5, 6\}$ this is not surprising as we know that the automorphism groups are non-trivial. Furthermore, the matrices ϕ_i are of rank at most 4 since they have to be even and at

most $n-1$. This might also lead to extra syzygies in degree 4 for these two values of n. For $n = 8$ the extra syzygies might indicate that there is more structure in even degree that is worth exploring. The fact that the amount of syzygies for $n = 7$ and $n = 9$ are a correct prediction should give us some reassurance for higher n.

The function ξ_ψ. Recall the function ξ_ψ that we introduced in Equation (5). Since every element in $\ker(\xi_\psi)$ leads to a syzygy in degree three, it is worthwhile to explore its size. Then we can give a lower bound on the amount of syzygies. As stated before, we used experiments to verify that this is surjective for random alternating trilinear forms for n up to 30 (except $n = 6$). For each of those we computed the vector space $L(\psi)$. Then we created a list of wedge products of $\omega \in L(\psi)$ and canonical basis vectors $\mathbf{e}_i$. This results in a list of elements from $\bigwedge^3 \mathbb{F}_q^n$. These are just 3-way arrays so we vectorized them to vectors of length n^3. Finally, we computed the dimension of the space spanned by these vectors and verified this is the same dimension as $\ker(\psi)$. We conclude that the functions ξ_ψ are surjective for all these random instances and assume this holds for the generic case.

7.2 Running Gröbner basis computations

As a final step in our experimental work, we solve concrete instances of the systems arising from the quadratic with inverse modelling from [BDN+23] and the improved modelling, using the F4 [Fau99] implementation in MAGMA [BCP97]. For parameter sizes $n = \{5, 6, 7\}$, we generate 50 random instances of ATFE with one planted solution. We do this by generating a random trilinear form ϕ and a random invertible matrix $\mathbf{A}$, and then applying the group action to compute ψ. Note that for these parameter sizes ($n < 9$) we expect to have many solutions to the systems, so instead of enumerating the solution space, we stop after the computation of the Gröbner basis.

Table 5. Running times (in seconds) of F4 using two modellings of ATFE.

n	Modelling in [BDN+23]	Our modelling
5	64.20	0.64
6	> 200000	679.46

Results shown in Table 5 are an average of 50 runs. All of the instances are over $\mathbb{F}_q$ with $q = 3$, however, we performed (fewer instances of) these experiments with $q = 31$ and obtained comparable results. We see that the improved modelling significantly outperforms the quadratic with inverse modelling, which is in line with our theoretical findings. For $n = 7$, the computation for both variants timed out after 72 hours. For $n = 6$, we were only able to solve the systems using the improved modelling. However, the authors of [BDN+23] report

that they were able to solve the system for $n = 6$ in about 25 hours with the quadratic with inverse modelling.

References

[ADFMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In *ASIACRYPT '20*, pages 411–439. Springer, 2020.

[Bar04] Magali Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.* PhD thesis, Université de Paris VI, 2004.

[BBC+20] Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray A. Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier A. Verbel. Improvements of algebraic attacks for solving the rank decoding and minrank problems. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020*, volume 12491 of *LNCS*, pages 507–536. Springer, 2020.

[BBPS21] Alessandro Barenghi, Jean-François Biasse, Edoardo Persichetti, and Paolo Santini. LESS-FM: fine-tuning signatures from the code equivalence problem. In Jung Hee Cheon and Jean-Pierre Tillich, editors, *PQCrypto 2021*, volume 12841 of *LNCS*, pages 23–43. Springer, 2021.

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma Algebra System. I. The User Language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[BDN+23] Markus Bläser, Dung Hoang Duong, Anand Kumar Narayanan, Thomas Plantard, Youming Qiao, Arnaud Sipasseuth, , and Gang Tang. The ALTEQ Signature Scheme: Algorithm Specifications and Supporting Documentation. NIST PQC Submission, 2023.

[Beu20] Ward Beullens. Not enough LESS: an improved algorithm for solving code equivalence problems over $\mathbb{F}_q$. In Orr Dunkelman, Michael J. Jacobson, and Colin O'Flynn, editors, *SAC 2020*, volume 12804 of *LNCS*, pages 387–403. Springer, 2020.

[Beu22] Ward Beullens. Graph-theoretic algorithms for the alternating trilinear form equivalence problem. Cryptology ePrint Archive, Paper 2022/1528, 2022. https://eprint.iacr.org/2022/1528.

[BFFP11] C. Bouillaguet, J.-C. Faugère, P.A. Fouque, and L. Perret. Practical Cryptanalysis of the Identification Scheme Based on the Isomorphism of Polynomial With One Secret Problem. In *Public Key Cryptography – PKC 2011*, volume 6571 of *Lecture Notes in Computer Science*, pages 441–458. Springer, 2011.

[BFSY05] Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Bo-Yin Yang. Asymptotic Behaviour of the Degree of Regularity of Semi-Regular Polynomial Systems. In *Proc. of MEGA 2005, Eighth In-*

ternational Symposium on Effective Methods in Algebraic Geometry, 2005.

[BFV13] Charles Bouillaguet, Pierre-Alain Fouque, and Amandine Véber. Graph-theoretic algorithms for the "isomorphism of polynomials" problem. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 211–227. Springer, 2013.

[BKP20] Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020*, volume 12492 of *LNCS*, pages 464–492. Springer, 2020.

[BMPS20] Jean-François Biasse, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. LESS is More: Code-Based Signatures Without Syndromes. In Abderrahmane Nitaj and Amr Youssef, editors, *AFRICACRYPT 2020*, volume 12174 of *LNCS*, pages 45–65. Springer, 2020.

[Bou11] Charles Bouillaguet. *Algorithms for some hard problems and cryptographic attacks against specific cryptographic primitives*. PhD thesis, Université Paris Diderot, 2011.

[CNP+22] Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Tovohery Hajatiana Randrianarisoa, Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. Take your meds: Digital signatures from matrix code equivalence. Cryptology ePrint Archive, Paper 2022/1559, 2022. https://eprint.iacr.org/2022/1559.

[CNP+23] Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Lars Ran, Tovohery Hajatiana Randrianarisoa, Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. MEDS – Matrix Equivalence Digital Signature, 2023. Submission to the NIST Digital Signature Scheme standardization process.

[Cop94] Don Coppersmith. Solving homogeneous linear equations over gf (2) via block wiedemann algorithm. *Mathematics of Computation*, 62:333–350, 1994.

[Cou06] Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Paper 2006/291, 2006.

[DFG19] Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019*, volume 11478 of *LNCS*, pages 759–789. Springer, 2019.

[Fau99] Jean-Charles Faugère. A New Efficient Algorithm for Computing Gröbner basis (F4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, 1999.

[FdVP08] Jean-Charles Faugère, Françoise Levy dit Vehel, and Ludovic Perret. Cryptanalysis of MinRank. In *CRYPTO 2008*, volume 5157 of *LNCS*, pages 280–296. Springer, 2008.

[FP06] Jean-Charles Faugère and Ludovic Perret. Polynomial equivalence problems: Algorithmic and theoretical aspects. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 30–47. Springer, 2006.

[GQ21] Joshua A. Grochow and Youming Qiao. On the Complexity of Isomorphism Problems for Tensors, Groups, and Polynomials I: Tensor Isomorphism-Completeness. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[Gre12] W.H. Greub. *Multilinear Algebra*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2012.

[Leo82] Jeffrey S. Leon. Computing automorphism groups of error-correcting codes. *IEEE Trans. Inf. Theory*, 28(3):496–510, 1982.

[Pat96] Jacques Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of asymmetric algorithms. In *EUROCRYPT '96*, volume 1070 of *LNCS*, pages 33–48. Springer, 1996.

[Per05] Ludovic Perret. A Fast Cryptanalysis of the Isomorphism of Polynomials with One Secret Problem. In *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 354–370. Springer, 2005.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Paper 2006/145, 2006.

[RST22] Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. Hardness estimates of the code equivalence problem in the rank metric. Cryptology ePrint Archive, Paper 2022/276, 2022.

[TDJ+22] Gang Tang, Dung Hoang Duong, Antoine Joux, Thomas Plantard, Youming Qiao, and Willy Susilo. Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In *EUROCRYPT 2022*, volume 13277 of *LNCS*, pages 582–612. Springer, 2022.